

[MS-OXCMAIL]:

RFC 2822 and MIME to Email Object Conversion Algorithm

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.

Preliminary

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	1.0.0	Major	Initial Availability.
4/25/2008	0.2	Editorial	Revised and updated property names and other technical content.
6/27/2008	1.0	Major	Initial Release.
8/6/2008	1.01	Editorial	Revised and edited technical content.
9/3/2008	1.02	Editorial	Revised and edited technical content.
12/3/2008	1.03	Editorial	Revised and edited technical content.
2/4/2009	1.04	Editorial	Revised and edited technical content.
3/4/2009	1.05	Editorial	Revised and edited technical content.
4/10/2009	2.0	Major	Updated technical content and applicable product releases.
7/15/2009	3.0	Editorial	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	13.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	14.0	Major	Significantly changed the technical content.
2/11/2013	15.0	Major	Significantly changed the technical content.
7/26/2013	16.0	Major	Significantly changed the technical content.
11/18/2013	16.1	Minor	Clarified the meaning of the technical content.
2/10/2014	16.1	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	16.1	No Change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
7/31/2014	16.1	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	17.0	Major	Significantly changed the technical content.
3/16/2015	18.0	Major	Significantly changed the technical content.
5/26/2015	18.0	No Change	No changes to the meaning, language, or formatting of the technical content.

Preliminary

Table of Contents

1	Introduction	10
1.1	Glossary	10
1.2	References	15
1.2.1	Normative References	16
1.2.2	Informative References	18
1.3	Overview	19
1.3.1	Data Models	19
1.4	Relationship to Protocols and Other Algorithms	20
1.5	Applicability Statement	21
1.6	Standards Assignments	21
2	Algorithm Details	22
2.1	MIME Generation Algorithm Details	22
2.1.1	Abstract Data Model	23
2.1.1.1	Global	23
2.1.1.2	Per Mailbox	23
2.1.1.3	Per Message Object	23
2.1.2	Initialization	23
2.1.3	Processing Rules	23
2.1.3.1	Address Elements	24
2.1.3.1.1	Recipients	24
2.1.3.1.1.1	To and Cc Recipients	25
2.1.3.1.1.2	Bcc Recipients	26
2.1.3.1.2	Reply-To	26
2.1.3.1.3	From	27
2.1.3.1.4	Sender	27
2.1.3.1.5	Return Receipt	27
2.1.3.1.6	Read Receipt	28
2.1.3.1.7	Directory Lookups	28
2.1.3.1.8	IMCEA Encapsulation	28
2.1.3.1.9	PidTagAddressType Property	29
2.1.3.2	Envelope Elements	30
2.1.3.2.1	Message Class	30
2.1.3.2.2	Content Class	31
2.1.3.2.3	Unified Messaging Properties	31
2.1.3.2.4	Arbitrary MIME Headers	32
2.1.3.2.5	Importance	32
2.1.3.2.6	Sensitivity	33
2.1.3.2.7	Sent Time	33
2.1.3.2.8	Subject	34
2.1.3.2.9	Conversation Topic	34
2.1.3.2.10	Conversation Index	34
2.1.3.2.11	Message ID	34
2.1.3.2.12	References	35
2.1.3.2.13	Categories	35
2.1.3.2.14	In-Reply-To Message ID	35
2.1.3.2.15	List Server Properties	35
2.1.3.2.16	Language Properties	35
2.1.3.2.17	Classification Properties	36
2.1.3.2.18	Payload Properties	36
2.1.3.2.19	Has Attach	37
2.1.3.2.20	Auto Response Suppress	37
2.1.3.2.21	Is Auto Forwarded	38

2.1.3.2.22	Sender Id Status	38
2.1.3.2.23	Purported Sender Domain	38
2.1.3.2.24	Spam Confidence Level	38
2.1.3.2.25	Flag Request.....	38
2.1.3.2.26	TNEF Correlation Key	38
2.1.3.2.27	Received Headers	39
2.1.3.2.28	ReplyBy Time.....	39
2.1.3.2.29	Content-ID	39
2.1.3.2.30	Content-Location.....	39
2.1.3.2.31	XRef	39
2.1.3.3	Body Text	39
2.1.3.3.1	Client Actions.....	40
2.1.3.3.2	Message Body in TNEF	41
2.1.3.3.3	Simple Plain Text Message Body.....	41
2.1.3.3.4	HTML Text Message Body Without Inline Attachments.....	42
2.1.3.3.5	HTML Text Message Body from RTF Without Inline Attachments	42
2.1.3.3.6	HTML Text Message Body with Inline Attachments.....	42
2.1.3.3.7	HTML Text Message Body from RTF with Inline (OLE) Attachments	43
2.1.3.3.8	Calendar Items and Meeting Messages	43
2.1.3.3.8.1	Plain Text Calendar Message	43
2.1.3.3.8.2	Calendar Message Without Inline Attachments.....	43
2.1.3.3.8.3	Calendar Message with Inline Attachments.....	44
2.1.3.3.9	Enriched Text Message Body.....	44
2.1.3.4	Attachments.....	44
2.1.3.4.1	Inline Attachments	46
2.1.3.4.1.1	Inline Attachments in RTF Messages.....	46
2.1.3.4.1.2	Inline Attachments in HTML Messages	46
2.1.3.4.2	Attached Files	47
2.1.3.4.2.1	File Name	47
2.1.3.4.2.2	Content-Type, Content-Description, Content-Disposition Headers	47
2.1.3.4.2.3	Content-ID, Content-Location, Content-Base.....	48
2.1.3.4.2.4	Content-Transfer-Encoding, MIME Part Body	48
2.1.3.4.3	MacBinary Attached Files.....	48
2.1.3.4.3.1	Application/Applefile.....	51
2.1.3.4.4	OLE Attachments.....	51
2.1.3.4.5	Embedded Message Attachments.....	51
2.1.3.4.6	vCard Generation	52
2.1.3.5	Generating Pure MIME Messages.....	52
2.1.3.5.1	Generation Process.....	52
2.1.3.6	Generating Report Messages	53
2.1.3.6.1	Generating Delivery Status Notification Messages.....	53
2.1.3.6.1.1	Generating a Value for the Action Field	54
2.1.3.6.1.2	Generating a Value for the Status Field.....	54
2.1.3.6.2	Generating Message Disposition Notification Messages	56
2.1.3.6.2.1	Generating a Value for the Disposition Field	57
2.1.3.7	Generating TNEF Messages	57
2.2	MIME Analysis Algorithm Details	58
2.2.1	Abstract Data Model.....	58
2.2.1.1	Global.....	58
2.2.1.2	Per Mailbox	58
2.2.1.3	Per Message Object.....	58
2.2.2	Initialization	58
2.2.3	Processing Rules.....	59
2.2.3.1	Address Elements	59
2.2.3.1.1	Mapping Internet E-Mail Address Elements to a Property Group.....	59
2.2.3.1.2	Recognizing and De-Encapsulating IMCEA-Encapsulated Addresses.....	59
2.2.3.1.3	From.....	60

2.2.3.1.4	Sender	60
2.2.3.1.5	To, Cc, Bcc	61
2.2.3.1.6	Reply Recipients	61
2.2.3.1.7	Disposition Notification Recipients	61
2.2.3.1.8	Return-Receipt-To	62
2.2.3.2	Envelope Elements	62
2.2.3.2.1	MessageID	62
2.2.3.2.2	Sent time	62
2.2.3.2.3	References	63
2.2.3.2.4	Sensitivity	63
2.2.3.2.5	Importance.....	63
2.2.3.2.6	Subject	64
2.2.3.2.6.1	Normalizing the Subject.....	64
2.2.3.2.7	Conversation Topic	65
2.2.3.2.8	Conversation Index.....	66
2.2.3.2.9	In-Reply-To Message ID	66
2.2.3.2.10	ReplyBy Time.....	66
2.2.3.2.11	Language Properties	66
2.2.3.2.12	Categories	66
2.2.3.2.13	Message Expiry Time	67
2.2.3.2.14	Suppression of Automatic Replies	67
2.2.3.2.15	Content Class.....	67
2.2.3.2.15.1	Requirements for Fax messages	68
2.2.3.2.15.2	Requirements for Voice and Voicemail messages	68
2.2.3.2.16	Message Flagging	69
2.2.3.2.17	List Server Properties.....	69
2.2.3.2.18	Payload Properties.....	70
2.2.3.2.19	Purported Sender Domain	70
2.2.3.2.20	Sender Id Status	70
2.2.3.2.21	Spam Confidence Level	70
2.2.3.2.22	Classification Properties.....	70
2.2.3.2.23	Unified Messaging Properties	71
2.2.3.2.24	Content-ID	71
2.2.3.2.25	Content-Base	72
2.2.3.2.26	Content-Location.....	72
2.2.3.2.27	XRef	72
2.2.3.2.28	PidTagTransportMessageHeaders.....	72
2.2.3.2.29	Generic Headers in PS_INTERNET_HEADERS	72
2.2.3.3	Body Text	74
2.2.3.3.1	Client Actions.....	74
2.2.3.3.2	Determining Which MIME Element Is the Message Body	75
2.2.3.3.2.1	Selecting the Primary Message Text MIME Element	75
2.2.3.3.2.2	Creating an Aggregate Body	76
2.2.3.4	Attachments.....	77
2.2.3.4.1	Regular File Attachment MIME Part Analysis.....	77
2.2.3.4.1.1	File Name	77
2.2.3.4.1.2	Content Type	79
2.2.3.4.1.3	Attachment Creation and Modification Date	79
2.2.3.4.1.4	Attachment Content-Id, Content-Base, and Content-Location	80
2.2.3.4.1.4.1	Inline Attachments	80
2.2.3.4.1.5	Attachment Content-Transfer-Encoding and MIME Part Body	80
2.2.3.4.2	Apple File Formats.....	81
2.2.3.4.2.1	Multipart/Appledouble	81
2.2.3.4.2.2	Application/Applefile.....	83
2.2.3.4.2.3	Application/Mac-binhex40	86
2.2.3.4.3	Attached Messages	86
2.2.3.4.4	Inbound vCard Conversion.....	88

2.2.3.4.4.1	Content-Type	88
2.2.3.4.4.2	General Parsing Guidelines.....	88
2.2.3.5	External Body Attachments	89
2.2.3.6	Reading Pure MIME Messages.....	90
2.2.3.7	Reading Report Messages	90
2.2.3.7.1	Reading Delivery Status Notification Messages	90
2.2.3.7.1.1	Determining the Value of the PidTagMessageClass Property	91
2.2.3.7.1.2	Calculating a Value for the PidTagSupplementaryInfo Property	91
2.2.3.7.1.3	Processing the Status Field	92
2.2.3.7.2	Reading Message Disposition Notification Messages	94
2.2.3.8	Reading TNEF Messages.....	95
2.2.3.9	Additional Content Types	95
2.2.3.9.1	Analysis of Non-MIME Content	95
2.2.3.9.2	Message/Partial.....	95
2.2.3.9.3	Multipart/Digest	96
2.3	Unconverted MIME Part Generation Algorithm Details	96
2.3.1	Abstract Data Model.....	96
2.3.1.1	Global.....	96
2.3.1.2	Per Mailbox	96
2.3.1.3	Per Message Object.....	96
2.3.2	Initialization.....	97
2.3.3	Processing Rules.....	97
2.3.3.1	Impact of Message Changes on the MIME Skeleton	97
2.4	Unconverted MIME Part Analysis Algorithm Details	101
2.4.1	Abstract Data Model.....	101
2.4.1.1	Global.....	102
2.4.1.2	Per Mailbox	102
2.4.1.3	Per Message Object.....	102
2.4.2	Initialization.....	102
2.4.3	Processing Rules.....	102
2.4.3.1	MIME Conversion	102
2.5	Message Object Properties	103
2.5.1	PidLidClassificationGuid	103
2.5.2	PidLidClassificationKeep.....	103
2.5.3	PidNameCrossReference	103
2.5.4	PidNameQuarantineOriginalSender.....	103
2.6	Recipient Property Groups.....	103
2.6.1	PidTagReadReceipt Property Group	104
2.6.2	PidTagReceivedBy Property Group	104
2.6.3	PidTagReceivedRepresenting Property Group.....	105
2.6.4	PidTagSender Property Group	105
2.6.5	PidTagSentRepresenting Property Group.....	105
2.6.6	Recipient Table Property Group	106
3	Algorithm Examples	107
3.1	MIME Examples	107
3.1.1	Simple MIME Message	107
3.1.2	MIME Message Containing Inline and Non-Inline Attachments	109
3.1.3	MIME Message Containing Only Inline Attachments.....	110
3.1.4	MIME Message Containing Only Non-Inline Attachments	110
3.1.5	E-Mail Message Without a MIME-Version Header	111
4	Security.....	112
4.1	Security Considerations for Implementers	112
4.1.1	Unsolicited Commercial E-Mail (Spam)	112
4.1.2	Information Disclosure	112
4.1.3	Content-Type Versus File Extension Mismatch	112

4.1.4	Do Not Support Message/Partial	113
4.1.5	Considerations for Message/External-Body.....	113
4.1.6	Preventing Denial of Service Attacks	113
4.1.6.1	Submission Limits	113
4.1.6.2	Complexity of Nested Entities	114
4.1.6.3	Number of Embedded Messages	114
4.1.6.4	Compressed Attachments.....	114
4.2	Index of Security Parameters	114
5	Appendix A: Product Behavior	115
6	Change Tracking.....	127
7	Index.....	129

Preliminary

1 Introduction

The RFC 2822 and MIME to Email Object Conversion Algorithm consists of a set of algorithms that applications use to convert data between these two representations of e-mail messages. The process of converting **Message object** data to **MIME** format is referred to as **MIME generation**, while the reverse process is referred to as **MIME analysis**.

Section 2 of this specification is normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [\[RFC2119\]](#). Section 1.6 is also normative but does 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 list: A collection of distinct Address Book objects.

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

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**.

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [\[RFC5234\]](#).

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable **ASCII** characters, as described in [\[RFC4648\]](#).

best body: The text format that provides the richest representation of a **message body (2)**. The algorithm for determining the best-body format is described in [\[MS-OXBBODY\]](#).

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

binary large object (BLOB): A discrete packet of data that is stored in a database and is treated as a sequence of uninterpreted bytes.

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 [\[RFC2045\]](#).

calendar: A date range that shows availability, meetings, and appointments for one or more users or **resources**. See also Calendar object.

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.

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

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 person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and attributes (2), such as an Internet message address or login name.

contact attachment: An attached message item that has a message type of "IPM.Contact" and adheres to the definition of a **Contact object**.

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

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).

cyclic redundancy check (CRC): An algorithm used to produce a checksum (a small, fixed number of bits) against a block of data, such as a packet of network traffic or a block of a computer file. The CRC is used to detect errors after transmission or storage. A CRC is designed to catch random errors, as opposed to intentional errors. If errors might be introduced by a motivated and intelligent adversary, a cryptographic hash function should be used instead.

delivery status notification (DSN): A message that reports the result of an attempt to deliver a message to one or more recipients, as described in [\[RFC3464\]](#).

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

distinguished name (DN): A name that uniquely identifies an object by using the relative distinguished name (RDN) for the object, and the names of container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in a tree.

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a domain controller (DC) and host a member list that identifies all members of the domain, as well as optionally hosting the Active Directory service. The domain controller provides authentication (2) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [\[MS-AUTHSOD\]](#) section 1.1.1.5 and [\[MS-ADTS\]](#).

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

encapsulation: A process of encoding one document in another document in a way that allows the first document to be re-created in a form that is nearly identical to its original form.

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.

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)**.

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

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 [\[HTML\]](#).

Internet Mail Connector Encapsulated Address (IMCEA): A means of encapsulating an email address that is not compliant with [\[RFC2821\]](#) within an email address that is compliant with [\[RFC2821\]](#).

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

Joint Photographic Experts Group (JPEG): A raster graphics file format for displaying high-resolution color graphics. JPEG graphics apply a user-specified compression scheme that can significantly reduce the file sizes of photo-realistic color graphics. A higher level of compression results in lower quality, whereas a lower level of compression results in higher quality. JPEG-format files have a .jpg or .jpeg file name extension.

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.

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

Mail User Agent (MUA): A client application that is used to compose and read email messages.

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

message body: (1) The content within an HTTP message, as described in [\[RFC2616\]](#) section 4.3.

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

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

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

addressed, and an **attachments table** that represents any files and other Message objects that are attached to it.

Messaging Application Programming Interface (MAPI): A messaging architecture that enables multiple applications to interact with multiple messaging systems across a variety of hardware platforms.

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.

MIME analysis: A process that converts data from an Internet wire protocol to a format that is suitable for storage by a server or a client.

MIME body: The content of a **MIME** entity, which follows the header of the MIME entity to which they both belong.

MIME content-type: A content type that is as described in [RFC2045], [RFC2046], and [\[RFC2047\]](#).

MIME entity: An entity that is as described in [RFC2045], [RFC2046], and [RFC2047].

MIME generation: A process that converts data held by a server or client to a format that is suitable for Internet wire protocols.

MIME message: A message that is as described in [RFC2045], [RFC2046], and [RFC2047].

MIME part: A message part that is as described in [RFC2045], [RFC2046], and [RFC2047].

MIME reader: An agent that performs MIME analysis. It can be a client or a server.

MIME writer: An agent that performs **MIME** generation. It can be a client or a server.

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

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

non-delivery report: A report message that is generated and sent by a server to the sender of a message if an email message could not be received by an intended recipient.

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.

one-off address: An email address that is encoded as a mail-type/address pair. Valid mail-types include values such as SMTP, X400, X500, and MSMAIL.

one-off EntryID: A special address object **EntryID** that encapsulates electronic address information, as described in [\[MS-OXCDATA\]](#).

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

Personal Information Manager (PIM): A category of software packages for managing commonly used types of personal information, including contacts, email messages, calendar appointments, and meetings.

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

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

primary SMTP proxy address: The **Simple Mail Transfer Protocol (SMTP)** email address that is used to designate a message server user in all SMTP traffic. Proxy addresses are stored in a user's **address book** entry, in the PidTagAddressBookProxyAddresses multivalued string property. The primary SMTP proxy address can be identified by its **address type** field, which is set to "SMTP" (uppercase). Non-primary SMTP proxy addresses have the **address type** field set to "smtp" (lowercase).

property set: A set of attributes (1), identified by a **GUID**. Granting access to a property set grants access to all the attributes in the set.

property type: A 16-bit quantity that specifies the data type of a property value.

PS_INTERNET_HEADERS: An extensible namespace that can store custom property headers.

pure MIME message: A **MIME** representation of an email message that does not contain a **Transport Neutral Encapsulation Format (TNEF)** body part.

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)**.

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

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

remote procedure call (RPC): A 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].

resource: Any component that a computer can access where data can be read, written, or processed. This resource could be an internal component such as a disk drive, or another computer on a network that is used to access a file.

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

S/MIME (Secure/Multipurpose Internet Mail Extensions): A set of cryptographic security services, as described in [\[RFC5751\]](#).

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

spam: An unsolicited email message.

stream: (1) An element of a compound file, as described in [\[MS-CFB\]](#). A stream contains a sequence of bytes that can be read from or written to by an application, and they can exist only in storages.

(2) A flow of data from one host to another host, or the data that flows between two hosts.

To recipient: See primary recipient.

top-level message: A message that is not included in another message as an **Embedded Message object**. Top-level messages are messaging objects.

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

Transport Neutral Encapsulation Format (TNEF) message: A **MIME** representation of an email message in which attachments and some message properties are carried in a **Transport Neutral Encapsulation Format (TNEF)** body part.

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).

Unified Messaging: A set of components and services that enable voice, fax, and email messages to be stored in a user's **mailbox** and accessed from a variety of devices.

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 [\[RFC3986\]](#).

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [\[RFC1738\]](#).

universally unique identifier (UUID): A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and **RPC** objects. UUIDs are highly likely to be unique. UUIDs are also known as **globally unique identifiers (GUIDs)** and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. 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 UUID.

UTF-16LE: The Unicode Transformation Format - 16-bit, Little Endian encoding scheme. It is used to encode **Unicode** characters as a sequence of 16-bit codes, each encoded as two 8-bit bytes with the least-significant byte first.

vCard: A format for storing and exchanging electronic business cards, as described in [\[RFC2426\]](#).

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.

[HTML] World Wide Web Consortium, "HTML 4.01 Specification", December 1999, <http://www.w3.org/TR/html4/>

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

[MacBin] Apple Corporation, "Macintosh Binary Transfer Format ("MacBinary III") Standard Proposal", December 1996, http://web.archive.org/web/20050305044255/www.lazerware.com/formats/macbinary/macbinary_iii.html

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

[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-OXCFOLD] Microsoft Corporation, "[Folder Object Protocol](#)".

[MS-OXCICAL] Microsoft Corporation, "[iCalendar to Appointment Object Conversion Algorithm](#)".

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

[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-OXCSPAM] Microsoft Corporation, "[Spam Confidence Level 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-OXOCNTC] Microsoft Corporation, "[Contact Object Protocol](#)".

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

[MS-OXOMSG] Microsoft Corporation, "[Email Object Protocol](#)".

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

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

[MS-OXOTASK] Microsoft Corporation, "[Task-Related Objects 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-OXVCARD] Microsoft Corporation, "[vCard to Contact Object Conversion Algorithm](#)".
- [MS-WMF] Microsoft Corporation, "[Windows Metafile Format](#)".
- [MSFT-RTF] Microsoft Corporation, "Rich Text Format (RTF) Specification", version 1.9.1, March 2008, <http://www.microsoft.com/en-us/download/details.aspx?id=10725>
- [RFC1738] Berners-Lee, T., Masinter, L., and McCahill, M., Eds., "Uniform Resource Locators (URL)", RFC 1738, December 1994, <http://www.ietf.org/rfc/rfc1738.txt>
- [RFC1740] Faltstrom, P., Crocker, D., and Fair, E., "MIME Encapsulation of Macintosh files - MacMIME", RFC 1740, December 1994, <http://www.rfc-editor.org/rfc/rfc1740.txt>
- [RFC1741] Faltstrom, P., Crocker, D., and Fair, E., "MIME Content Type for BinHex Encoded Files", RFC 1741, December 1994, <http://www.rfc-editor.org/rfc/rfc1741.txt>
- [RFC1896] Resnick, P., and Walker, A., "The text/enriched MIME Content-type", RFC 1896, February 1996, <http://www.rfc-editor.org/rfc/rfc1896.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>
- [RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996, <http://ietf.org/rfc/rfc2047.txt>
- [RFC2076] Palme, J., "Common Internet Message Headers", RFC 2076, February 1997, <http://www.rfc-editor.org/rfc/rfc2076.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>
- [RFC2156] Kille, S., "MIXER (Mime Internet X.400 Enhanced Relay): Mapping between X.400 and RFC 822/MIME", RFC 2156, January 1998, <http://www.rfc-editor.org/rfc/rfc2156.txt>
- [RFC2369] Neufeld, G., and Baer, J., "The Use of URLs as Meta-Syntax for Core Mail List Commands and their Transport through Message Header Fields", RFC 2369, July 1998, <http://www.rfc-editor.org/rfc/rfc2369.txt>
- [RFC2387] Levinson, E., "The MIME Multipart/Related Content-type", RFC 2387, August 1998, <http://www.rfc-editor.org/rfc/rfc2387.txt>
- [RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, August 1998, <http://www.rfc-editor.org/rfc/rfc2392.txt>
- [RFC2426] Dawson, F., and Howes, T., "vCard MIME Directory Profile", RFC 2426, September 1998, <http://www.rfc-editor.org/rfc/rfc2426.txt>
- [RFC2445] Dawson, F., and Stenerson, D., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 2445, November 1998, <http://www.rfc-editor.org/rfc/rfc2445.txt>

[RFC2557] Palme, J., Hopmann, A., and Shelness, N., "MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)", RFC 2557, March 1999, <http://www.rfc-editor.org/rfc/rfc2557.txt>

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

[RFC3030] Vaudreuil, G., "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", RFC 3030, December 2000, <http://www.rfc-editor.org/rfc/rfc3030.txt>

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

[RFC3464] Moore, K., and Vaudreuil, G., "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003, <http://www.rfc-editor.org/rfc/rfc3464.txt>

[RFC3516] Nerenberg, L., "IMAP4 Binary Content Extension", RFC 3516, April 2003, <http://www.rfc-editor.org/rfc/rfc3516.txt>

[RFC3798] Hansen, T., and Vaudreuil, G., Eds., "Message Disposition Notification", RFC 3798, May 2004, <http://www.rfc-editor.org/rfc/rfc3798.txt>

[RFC4646] Phillips, A., and Davis, M., Eds., "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006, <http://www.rfc-editor.org/rfc/rfc4646.txt>

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

[RFC5751] Ramsdell, B., and Turner, S., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010, <http://www.rfc-editor.org/rfc/rfc5751.txt>

[UNICODE5.1] The Unicode Consortium, "The Unicode Standard, Version 5.1.0, defined by: The Unicode Standard, Version 5.0 (Boston, MA, Addison-Wesley, 2007. ISBN 0-321-48091-0), as amended by Unicode 5.1.0", <http://www.unicode.org/versions/Unicode5.1.0/>

1.2.2 Informative References

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

[MSKB2276479] Microsoft Corporation, "Description of the Office Outlook 2007 hotfix package (Outlook-x-none.msp): August 31, 2010", August 2010, <http://support.microsoft.com/kb/2276479>

[MSKB908027] Microsoft Corporation, "Exchange Server 2003 does not include the 'In-reply-to' header field when you use Outlook or Outlook Web Access to reply to an e-mail message that originated from outside the organization", version 3.0, October 2011, <http://support.microsoft.com/kb/908027>

[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>

[RFC2046] Freed, N., and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996, <http://www.rfc-editor.org/rfc/rfc2046.txt>

[RFC2048] Freed, N., Klensin, J., and Postel, J., "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", BCP 13, RFC 2048, November 1996, <http://www.rfc-editor.org/rfc/rfc2048.txt>

[RFC2049] Freed, N., and Borenstein N., "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, November 1996, <http://www.rfc-editor.org/rfc/rfc2049.txt>

[RFC3463] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, January 2003, <http://www.rfc-editor.org/rfc/rfc3463.txt>

[X25] ITU-T, "X25: Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit", ITU-T Recommendation X.25, October 1996, <http://www.itu.int/rec/T-REC-X.25-199610-I/en>

1.3 Overview

This algorithm enables conversion between Message objects and MIME-formatted textual **streams (2)**. Similarly, the agent that performs MIME generation (which might be either a client or server) is referred to as a **MIME writer**, and the agent that performs MIME analysis is referred to as a **MIME reader**.

1.3.1 Data Models

Message objects model e-mail messages and other **PIM** objects after a business memo: there is a single **message body (2)** with zero or more attachments and zero or more **recipients (2)**. Each Message object has a **message class** property that indicates its type and an arbitrary collection of properties. Attached messages allow for the nesting of content.

MIME, in contrast, models e-mail messages as a nested set of **MIME entities**, each of which has **headers** and a (possibly empty) body. No entity is distinguished as the message body (2). The **Content-Type** header indicates the type of each **body part**; other headers indicate whether a body part is intended as a message body or an attachment. Recipients (2) are modeled by e-mail addresses in certain headers on the top-level body part. Multipart body parts allow for grouping and nesting of content, including attached messages.

How the parts of each data model correspond is shown, at a high level, in the following table.

MIME	Message object
E-mail address	Recipient
Header	Property
Body part	Message body
Body part	Attachment

At the next level of detail, some problems become apparent. Because the data models do not match exactly, it becomes more difficult to convert lower-level items between the Message object and MIME formats.

One of the challenges in mapping the Message object content to the MIME format comes from the need to generate human-readable text. A Message object property can be of a data type, such as a **binary large object (BLOB)**, that does not lend itself to representation as text. Two solutions are available for this problem:

1. Generate a **pure MIME message**, in which data that does not lend itself to representation in MIME format is simply omitted from the MIME representation.

2. Generate a **Transport Neutral Encapsulation Format (TNEF) message**, in which data that does not lend itself to representation in MIME format is placed in a **TNEF** body part with a **Content-Type** header of "application/ms-tnef".

Challenges in mapping MIME content to Message objects include distinguishing the message body from attachments, analyzing multipart structures that do not fit the Message object data model, and mapping headers or header parameters that do not have any corresponding property.

Each Message object has a single **character set** (although nested Message objects can have different character sets). MIME, on the other hand, permits the character set of each header and message body to be specified separately.

A wide variety of possible structures exist for **MIME messages**. One particular structure carries a TNEF MIME element, which provides a high level of fidelity to the original Message object content. All TNEF messages have the same structure, as follows:

- At the top level, a MIME entity with a **Content-Type** header value of "multipart/mixed" that specifies all address elements, as well as the following two child entities:
 - A MIME entity with a **Content-Type** header value of "text/plain" that contains a **plain text** rendering of the message body (2).
 - A MIME entity with a **Content-Type** header value of "application/ms-tnef" that contains all attachment content and the **Hypertext Markup Language (HTML)** or **Rich Text Format (RTF)** message body (2). Any Message object properties for which no mapping to MIME headers is defined is encoded as specified in [\[MS-OXTNEF\]](#).

Because a TNEF message is a MIME structure, MIME messages without a TNEF element are sometimes referred to as pure MIME messages to distinguish them from TNEF messages.

1.4 Relationship to Protocols and Other Algorithms

The representation of e-mail messages, **calendar** items, and other Personal Information Manager (PIM) objects by Message objects and their properties is described in [\[MS-OXOMSG\]](#), [\[MS-OXOCAL\]](#), and related specifications.

In contrast, e-mail messages, calendar items, and other PIM objects are represented as textual streams (2) when sent over the Internet using established e-mail protocols. The textual representation of these streams (2) is commonly referred to as RFC2822 format and/or MIME format, as described in [\[RFC2822\]](#), [\[RFC2045\]](#), [\[RFC2046\]](#), [\[RFC2047\]](#), [\[RFC2048\]](#), and [\[RFC2049\]](#).

Conversion between the MIME format and the Message object format is performed in the context of several different protocols. For example:

- Clients and servers perform MIME generation for mail outbound to **Simple Mail Transfer Protocol (SMTP)**.
- Servers perform MIME analysis for mail inbound from SMTP.
- Servers perform MIME generation for Message objects that are downloaded via the **Post Office Protocol - Version 3 (POP3)** or the **Internet Message Access Protocol - Version 4 (IMAP4)**. Clients perform MIME generation for messages that are uploaded via IMAP4.
- Servers perform MIME analysis for Message objects that are uploaded via IMAP4. Clients perform MIME analysis for Message objects that are downloaded via POP3 or IMAP4.

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

1.5 Applicability Statement

These algorithms are used by any application that translates e-mail messages between the MIME and Message object formats, such as an e-mail client that retrieves e-mail from a server that utilizes the Message object format.

1.6 Standards Assignments

None.

Preliminary

2 Algorithm Details

2.1 MIME Generation Algorithm Details

This section specifies both conversion to pure MIME messages and conversion to TNEF messages from Message objects.

When generating a MIME rendering of a Message object, whether a pure MIME message or a TNEF message, MIME writers retrieve all properties of the Message object by issuing one of the following **remote operation (ROP)** sequences:

- **RopGetPropertiesList** ([\[MS-OXCROPS\]](#) section 2.2.8.5) followed by **RopGetPropertiesSpecific** ([\[MS-OXCROPS\]](#) section 2.2.8.3)
- **RopGetPropertiesAll** ([\[MS-OXCROPS\]](#) section 2.2.8.4)

Clients can explicitly request conversion to a pure MIME message or to a TNEF message by doing one of the following. A MIME writer SHOULD return the message format requested by the client.

- A client can request conversion to a pure MIME message for all recipients (2) by setting the value of the **PidTagSendRichInfo** property ([\[MS-EXOABK\]](#) section 2.2.3.18) to FALSE on the Message object, and request conversion to a TNEF message for all recipients (2) by setting the same property value to **TRUE**.
- A client can request conversion to a pure MIME message for an individual recipient (2) by setting the value of the **PidTagSendRichInfo** property to FALSE on that recipient (2), and request conversion to a TNEF message for an individual recipient (2) by setting the same property value to **TRUE**.
- A client can request conversion to a pure MIME message for a message sent using a **one-off address** by setting the **M** bit in the **one-off EntryID**, as specified in [\[MS-OXCDATA\]](#) section 2.2.5.1, and request conversion to a TNEF message by resetting the same bit.

Similarly, when conversion to a pure MIME message is requested, clients can explicitly request plain text or HTML message body generation by one of the following means. Again, a MIME writer SHOULD honor such a client request for message format.

- A client can request a specific **MIME body** format for all recipients (2) by setting the value of the **PidTagSendInternetEncoding** property ([\[MS-EXOABK\]](#) section 2.2.3.19) on the Message object to one of the values in the following table.
- A client can request a specific MIME body format for an individual recipient (2) by setting the value of the **PidTagSendInternetEncoding** property on the recipient (2) to one of the values in the following table.

The valid values of the **PidTagSendInternetEncoding** property are specified in the following table. The first value is the hexadecimal representation of the property value; the second is its representation in **Augmented Backus-Naur Form (ABNF)**, as specified in [\[RFC5234\]](#).

PidTagSendInternetEncoding property value (hexadecimal, ABNF)	Requested format
0x00060000 %x00.00.06.00	Plain text only
0x000E0000, %x00.00.0E.00	HTML only<1>

PidTagSendInternetEncoding property value (hexadecimal, ABNF)	Requested format
0x00160000, %x00.00.16.00	Both plain text and HTML

2.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this algorithm. The described organization is provided to facilitate the explanation of how the algorithm 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) types are defined in this section:

Global

Mailbox

MessageObject

2.1.1.1 Global

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

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

2.1.1.2 Per Mailbox

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

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

2.1.1.3 Per Message Object

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

Mailbox.MessageObject.MIMESkeleton: The original MIME structure of the message.

2.1.2 Initialization

None.

2.1.3 Processing Rules

In the following sections, a client is specified as an agent that sets properties in a Message object, while a MIME writer is specified as the agent that converts Message object properties into MIME format.

2.1.3.1 Address Elements

In general, address elements are generated in a pure MIME message only and not in a TNEF message. However, when a TNEF message is generated, all address elements of messages that are included as attachments of the **top-level message** are generated in the TNEF message only, as specified in [\[MS-OXTNEF\]](#). This is because no MIME entity corresponds to the attached messages; the attached messages are wholly contained in the TNEF message.

MIME writers MUST generate e-mail addresses for MIME recipients (2) in compliance with the address requirements specified in [\[RFC2822\]](#). For example, in cases where a **display name** is generated in a MIME address header, MIME writers SHOULD use the encoding specified by [\[RFC2047\]](#) to encode any display name value that has characters that are not allowed in a MIME header, as specified in [\[RFC2822\]](#); however, MIME writers MAY<2> generate display names with quotes around all but the non-**ASCII** characters, in order to accommodate clients that are not compliant with the display name encoding specified in [\[RFC2047\]](#). MIME writers MAY<3> only generate display names for recipients (2) when the SMTP policy for the **domain** is configured to preserve the sender's address on the message, which is turned off by default. These addresses are SMTP addresses. When a client supports other types of e-mail addresses through the **PidTagAddressType** property ([\[MS-OXOABK\]](#) section 2.2.3.13), MIME writers SHOULD<4> use **Internet Mail Connector Encapsulated Address (IMCEA) encapsulation** of the e-mail address to form an SMTP address, as specified in section [2.1.3.1.8](#).

2.1.3.1.1 Recipients

To create a recipient (2) in a MIME recipient header, clients create a Message object recipient (2) with either a **PidTagEntryId** property ([\[MS-OXCPERM\]](#) section 2.2.4) or both the **PidTagAddressType** property ([\[MS-OXOABK\]](#) section 2.2.3.13) and **PidTagEmailAddress** property ([\[MS-OXOABK\]](#) section 2.2.3.14), which suffice to fully represent the recipient's (2) e-mail **address type** and e-mail address. Clients SHOULD, in addition, set the **PidTagSmtpAddress** property ([\[MS-OXOABK\]](#) section 2.2.3.21), particularly to save the SMTP address when the value of the **PidTagAddressType** property is not SMTP.

Clients MUST set the value of the **PidTagRecipientType** property ([\[MS-OXOMSG\]](#) section 2.2.3.1) for each recipient (2) as specified in the following table to indicate whether a recipient (2) is a **To recipient**, a **carbon copy (Cc) recipient**, or a **blind carbon copy (Bcc) recipient**.

PidTagRecipientType property value	Recipient header
0x00000001	To
0x00000002	Cc
0x00000003	Bcc

When generating MIME messages or TNEF messages, MIME writers SHOULD<5> ignore recipients that have a value for the **PidTagRecipientType** property that is not included in the preceding table. Alternatively, MIME writers MAY<6> perform a bitwise **AND** of the value of the **PidTagRecipientType** property and 0x00000003 and use that value to determine the recipient type. MIME writers SHOULD generate one MIME recipient (2) for a Message object recipient (2) that has a value of To, Cc, or Bcc but MAY<7> generate no MIME recipients (2) for Bcc recipients when generating a message to send via SMTP. Each MIME recipient (2) MUST be generated in the header that corresponds to the **PidTagRecipientType** property value, as specified by the **PidTagRecipientType** property value table.

Clients SHOULD set the **PidTagDisplayName** property ([\[MS-OXCFOLD\]](#) section 2.2.2.2.5) for recipients (2), where that information is available. MIME writers SHOULD copy the **PidTagDisplayName** property value, when it exists, when generating the display name of an address

as specified in [\[RFC2822\]](#). If the **PidTagDisplayName** property is not assigned, MIME writers SHOULD NOT generate a value for the display name. MIME writers can encode the display name, as specified in [\[RFC2047\]](#), in order to preserve non-ASCII characters.

MIME writers SHOULD generate the angle-address portion (angle-addr) of an address, as specified in [\[RFC2822\]](#) section 3.4, from addressee properties, used in the following order of preference:

PidTagEntryId, **PidTagAddressType** / **PidTagEmailAddress**, **PidTagSmtpAddress**. More specifically, MIME writers SHOULD do the following:

1. If the **PidTagEntryId** property is present and bytes 4-19 are equal to the MUIDEMSAB **universally unique identifier (UUID)** value "{%xdc.a7.40.c8.c0.42.10.1a.b4.b9.08.00.2b.2f.e1.82}", it is an **address book EntryID**. (The MUIDEMSAB value identifies the value as an EntryID.) In this case, MIME writers SHOULD look up the address book entry that corresponds to the **distinguished name (DN)** that is contained in the EntryID and use the **primary SMTP proxy address** that is found on the address book entry. (EntryID format is specified in [\[MS-OXCDATA\]](#) section 2.2, and the procedure for looking up address book entries is specified in [\[MS-OXOABK\]](#) section 2.1.)
2. Otherwise, if the **PidTagEntryId** property is present and bytes 4-19 are equal to the MUIDOOP UUID value "{%x81.2b.1f.a4.be.a3.10.19.9d.6e.00.dd.01.0f.54.02}", it is a one-off EntryID. The e-mail type and address are encoded in the EntryID, as specified in [\[MS-OXCDATA\]](#) section 2.2.5.1. If the e-mail type is SMTP, use this e-mail address; otherwise, continue to step 6.
3. If the **PidTagEntryId** property is present and bytes 4-19 are some value other than the values that are shown in steps 1 and 2, the MIME writer rejects the recipient (2). If MIME generation is being done for SMTP, a failure **delivery status notification (DSN)** is generated for that recipient (2). The format of a failure DSN is specified in [\[RFC3464\]](#). The corresponding Message object structure is referred to as a **non-delivery report**.
4. Otherwise, if both the **PidTagAddressType** property and the **PidTagEmailAddress** property are present and the **PidTagAddressType** property matches "SMTP", use the value of the **PidTagEmailAddress** property.
5. Otherwise, if the **PidTagSmtpAddress** property is present, use its value.
6. Otherwise, if an e-mail address and address type are present, whether obtained from the **PidTagAddressType** property and the **PidTagEmailAddress** property or from an EntryID, but the address type does not match "SMTP", the MIME writer SHOULD attempt IMCEA encapsulation of the e-mail address, as specified in section [2.1.3.1.8](#).
7. Finally, if all of the previous conditions fail, the MIME writer MUST reject the recipient (2). If MIME generation is being done for outbound SMTP, a failure DSN is generated for that recipient (2). The format of a failure DSN is specified in [\[RFC3464\]](#). The corresponding Message object structure is referred to as a non-delivery report.

A MIME writer MAY [<8>](#) use these steps in the following order, omitting step 6: 4, 5, 1, 2, 3, 7.

2.1.3.1.1.1 To and Cc Recipients

To generate a **To** or **Cc** header, clients add a recipient (2) to the Message object and set the **PidTagRecipientType** property ([\[MS-OXOMSG\]](#) section 2.2.3.1) to the value that corresponds to the individual recipient type, as specified by the table of **PidTagRecipientType** property values in section [2.1.3.1.1](#).

MIME writers map recipients (2) to the **To** or **Cc** headers as requested by clients. An exceptional situation occurs when generating MIME for an attached DSN message. A DSN message is one that has the following value for the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3).

```
; The most common values are "REPORT.IPM.Note.NDR" and "REPORT.IPM.Note.DR"
ReportMsgClass = "REPORT" 1*("." MsgClassToken) (".NDR" / ".DR")
MsgClassToken = ALPHA *(ALPHA / DIGIT)
```

In that case, MIME writers ignore the recipients (2) of the attached message and instead populate the **To** header of the attached message by using the **PidTagReceivedRepresenting** property group (section 2.6.3) of the attached message. If the properties in the **PidTagReceivedRepresenting** property group are not present, MIME writers use the **PidTagReceivedBy** property group (section 2.6.2) of the attached message.

When generating TNEF messages, MIME writers SHOULD NOT generate the **attRecipTable** attribute, as specified in [MS-OXTNEF] section 2.1.3.3.22, for the top-level message but MAY<9> generate this attribute for the top-level summary and legacy TNEF messages as specified in [MS-OXTNEF] section 2.1.3.3.22. For attached messages, MIME writers MUST copy all recipients (2), along with all their properties, into the **attRecipTable** attribute in the TNEF body part. This applies to attached DSN messages as well.

2.1.3.1.1.2 Bcc Recipients

To generate a **Bcc** header, clients add a recipient (2) to the Message object and set the value of the **PidTagRecipientType** property ([MS-OXOMSG] section 2.2.3.1) for that recipient (2) to 0x00000003.

When generating a message for outbound submission to SMTP, MIME writers MUST NOT copy Bcc recipients to the MIME **Bcc** header except for meeting and task messages. This also applies to the MIME **Bcc** header of attached messages. MIME writers MUST NOT copy Bcc recipients to the TNEF **attRecipTable** attribute, as specified in [MS-OXTNEF] section 2.1.3.3.22, for attached messages.

When generating a message for protocols such as POP3 and IMAP4, MIME writers SHOULD<10> copy Bcc recipients to the MIME **Bcc** header. This also applies to the MIME **Bcc** header of the attached messages. MIME writers SHOULD<11> copy Bcc recipients to the TNEF **recipient table** for attached messages.

2.1.3.1.2 Reply-To

To generate a **Reply-To** header, clients set the values of the **PidTagReplyRecipientEntries** property ([MS-OXOMSG] section 2.2.1.43) and the **PidTagReplyRecipientNames** property ([MS-OXOMSG] section 2.2.1.44).

When generating a MIME message, MIME writers generate a **Reply-To** header by using the **PidTagReplyRecipientEntries** and the **PidTagReplyRecipientNames** properties. MIME writers SHOULD ignore the value of the **PidTagReplyRecipientNames** property if the count of names does not match the count of entries in the **PidTagReplyRecipientEntries** property. Assuming the counts do match, each entry in the value of the **PidTagReplyRecipientNames** property maps to one display name, and each EntryID in the value of the **PidTagReplyRecipientEntries** property maps to one address, as follows:

1. If bytes 4-19 are equal to the MUIDEMSAB UUID value "{%xdc.a7.40.c8.c0.42.10.1a.b4.b9.08.00.2b.2f.e1.82}", it is an address book EntryID. (The MUIDEMSAB value identifies the value as an EntryID.) In this case, the MIME writer SHOULD look up the address book entry that corresponds to the DN that is contained in the EntryID, and use its primary SMTP proxy address.
2. If bytes 4-19 are equal to the MUIDOOP UUID value "{%x81.2b.1f.a4.be.a3.10.19.9d.6e.00.dd.01.0f.54.02}", it is a one-off EntryID. The e-mail type and address are encoded in the one-off EntryID, as specified in [MS-OXCDATA] section 2.2.5.1,

and SHOULD be extracted. If the e-mail type is SMTP, the e-mail address SHOULD be used as is; otherwise, the address MUST be IMCEA-encapsulated, as specified in section [2.1.3.1.8](#).

When generating a TNEF message, MIME writers SHOULD [<12>](#) also copy the values of the **PidTagReplyRecipientEntries** and the **PidTagReplyRecipientNames** properties to the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, in the TNEF body part.

2.1.3.1.3 From

To generate a **From** header, clients set the **PidTagSentRepresenting** property group (section [2.6.5](#)).

When generating a MIME message, MIME writers generate a **From** header by using the values of the **PidTagSentRepresenting** property group. The order of preference in that property group is as specified in section [2.1.3.1.1](#).

When generating a TNEF message, MIME writers SHOULD [<13>](#) also copy the values of the **PidTagSentRepresenting** property group to the **attSentFor** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.17, and the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, in the TNEF body part.

2.1.3.1.4 Sender

To generate a **Sender** header, clients set the value of the **PidTagSender** property group (section [2.6.4](#)).

MIME writers generate a **Sender** header by using the values of the **PidTagSender** property group. The order of preference in that property group is as specified in section [2.1.3.1.1](#). MIME writers SHOULD NOT generate the **Sender** header if the **PidTagSender** property group and the **PidTagSentRepresenting** property group (section [2.6.5](#)) represent the same user.

When generating a TNEF message, MIME writers SHOULD [<14>](#) also copy the values of the **PidTagSender** property group to the **attFrom** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.3, and the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, in the TNEF body part.

2.1.3.1.5 Return Receipt

A MIME writer generates a (nonstandard) **Return-Receipt-To** header when the **PidTagOriginatorDeliveryReportRequested** property ([\[MS-OXOMSG\]](#) section 2.2.1.20) is set to TRUE.

If the **PidTagOriginatorDeliveryReportRequested** property is set and its value is **TRUE**, MIME writers SHOULD copy the **Return-Receipt-To** header from one of the following property groups in the specified order of priority and according to the method specified in section [2.1.3.1.1](#):

- **PidTagReadReceipt** property group (section [2.6.1](#))
- **PidTagSender** property group (section [2.6.4](#))
- **PidTagSentRepresenting** property group (section [2.6.5](#))

A MIME writer MAY [<15>](#) copy the values in the **PidTagSentRepresenting** property group to the **Return-Receipt-To** header, regardless of the values of the other properties.

When generating a TNEF message, MIME writers SHOULD [<16>](#) copy the values of the **PidTagOriginatorDeliveryReportRequested** property and one of the e-mail address properties

specified in the previous list to the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, in the TNEF body part.

2.1.3.1.6 Read Receipt

To generate a **Disposition-Notification-To** header, clients set the **PidTagReadReceiptRequested** property ([\[MS-OXOMSG\]](#) section 2.2.1.29) to "TRUE" and also set the values of either the **PidTagReadReceipt** property group (section [2.6.1](#)) or the **PidTagSentRepresenting** property group (section [2.6.5](#)).

MIME writers check the **PidTagReadReceiptRequested** property value first. If the property is not set or the value is "FALSE", MIME writers MUST NOT generate the **Disposition-Notification-To** header.

If the **PidTagReadReceiptRequested** property is set and its value is "TRUE", MIME writers generate the **Disposition-Notification-To** header from the **PidTagReadReceipt** property group, if that property group is set. The order of preference in that property group is as specified in section [2.1.3.1.1](#). If the **PidTagReadReceipt** property group is not set, servers SHOULD generate the **Disposition-Notification-To** header from the **PidTagSentRepresenting** property group.

MIME writers MUST generate the **Disposition-Notification-To** header as specified in [\[RFC3798\]](#).

When generating a TNEF message, MIME writers SHOULD [<17>](#) also copy the values of the **PidTagReadReceiptRequested**, the **PidTagReadReceipt**, and the **PidTagSentRepresenting** property groups to the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, in the TNEF body part.

2.1.3.1.7 Directory Lookups

Clients SHOULD specify the primary SMTP proxy address or the address book (EX) proxy address in all address elements of a message. But clients can use any proxy address. MIME writers SHOULD perform a lookup on the proxy address in the address book directory, as specified in [\[MS-OXOABK\]](#). If a matching address book entry is found, a MIME writer SHOULD substitute its primary SMTP proxy address for the address specified by the client.

2.1.3.1.8 IMCEA Encapsulation

When no SMTP proxy address is available for an address element, servers SHOULD [<18>](#) encapsulate any other address type to produce the required SMTP address, by using the IMCEA encapsulation mechanism. The domain part of the encapsulated SMTP address SHOULD be the MIME writer's local domain or the domain of another mail server that can de-encapsulate, and deliver to, the encapsulated address.

The IMCEA encapsulation mechanism is defined for the address types listed in the following table.

Address type	Value of PidTagAddressType or related property
Address Book	EX
Facsimile	FAX
X.400	X400

The following Augmented Backus-Naur Form (ABNF) notation specifies the format of an encapsulated address:

```
Encapsulated-address = "IMCEA" address-type "-" encoded-address "@" domain
address-type = *VCHAR
domain = dot-atom-text; see [RFC2822] section 3.2.4 for the definition.
```

```

encoded-address = * (Escaped-chars/ Normal-chars)

Escaped-chars = (ESCSLASH / ESCCHARS)

; Encoded form for "/" (%x2F) is "_"
ESCSLASH = %x5F

; All OCTETS not ALPHA, DIGIT, or in "-=/"
; These are a "+" and the two hex digits of the OCTET's value.
ESCCHARS = "+" 2(HEXDIG)

; All other characters
Normal-chars= (ALPHA / DIGIT / HYPHEN / EQUALSIGN)
HYPHEN = %x2D
EQUALSIGN = %x3D

```

Encapsulated addresses MUST NOT include line breaks and therefore can require longer line lengths than those recommended by [\[RFC2822\]](#).

A MIME writer MAY [<19>](#) encode only the address part and use the **address-type** part as is. On decoding, the MIME writer MAY scan for the first hyphen ("-") after "IMCE" and use the prefix as is without parsing for any escaped characters.

A MIME writer MAY [<20>](#) build the entire string including the **address-type** part and the encapsulated address, and then encode the whole string, escaping any non-alphanumeric characters contained in the **address-type** part. On decoding, the MIME writer unescapes the entire string and scans the first nine characters for a hyphen ("-"). If a hyphen is not found, the address is not de-encapsulated. The **address-type** part is limited to eight characters. In this circumstance, only ASCII alphanumeric characters are allowed in the **address-type** part.

2.1.3.1.9 PidTagAddressType Property

The value of the **PidTagAddressType** property ([\[MS-OXOABK\]](#) section 2.2.3.13) is a string that names the messaging system that the address is destined for. It is used to assign responsibility for an e-mail address to the right transport provider. The string value provided by the **PidTagAddressType** property contains only uppercase alphabetic characters from "A" through "Z", and the numbers from "0" through "9". The value of the **PidTagAddressType** property is also used to designate the correct format for the **PidTagEmailAddress** property ([\[MS-OXOABK\]](#) section 2.2.3.14).

If a client tries to compose a message to a user whose address type is not in the server's list of known address types, the message will produce a non-delivery report unless the client itself, acting as the message transfer agent, is able to deliver the message by using an alternate transport that bypasses the server.

The following table lists the address types that are known at this time. The common address types include "EX", "SMTP", "X400", and "X500".

Messaging system	Value of the PidTagAddressType property
Microsoft Exchange	"EX"
Internet	"SMTP"
X.400 Message Handling System	"X400"
X.500 Directory Services	"X500"

2.1.3.2 Envelope Elements

Many Message object properties that map to MIME headers have string values. Unless otherwise specified, the string values are simply copied from the property to the header. When MIME writers generate MIME header values, the encoding specified in [\[RFC2047\]](#) MUST be used to encode **Unicode** characters according to the conditions specified in section 1 of that document.

Likewise, unless otherwise specified, when a MIME message with a TNEF body part is being generated, all Message object properties SHOULD [<21>](#) be copied to the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, of the TNEF body part, even if there is also a corresponding MIME header.

2.1.3.2.1 Message Class

When generating a TNEF message, MIME writers copy the value of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) to the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, in the TNEF body part. In addition, MIME writers SHOULD map the value of the **PidTagMessageClass** property to the **attMessageClass** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.5.

When generating pure MIME messages, the value of the **PidTagMessageClass** property SHOULD NOT be copied to the MIME messages. Instead, its value is reflected in the structure of the MIME message, as specified in the following table. The MIME structure is indicated by listing the value of the **Content-Type** header, indented according to how the MIME entities are nested.

PidTagMessageClass property value	MIME structure
"IPM.Note.SMIME.MultipartSigned", or begins with "IPM.InfoPathForm." and ends with ".SMIME.MultipartSigned"	Multipart/signed, as specified in [RFC5751] and [MS-OXOSMIME] .
"IPM.Note.SMIME", or begins with "IPM.InfoPathForm." and ends with ".SMIME"	Application/pkcs7-MIME, as specified in [RFC5751] and [MS-OXOSMIME] .
"REPORT.IPM.Note.DR" or "REPORT.IPM.Note.NDR" (other values MAY be substituted for "IPM.Note")	As specified in [RFC3464] : "multipart/report" "text/HTML" "message/delivery-status" <original message structure> For details, see section 2.1.3.6.1 .
"REPORT.IPM.Note.IPNRN" or "REPORT.IPM.Note.IPNNRN" (other values MAY be substituted for "IPM.Note")	As specified in [RFC3798] : "multipart/report" "text/HTML" Message/disposition-notification For details, see section 2.1.3.6.2 .
Equals "IPM.Appointment" or begins with "IPM.Appointment."	"text/calendar", as specified in [RFC2445] and [MS-OXCICAL] . <22>
Begins with "IPM.Schedule.Meeting."	Content mapped to "text/calendar", as specified in [RFC2445] and [MS-OXCICAL] . Top-level message structure is "multipart/alternative" or "multipart/mixed", depending on the presence and type of message body and attachments. For details, see section 2.1.3.3 .

PidTagMessageClass property value	MIME structure
"IPM.Note" or any other value	"text/plain", "text/HTML", "multipart/alternative", "multipart/related", as specified in [RFC2387] , or "multipart/mixed", depending on the presence and type of message body and attachments. For details, see section 2.1.3.3.

2.1.3.2.2 Content Class

MIME writers SHOULD generate the following values for a **Content-Class** header, based on the value of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3).<23>

PidTagMessageClass property value	Content-class header value
"IPM.Note.Microsoft.Fax"	"fax".
"IPM.Note.Microsoft.Fax.CA"	"fax-ca". A MIME writer MAY<24> write a value of "fax".
"IPM.Note.Microsoft.Missed.Voice"	"missedcall".
"IPM.Note.Microsoft.Conversation.Voice"	"voice-uc".
"IPM.Note.Microsoft.Voicemail.UM.CA"	"voice-ca". A MIME writer MAY<25> write a value of "voice".
"IPM.Note.Microsoft.Voicemail.UM"	"voice".

PidTagMessageClass property prefix	Content-class header value
"IPM.Note.Custom."	"urn:content-class:custom.", followed by the value of the PidTagMessageClass property with the "IPM.Note.Custom." prefix removed.
"IPM.InfoPathForm."	<p>If the PidLidInfoPathFormName property ([MS-OXPROPS] section 2.149) has some value, the Content-class header SHOULD be generated with the value of "InfoPathForm.", followed by a string, which is generated as follows:</p> <p>MIME writers SHOULD take the value of the PidTagMessageClass property and remove the "IPM.InfoPathForm." prefix.</p> <p>If the remaining string contains a '.' symbol, the value SHOULD be truncated before the period '.'.</p> <p>The value of the PidLidInfoPathFormName property SHOULD be appended to this string, preceded by a '.' character.</p>

If the MIME writer was unable to find a mapping between the value of the **PidTagMessageClass** property and a value of the **Content-class** header, it SHOULD look up the value of the **PidNameContentClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.48). If this property has a value, that value SHOULD be used as the value of the **Content-class** header; otherwise, a header SHOULD NOT be generated.

2.1.3.2.3 Unified Messaging Properties

To generate an **X-CallingTelephoneNumber** header, clients SHOULD<26> set the value of the **PidTagSenderTelephoneNumber** property ([\[MS-OXPROPS\]](#) section 2.996). They MAY instead use the **PidNameXSenderTelephoneNumber** property ([\[MS-OXPROPS\]](#) section 2.477). MIME writers

SHOULD [<27>](#) copy either property to the **X-CallingTelephoneNumber** header, preferring the **PidTagSenderTelephoneNumber** property.

To generate an **X-VoiceMessageDuration** header, clients SHOULD set the value of the **PidTagVoiceMessageDuration** property ([MS-OXPROPS] section 2.1048). They MAY instead use the **PidNameXVoiceMessageDuration** property ([MS-OXPROPS] section 2.494). MIME writers SHOULD [<28>](#) map either property to the **X-VoiceMessageDuration** header, preferring the **PidTagVoiceMessageDuration** property. The value of the **PidTagVoiceMessageDuration** property is a positive valued **PtypInteger32** ([MS-OXCADATA] section 2.11.1) and is formatted as a decimal string in the header without sign or separator characters.

To generate an **X-VoiceMessageSenderName** header, clients SHOULD set the value of the **PidTagVoiceMessageSenderName** property ([MS-OXPROPS] section 2.1049). They MAY instead use the **PidNameXVoiceMessageSenderName** property ([MS-OXPROPS] section 2.495). MIME writers SHOULD [<29>](#) copy either property to the **X-VoiceMessageSenderName** header, preferring the **PidTagVoiceMessageSenderName** property.

To generate an **X-FaxNumberOfPages** header, clients SHOULD set the value of the **PidTagFaxNumberOfPages** property ([MS-OXPROPS] section 2.686). They MAY instead use the **PidNameXFaxNumberOfPages** property ([MS-OXPROPS] section 2.475). MIME writers SHOULD [<30>](#) map either property to the **X-FaxNumberOfPages** header, preferring the **PidTagFaxNumberOfPages** property. The value of the **PidTagFaxNumberOfPages** property is a positive valued **PtypInteger32** and is formatted as a decimal string in the header without sign or separator characters.

To generate an **X-AttachmentOrder** header, clients SHOULD set the value of the **PidTagVoiceMessageAttachmentOrder** property ([MS-OXPROPS] section 2.1047). They MAY instead use the **PidNameXVoiceMessageAttachmentOrder** property ([MS-OXPROPS] section 2.493). MIME writers SHOULD [<31>](#) copy either property to the **X-AttachmentOrder** header, preferring the **PidTagVoiceMessageAttachmentOrder** property.

To generate an **X-CallID** header, clients SHOULD set the value of the **PidTagCallId** property ([MS-OXPROPS] section 2.619). They MAY instead use the **PidNameXCallId** property ([MS-OXPROPS] section 2.474). MIME writers SHOULD [<32>](#) copy either property to the **X-CallID** header, preferring the **PidTagCallId** property.

2.1.3.2.4 Arbitrary MIME Headers

To generate an arbitrary header on a MIME message, a client creates a **named property** in the **PS_INTERNET_HEADERS property set**, with the property name equal to the header name and the data type equal to the **PtypString** data type ([MS-OXCADATA] section 2.11.1). The value of this property is set to the MIME header value.

MIME writers use the name and value of such a property to create a header on the generated MIME message with the corresponding name and value. MIME writers encode the header value as specified in [\[RFC2047\]](#), according to the conditions specified in section 1 of that document. [<33>](#) MIME writers MUST NOT create such a header if a different Message object property is already mapped to the same header (2). MIME writers SHOULD NOT [<34>](#) create such a header if the header name begins with one of the reserved name prefixes "X-Microsoft-Exchange-Organization" or "X-Microsoft-Exchange-Forest".

2.1.3.2.5 Importance

To generate an **Importance** header, a client sets the value of the **PidTagImportance** property ([MS-OXCMSG] section 2.2.1.11) as specified in the following table.

PidTagImportance property value	Importance header value
0x00000000	Low
0x00000001	Normal
0x00000002	High

MIME writers MUST map the value of the **PidTagImportance** property to the **Importance** header, as specified in the table. MIME writers MAY omit the **Importance** header for a **PidTagImportance** value of 1 (normal) or for values other than 0, 1, or 2.

When generating a TNEF message, MIME writers also map the value of the **PidTagImportance** property to the **attPriority** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.10, and the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21.

2.1.3.2.6 Sensitivity

To generate a **Sensitivity** header, a client sets the value of the **PidTagSensitivity** property ([\[MS-OXCMMSG\]](#) section 2.2.1.13) as specified in the following table.

PidTagSensitivity property value	Sensitivity header value
0x00000000	Normal
0x00000001	Personal
0x00000002	Private
0x00000003	Company-Confidential

MIME writers MUST map the value of the **PidTagSensitivity** property to the **Sensitivity** header, as specified in the table. MIME writers SHOULD omit the **Sensitivity** header value when the value of the **PidTagSensitivity** property is 0, which is the default. MIME writers SHOULD also omit the **Sensitivity** header value when the value of the **PidTagSensitivity** property is any value less than 0 or greater than 3.

When generating a TNEF message, MIME writers also copy the value of the **PidTagSensitivity** property to the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21.

2.1.3.2.7 Sent Time

To generate a **Date** header, clients set the value of the **PidTagClientSubmitTime** property ([\[MS-OXOMSG\]](#) section 2.2.3.11). The property value is expressed in **Coordinated Universal Time (UTC)**.

MIME writers copy the value of the **PidTagClientSubmitTime** property to the **Date** header, formatting it as specified by [\[RFC2822\]](#) section 3.3. MIME writers SHOULD include hours, minutes, and seconds in the generated **Date** header value. MIME writers MAY convert the date and time value from UTC to another time zone of their choice.

If no value is specified for the **PidTagClientSubmitTime** property when a message is submitted to SMTP, MIME writers SHOULD generate a **Date** header with a value of the current time.

When generating a TNEF message, if the **PidTagClientSubmitTime** property is present on the Message object, MIME writers SHOULD [<35>](#) copy the value of the **PidTagClientSubmitTime** property to the **attDateSent** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.4, and the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21.

2.1.3.2.8 Subject

To generate a **Subject** header, clients SHOULD set the **PidTagSubjectPrefix** property ([MS-OXOMSG] section 2.2.1.60) and the **PidTagNormalizedSubject** property ([MS-OXCMSG] section 2.2.1.10) on the Message object. Clients MAY set the **PidTagSubject** property ([MS-OXPROPS] section 2.1021) instead, but in that case, the separation of the subject from the subject prefix is vulnerable to limitations of the server's parsing procedure, as specified in section 2.2.3.2.6.1. The values of the **PidTagSubject** and **PidTagNormalizedSubject** properties SHOULD NOT contain line breaks.

MIME writers SHOULD generate the **Subject** header by combining the values of the **PidTagSubjectPrefix** and **PidTagNormalizedSubject** properties. MIME writers MAY <36> instead copy the value of the **PidTagSubject** property ([MS-OXCMSG] section 2.2.1.46) to the **Subject** header.

If those two properties are not available, MIME writers MUST copy the value of the **PidTagSubject** property to the **Subject** header. MIME writers MAY truncate the subject value. The property value SHOULD NOT be truncated in the middle of a multiple-byte character.

When generating a TNEF message, MIME writers SHOULD <37> also copy the message subject (however it is obtained) to the **attSubject** attribute, as specified in [MS-OXTNEF] section 2.1.3.3.7, and the **attMsgProps** attribute, as specified in [MS-OXTNEF] section 2.1.3.3.21. MIME writers SHOULD also copy the **PidTagSubjectPrefix** property, with its value, to the **attMsgProps** attribute.

2.1.3.2.9 Conversation Topic

To generate a **Thread-Topic** header, clients set the value of the **PidTagConversationTopic** property ([MS-OXOMSG] section 2.2.1.5). Clients SHOULD set this property value to the same value as the **PidTagNormalizedSubject** property ([MS-OXCMSG] section 2.2.1.10) has, with any subject prefix removed, as specified in section 2.2.3.2.6.1.

MIME writers copy the value of the **PidTagConversationTopic** property to the **Thread-Topic** header.

2.1.3.2.10 Conversation Index

To generate a **Thread-Index** header, clients set the value of the **PidTagConversationIndex** property ([MS-OXOCFG] section 2.2.8.8).

MIME writers copy the value of the **PidTagConversationIndex** property to the **Thread-Index** header. The **property type** is binary; the value is encoded using **base64 encoding**, as specified in [RFC2045].

2.1.3.2.11 Message ID

MIME writers SHOULD copy the value of the **PidTagInternetMessageId** property ([MS-OXOMSG] section 2.2.1.13) to the **Message-ID** header. A MIME writer MAY <38> generate a new **Message-ID** header if it is generating a message to send via SMTP. If no value is specified for the **PidTagInternetMessageId** property when a message is submitted to SMTP, MIME writers SHOULD generate a value as specified in [RFC2822].

Clients SHOULD NOT set the **PidTagInternetMessageId** property when submitting a message via **remote procedure call (RPC)**. As specified in [RFC2822], the value of the **Message-ID** header is unique and is assigned by the server that originated the message. Servers MAY overwrite the **PidTagInternetMessageId** property from a client before submitting the message to SMTP.

Once set, the value of the **Message-ID** header and the corresponding value of the **PidTagInternetMessageId** property SHOULD remain constant. MIME writers SHOULD NOT overwrite

the value of the **PidTagInternetMessageId** property when generating MIME messages for protocols such as POP3 or IMAP4.

2.1.3.2.12 References

To generate a **References** header, clients set the value of the **PidTagInternetReferences** property ([\[MS-OXCMSG\]](#) section 2.2.1.26).

MIME writers copy the value of the **PidTagInternetReferences** property to the **References** header.

2.1.3.2.13 Categories

To generate a **Keywords** header, clients set the value of the **PidNameKeywords** property ([\[MS-OXCMSG\]](#) section 2.2.1.17). The type of the **PidNameKeywords** property is multiple strings; each category SHOULD be mapped to a single keyword.

MIME writers SHOULD copy each subvalue of the **PidNameKeywords** property to a separate keyword in the **Keywords** header, with a comma (U+002C) and space (U+0020) separating each keyword. To avoid conflict among different sets of categories in different organizations, MIME writers can drop the **PidNameKeywords** property instead of copying it to the **Keywords** header.

2.1.3.2.14 In-Reply-To Message ID

To generate an **In-Reply-To** header, clients set the value of the **PidTagInReplyToId** property ([\[MS-OXOMSG\]](#) section 2.2.1.13).

MIME writers SHOULD [<39>](#) copy the value of the **PidTagInReplyToId** property ([MS-OXOMSG] section 2.2.1.13) to the **In-Reply-To** header.

2.1.3.2.15 List Server Properties

To generate a **List-Help** header, clients set the value of the **PidTagListHelp** property ([\[MS-OXPROPS\]](#) section 2.758).

MIME writers copy the value of the **PidTagListHelp** property to the **List-Help** header.

To generate a **List-Subscribe** header, clients set the value of the **PidTagListSubscribe** property ([MS-OXPROPS] section 2.759).

MIME writers copy the value of the **PidTagListSubscribe** property to the **List-Subscribe** header.

To generate a **List-Unsubscribe** header, clients set the value of the **PidTagListUnsubscribe** property ([MS-OXPROPS] section 2.760).

MIME writers copy the value of the **PidTagListUnsubscribe** property to the **List-Unsubscribe** header.

The **List-Help**, **List-Subscribe**, and **List-Unsubscribe** headers are specified in [\[RFC2369\]](#).

2.1.3.2.16 Language Properties

To generate an **Accept-Language** header, as specified in [\[RFC3282\]](#), clients set the value of the **PidNameAcceptLanguage** property ([\[MS-OXCMSG\]](#) section 2.2.1.42).

MIME writers SHOULD [<40>](#) copy the value of the **PidNameAcceptLanguage** property to the **Accept-Language** header. If the **PidNameAcceptLanguage** property is missing, MIME writers should check the value of the **PidTagMessageFlags** property ([MS-OXCMSG] section 2.2.1.6). If the **mfSubmitted** flag is set, MIME writers SHOULD generate a value for the **Accept-Language** header

by identifying the acceptable **locales** of the sender's mailbox and determining the corresponding language tag, as specified by [\[RFC4646\]](#). If the **mfSubmitted** flag is not set, MIME writers MAY [<41>](#) generate a value for the **Accept-Language** header.

To generate a **Content-Language** header, as specified in [\[RFC3282\]](#), clients SHOULD [<42>](#) set the value of the **PidTagMessageLocaleId** property ([\[MS-OXCMSG\]](#) section 2.2.1.5) to a locale ID.

MIME writers use the value of the **PidTagMessageLocaleId** property to write the **Content-Language** header. The value of **PidTagMessageLocaleId** is a **language code identifier (LCID)**, but the header value is a language tag, as specified by [\[RFC4646\]](#). The mapping between the LCID value and the language tag is specified in [\[MS-LCID\]](#).

2.1.3.2.17 Classification Properties

To generate headers related to message classification, clients both set the value of the **PidLidClassified** property ([\[MS-OXCMSG\]](#) section 2.2.1.25) to TRUE and set the values of the following properties: **PidLidClassification** ([\[MS-OXCMSG\]](#) section 2.2.1.23), **PidLidClassificationDescription** ([\[MS-OXCMSG\]](#) section 2.2.1.24), **PidLidClassificationGuid** ([\[MS-OXPROPS\]](#) section 2.54), and **PidLidClassificationKeep** ([\[MS-OXPROPS\]](#) section 2.55).

When the value of the **PidLidClassified** property is TRUE, MIME writers SHOULD [<43>](#) copy all classification property values to their corresponding headers, as specified in the following table. If the value of **PidLidClassified** is FALSE, no value is written for any of the five headers listed in the following table.

Classification property name	Classification header name	Property value mapping
PidLidClassified	X-Microsoft-Classified	TRUE maps to "true". If the value of PidLidClassified is FALSE, then there is no header.
PidLidClassificationKeep	X-Microsoft-ClassKeep	TRUE maps to "true". FALSE maps to no header.
PidLidClassification	X-Microsoft-Classification	No mapping. The string value is copied directly.
PidLidClassificationDescription	X-Microsoft-ClassDesc	No mapping. The string value is copied directly.
PidLidClassificationGuid	X-Microsoft-ClassID	No mapping. The string value is copied directly.

2.1.3.2.18 Payload Properties

To generate an **X-Payload-Provider-GUID** header, clients SHOULD set the value of the **PidTagAttachPayloadProviderGuidString** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) for messages and message attachments. [<44>](#) Clients MAY [<45>](#) generate this header for file attachments.

MIME writers SHOULD [<46>](#) copy the value of the **PidTagAttachPayloadProviderGuidString** property to the **X-Payload-Provider-GUID** header. MIME writers MAY [<47>](#) set this value as a property of the attachment or as a property of the message.

To generate an **X-Payload-Class** header, clients set the value of the **PidTagAttachPayloadClass** property ([\[MS-OXCMSG\]](#) section 2.2.2.26).

MIME writers SHOULD<48> copy the value of the **PidTagAttachPayloadClass** property to the **X-Payload-Class** header.

2.1.3.2.19 Has Attach

To generate an **X-MS-Has-Attach** header, clients MUST add at least one **Attachment object** to the **attachments table** of the Message object.

When the Message object's attachments table contains at least one Attachment object, MIME writers SHOULD generate an **X-MS-Has-Attach** header with a value of "Yes". MIME writers MAY<49> emit this header with no output. MIME writers MAY<50> omit this header if it was not included in the original message. When the Message object's attachments table is empty, MIME writers generate an **X-MS-Has-Attach** header with no value.

2.1.3.2.20 Auto Response Suppress

To generate an **X-Auto-Response-Suppress** header, clients set the value of the **PidTagAutoResponseSuppress** property ([\[MS-OXPROPS\]](#) section 2.606).

When the **PidTagAutoResponseSuppress** property has a value of 0 (zero) or -1, MIME writers SHOULD<51> map its value to the **X-Auto-Response-Suppress** header as shown in the following table.

PidTagAutoResponseSuppress property value	X-Auto-Response-Suppress header value
0	None
-1	All

When the **PidTagAutoResponseSuppress** property has a value other than 0 (zero) or -1, MIME writers MUST construct the value of the **X-Auto-Response-Suppress** header as follows: For each bit of the value of **PidTagAutoResponseSuppress** that is set (left-hand column), append the string in the center column to the header value. If the header value was nonempty, append a comma (U+002C) and space (U+0020) before the new value.

PidTagAutoResponseSuppress property value	X-Auto-Response-Suppress header value	Meaning
0x00000001	DR	Suppress delivery reports from transport.
0x00000002	NDR	Suppress non-delivery reports from transport.
0x00000004	RN	Suppress read notifications from receiving client.
0x00000008	NRN	Suppress non-read notifications from receiving client.
0x00000010	OOF	Suppress Out of Office (OOF) notifications.
0x00000020	AutoReply	Suppress auto-reply messages other than OOF notifications.

For example, if the value of the **PidTagAutoResponseSuppress** property is 0x000C, the header MUST be written as follows:

X-Auto-Response-Suppress: RN, NRN

The order of these values in the header is not important.

2.1.3.2.21 Is Auto Forwarded

To generate an **X-MS-Exchange-Organization-AutoForwarded** header, clients set the value of the **PidTagAutoForwarded** property ([MS-OXCMMSG] section 2.2.1.20) to TRUE.

If the value of the **PidTagAutoForwarded** property is TRUE, MIME writers SHOULD<52> generate the following header:

```
X-MS-Exchange-Organization-AutoForwarded: true
```

If the property is absent or the property value is false, a header SHOULD NOT be generated.

2.1.3.2.22 Sender Id Status

To generate an **X-MS-Exchange-Organization-SenderIdResult** header, clients set the value of the **PidTagSenderIdStatus** property ([MS-OXPROPS] section 2.992).

MIME writers SHOULD<53> copy the value of the **PidTagSenderIdStatus** property, which is a **PtypInteger32** ([MS-OXCDATA] section 2.11.1), to the **X-MS-Exchange-Organization-SenderIdResult** header, formatting it as a string, without separator characters.

2.1.3.2.23 Purported Sender Domain

To generate an **X-MS-Exchange-Organization-PRD** header, clients set the value of the **PidTagPurportedSenderDomain** property ([MS-OXCMMSG] section 2.2.1.43).

MIME writers SHOULD<54> copy the value of the **PidTagPurportedSenderDomain** property to the **X-MS-Exchange-Organization-PRD** header.

2.1.3.2.24 Spam Confidence Level

To generate an **X-MS-Exchange-Organization-SCL** header, clients set the value of the **PidTagContentFilterSpamConfidenceLevel** property ([MS-OXCSPAM] section 2.2.1.3) to a value in the range -1 to 10. The value of -1 indicates the message is from a trusted sender and is never treated as **spam**. Values 0 through 10 indicate the confidence levels calculated from the message content, as specified in [MS-OXCSPAM].

MIME writers SHOULD<55> copy the value of the **PidTagContentFilterSpamConfidenceLevel** property, which is a **PtypInteger32** ([MS-OXCDATA] section 2.11.1), to the **X-MS-Exchange-Organization-SCL** header, formatting it as a decimal numeric string without separator characters.

2.1.3.2.25 Flag Request

To generate an **X-Message-Flag** header, clients set the value of the **PidLidFlagRequest** property ([MS-OXCSPAM] section 2.2.1.3).

MIME writers copy the value of the **PidLidFlagRequest** property to the **X-Message-Flag** header.

2.1.3.2.26 TNEF Correlation Key

When creating a new TNEF message, MIME writers choose a unique key relating the TNEF body part to its parent message. (MIME writers SHOULD use the value of the **PidTagInternetMessageId** property ([\[MS-OXOMSG\]](#) section 2.2.1.12) for this purpose.) The chosen value SHOULD [<56>](#) be written in the following two places:

- As the value of the **X-MS-TNEF-Correlator** header on the MIME message.
- As the value of the **PidTagTnefCorrelationKey** property ([\[MS-OXCMSG\]](#) section 2.2.1.29) in the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, of the TNEF body part itself.

This pair of values SHOULD be used by MIME writers to validate that the top-level message and its TNEF body part do, in fact, belong to each other and are not (for example) the result of a non-TNEF-aware **Mail User Agent (MUA)** forwarding a message with an attached TNEF body part and retaining the attachment.

2.1.3.2.27 Received Headers

If a MIME message is bound for SMTP, MIME writers MUST NOT copy **Received** headers from the **PidTagTransportMessageHeaders** property ([\[MS-OXPROPS\]](#) section 2.1038) to the generated MIME header. If a MIME message is bound for POP3 or IMAP4, MIME writers SHOULD copy all **Received** headers from the **PidTagTransportMessageHeaders** property to the generated MIME header.

2.1.3.2.28 ReplyBy Time

To generate a **Reply-By** header, clients set the **PidTagReplyTime** property ([\[MS-OXOFLAG\]](#) section 2.2.3.1) to the date and time by which a reply is requested. Clients also MUST set the **PidLidFlagRequest** property ([\[MS-OXOFLAG\]](#) section 2.2.1.9), as specified in section [2.1.3.2.25](#), to any non-empty string value. If the **PidLidFlagRequest** property is not set, the **Reply-By** header will not be generated.

MIME writers SHOULD [<57>](#) copy the value of the **PidTagReplyTime** property to the **Reply-By** header.

2.1.3.2.29 Content-ID

To generate the **Content-ID** header, clients set the value of the **PidTagBodyContentId** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.7) [<58>](#). For guidance about how to format the property value, please refer to [\[RFC2392\]](#)

2.1.3.2.30 Content-Location

To generate the **Content-Location** header, clients set the value of the **PidTagBodyContentLocation** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.8). [<59>](#) For guidance about how to format the property value, please refer to [\[RFC2557\]](#)

2.1.3.2.31 XRef

A MIME writer SHOULD [<60>](#) generate an **XRef** header when it detects the **PidNameCrossReference** property (section [2.5.3](#)). The value of this property becomes the header value.

2.1.3.3 Body Text

When generating a pure MIME message, MIME writers generate a single MIME entity for the message body, and it MUST be the first entity generated. (For Message objects without attachments, it SHOULD

be the only MIME entity generated.) The MIME entity generated for the message body can have several different structures, some of them fairly complex.

For diagrams of the message structure with attachments, and for details about how to determine whether an Attachment object represents an inline attached file, see section [2.1.3.4](#).

2.1.3.3.1 Client Actions

To create a plain text message body in MIME format, clients SHOULD set the value of the **PidTagBody** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.1). Additionally, clients SHOULD set the value of the **PidTagInternetCodepage** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.6) to a **code page** that corresponds to the character set that the client wants to appear in the MIME message. If the client does not set the **PidTagInternetCodepage** property for a plain text message body, the server SHOULD default this property to "ISO-8859-15", but MAY [<61>](#) default this property to "ISO-8859-1". Clients SHOULD NOT create inline Attachment objects when the **best body** format of the Message object is plain text.

To create an HTML message body in MIME format, clients SHOULD set the value of the **PidTagHtml** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.9) to the HTML message text. When this property is set, clients MUST set the value of the **PidTagInternetCodepage** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.6) to the code page of the HTML message text. Note that the **PidTagHtml** property is a **PtypBinary** property ([\[MS-OXCDATA\]](#) section 2.11.1), not a **PtypString** property. Clients can, instead, set the value of the **PidTagRtfCompressed** property ([\[MS-OXPROPS\]](#) section 2.930) to the body text in compressed RTF format, depending on the MIME writer that is used to convert this text to HTML format. Clients MUST NOT create HTML message text using Unicode characters (**UTF-16LE** scheme), and the value of the **PidTagInternetCodepage** property MUST NOT be set to "1200". The UTF-32 form and the UTF-16GE scheme are also not acceptable for this purpose; UTF-7 (code page 65000) and the UTF-8 form (code page 65001) are acceptable.

To create a multipart/related message body in MIME format with HTML body text and inline images, clients SHOULD set the value of the **PidTagHtml** property to the HTML message text. When this property is set, the value of the **PidTagInternetCodepage** property is set to the code page of the HTML message text. Clients supply a value for either the **PidTagAttachContentId** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) or the **PidTagAttachContentLocation** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) on related file attachments such as images; the **PidTagAttachContentId** property SHOULD be chosen for this purpose. Depending on the choice of attachment property, inline image links in the HTML body MUST use one of the following:

1. The "cid:" **Uniform Resource Identifier (URI)** scheme and a unique content identifier that matches the value of the **PidTagAttachContentId** property on the corresponding Attachment object.
2. A copy of the value of the **PidTagAttachContentLocation** property on the corresponding Attachment object.

Instead of setting the value of the **PidTagHtml** property, clients can set the value of the **PidTagRtfCompressed** property and include **Object Linking and Embedding (OLE)** attachments, depending on the server, to convert the RTF text to HTML and the static renderings of the OLE attachments to image attachments. For details, see section [2.1.3.3.7](#).

For plain text messages, clients SHOULD write the value of the **PidTagBody** property in Unicode characters and SHOULD set the value of the **PidTagInternetCodepage** property to the code page that matches the sender's preferred character set. When generating a MIME element for the plain text body, MIME writers map this code page to a character set name, convert the Unicode text into that character set, and write that character set name to the value of the character set parameter of the **Content-Type** header. The plain text MIME element generated for a TNEF message SHOULD be treated in the same way.

For HTML messages, clients SHOULD write the value of the **PidTagHtml** property by using text in the sender's preferred character set. Clients set the value of the **PidTagInternetCodepage** property to the code page that corresponds to the preferred character set. Clients MUST NOT use UTF-16 form (code page 1200) as the preferred character set. If the HTML document contains a content-type meta tag, its *charset* parameter value SHOULD match the preferred character set.

When generating a MIME element or elements for an HTML message body, MIME writers map the value of the **PidTagInternetCodepage** property to a character set name, write the MIME element body in that character set, and write that character set name as the value of the **Content-Type** header's *charset* parameter, as specified in section 5.1 of [\[RFC2045\]](#). If the HTML document contains a content-type meta tag, its *charset* parameter value SHOULD match the **Content-Type** header's character set parameter value.

For RTF messages, clients SHOULD write the value of the **PidTagRtfCompressed** property by using text in the sender's preferred character set. Clients SHOULD set the value of the **PidTagInternetCodepage** property to the code page that corresponds to the preferred character set. The preferred character set MUST NOT be the UTF-16 form (code page 1200). MIME writers MUST NOT rely on the value of the **PidTagInternetCodepage** property but treat it as a preference; MIME writers SHOULD instead rely on the value of one or more **\ansicpg** control words in the RTF stream (2), as described in [\[MSFT-RTF\]](#), to determine the actual body code page.

When generating a MIME element or elements for an RTF message body, MIME writers SHOULD convert the RTF text to plain text or HTML. MIME writers MAY [<62>](#) instead generate a TNEF attachment that contains the RTF body. MIME writers also SHOULD map the body code page to a character set name, SHOULD write the MIME element body in that character set, and SHOULD write that character set name as the value of the **Content-Type** header's *charset* parameter.

Even if a Message object has no body, clients SHOULD set the value of the **PidTagInternetCodepage** property to indicate a preferred character set for header text, to be used in encoding as specified in [\[RFC2047\]](#).

When generating headers for a MIME entity, it can be necessary to encode the characters as specified in [\[RFC2047\]](#). MIME writers SHOULD use the same character set for all headers and the message body. MIME writers MAY [<63>](#) re-encode the message body with the same character set as the MIME message headers. MIME writers MAY [<64>](#) preserve the character set of the original MIME message body. If a different character set is supplied for different body types, MIME writers MAY [<65>](#) identify the body format that can generate the others and apply its character set to all body formats. Attachments that are themselves messages are independent and can have a different character set.

2.1.3.3.2 Message Body in TNEF

When generating a TNEF message, MIME writers SHOULD identify the best body property of the Message object, as specified in [\[MS-OXBBODY\]](#), and copy its value to the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, of the TNEF body part. MIME writers also place a plain text version of the message body in the first child body part of the TNEF message, as specified in section [2](#), generating plain text from the value of the best body property if the best body format is other than plain text. Finally, when the best body is plain text, MIME writers SHOULD [<66>](#) also write a matching value for the **PidTagRtfCompressed** property ([\[MS-OXPROPS\]](#) section 2.930) to the **attMsgProps** attribute of the TNEF body part.

2.1.3.3.3 Simple Plain Text Message Body

When the best body format type is plain text, MIME writers SHOULD generate a single MIME entity with the value of its **Content-Type** header set to "text/plain".

The *charset* parameter value of this MIME entity's **Content-Type** header SHOULD be set to a character set that corresponds to the value of the **PidTagInternetCodepage** property ([\[MS-OXCMMSG\]](#) section 2.2.1.56.6), which is a code page number. If there is no **PidTagInternetCodepage**

property, the value of the **PidTagMessageCodepage** property ([MS-OXCMSG] section 2.2.1.56.6) can be used instead, but in that case the message code page SHOULD first be mapped to the corresponding Internet code page. MIME writers SHOULD verify that the plain text, which is stored in UTF-16 form, can actually be encoded in this character set and SHOULD, if necessary, choose a different character set that can in fact encode the entire message body; the code page properties express a preference rather than a requirement.

The value of the **PidTagBody** property ([MS-OXCMSG] section 2.2.1.56.1) is written to the content of the "text/plain" MIME element, after being converted to the chosen character set.

2.1.3.3.4 HTML Text Message Body Without Inline Attachments

When the best body format type is HTML and no inline Attachment objects exist, MIME writers SHOULD generate a MIME entity that has "multipart/alternative" for the value of its **Content-Type** header, and that has the following two child entities:

1. The first child entity has "text/plain" for the value of its **Content-Type** header. Its body SHOULD be plain text generated from the value of the **PidTagHtml** property ([MS-OXCMSG] section 2.2.1.56.9). MIME writers that rely on the store for text conversions MAY [instead](#) copy the body from the value of the **PidTagBody** property ([MS-OXCMSG] section 2.2.1.56.1), assuming that the value of the **PidTagBody** property is equal to the value of the **PidTagHtml** property [%5BMS-OXPROPS%5D.pdf](#).
2. The second child entity has "text/HTML" for the value of its **Content-Type** header. Its body is the value of the **PidTagHtml** property.

The plain text and *charset* parameters SHOULD be processed as specified in section [2.1.3.3.3](#). HTML text MAY be processed in exactly the same way, or characters that do not fit the preferred character set can instead be encoded within the HTML.

2.1.3.3.5 HTML Text Message Body from RTF Without Inline Attachments

When the best body format type is RTF and no inline (OLE) Attachment objects exist, MIME writers SHOULD generate a multipart/alternative MIME entity with the following two child entities:

1. The first child entity has "text/plain" for the value of its **Content-Type** header. Its body SHOULD be plain text generated from the value of the **PidTagRtfCompressed** property ([MS-OXCMSG] section 2.2.1.56.4). MIME writers that rely on the store for text conversion MAY instead copy the body from the value of the **PidTagBody** property ([MS-OXCMSG] section 2.2.1.56.1), assuming that both properties contain substantially similar text. The format of the RTF data is specified in [\[MSFT-RTF\]](#).
2. The second child entity has "text/HTML" for the value of its **Content-Type** header. Its body is HTML text. The HTML text SHOULD be generated from the value of the **PidTagRtfCompressed** property ([\[MS-OXPROPS\]](#) section 2.930). MIME writers that rely on the store for text conversion MAY instead copy the body from the value of the **PidTagHtml** property ([MS-OXCMSG] section 2.2.1.56.9), assuming that both properties contain substantially similar text.

MIME content-types of type "text/" and the *charset* parameter SHOULD be processed as specified in section [2.1.3.3.4](#).

2.1.3.3.6 HTML Text Message Body with Inline Attachments

When the best body format type is HTML and inline Attachment objects exist, MIME writers SHOULD generate a MIME entity with "multipart/related" for the value of its **Content-Type** header and two or more child elements, as follows:

1. The first child entity is a structure with a MIME content-type of "multipart/alternative", as specified in section [2.1.3.3.4](#).
2. Subsequent child entities are generated from the Message object's inline Attachment objects. A child entity MUST be generated if and only if the Attachment object is marked as specified in section [2.1.3.4.1](#).

MIME writers SHOULD verify that the HTML text actually contains a reference to each inline Attachment object, either by its **PidTagAttachContentId** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) or its **PidTagAttachContentLocation** property ([\[MS-OXCMSG\]](#) section 2.2.2.26), as specified in [2.1.3.3.1](#). If the HTML text contains no such reference, the MIME writer SHOULD [<68>](#) consider this Attachment object as not inline and generate its MIME entity as a peer of the MIME entity with a MIME content-type of " multipart/related" instead of as its child.

2.1.3.3.7 HTML Text Message Body from RTF with Inline (OLE) Attachments

When the best body format type is RTF and inline (OLE) Attachment objects exist, MIME writers SHOULD generate a MIME entity with multipart/related for the value of its **Content-Type** header and three or more child entities, as follows:

1. The first child entity is a structure with a MIME content-type of "multipart/alternative", as specified in section [2.1.3.3.5](#).
2. Subsequent child entities are generated from the Message object's inline Attachment objects. Each entity is generated as specified in [2.1.3.4.4](#), because inline Attachment objects in RTF messages are OLE attachments.

2.1.3.3.8 Calendar Items and Meeting Messages

A Message object is a calendar item when the value of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) starts with "IPM.Appointment." or equals "IPM.Appointment". A Message object is a meeting message when the value of the **PidTagMessageClass** property starts with "IPM.Schedule.Meeting.". Clients SHOULD create items of these types with a best body format type of RTF. Clients can use a plain text body instead but SHOULD NOT create calendar items or meeting messages with a best body format type of HTML.

Each of the leaf MIME entities specified in this section SHOULD use the UTF-8 form as its character set, as specified in [\[RFC2445\]](#). A MIME writer MAY [<69>](#) enable a user or a system administrator to select a character set encoding other than the UTF-8 form for the MIME headers and MIME entities generated from the properties of a Message object. The iCalendar message body, however, remains encoded in the UTF-8 form.

2.1.3.3.8.1 Plain Text Calendar Message

When the best body format type of a calendar item or meeting message is plain text, MIME writers SHOULD generate a MIME entity with multipart/alternative for the value of its **Content-Type** header and two child entities, as follows:

1. The first child entity has "text/plain" for the value of its **Content-Type** header, and its content is copied from the **PidTagBody** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.1).
2. The second child entity has "text/calendar" for the value of its **Content-Type** header, and its content is generated as specified in [\[MS-OXCICAL\]](#).

2.1.3.3.8.2 Calendar Message Without Inline Attachments

When the best body format type of a calendar item or meeting message is RTF and there are no inline attachments, MIME writers SHOULD generate a MIME entity with "multipart/alternative" for the value of its **Content-Type** header and three child entities, as follows:

1. The first child entity has "text/plain" for the value of its **Content-Type** header. Its content SHOULD be plain text generated from the value of the **PidTagRtfCompressed** property ([\[MS-OXPROPS\]](#) section 2.930) but can instead be copied from the value of the **PidTagBody** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.1), assuming that the values of the two properties are equal.
2. The second child entity has "text/HTML" for the value of its **Content-Type** header. Its content SHOULD be HTML text generated from the value of the **PidTagRtfCompressed** property but MAY<70> instead be copied from the value of the **PidTagHtml** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.9), assuming that the values of the two properties are equal.
3. The third child entity has "text/calendar" for the value of its **Content-Type** header, and its content SHOULD be generated as specified in [\[MS-OXCICAL\]](#). A MIME writer MAY<71> use a **Content-Transfer-Encoding** header value of "8bit" for the "text/calendar" MIME content-type, or a MIME writer MAY<72> use a **Content-Transfer-Encoding** header value of "base64". Both conform to MIME, and both are consistent with the description in this section.

A MIME writer MAY<73> use a "quoted-printable" value for the **Content-Transfer-Encoding** header, or a MIME writer MAY<74> use a **Content-Transfer-Encoding** value of "base64". Both encodings produce conformant MIME entities, and both encodings are consistent with the description in this section.

2.1.3.3.8.3 Calendar Message with Inline Attachments

When the best body format type of a calendar item or meeting message is RTF and there are inline attachments, MIME writers SHOULD generate a MIME entity with "multipart/related" for the value of its **Content-Type** header and two or more child entities, as follows:

1. The first child entity is a "multipart/alternative" MIME content-type structure generated as specified in section [2.1.3.3.8.2](#).
2. Subsequent child entities are generated from the Message object's inline Attachment objects. Each entity MUST be generated as specified in section [2.1.3.4.1](#).

A MIME writer MAY<75> generate a MIME entity that has "multipart/mixed" for the value of its **Content-Type** header.

2.1.3.3.9 Enriched Text Message Body

MIME writers MAY<76> generate a MIME entity with "text/enriched" for the value of its **Content-Type** header. In that case, the MIME writer SHOULD convert the best body text format to the enriched text format, as specified in [\[RFC1896\]](#).

2.1.3.4 Attachments

Each Attachment object in a Message object represents one attachment. MIME writers SHOULD classify Attachment objects based on the value of the **PidTagAttachMethod** property ([\[MS-OXCMSG\]](#) section 2.2.2.9), as specified in the following table.

PidTagAttachMethod property value	Attachment object classification
5	Embedded Message object attachments
6	OLE attachments

PidTagAttachMethod property value	Attachment object classification
All other values	Ordinary file attachments

Note that ordinary file attachments can contain additional Macintosh-specific data. These attachments require special handling, as specified in section [2.1.3.4.3](#).

MIME writers SHOULD generate a **vCard** 3.0 attachment when generating **contact** information in a MIME message, as specified in section [2.1.3.4.6](#).

Additionally, MIME writers SHOULD classify Attachment objects as inline or not inline, as specified in section [2.1.3.4.1](#).

MIME writers SHOULD generate different MIME structures for the message depending on the presence of inline and non-inline attachments, as specified in the following three examples:

If both inline and non-inline attachments are present, MIME writers SHOULD generate the structure shown in the following figure. For an example of this structure, see section [3.1.2](#).

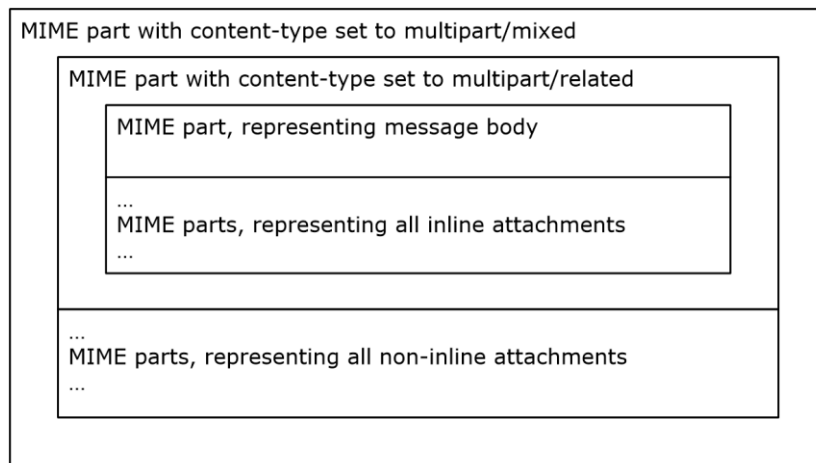


Figure 1: Inline and non-inline attachments present

If only inline attachments are present, MIME writers SHOULD generate the structure shown in the following figure. For an example of this structure, see section [3.1.3](#).

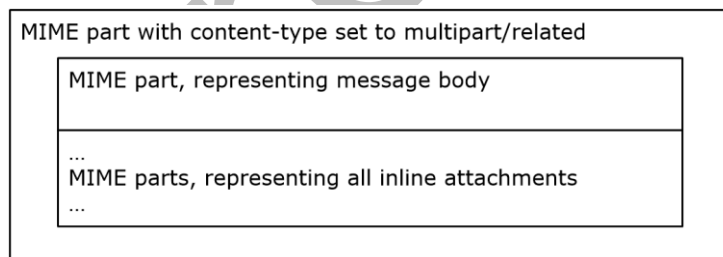


Figure 2: Only inline attachments present

If only non-inline attachments are present, MIME writers SHOULD generate the structure shown in the following figure. For an example of this structure, see section [3.1.4](#).

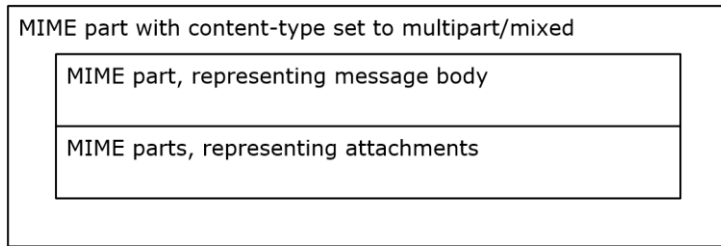


Figure 3: Only non-inline attachments present

2.1.3.4.1 Inline Attachments

Inline attachments are attachments that are rendered by clients within the body of the message. Clients SHOULD NOT create inline attachments if the best body text format is plain text. MIME writers SHOULD ignore indications that an attachment is inline for plain text messages by ignoring the presence of these properties on the Attachment object:

- **PidTagAttachFlags** ([MS-OXCMSG] section 2.2.2.18)
- **PidTagAttachContentId** ([MS-OXCMSG] section 2.2.2.26)
- **PidTagAttachContentLocation** ([MS-OXCMSG] section 2.2.2.26)

Likewise, clients SHOULD NOT designate attached Message objects as inline, and MIME writers SHOULD NOT [treat](#) attached Message objects as inline.

2.1.3.4.1.1 Inline Attachments in RTF Messages

If the best body text format is RTF, MIME writers SHOULD [treat](#) all OLE attachments, and only OLE attachments, as inline attachments. OLE attachments have 0x00000006 for the value of the **PidTagAttachMethod** property ([MS-OXCMSG] section 2.2.2.9).

RTF text does not contain explicit references to inline attachments, as HTML text does. Instead, the position of an inline attachment in the RTF text is indicated by an **\objattph** control word; clients insert such a tag into the RTF text for each inline attachment, as specified in [MS-OXRTFEX]. Clients also set the value of the **PidTagRenderingPosition** property ([MS-OXCMSG] section 2.2.2.16) to indicate the order of inline attachments: the attachment with the lowest value of this property matches the first **\objattph** control word; the next lowest matches the second **\objattph** control word, and so on. MIME writers sort inline attachments by the value of the **PidTagRenderingPosition** property when converting RTF text with inline attachments to HTML, and insert an HTML **IMG** element into the generated HTML at the position corresponding to the RTF **\objattph** control word.

2.1.3.4.1.2 Inline Attachments in HTML Messages

To mark an Attachment object as inline in a message for which the best body text format is HTML, clients do the following:

1. Set bit 3 (0x00000004) in the value of the Attachment object's **PidTagAttachFlags** property ([MS-OXCMSG] section 2.2.2.18) to TRUE.
2. Set the value of either the **PidTagAttachContentId** property ([MS-OXCMSG] section 2.2.2.26) (preferred) or the **PidTagAttachContentLocation** property ([MS-OXCMSG] section 2.2.2.26) on the Attachment object. If the **PidTagAttachContentLocation** property is used, the **PidTagAttachContentBase** property ([MS-OXCMSG] section 2.2.2.26) MAY be set to fully qualify a relative URI in the **PidTagAttachContentLocation** property. For details, see [RFC2557].

3. Clients MAY<79> include a tag that refers to the URI specified in step 2 in the HTML message text. If the **PidTagAttachContentId** property is used, the URI MUST use the "cid:" scheme.

MIME writers SHOULD NOT<80> rely entirely on bit 0x00000004 of the **PidTagAttachFlags** property value to be set correctly for all Attachment objects. Instead, MIME writers SHOULD verify all three conditions specified when deciding whether to treat an attachment as inline.

2.1.3.4.2 Attached Files

This section concerns generating attachments for pure MIME messages. When generating a TNEF message, all attachment data is written to the TNEF body part, as specified in [\[MS-OXTNEF\]](#).

2.1.3.4.2.1 File Name

For the file name in a MIME representation of an attached file, MIME writers SHOULD use the value of the **PidTagAttachLongFilename** property ([\[MS-OXCMSG\]](#) section 2.2.2.10). If this value is not available, MIME writers SHOULD use the value of the **PidTagAttachFilename** property ([\[MS-OXCMSG\]](#) section 2.2.2.11). If the value of the **PidTagAttachFilename** property is also not available, MIME writers SHOULD use a default value or MAY use an empty string.<81> The attached file name SHOULD be written to several different MIME headers, as specified in section [2.1.3.4.2.2](#).

If a file extension is needed for mapping the attachment content type, it SHOULD be obtained by copying all characters after the last "." (U+002E) character in the file name.

2.1.3.4.2.2 Content-Type, Content-Description, Content-Disposition Headers

MIME writers SHOULD determine the primary value of the **Content-Type** header (2) for an attached file by using the following steps:

1. Acquire the value of the **PidTagAttachMimeTag** property ([\[MS-OXCMSG\]](#) section 2.2.2.26). If this value is not available, MIME writers determine the value of the **Content-Type** header by mapping it from the file extension (which is determined from the attachment file name, as specified in section [2.1.3.4.2.1](#)), or by examining the file content itself. A MIME writer MAY<82> only generate the value of the **PidTagAttachMimeTag** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) on inbound MIME messages and not provide the property value for an outbound MIME message. As a last resort, the MIME writer uses the value "application/octet-stream".
2. If the value acquired in the previous step does not match requirements for MIME content-type, as specified in [\[RFC2045\]](#), or if the value represents any multipart MIME content-type, or if the value matches one of the following values, MIME writers replace it with "application/octet-stream":
 - application/applefile
 - application/mac-binhex40
 - message/rfc822

The value acquired as a result is then used as the value of the **Content-Type** header. MIME writers SHOULD also generate the *name* parameter for this header, by using the attachment file name (determined as specified in section 2.1.3.4.2.1) as a value.

MIME writers SHOULD<83> generate a **Content-Description** header by using the value of the **PidTagDisplayName** property ([\[MS-OXCFOld\]](#) section 2.2.2.2.5). If the property has no value, an empty header can be generated. If any of the conditions specified in [\[RFC2047\]](#) section 1 apply, the value of the **Content-Description** header (2) SHOULD be encoded as specified in that document. A MIME writer MAY<84> skip any encoding of the non-ASCII characters in the **Content-Description** header and use the value of an attachment's **PidTagDisplayName** property ([\[MS-OXCFOld\]](#) section 2.2.2.2.5) with no further encoding.

The value for the **Content-Disposition** header SHOULD<85> be generated based on whether the attachment is inline or not, as specified in section 2.1.3.4.1. For inline attachments, the value is "inline", and for non-inline attachments, the value is "attachment". MIME writers SHOULD generate the following parameters for this header:

- *filename*: the attachment file name determined as specified in 2.1.3.4.2.1 is used as a value.
- *size*: the value of the **PidTagAttachSize** property ([MS-OXCMSG] section 2.2.2.5) SHOULD be used as a parameter value. The *size* parameter SHOULD<86> be generated only if this property value is available and greater than 0 (zero).
- *creation-date*: the value of the **PidTagCreationTime** property ([MS-OXCMSG] section 2.2.2.3) SHOULD<87> be used as the parameter value; if the property value is not available, the current time SHOULD be used. In either case, the creation time SHOULD<88> be converted from UTC to a local time zone of the MIME writer's choice and formatted as specified in [RFC2822].
- *modification-date*: the value of the **PidTagLastModificationTime** property ([MS-OXCMSG] section 2.2.2.2) SHOULD<89> be used as the parameter value; if the property value is not available, the current time SHOULD be used. In either case, the modification time SHOULD<90> be converted from UTC to a local time zone of the MIME writer's choice and formatted as specified in [RFC2822].
- When a MIME skeleton is present as described in section 2.1.3.5.1, a MIME writer MAY<91> copy the entire original value of the **Content-Disposition** header to the message instead of generating individual parameter values as described in the preceding points.

2.1.3.4.2.3 Content-ID, Content-Location, Content-Base

MIME writers SHOULD generate a **Content-ID** header if the value of the **PidTagAttachContentId** property ([MS-OXCMSG] section 2.2.2.26) contains nonwhitespace characters. All trailing and leading whitespace characters SHOULD be removed from this value. If the resulting value does not start with "<" (U+003C), or does not end with ">" (U+003E), it SHOULD be enclosed in angle brackets. The resulting string becomes the value of the **Content-ID** header.

MIME writers SHOULD generate a **Content-Location** header if the **PidTagAttachContentLocation** property ([MS-OXCMSG] section 2.2.2.26) contains a value that is a valid URI. This value SHOULD be copied to the value of the **Content-Location** header.

MIME writers SHOULD generate a **Content-Base** header if the **PidTagAttachContentBase** property ([MS-OXCMSG] section 2.2.2.26) contains a value that is a valid absolute URI. This value SHOULD be copied to the value of the **Content-Base** header.

2.1.3.4.2.4 Content-Transfer-Encoding, MIME Part Body

MIME writers SHOULD use base64 encoding, as specified in see [RFC2045], for all ordinary file attachment **MIME part** bodies. As specified in [RFC2045], this also means that MIME writers SHOULD correspondingly generate the **Content-Transfer-Encoding** header, and set its value to "base64".

MIME writers use the value of the **PidTagAttachDataBinary** property ([MS-OXCMSG] section 2.2.2.7) to generate the MIME entity body for this attachment. If the property does not exist or has 0 (zero) length, an empty MIME entity body SHOULD be generated.

2.1.3.4.3 MacBinary Attached Files

For interoperability with Macintosh-based mail clients, MIME message attachments can be encoded in MacBinary format, which is signified by using one of the following **Content-Type** header values:

- application/applefile, as specified in [RFC1740].

- application/mac-binhex40, as specified in [\[RFC1741\]](#).
- multipart/appledouble, as specified in [\[RFC1740\]](#).

MIME writers SHOULD<92> generate a "multipart/appledouble" format, as this MIME content-type is recommended by [\[RFC1740\]](#) for use in most cases.

As specified in [\[RFC1740\]](#), the MIME part with a MIME content-type of "multipart/appledouble" contains two subparts: a header part, with a **Content-Type** header value of "application/applefile", and a data part that contains actual file data (with the **Content-Type** header set to the value that corresponds to the actual MIME content-type of the file that is encoded).

To trigger encoding of an Attachment object with a MIME content-type of "multipart/appledouble", clients set property values on the Attachment object as follows:

1. The value of the **PidTagAttachMethod** property ([\[MS-OXCMSG\]](#) section 2.2.2.9) is 0x00000001 (file attachment).
2. The value of the **PidTagAttachEncoding** property ([\[MS-OXCMSG\]](#) section 2.2.2.20) is the following byte string (expressed in hexadecimal): "%x2A.86.48.86.F7.14.03.0B.01".
3. The attachment content, which is the value of the **PidTagAttachDataBinary** property ([\[MS-OXCMSG\]](#) section 2.2.2.7), is encoded in MacBinary format.

MacBinary is a way of serializing all attributes of a Macintosh file, including both data and resource forks, into a single stream (2). The MacBinary format elements relied upon in this algorithm are summarized very briefly by the following two tables. What follows is intended to specify server behavior with respect to MacBinary data; it is not normative with respect to the MacBinary format itself. Additional details on the MacBinary format are specified in [\[MacBin\]](#).

The following table describes the MacBinary data fields.

MacBinary data field	Field length	Description
MacBinary header	128 bytes.	See more detail later in this section.
Secondary header data	Length is specified in bytes 120:121 of the MacBinary header.	SHOULD<93> be ignored by MIME writers. If bytes 120:121 have nonzero values, MIME writers MAY<94> write the attachment MIME content-type as "application/octet-stream" instead of "application/appledouble". The resource and data forks are not returned as separate attachments.
Data fork	Length is specified in bytes 83:86 of the MacBinary header; begins on an even multiple of 128 bytes.	Contents of the file.
Resource fork	Length is specified in bytes 87:90 of the MacBinary header; begins on an even multiple of 128 bytes.	Resources associated with the file.
Get Info comment	Length is specified in byte 99 of the MacBinary header.	SHOULD<95> be ignored by MIME writers. If byte 99 has a nonzero value, MIME writers MAY<96> write the attachment MIME content-type as "application/octet-stream" instead of "application/appledouble". The resource and data fork are not returned as separate attachments.

The following table describes the MacBinary header.

Byte offset and length	Value
Byte 0	Old version number; MUST be zero.
Byte 1	Length of file name; MUST be less than 64.
Bytes 2:64	File name, in us-ASCII character set; characters beyond the length specified in byte 1 MUST be ignored. <97>
Bytes 65:68	File type, signed integer.
Bytes 69:72	File creator, signed integer.
Byte 74	Pad; MUST be 0 (zero).
Byte 82	Pad; MUST be 0 (zero).
Bytes 83 : 86	Data fork length, signed 32-bit integer in big-endian format.
Bytes 87 : 90	Resource fork length, signed 32-bit integer in big-endian format.

MIME writers MUST create a MIME entity with a **Content-Type** header value of "multipart/appldouble", as specified in [RFC1740]. MIME writers SHOULD NOT write a *name* parameter for the **Content-Type** header in this MIME part. (This parameter is optional, as specified in [RFC1740].) All additional information (other than the file contents) for a file that is to be transmitted by using the "MIME content-type of "multipart/appldouble" SHOULD be put into a subpart with the **Content-Type** header value "application/applefile".

If the Attachment object's **PidNameAttachmentMacInfo** property ([MS-OXCMSG] section 2.2.2.26) has a value, MIME writers MUST use it as the body of the "application/applefile" body part. The value of this property SHOULD be "application/applefile" data, as specified in [RFC1740] and further detailed in section [2.2.3.4.2.2](#), but containing only the header and resource fork sections.

If the Attachment object's **PidNameAttachmentMacInfo** property has no value, MIME writers SHOULD generate the body of the "application/applefile" body part from the resource fork and header data present in the MacBinary structure from the **PidTagAttachDataBinary** property, by using the mappings specified in section 2.2.3.4.2.2. This MIME part is written out in the same way as in the case of an ordinary file attachment, with the following exceptions:

1. MIME writers MUST generate this part's MIME body by extracting only the file's data fork from the MacBinary structure in the **PidTagAttachDataBinary** property on the Attachment object, instead of just using raw data from this property.
2. MIME writers SHOULD copy the value of the **PidNameAttachmentMacContentType** property ([MS-OXCMSG] section 2.2.2.26) to the MIME data part's **Content-Type** header.

If the **PidNameAttachmentMacContentType** property has no value, MIME writers SHOULD write a **Content-Type** header value of "application/octet-stream". An "application/octet-stream" value SHOULD also be written if the **PidNameAttachmentMacContentType** property has one of the following values:

- message/rfc822
- application/applefile

- application/mac-binhex40
- any multipart MIME content-type

2.1.3.4.3.1 Application/Applefile

MIME messages can contain attachments in AppleSingle format. Such attachments have a **Content-Type** header value of "application/applefile" that are not subparts of a MIME part with a **Content-Type** header value of "multipart/appledouble". The format of such messages is specified in [\[RFC1740\]](#) and in section [2.2.3.4.2.2](#).

For the MacBinary header used by the AppleSingle format, MIME writers SHOULD NOT validate the **cyclic redundancy check (CRC)** value (byte 124 of the header).

MIME writers SHOULD NOT write out a Macintosh file resource fork for this format if the length of the resource fork is zero (0) in the attachment.

2.1.3.4.4 OLE Attachments

This section describes the generation of MIME entities that correspond to OLE attachments. An Attachment object is an OLE attachment if its **PidTagAttachMethod** property ([\[MS-OXCMSG\]](#) section 2.2.2.9) is set to 0x00000006.

MIME writers SHOULD generate a MIME part that has "image/jpeg" for the value of its **Content-Type** header to represent an OLE attachment in a MIME message or MAY<98> use "image/BMP" for the value of the **Content-Type** header. MIME writers SHOULD<99> generate a description string for an OLE attachment by using the value of the **PidTagDisplayName** property ([\[MS-OXCFOld\]](#) section 2.2.2.2.5) but ensuring that this value ends with ".jpg". The description string SHOULD be used as the *name* parameter of the **Content-Type** header, and the value of the **Content-Description** header SHOULD be generated with the same value.

A **Content-Disposition** header SHOULD<100> be generated in the same way as for ordinary file attachments, with the following exceptions:

1. The *size* parameter SHOULD NOT be generated.
2. The *filename* parameter value SHOULD be set to description string (see section [2.1.3.4.2.1](#)).

The rest of the MIME part headers SHOULD be generated in the same way as for ordinary file attachments, as specified in section [2.1.3.4.2](#).

OLE attachments SHOULD NOT have the **PidTagAttachDataBinary** property ([\[MS-OXCMSG\]](#) section 2.2.2.7) set, so the MIME part body cannot be generated in the same way as for ordinary file attachments. Instead, the **PidTagAttachDataObject** property ([\[MS-OXCMSG\]](#) section 2.2.2.8) SHOULD be used. This property SHOULD contain a static rendition of an OLE object in **metafile** format, as specified in [\[MS-WMF\]](#). MIME writers SHOULD<101> use this data to generate a **Joint Photographic Experts Group (JPEG)** image that represents this OLE object, and generate the MIME part body by using this image data. If image generation fails, the server SHOULD use a generic icon representing an attachment.

2.1.3.4.5 Embedded Message Attachments

This section describes the generation of MIME entities that correspond to embedded message attachments. An Attachment object is considered by MIME writers to be an embedded message attachment if the value of its **PidTagAttachMethod** property ([\[MS-OXCMSG\]](#) section 2.2.2.9) is 0x00000005. MIME writers SHOULD generate a MIME entity with the **Content-Type** header (2) set to "message/rfc822" (without parameters being generated). No other MIME headers SHOULD be generated. Instead, MIME writers SHOULD use properties of the embedded message to generate a

pure MIME message representation of this message, exactly as specified for ordinary messages, and use this data as the content of the MIME entity whose **Content-Type** header value is "message/rfc822". This MIME representation SHOULD be generated exactly as specified for ordinary messages, with the following exception: when writing MIME message headers by using PS_INTERNET_HEADERS properties, as specified in section [2.1.3.2.4](#), properties whose names begin with "X-MS-Exchange-Organization-" or "X-MS-Exchange-Forest-" SHOULD NOT be excluded from MIME generation (as they are for ordinary messages).

2.1.3.4.6 vCard Generation

To generate a vCard attachment, clients attach a **Contact object**, as specified in [\[MS-OXOCNTC\]](#), to a message. For MIME writers, an attached message with a value for the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) that begins with "IPM.Contact" is the trigger for generating a vCard attachment in a MIME message. [<102>](#) The vCard format is generated from Contact object properties, as specified in [\[MS-OXVCARD\]](#). vCard information is included in outbound MIME messages as a file attachment, as specified in section [2.1.3.4.2.2](#).

The vCard MIME part MUST use a **Content-Type** header value of "text/directory" with a profile value of "vCard". [<103>](#) The character set is set to UTF-8. The vCard content uses quoted-printable encoding, as specified in [\[RFC2045\]](#) section 6.7.

2.1.3.5 Generating Pure MIME Messages

When generating pure MIME messages, MIME writers SHOULD [<104>](#) combine stored Message object content with the contents of the **PidTagMimeSkeleton** property ([\[MS-OXCMSG\]](#) section 2.2.1.28). The purpose of the MIME skeleton is to make the MIME output more accurately resemble the original MIME message when using the messaging server as the conduit between two MIME-based protocols, such as SMTP and POP3 or IMAP4. The MIME skeleton contains the MIME structure and the headers of the original MIME message without any of the body part content, with some exceptions, as specified in section [2.3](#).

To generate a pure MIME message, the MIME writer reads the contents of the **PidTagMimeSkeleton** property associated with the Message object, and then generates the pure MIME message by combining the contents of the skeleton with the saved Message object content and attachments. The coupling of the generated MIME message to the original saved message by using the skeleton enables a more accurate reproduction of the MIME message provided by the server to clients. For more details about how inbound MIME content is stored and saved, see section 2.3.

The behavior specified in this section applies only if the **PidTagMimeSkeleton** property is defined. If it is not, MIME writers MUST follow the guidelines set forth in sections [2.1.3.1](#), [2.1.3.2](#), [2.1.3.3](#), and [2.1.3.4](#), in that order.

2.1.3.5.1 Generation Process

Generating a pure MIME message using the saved Message object contents in conjunction with the **PidTagMimeSkeleton** property ([\[MS-OXCMSG\]](#) section 2.2.1.28) follows these steps.

1. The contents of the **PidTagMimeSkeleton** property are combined with the saved best body and message attachments from the Message object, using the order of the headers from the **PidTagMimeSkeleton** property to place the message contents in the original order in the generated message.
 1. Use the **Content-ID** (or **X-ExchangeMime-Skeleton-Content-Id**) header in the **PidTagMimeSkeleton** property to map the MIME attachment to the message body part with a matching value of the **PidTagAttachContentId** property ([\[MS-OXCMSG\]](#) section 2.2.2.26).

2. Use the **Content-ID** (or **X-ExchangeMime-Skeleton-Content-Id**) header to map the MIME body part to the message body part with a matching value of the **PidTagAttachContentId** property.
3. The following headers are ignored in the skeleton and are regenerated from Message object properties:
 - **Keywords**
 - **Importance**
 - **Priority**
 - **X-MsMail-Priority**
 - **X-Priority**
 - **X-Message-Flag**
 - The **MIME-Version** header MAY [<105>](#) be ignored in the skeleton and be set to "1.0".
 - Encode attachment content using the **Content-Transfer-Encoding** header value specified in the headers for the attachment in the **PidTagMimeSkeleton** property. If the specified **Content-Transfer-Encoding** header is not supported by the server, then use the default base64 encoding for the attachment content and modify the value of the **Content-Transfer-Encoding** header in the **PidTagMimeSkeleton** property and in the outbound message to reflect this change.
2. The **X-ExchangeMime-Skeleton-Content-Id** headers SHOULD be deleted from the generated MIME message.

2.1.3.6 Generating Report Messages

When the value of **PidTagMessageClass** requires generation of a delivery status notification or message disposition notification, as specified in section [2.1.3.2.1](#), additional processing is required, as specified in the following sections:

- Delivery status notifications are specified in section [2.1.3.6.1](#).
- Message disposition notifications are specified in section [2.1.3.6.2](#).

2.1.3.6.1 Generating Delivery Status Notification Messages

To generate a delivery status notification, as specified in [\[RFC3464\]](#), the MIME writer does the following additional processing to generate the message/delivery-status body part:

1. The MIME writer writes the value of the **PidTagReportingMessageTransferAgent** property ([\[MS-OXOMSG\]](#) section 2.2.2.35) to the **Reporting-MTA** field. If the **PidTagReportingMessageTransferAgent** property is not present, the MIME writer uses the name of the computer where the MIME writer is located.
2. For each recipient in the message's recipient table:
 1. The MIME writer writes the recipient's SMTP address to the **Final-Recipient** field for the recipient.
 2. The MIME writer writes a value to the **Action** field for the recipient based on the value of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) on the message, as specified in section [2.1.3.6.1.1](#).

3. The MIME writer writes a value to the **Status** field for the recipient based on the values of the **PidTagNonDeliveryReportDiagCode** property ([MS-OXOMSG] section 2.2.2.30) and **PidTagNonDeliveryReportReasonCode** property ([MS-OXOMSG] section 2.2.2.31) on the recipient, as specified in section [2.1.3.6.1.2](#).
4. If the **PidTagRemoteMessageTransferAgent** property ([MS-OXOMSG] section 2.2.2.34) is present on the recipient, the MIME writer writes its value to the **Remote-MTA** field for the recipient.
5. If the **PidTagSupplementaryInfo** property ([MS-OXOMSG] section 2.2.2.36) is present on the recipient, the MIME writer writes its value to the **X-Supplementary-Info** field for the recipient.
6. If the **PidTagDisplayName** property is present on the recipient and its value is not the same as the recipient's SMTP address, the MIME writer writes its value to the **X-Display-Name** field for the recipient.

2.1.3.6.1.1 Generating a Value for the Action Field

The value of the **Action** field, as specified in [RFC3464](#), is determined by the value of the **PidTagMessageClass** property ([MS-OXCMSG](#) section 2.2.1.3) on the message, as specified in the following table.

Value of PidTagMessageClass property	Value of Action field
*.DR	delivered
*.Expanded.DR	expanded
*.Relayed.DR	relayed
*.Delayed.DR	delayed
*.NDR	failed

2.1.3.6.1.2 Generating a Value for the Status Field

The value of the **Status** field, as specified in [RFC3464](#), is determined by the values of the **PidTagNonDeliveryReportDiagCode** ([MS-OXOMSG](#) section 2.2.2.30) and **PidTagNonDeliveryReportReasonCode** ([MS-OXOMSG] section 2.2.2.31) properties. If the value of the **PidTagNonDeliveryReportDiagCode** property is -1, the value of the **Status** field is determined using the following table.

Value of PidTagNonDeliveryReportReasonCode property	Value of Status field
1	510
2	565
3	565
4	520
5	530
6	443

Value of PidTagNonDeliveryReportReasonCode property	Value of Status field
Any other value	540

If the value of the **PidTagNonDeliveryReportDiagCode** is not -1, the value of the **Status** field is determined using the following table.

Value of PidTagNonDeliveryReportDiagCode property	Value of Status field
0	510
1	514
2	445
3	446
4	510
5	447
6	530
7	560
8	563
9	562
10	560
11	554
12	560
13	534
14	560
15	561
16	553
17	555
18	533
19	562
20	560
21	560
22	464
23	464
24	464
25	464

Value of PidTagNonDeliveryReportDiagCode property	Value of Status field
26	573
27	446
28	572
29	571
30	524
31	561
32	513
33	510
34	513
35	511
36	510
37	530
38	530
39	510
40	516
41	510
42	510
43	516
46	570
48	512
Any other value	500

2.1.3.6.2 Generating Message Disposition Notification Messages

To generate a message disposition notification, as specified in [\[RFC3798\]](#), the MIME writer does the following additional processing to generate the message/disposition-notification body part.

1. The MIME writer writes a value using the SMTP address of the user represented by the **PidTagSender** recipient property group (section [2.6.4](#)) to the **Final-Recipient** field. If the **PidTagSender** recipient property group is not present on the message, the **PidTagSentRepresenting** recipient property group (section [2.6.5](#)) is used instead. If neither recipient property group is present, the MIME writer writes an empty **Final-Recipient** field.
2. The MIME writer writes a value to the **Disposition** field based on the value of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) on the message, as specified in section [2.1.3.6.2.1](#).

3. If the **PidTagParentKey** property ([\[MS-OXOMSG\]](#) section 2.2.2.18) is present and contains a value on the message, the MIME writer encodes the value using base64 encoding, and writes the resulting value to the **X-MSExch-Correlation-Key** field.
4. If the **PidTagOriginalMessageId** property ([\[MS-OXOMSG\]](#) section 2.2.1.85) is present and contains a value on the message, the MIME writer writes the value to the **Original-Message-ID** field.
5. If a display name is available for the recipient that was used to generate a value for the **Final-Recipient** field in step 1, and the display name is not the same as the SMTP address, the MIME writer writes the display name to the **X-Display-Name** field.

2.1.3.6.2.1 Generating a Value for the Disposition Field

The value of the **Disposition** field, as specified in [\[RFC3798\]](#), conforms to the following ABNF rule.

```
disposition = "automatic-action/MDN-sent-automatically;" action_value
action_value = "displayed" / "dispatched" / "processed" / "deleted" / "denied" / "failed"
```

The value of the *action_value* parameter is determined by the value of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3), as specified in the following table.

Value of PidTagMessageClass property	Value of action_value parameter
*.IPNRN	displayed
*.IPNNRN	deleted

The remaining 4 "disposition" values are not mapped to a specific message class and need not be supported by MIME writers.

2.1.3.7 Generating TNEF Messages

When generating TNEF messages, the MIME writer uses the following process:

1. The MIME writer generates a root MIME part with a **Content-Type** header value of "multipart/mixed".
2. The MIME writer generates a body part as a child of the root MIME part with a **Content-Type** header value of "text/plain". The content of this body part is a plain text rendering of the body of the message.
3. The MIME writer generates a body part as a child of the root MIME part with a **Content-Type** header value of "application/ms-tnef". The content of this body part is a TNEF data stream (2), as specified in [\[MS-OXTNEF\]](#). The MIME writer generates the body part using the following process:
 1. The MIME writer sets the attributes in the TNEF data stream (2) as specified in section [2.1.3.1](#), section [2.1.3.2](#), section [2.1.3.3](#), section [2.1.3.4](#), and their respective subsections.
 2. Before writing to the TNEF data stream (2), the MIME writer SHOULD first convert all address book EntryID values (bytes 4-19 are equal to the MUIDEMSAB UUID value "{%xdc.a740.c8.c0.42.10.1a.b4.b9.08.00.2b.2f.e1.82}") to an equivalent one-off EntryID, as specified in [\[MS-OXCDATA\]](#) section 2.2.5.1.

2.2 MIME Analysis Algorithm Details

This section specifies both conversion from pure MIME messages to Message objects, and from TNEF messages to Message objects. The agent that performs the conversion is referred to as a MIME reader for clarity, because both clients and servers perform this conversion for different protocols.

As a general rule, when data occurs both in MIME format and in a TNEF body part, the version found in MIME format SHOULD [<106>](#) be preferred. The message body is an exception to this rule: the plain text rendering found in MIME messages SHOULD NOT be used in preference to a richer (HTML or RTF) rendering found in TNEF messages. As an implementation guideline, MIME readers can process the TNEF body part before processing the remaining MIME data so that data from the MIME message overwrites the conflicting data from the TNEF message.

2.2.1 Abstract Data Model

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

The following ADM types are defined in this section:

Global

Mailbox

MessageObject

2.2.1.1 Global

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

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

2.2.1.2 Per Mailbox

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

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

2.2.1.3 Per Message Object

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

Mailbox.MessageObject.MIMESkeleton: The original MIME structure of the message.

2.2.2 Initialization

None.

2.2.3 Processing Rules

In the following sections, a client is specified as an agent that sets properties in a Message object, while a MIME reader is specified as an agent that parses MIME headers and assigns their values to Message object properties.

2.2.3.1 Address Elements

Most MIME address elements correspond to a recipient property group in the Message object, as specified in section 2.6. The MIME address element has three parts, as specified in [\[RFC2822\]](#): display name, comment, and e-mail address.

2.2.3.1.1 Mapping Internet E-Mail Address Elements to a Property Group

In general, MIME readers map the three elements of an Internet e-mail address to the four properties of the corresponding recipient property group as follows.

The comment part of the Internet e-mail address SHOULD be ignored.

Before setting properties, a MIME reader SHOULD [<107>](#) check whether the Internet e-mail address is encoded by using IMCEA encapsulation, as specified in section 2.1.3.1.8. If it is, the MIME reader performs de-encapsulation, as specified in section 2.2.3.1.2, to obtain the Internet e-mail address and type.

Next, a MIME reader SHOULD perform a lookup against the address book for an entry that has a proxy address that matches the Internet e-mail address.

The MIME reader sets the properties in the corresponding recipient property group, as specified in section 2.6, according to the following rules.

- Display name: If an address book entry was found with a matching proxy address, copy this value from the address book entry. Otherwise, if the Internet e-mail address has a display name part, convert it to a Unicode string, performing decoding as specified in [\[RFC2047\]](#) if required, and write it to this property value. If there is not a matching address book entry and there is no display name part, use the Internet e-mail address part.
- EntryID: If an address book entry was found with a matching proxy address, copy this value from the address book entry. If no entry is found, construct a one-off EntryID from the display name, address type, and Internet e-mail address property values, according to the format of the one-off EntryID as specified in [\[MS-OXCDATA\]](#) section 2.2.5.1.
- Address type: If an address book entry was found with a matching proxy address, copy this value from the address book entry. Otherwise, if the original Internet e-mail address was IMCEA-encapsulated and the MIME reader performed de-encapsulation, the MIME reader SHOULD write the type obtained from IMCEA de-encapsulation to this property. Otherwise, write "SMTP" to this property value. If there is no Internet e-mail address part, do not set this property value.
- E-mail address: If an address book entry was found with a matching proxy address, copy this value from the address book entry. Otherwise, if the original Internet e-mail address was IMCEA-encapsulated and the MIME reader performed de-encapsulation, use the Internet e-mail address obtained by de-encapsulation. Otherwise, convert the entire Internet e-mail address part to Unicode and write it to this property value. If there is no Internet e-mail address part, do not set this property value.

2.2.3.1.2 Recognizing and De-Encapsulating IMCEA-Encapsulated Addresses

For details about IMCEA encapsulation, see section [2.1.3.1.8](#). De-encapsulation SHOULD<108> be attempted only if the domain part of the encapsulated address is recognized as local or otherwise able to deliver mail to the de-encapsulated address.

An IMCEA-encapsulated SMTP address consists of the following six elements:

1. The literal string "IMCEA" in any combination of uppercase or lowercase letters.
2. The original address type; one or more ASCII characters.
3. A literal hyphen character, U+002D.
4. The encoded original address. Legal characters are uppercase and lowercase ASCII letters, digits, hyphen (U+002D), equal sign (U+003D), underscore (U+005F), and plus sign (U+002B). Any other characters MUST be encoded as a plus sign (U+002B) followed by two hexadecimal digits.
5. A literal "@" sign, U+0040.
6. The encapsulation domain, such as "example.com".

To identify an e-mail address as IMCEA-encapsulated, it is sufficient to match elements 1-3.

To obtain the original e-mail address and type from an encapsulated address, use the following procedure:

1. Copy element 2 to the e-mail address type.
2. Extract element 4, the encoded e-mail address.
3. Decode element 4 by replacing any underscore (U+005F) with a forward slash (U+002F), and replacing any sequence of plus sign (U+002B) followed by two hexadecimal digits with the single character that the two hexadecimal digits represent.

2.2.3.1.3 From

To set the values of the properties in the **PidTagSentRepresenting** property group (section [2.6.5](#)), clients MUST set the **From** header value, as specified in [\[RFC2822\]](#).

MIME readers MUST set the value of the **PidTagSentRepresenting** property group to the value of the first e-mail address component of the **From** header (which can contain multiple e-mail addresses). If the **From** header contains multiple addresses, the first address MUST be used; the others are ignored.

When reading TNEF messages, MIME readers SHOULD<109> use a **From** header value specified in MIME format in preference to the **attSentFor** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.17, or the **PidTagSentRepresenting** property group values of the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, specified in a TNEF message, except for messages attached to a TNEF message, where a MIME header does not exist.

2.2.3.1.4 Sender

To set the values of the properties in the **PidTagSender** property group (section [2.6.4](#)), clients MUST set the value of either the **Sender** header or the **From** header, as specified in [\[RFC2822\]](#).

MIME readers set the value of the **PidTagSender** property group to the value of the **Sender** header, if the **Sender** header is present in the MIME header. Otherwise, servers SHOULD set the **PidTagSender** property group to the value of the first mailbox, as specified in [\[RFC2822\]](#), of the **From** header.

When processing a TNEF message, MIME readers SHOULD use values specified in MIME format in preference to the **attFrom** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.3, or the **PidTagSender** property group values of the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, specified in a TNEF message, except for messages attached to a TNEF message, where a MIME header does not exist.

2.2.3.1.5 To, Cc, Bcc

To set the values of the properties in the Recipient Table property group (section [2.6.6](#)), clients MUST set one of the **To**, **Cc**, or **Bcc** header values, as specified in [\[RFC2822\]](#), that corresponds to the recipient type, as specified in the following table.

PidTagRecipientType property value	Recipient type name
0x00000001	To
0x00000002	Cc
0x00000003	Bcc

MIME readers MUST add one recipient (2) to the Message object for each address in the **To**, **Cc**, and **Bcc** headers. MIME readers map the value of the Recipient table property group from address elements, as specified in section [2.2.3.1.1](#). Clients can specify multiple **To**, **Cc**, or **Bcc** headers, and MIME readers SHOULD process all of them.

MIME readers set the value of the **PidTagRecipientType** property ([\[MS-OXOMSG\]](#) section 2.2.3.1) for each recipient row to the value specified in the table.

When processing a TNEF message, MIME readers SHOULD use values specified in MIME format in preference to the value of the **attRecipTable** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.22, specified in TNEF format, except for TNEF DSN messages and any messages attached to a TNEF message.

2.2.3.1.6 Reply Recipients

To set the values of the **PidTagReplyRecipientEntries** property ([\[MS-OXOMSG\]](#) section 2.2.1.43) and the **PidTagReplyRecipientNames** property ([\[MS-OXOMSG\]](#) section 2.2.1.44), clients MUST set the **Reply-To** header value, as specified in [\[RFC2822\]](#).

Note that because the value of the **Reply-to** header is an **address list** and not a single address, the property mapping is not a recipient property group.

MIME readers set the values of the **PidTagReplyRecipientEntries** property and the **PidTagReplyRecipientNames** property by mapping addresses from the **Reply-To** header.

When processing a TNEF message, MIME readers SHOULD use a **Reply-To** header value specified in MIME format in preference to the **PidTagReplyRecipientEntries** property value and the **PidTagReplyRecipientNames** property values of the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, specified in a TNEF message (except for messages attached to a TNEF message, where the MIME format counterpart is not available).

2.2.3.1.7 Disposition Notification Recipients

To set the value of the **PidTagReadReceiptRequested** property ([\[MS-OXOMSG\]](#) section 2.2.1.29) and the values of the properties in the **PidTagReadReceipt** property group (section [2.6.1](#)), clients MUST set the **Disposition-Notification-To** header value, as specified in [\[RFC3798\]](#).

MIME readers set the value of the **PidTagReadReceiptRequested** property to "TRUE" if the MIME header contains the **Disposition-Notification-To** header. If the MIME header does not contain the **Disposition-Notification-To** header, the value of the **PidTagReadReceiptRequested** property is not set.

MIME readers map the value of the **PidTagReadReceipt** property group from the value of the **Disposition-Notification-To** header, if the field exists.

When processing a TNEF message, MIME readers SHOULD use a **Disposition-Notification-To** header value specified in MIME format in preference to the **PidTagReadReceiptRequested** property and **PidTagReadReceipt** property group values of the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, specified in a TNEF message (except for messages attached to a TNEF message, where the MIME counterpart is not available).

2.2.3.1.8 Return-Receipt-To

To set the value of the **PidTagOriginatorDeliveryReportRequested** property ([\[MS-OXOMSG\]](#) section 2.2.1.20), clients MUST set the (non-standard) **Return-Receipt-To** header value.

MIME readers set the value of the **PidTagOriginatorDeliveryReportRequested** property to **TRUE** if the message contains the **Return-Receipt-To** header. The actual value of the header is ignored, and receipts will be returned to the sender.

When processing a TNEF message, MIME readers SHOULD use a **Return-Receipt-To** header value specified in a MIME message in preference to the value of the **PidTagOriginatorDeliveryReportRequested** property ([\[MS-OXOMSG\]](#) section 2.2.1.20) of the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, specified in a TNEF message (except for messages attached to a TNEF message, where the MIME counterpart is not available).

2.2.3.2 Envelope Elements

Many MIME headers that map directly to Message object properties have string values. Unless otherwise specified, the string values are copied directly. All string values SHOULD be converted to Unicode (UTF-16 form) before they are copied to property values, and where applicable, the decoding specified in [\[RFC2047\]](#) is applied before generating the Unicode characters.

If there are multiple instances of a header, MIME readers SHOULD use the first instance to set the value of the corresponding property but MAY [<110>](#) use the last instance instead. However, in the case of multiple recipient fields, MIME readers SHOULD combine the content of all instances to set the value of the corresponding property.

2.2.3.2.1 MessageID

To set the value of the **PidTagInternetMessageId** property ([\[MS-OXOMSG\]](#) section 2.2.1.12), clients MUST set the **Message-ID** header value, as specified in [\[RFC2822\]](#). MIME readers copy the value of the **Message-ID** header to the **PidTagInternetMessageId** property.

2.2.3.2.2 Sent time

To set the value of the **PidTagClientSubmitTime** property ([\[MS-OXOMSG\]](#) section 2.2.3.11), clients MUST set the **Date** header value, as specified in [\[RFC2822\]](#).

MIME readers set the value of the **PidTagClientSubmitTime** property to the value of the **Date** header, converted to UTC. Full precision of the **Date** header value, including seconds, MUST be preserved. If the **Date** header is missing or contains an invalid value, MIME readers set the value of the **PidTagClientSubmitTime** property to the current UTC time.

When processing TNEF messages, MIME readers use a **Date** header value specified in a MIME message in preference to an **attDateSent** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.4, or the value of the **PidTagClientSubmitTime** property specified in a TNEF message.

2.2.3.2.3 References

To set the value of the **PidTagInternetReferences** property ([\[MS-OXCMSG\]](#) section 2.2.1.26), clients write the value to a **References** header.

MIME readers copy the value of the **References** header to the value of the **PidTagInternetReferences** property. MIME readers MAY truncate the value of the **PidTagInternetReferences** property if it exceeds 64 kilobytes in length.

2.2.3.2.4 Sensitivity

To set the value of the **PidTagSensitivity** property ([\[MS-OXCMSG\]](#) section 2.2.1.13) to a value other than normal, clients MUST write the value to a **Sensitivity** header.

MIME readers map **Sensitivity** header values to values of the **PidTagSensitivity** property as specified in the following table.

PidTagSensitivity property value	Sensitivity header value
0x00000000	Normal
0x00000001	Personal
0x00000002	Private
0x00000003	Company-Confidential

2.2.3.2.5 Importance

To set the value of the **PidTagImportance** property ([\[MS-OXCMSG\]](#) section 2.2.1.11), clients SHOULD write the value to an **Importance** header.

MIME readers map **Importance** header values to values of the **PidTagImportance** property as specified in the following table.

Importance header value	PidTagImportance property value
Low	0x00000000
Normal	0x00000001
High	0x00000002

Clients MAY [<111>](#) use a **Priority**, **X-Priority**, or **X-MSMail-Priority** header instead of an **Importance** header to set the value of the **PidTagImportance** property. In that case, MIME readers map the header values to values of the **PidTagImportance** property as specified in the following tables. However, if an **Importance** header is present, MIME readers SHOULD use its value in preference to any of the others.

Priority header value	PidTagImportance property value
Non-Urgent	0x00000000
Normal	0x00000001
Urgent	SHOULD be 0x00000002, MAY <112> be 0x00000001

X-Priority header value	PidTagImportance property value
5	0x00000000
4	0x00000000
3	0x00000001
2	0x00000002
1	0x00000002

X-MSMail-Priority header value	PidTagImportance value
Low	0x00000000
Normal	0x00000001
High	0x00000002

2.2.3.2.6 Subject

To set the value of the **PidTagSubjectPrefix** property ([\[MS-OXOMSG\]](#) section 2.2.1.60) and the **PidTagNormalizedSubject** property ([\[MS-OXCMSG\]](#) section 2.2.1.10), clients MUST set the **Subject** header value, as specified in [\[RFC2822\]](#).

MIME readers SHOULD analyze the **Subject** header value into a prefix and a normalized subject value, as specified in section [2.2.3.2.6.1](#), and then set the values of the **PidTagSubjectPrefix** and **PidTagNormalizedSubject** properties, rather than simply setting the value of the **PidTagSubject** property ([\[MS-OXPROPS\]](#) section 2.1021). MIME readers can truncate the **Subject** value.

MIME readers use a **Subject** header value specified in a MIME message in preference to an **attSubject** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.7, or to the value of the **PidTagSubject** property specified in a TNEF message. They SHOULD, however, use the values of the **PidTagSubjectPrefix** and **PidTagNormalizedSubject** properties instead of the **PidTagSubject** property from TNEF messages when they match the MIME message subject because of limitations in the subject normalization algorithm, as specified in section 2.2.3.2.6.1.

2.2.3.2.6.1 Normalizing the Subject

If no values are available for the **PidTagNormalizedSubject** property ([\[MS-OXCMSG\]](#) section 2.2.1.10) and the **PidTagSubjectPrefix** property ([\[MS-OXOMSG\]](#) section 2.2.1.60) in the MIME message, servers SHOULD parse the **Subject** header value and set those values as follows. If the **Subject** header value consists of one, two, or three characters (exclusive of colon (U+003A), blank

(U+0020), or digits (U+0030 through U+0039)), followed by a colon (U+003A) and any number of blanks (U+0020), the server SHOULD set the value of the **PidTagSubjectPrefix** property to the aforementioned one, two, or three characters appended with a colon and a space (": "), and SHOULD set the value of the **PidTagNormalizedSubject** property to the remainder of the **Subject** header value beginning immediately after the aforementioned blanks.

2.2.3.2.7 Conversation Topic

To set the value of the **PidTagConversationTopic** property ([\[MS-OXCMSG\]](#) section 2.2.1.10), clients MUST write the value to a **Thread-Topic** header. This value SHOULD be the same as the value of the **Subject** header, normalized as specified in section [2.2.3.2.6.1](#) to remove any prefix.

MIME readers copy the value of a **Thread-Topic** header to the value of the **PidTagConversationTopic** property. A MIME reader MAY<113> set the **PidTagConversationTopic** property differently depending on which MIME header, **Subject** header, or **Thread-Topic** header is present; the value of the header; and which header comes first. In this case, the MIME writer uses the following procedure.

1. If neither header is available, the **PidTagConversationTopic** property is not set.
2. If only the **Subject** header is available and the header can be parsed to set the **PidTagNormalizedSubject** property ([\[MS-OXCMSG\]](#) section 2.2.1.10), then the **PidTagConversationTopic** property is set to the value of the **PidTagNormalizedSubject** property.
3. However, if the message class is "IPM.Post" (denoting a Post object as specified in [\[MS-OXOPOST\]](#)), then the **PidTagConversationTopic** property is not set.
4. If only a **Thread-Topic** header is available, the **PidTagConversationTopic** property is set from the value of the **Thread-Topic** header.
5. If both the **Subject** and **Thread-Topic** headers are present, and the **Subject** header precedes the **Thread-Topic** header, then the MIME writer attempts to normalize the **Subject** header value, as specified in section 2.2.3.2.6.1.
 1. If normalization is successful, the MIME writer sets the **PidTagConversationTopic** property to the value of the **PidTagNormalizedSubject** property.
 2. If unsuccessful, the MIME writer looks to see whether the **Subject** header value ends with the **Thread-Topic** header value. If this is successful, the MIME writer sets the **PidTagConversationTopic** property to the value of the **Thread-Topic** header.
 3. If neither of the above conditions are successful, the MIME writer sets the **PidTagConversationTopic** property to the value of the **Subject** header.
6. If both the **Subject** and the **Thread-Topic** header are present, and the **Thread-Topic** header precedes the **Subject** header, then the MIME writer first looks to see whether the **Subject** header value ends with the **Thread-Topic** header value.
 1. If successful, then the MIME writer sets the value of the **PidTagConversationTopic** property to the value of the **Thread-Topic** header.
 2. If unsuccessful, then the MIME writer normalizes the value of the **Subject** header, as specified in section 2.2.3.2.6.1, producing values for the **PidTagSubjectPrefix** property ([\[MS-OXOMSG\]](#) section 2.2.1.60) and the **PidTagNormalizedSubject** property, and sets the **PidTagConversationTopic** property to the value of the **PidTagNormalizedSubject** property.

3. If neither of the above conditions are successful, the MIME writer sets the **PidTagConversationTopic** property to the value of the **Subject** header.

MIME readers SHOULD also use this header value as a hint to normalize the subject, as specified in section 2.2.3.2.6.1, if this value matches the tail of the **Subject** header value.

2.2.3.2.8 Conversation Index

To set the value of the **PidTagConversationIndex** property ([MS-OXOCFG] section 2.2.8.8), clients MUST write the value to a **Thread-Index** header. The property data type is binary, and clients encode the header value using base64 encoding, as specified in [RFC2045]. The format of the value is specified in [MS-OXOMSG].

MIME readers copy the value of a **Thread-Index** header to the value of the **PidTagConversationIndex** property, assuming the base64-encoded text can be successfully decoded to binary data. MIME readers SHOULD ignore a **Thread-Index** header that does not contain base64-encoded binary data.

2.2.3.2.9 In-Reply-To Message ID

To set the value of the **PidTagInReplyToId** property ([MS-OXOMSG] section 2.2.1.13), clients MUST write the value to an **In-Reply-To** header, as specified in [RFC2822].

MIME readers copy the value of an **In-Reply-To** header to the value of the **PidTagInReplyToId** property.

2.2.3.2.10 ReplyBy Time

To set the value of the **PidTagReplyTime** property ([MS-OXOFLAG] section 2.2.3.1), clients MUST set the **Reply-By** header value, as specified in [RFC2156].

MIME readers MUST set the value of the **PidTagReplyTime** property to the value of the **Reply-By** header, converted to UTC time.

When processing TNEF messages, MIME readers MUST use a **Reply-By** header value specified in the MIME message in preference to the value of the **PidTagReplyTime** property specified in a TNEF message.

2.2.3.2.11 Language Properties

To set the value of the **PidTagMessageLocaleId** property ([MS-OXCMSG] section 2.2.1.5), clients MUST set the **Content-Language** header, as specified in [RFC3282].

MIME readers set the value of the **PidTagMessageLocaleId** property by extracting the first language tag from the value of the **Content-Language** header and mapping it to an LCID, as specified in [MS-LCID]. MIME readers SHOULD use the value of a **Content-Language** header (2) in preference to the value of the **PidTagMessageLocaleId** property found in the **attMsgProps** attribute, as specified in [MS-OXTNEF] section 2.1.3.3.21, of a TNEF message.

To set the value of the **PidNameAcceptLanguage** property ([MS-OXCMSG] section 2.2.1.42), clients SHOULD write the value to an **Accept-Language** header. Clients MAY write an **X-Accept-Language** header instead.

MIME readers SHOULD <114> copy the value of either header to the value of the **PidNameAcceptLanguage** property. If both headers are present, MIME readers SHOULD use the **Accept-Language** header.

2.2.3.2.12 Categories

To set the value of the **PidNameKeywords** property ([MS-OXCMSG] section 2.2.1.17), clients MUST set the **Keywords** header, as specified in [RFC2076].

MIME readers SHOULD map the value of a **Keywords** header to the value of the **PidNameKeywords** property by splitting the **Keywords** header value at each comma (U+0032), trimming white space, and storing each keyword as an individual value of the multiple string property.

To prevent conflicts among category schemes in different organizations, MIME readers MAY omit mapping the **Keywords** header to the **PidNameKeywords** property.

2.2.3.2.13 Message Expiry Time

To set the value of the **PidTagExpiryTime** property ([MS-OXOMSG] section 2.2.3.7), clients MUST write the value to the **Expires** header.

MIME readers copy the value of the **Expires** header to the value of the **PidTagExpiryTime** property, after converting it to UTC time.

Clients MAY use an **Expiry-Date** header instead of an **Expires** header. Servers SHOULD use the value of the **Expires** header in preference to the **Expiry-Date** header, if both headers are present. A client MAY <115> use whichever header shows up last in the list of header information, either the **Expires** header or the **Expiry-Date** header, as the header used to set the **PidTagExpiryTime** property. However, this is not recommended.

2.2.3.2.14 Suppression of Automatic Replies

To set the value of the **PidTagAutoResponseSuppress** property (section 2.1.3.2.20) to -1, indicating that all automatic replies to the message are to be suppressed, clients SHOULD write an **X-Auto-Response-Suppress** header with the value "All". Clients MAY, instead, write a **Precedence** header (2) with any value.

To set the value of the **PidTagAutoResponseSuppress** property to a more specific value, clients write an **X-Auto-Response-Suppress** header with one or more values selected from the table in section 2.1.3.2.20.

MIME readers SHOULD <116> map individual elements of an **X-Auto-Response-Suppress** header to bits in the value of the **PidTagAutoResponseSuppress** property according to the table. If both the **X-Auto-Response-Suppress** and **Precedence** headers are present, the **PidTagAutoResponseSuppress** property value SHOULD be 0xFFFFFFFF but MAY <117> be 0x00000000. If the value of the **X-Auto-Response-Suppress** header is other than as specified in the table in section 2.1.3.2.20, MIME readers SHOULD <118> ignore the entire header.

2.2.3.2.15 Content Class

To set the value of the **PidNameContentClass** property ([MS-OXCMSG] section 2.2.1.48), clients SHOULD <119> write the value to a **Content-Class** header.

MIME readers copy the value of a **Content-Class** header to the value of the **PidNameContentClass** property.

MIME readers SHOULD <120> also set the value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) for certain **Content-Class** header values as specified in the following table, but only if the value of the **PidTagMessageClass** property would otherwise be set to "IPM.Note". MIME readers SHOULD verify that the content of the MIME message conforms to the additional requirements specified in the corresponding sections listed in the "Additional requirements" column. If the MIME message does not conform to the additional requirements, the **PidTagMessageClass** property SHOULD be set to "IPM.Note".

Content-Class header value	PidTagMessageClass property value	Additional requirements
"fax"	"IPM.Note.Microsoft.Fax"	See section 2.2.3.2.15.1 .
"fax-ca"	"IPM.Note.Microsoft.Fax.CA"	See section 2.2.3.2.15.1 .
"missedcall"	"IPM.Note.Microsoft.Missed.Voice"	See section 2.2.3.2.15.2 .
"voice-uc"	"IPM.Note.Microsoft.Conversation.Voice"	See section 2.2.3.2.15.2 .
"voice-ca"	"IPM.Note.Microsoft.Voicemail.UM.CA"	See section 2.2.3.2.15.2 .
"voice"	"IPM.Note.Microsoft.Voicemail.UM"	See section 2.2.3.2.15.2 .
Starts with "urn:content-class:custom."	"IPM.Note.Custom.", followed by the value of the Content-Class header, with the "urn:content-class:custom." prefix removed. <121>	None.

Additionally, if the **Content-Class** header value begins with "InfoPath.", MIME readers SHOULD extract a substring from the header value beginning immediately after the prefix and ending at the end of the header value. If this string contains a period character (U+002E), and the first occurrence of this character is not the last one in the string, this string SHOULD be further separated into two substrings. The delimiting period is not included into either one of the substrings.

The first substring SHOULD be additionally checked to match the string format of a **GUID** string, as specified in [\[MS-DTYP\]](#). If this check succeeds, the second substring SHOULD be saved as a value of the **PidLidInfoPathFormName** property ([MS-OXCMSG] section 2.2.1.27). In addition, the first substring SHOULD be appended to "IPM.InfoPathForm." and written to the value of the **PidTagMessageClass** property.

If a message that is being processed by a MIME reader is clear signed or opaque signed, as specified in [\[MS-OXOSMIME\]](#), the appropriate suffix (".SMIME.MultipartSigned" or ".SMIME") SHOULD be appended to the value of the **PidTagMessageClass** property.

2.2.3.2.15.1 Requirements for Fax messages

MIME readers should verify that MIME messages with a **Content-Type** header value of "fax" or "fax-ca" meet all of the following requirements.

- The root MIME part MUST have a **Content-Type** header value of "multipart/mixed".
- The root MIME part MUST contain exactly two MIME parts:
 - The first MIME part contained by the root MIME part MUST have a **Content-Type** header value of "text/html".
 - The second MIME part contained by the root MIME part MUST have a **Content-Type** header value of "image/tiff".

2.2.3.2.15.2 Requirements for Voice and Voicemail messages

MIME readers should verify that MIME messages with a **Content-Type** header value of "missedcall", "voice-uc", "voice-ca", or "voice" have no attachments with a **Content-Type** header value that is not in the following list.

- audio/gsm
- audio/mp3
- audio/wav
- audio/wma

2.2.3.2.16 Message Flagging

To set the value of the **PidLidFlagRequest** property ([\[MS-OXCSPAM\]](#) section 2.2.1.3), clients MUST write the value to an **X-Message-Flag** header.

MIME readers copy the value of an **X-Message-Flag** header to the value of the **PidLidFlagRequest** property. In addition, when an **X-Message-Flag** header is present, MIME readers SHOULD [<122>](#) do all the following:

1. Set the value of the **PidTagFlagStatus** property ([\[MS-OXOFLAG\]](#) section 2.2.1.1) to 2 (denoting that the message is flagged).
2. Copy the value of the **PidTagSubject** property ([\[MS-OXPROPS\]](#) section 2.1021) to the value of the **PidLidToDoTitle** property ([\[MS-OXOFLAG\]](#) section 2.2.1.12).
3. Set the value of the **PidLidTaskStatus** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.2) to 0 (zero) (denoting that a task is not started).
4. Delete or disregard any existing property values for the following properties:
 - **PidLidTaskDueDate** ([\[MS-OXOTASK\]](#) section 2.2.2.2.5)
 - **PidLidTaskStartDate** ([\[MS-OXOTASK\]](#) section 2.2.2.2.4)
 - **PidTagFlagCompleteTime** ([\[MS-OXOFLAG\]](#) section 2.2.1.3)
 - **PidLidTaskDateCompleted** ([\[MS-OXOTASK\]](#) section 2.2.2.2.9)
5. Set the value of the **PidLidTaskComplete** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.20) to FALSE.
6. Set the value of the **PidLidPercentComplete** property ([\[MS-OXOFLAG\]](#) section 2.2.2.3) to 0.0.
7. Set the value of the **PidTagToDoItemFlags** property ([\[MS-OXPROPS\]](#) section 2.1036) to 8.

2.2.3.2.17 List Server Properties

To set the values of list server-related properties, clients MUST write headers as specified in the following table.

Property name	Preferred header name	Alternate header name
PidTagListHelp	List-Help	X-List-Help
PidTagListSubscribe	List-Subscribe	X-List-Subscribe
PidTagListUnsubscribe	List-Unsubscribe	X-List-Unsubscribe

MIME readers SHOULD [<123>](#) copy header values to property values as specified in the table.

2.2.3.2.18 Payload Properties

To set the value of the **PidTagAttachPayloadClass** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) or the **PidTagAttachPayloadProviderGuidIdString** property ([\[MS-OXCMSG\]](#) section 2.2.2.26), clients SHOULD [<124>](#) write an **X-Payload-Class** and an **X-Payload-Provider-GUID** header, respectively. Such headers SHOULD [<125>](#) be written to a MIME entity that will be analyzed as an attachment, as specified in section [2.2.3.4](#).

MIME readers SHOULD [<126>](#) [<127>](#) copy these header values to the values of the corresponding properties. MIME readers SHOULD [<128>](#) copy these headers when they appear on a MIME entity that is analyzed as a message or message body, rather than as an attachment.

2.2.3.2.19 Purported Sender Domain

MIME readers SHOULD [<129>](#) copy the value of the **X-MS-Exchange-Organization-PRD** header to the **PidTagPurportedSenderDomain** property ([\[MS-OXCMSG\]](#) section 2.2.1.43).

2.2.3.2.20 Sender Id Status

MIME readers SHOULD [<130>](#) copy the value of the **X-MS-Exchange-Organization-SenderIdResult** header to the **PidTagSenderIdStatus** property ([\[MS-OXPROPS\]](#) section 2.992). The values of the header are mapped to the **PidTagSenderIdStatus** property as follows:

Symbolic name	Value
Neutral	0x00000001
Pass	0x00000002
Fail	0x00000003
SoftFail	0x00000004
None	0x00000005
TempError	0x80000006
PermError	0x80000007

2.2.3.2.21 Spam Confidence Level

MIME readers SHOULD [<131>](#) parse the value of the **X-MS-Exchange-Organization-SCL** header as an integer value in the range -1 to 10 and assign that value to the **PidTagContentFilterSpamConfidenceLevel** property ([\[MS-OXCSPAM\]](#) section 2.2.1.3).

2.2.3.2.22 Classification Properties

In order to preserve full client/server data fidelity in the MIME content, if the **PidLidClassified** property ([\[MS-OXCMSG\]](#) section 2.2.1.25) is present in the mail object and is set to TRUE, then clients SHOULD [<132>](#) write the following header:

```
X-Microsoft-Classified: true
```

In addition, clients SHOULD write header values for all of the **X-Microsoft-Classification**, **X-Microsoft-ClassDesc**, **X-Microsoft-Classification-GUID**, and **X-Microsoft-Classification-Keep** headers.

When the **X-Microsoft-Classified** header is present and has the value "true", MIME readers SHOULD map or copy all classification header values to their corresponding property values, as specified in the following table. If the **X-Microsoft-Classified** header is missing or has a value other than "true", MIME readers SHOULD NOT [<133>](#) set any of the five property values listed in the table.

Classification header name	Classification property name	Header value mapping
X-Microsoft-Classified	PidLidClassified ([MS-OXCMSG] section 2.2.1.25)	"true" maps to TRUE.
X-Microsoft-ClassKeep	PidLidClassificationKeep ([MS-OXPROPS] section 2.55)	"true" maps to TRUE. "false" maps to FALSE.
X-Microsoft-Classification	PidLidClassification ([MS-OXCMSG] section 2.2.1.23)	No mapping. The string value is copied directly.
X-Microsoft-ClassDesc	PidLidClassificationDescription ([MS-OXCMSG] section 2.2.1.24)	No mapping. The string value is copied directly.
X-Microsoft-ClassID	PidLidClassificationGuid ([MS-OXPROPS] section 2.54)	No mapping. The string value is copied directly.

2.2.3.2.23 Unified Messaging Properties

To set the values of **Unified Messaging** properties, clients SHOULD write the value to the corresponding header, as specified in the following table.

Header name	Property name
X-CallingTelephoneNumber	PidTagSenderTelephoneNumber ([MS-OXPROPS] section 2.996).
X-VoiceMessageSenderName	PidTagVoiceMessageSenderName ([MS-OXPROPS] section 2.1049).
X-AttachmentOrder	PidTagVoiceMessageAttachmentOrder ([MS-OXPROPS] section 2.1047).
X-CallID	PidTagCallId ([MS-OXPROPS] section 2.619).
X-VoiceMessageDuration	PidTagVoiceMessageDuration ([MS-OXPROPS] section 2.1048); header value MUST be parsed as PtypInteger32 ([MS-OXCDATA] section 2.11.1).
X-FaxNumberOfPages	PidTagFaxNumberOfPages ([MS-OXPROPS] section 2.686); header value MUST be parsed as PtypInteger32 .

MIME readers SHOULD [<134>](#) copy header values to property values, as specified in the table.

2.2.3.2.24 Content-ID

To set the value of the **PidTagBodyContentId** property ([MS-OXCMSG] section 2.2.1.56.7), clients MUST write the value to a **Content-ID** header on a MIME entity that maps to a message body, as specified in section [2.2.3.3](#). If this value starts with "<" (U+003C) and/or ends with ">" (U+003E),

these characters SHOULD be removed. Leading and trailing whitespace characters SHOULD also be removed.

MIME readers SHOULD [<135>](#) copy the value of a **Content-ID** header on such a MIME entity to the value of the **PidTagBodyContentId** property.

Clients can write either a **Content-ID** or a **Content-Location** header, but SHOULD NOT write both on a single MIME entity.

2.2.3.2.25 Content-Base

To set the value of the **PidNameContentBase** property ([\[MS-OXCMSG\]](#) section 2.2.1.41), clients MUST write the value to a **Content-Base** header on a MIME entity that maps to a message body (top-level or nested), as specified in section [2.2.3.3](#), or on a Message object that corresponds to an attachment.

MIME readers SHOULD [<136>](#) copy the value of a **Content-Base** header on a top-level message to the value of the **PidNameContentBase** property.

2.2.3.2.26 Content-Location

To set the value of the **PidTagBodyContentLocation** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.8), clients SHOULD write the value to a **Content-Location** header on a MIME entity that maps to a message body, as specified in section [2.2.3.3](#).

MIME readers SHOULD [<137>](#) copy the value of a **Content-Location** header on such a MIME entity to the value of the **PidTagBodyContentLocation** property.

2.2.3.2.27 XRef

MIME readers SHOULD [<138>](#)[<139>](#) copy the value of an **XRef** header to the value of the **PidNameCrossReference** property (section [2.5.3](#)).

2.2.3.2.28 PidTagTransportMessageHeaders

MIME readers SHOULD copy headers from an inbound message to the value of the **PidTagTransportMessageHeaders** property ([\[MS-OXPROPS\]](#) section 2.1038). The rules for including and excluding headers are as follows:

1. Headers whose names do not begin with the reserved name prefixes "X-MS-Exchange-Organization-" and "X-MS-Exchange-Forest-" SHOULD be copied to the **PidTagTransportMessageHeaders** property.
2. Headers specifically mentioned in section [2.1.3.2](#) SHOULD be copied to the **PidTagTransportMessageHeaders** property, even if their names begin with the reserved name prefixes "X-MS-Exchange-Organization-" or "X-MS-Exchange-Forest-".
3. Headers that are not specifically mentioned in section [2.1.3.2](#), and whose names begin with the reserved name prefixes "X-MS-Exchange-Organization-" and "X-MS-Exchange-Forest-", SHOULD NOT be copied to the **PidTagTransportMessageHeaders** property.

This property value SHOULD be set only by MIME readers upon delivery of a message from SMTP, in which case it SHOULD be set to the header of the top-level message (with exceptions as already specified). A MIME reader MAY [<140>](#) also set the **PidTagTransportMessageHeaders** property ([\[MS-OXOMSG\]](#) section 2.2.1.61) when downloading messages via IMAP4.

2.2.3.2.29 Generic Headers in PS_INTERNET_HEADERS

To create a named property in the PS_INTERNET_HEADERS property set, whose name is a header name and whose value is a header value, clients write the name and value to a header.

For each such header, MIME readers SHOULD [<141>](#) create a named property as follows:

- The property set GUID is "%X86.03.02.00.00.00.00.00.C0.00.00.00.00.46".
- The property name is the header name.
- The property value is the header value. If the header value was encoded according to [\[RFC2047\]](#), MIME readers MUST decode it.

MIME readers MUST NOT create such named properties for any MIME header that is mapped to a different property, as specified elsewhere in this section. MIME readers SHOULD NOT create such named properties for any of the following MIME headers:

- **Received**
- **Resent-From**
- **Resent-Sender**
- **Resent-Date**
- **Resent-Message-Id**
- **Content-Type**
- **Content-Disposition**
- **Content-Description**
- **Content-Transfer-Encoding**
- **Content-ID**
- **Content-MD5**
- **MIME-Version**
- **Return-Path**
- **Comments**
- **AdHoc**
- **Apparently-To**
- **Approved**
- **Control**
- **Distribution**
- **Encoding**
- **FollowUp-To**
- **Lines**
- **Bytes**

- **Article**
- **Supersedes**
- **NewsGroups**
- **NntpPostingHost**
- **Organization**
- **Path**
- **RR**
- **Summary**
- **Trace**
- **Encrypted**
- **X-MimeOle**
- **X-MS-TNEF-Correlator**
- Any header the name of which begins with "X-MS-Exchange-Organization-" or "X-MS-Exchange-Forest-", except for the following:
 - **X-MS-Exchange-Organization-AuthAs**
 - **X-MS-Exchange-Organization-AuthDomain**
 - **X-MS-Exchange-Organization-AuthMechanism**
 - **X-MS-Exchange-Organization-AuthSource**

2.2.3.3 Body Text

Unlike MIME, which allows an arbitrary number of inline text body parts, Message objects distinguish one text body part as the message body.

2.2.3.3.1 Client Actions

To send the value of the **PidTagBody** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.1) as the definitive body text, clients SHOULD create a MIME message in which the first or only element has "text/plain" as the value of the **Content-Type** header and that element's body contains the text. Clients SHOULD specify the character set of the message body text on the corresponding MIME element.

To send the value of the **PidTagHtml** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.9) as the definitive body text, clients SHOULD create a MIME message in which the first or only MIME element has "text/HTML" as the value of the **Content-Type** header and that element's body contains a well-formed HTML document. Clients SHOULD generate a MIME structure with a MIME content-type of "multipart/alternative", and include a plain text representation, so that a greater number of clients can process the message.

To send the value of the **PidTagRtfCompressed** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.4) as the definitive body text, clients SHOULD create a MIME message that contains a TNEF body part, as specified at the beginning of section 2, and write the value of the **PidTagRtfCompressed** property into the **attMsgProps** attribute, as specified in [\[MS-OXTNEF\]](#) section 2.1.3.3.21, of the TNEF message.

2.2.3.3.2 Determining Which MIME Element Is the Message Body

The rules a MIME reader follows for selecting a message body are both qualifying, or positive, and disqualifying, or negative. To qualify as a message body, a MIME entity MUST meet at least one of the following conditions:

- **Content-Type** header value is "text/plain", "text/HTML", "text/enriched", or "text/calendar".
- **Content-Type** header value is "multipart/alternative" and at least one child MIME entity is "text/plain", "text/HTML", "text/enriched", or "text/calendar".
- **Content-Type** header value is "multipart/related", and its first child MIME entity is either "text/HTML" or "multipart/alternative" with at least one "text/HTML" child MIME entity.
- **Content-Type** header value is "multipart/mixed", and its first child MIME entity is "text/plain", "text/HTML", "text/enriched", or "text/calendar".

To qualify as a message body, a MIME entity MUST NOT have a **Content-Disposition** header (2) with the value "attachment".

In all cases, it is the text body part and not the containing multipart itself that is mapped to the message body.

MIME readers select the first MIME entity that qualifies according to the rules as the message body. MIME readers SHOULD then map the content of the selected MIME entity to a Message object property value according to the following rules:

1. If the body MIME entity is a single "text/plain", copy its content to the value of the **PidTagBody** property ([MS-OXCMSG] section 2.2.1.56.1).
2. If the body MIME entity is a single "text/HTML", copy its content to the value of the **PidTagHtml** property ([MS-OXCMSG] section 2.2.1.56.9).
3. If the body MIME entity is a single "text/enriched", convert its content to HTML and copy the result to the value of the **PidTagHtml** property. The format of the "text/enriched" MIME type is specified in [MSFT-RTF], [MS-OXRTFEX], and [MS-OXRTFCP].
4. If the body MIME entity is a single "text/calendar", parse the iCalendar document and copy the value of the **DESCRIPTION** property to the **PidTagBody** property. If the **DESCRIPTION** property is missing, MIME readers can use the value of the **COMMENT** property instead. For details, see [MS-OXCICAL].
5. If the body MIME entity is "multipart/alternative", MIME readers SHOULD select the last child entity that has one of the four eligible types ("text/plain", "text/HTML", "text/enriched", or "text/calendar") and map it. However, if the last child entity is "text/calendar" and one of the preceding entities is "text/HTML", MIME readers SHOULD map the "text/HTML" value, instead of the **DESCRIPTION** property of the "text/calendar" value, to the **PidTagRtfCompressed** property ([MS-OXCMSG] section 2.2.1.56.4).
6. If the body MIME entity is "multipart/related", identify the first child MIME entity that is either "text/HTML" or "multipart/alternative" and map it according to rules 1-5.
7. If the body MIME entity is "multipart/mixed", MIME readers SHOULD identify the first MIME entity that has one of the four eligible types ("text/plain", "text/HTML", "text/enriched", or "text/calendar") and map it. Alternatively, MIME readers MAY <142> create an aggregate body as specified in section 2.2.3.3.2.2.

2.2.3.3.2.1 Selecting the Primary Message Text MIME Element

When alternative text MIME elements are present and eligible for use as the message body, as specified in section [2.2.3.3.2](#), MIME readers SHOULD choose a MIME element to populate the message body text by using the following content types, in descending order of preference:

- "text/HTML"
- "text/enriched"
- "text/plain"
- "text/calendar" (but only if the **METHOD** property value of the "text/calendar" body part is PUBLISH, REQUEST, REPLY, or CANCEL)

If "text/HTML" is selected, MIME readers copy the MIME element body text to the value of the **PidTagHtml** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.9), map the charset parameter of the MIME element's **Content-Type** header to a code page, and set the value of the **PidTagInternetCodepage** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.6) to that code page. If the *charset* parameter is not present, MIME readers MAY use the value of a **Content-Type META** element in the HTML document but SHOULD verify its validity before using it.

If "text/plain" is selected, MIME readers convert the plain text to the UTF-16LE scheme and write the resulting text to the value of the **PidTagBody** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.1). MIME readers SHOULD, in addition, map the value of the *charset* parameter of the MIME element's **Content-Type** header to a code page and set the value of the **PidTagInternetCodepage** property to that code page.

If "text/enriched" is selected, MIME readers convert the MIME element body text to either "text/plain", "text/HTML", or "text/RTF" and handle that as previously specified. The conversion of enriched text format to other formats is specified in Appendix B of [\[RFC1896\]](#).

If both "text/HTML" and "text/calendar" body parts are present and eligible for use as a message body, instead of writing text to the **PidTagHtml** property, MIME readers SHOULD convert the HTML text to RTF and write it to the value of the **PidTagRtfCompressed** property ([\[MS-OXCMSG\]](#) section 2.2.1.56.4). Alternatively, MIME readers can choose to use plain text from a "text/plain" body part or from data in the "text/calendar" body part, as specified in [\[MS-OXCICAL\]](#). MIME readers MUST NOT set the **PidTagHtml** property on a calendar or meeting Message object.

2.2.3.3.2.2 Creating an Aggregate Body

Aggregate bodies SHOULD [<143>](#) only be created when one of the following values appears as a substring in the **MIME-Version**, the **X-User-Agent**, or the **X-Mailer** header values: "apple", "ipad", or "ipod". The aggregate body is in HTML format and is constructed by parsing the child MIME entities of a "multipart/mixed" MIME entity. MIME readers use the following procedure to process each child MIME entity.

1. If the **Content-Type** header value is "text/html" or "text/plain", then the contents are appended to the aggregate body.
2. If the **Content-Type** header value is one of the values specified in section [2.2.3.4.1.4.1](#), then the MIME entity is treated as an inline attachment. The value of the **Content-Id** header is copied to the **PidTagAttachContentId** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) for this attachment. If there is no **Content-Id** header, then the MIME reader SHOULD generate a value for the **PidTagAttachContentId** property. An HTML element using the value of the **PidTagAttachContentId** property, as specified in [\[HTML\]](#), is appended to the aggregate body.
3. If the **Content-Type** header value is anything else, then the MIME entity is treated as a normal attachment.

2.2.3.4 Attachments

During MIME analysis, MIME readers classify all non-multipart MIME entities and "multipart/appledouble" MIME entities (that contain appropriate child MIME subparts) into the following three categories:

1. MIME entities that can potentially represent the message body, as specified in section [2.2.3.3.2](#).
2. MIME entities that represent non-inline attachments.
3. MIME entities that represent attachments that can potentially be inline.

All MIME entities that can be classified as attachments (category 2 or 3) SHOULD be treated by MIME readers as attachment MIME parts, and an entry in an attachments table SHOULD be created for each such MIME part. However, depending on the value of the **Content-Type** header, analysis SHOULD be done differently, as follows:

1. "message/rfc822" MIME entities SHOULD be treated as embedded message attachments, as specified in section [2.2.3.4.3](#).
2. "Multipart/appledouble", "application/applefile", and "application/mac-binhex40" MIME entities SHOULD be treated as Macintosh attachments, as specified in section [2.2.3.4.2](#).
3. "Message/external-body" attachments SHOULD be treated as external body attachments, as specified in section [2.2.3.5](#).
4. All other attachments SHOULD be treated as regular file attachments, as specified in section [2.2.3.4.1](#).

If no **Content-Type** header is present on a MIME entity, MIME readers SHOULD treat it as a "text/plain" MIME content-type (unless this MIME entity is a subpart of multipart/digest, in which case the default value for the **Content-Type** header is "message/rfc822").

2.2.3.4.1 Regular File Attachment MIME Part Analysis

When creating an Attachment object for a regular file attachment, MIME readers set the value of the **PidTagAttachMethod** property ([\[MS-OXCMSG\]](#) section 2.2.2.9) to 0x00000001.

2.2.3.4.1.1 File Name

The attachment file name SHOULD be determined by MIME readers in the following order:

1. If the **Content-Disposition** header exists on the attachment MIME entity, and a non-empty filename parameter is available on this header, the filename parameter value SHOULD be used.
2. Otherwise, if the **Content-Type** header is available on the attachment MIME entity, and a non-empty name parameter is available on this header, the name parameter value SHOULD be used.
3. Otherwise, if the **Content-Transfer-Encoding** header is set to "mac-binhex40", MIME readers SHOULD try to parse the MIME part body as MacBinary structure, as specified in section [2.2.3.4.2.3](#). Only the first 128 bytes of the MIME body (decoded with binhex, as specified in [\[RFC1741\]](#)) SHOULD be parsed. If parsing of MacBinary structure succeeds, file name data from this structure SHOULD be used.
4. Otherwise, if the attachment MIME part body is encoded with uuencode (see section [2.2.3.9.1](#) and [\[IEEE1003.1\]](#)), and it contains file name data, this file name SHOULD be used.
5. Otherwise, if the **Content-Description** header is available on the attachment and its value is non-empty, it SHOULD be used as the file name value for an attachment. (Even if a file name for an

attachment was found in one of the previous steps, this value SHOULD be written to the **PidTagDisplayName** property ([\[MS-OXCFOld\]](#) section 2.2.2.2.2.5) for an attachment.)

MIME readers SHOULD sanitize the resulting file name by removing characters that are not legal but MAY<144> instead replace illegal characters with an underscore "_" (U+005F). MIME readers SHOULD<145> apply the same sanitization to the display name. Invalid characters are listed in the following table.

Description	Code point	Character
Control characters	U+0000 through U+001F<146>	
Double quote	U+0022	"
Forward slash	U+002F	/
Colon	U+003A	:
Left angle bracket	U+003C	<
Right angle bracket	U+003E	>
Pipe	U+007C	
Backslash	U+005C	\

The following steps SHOULD<147> then be applied both to the attachment file name and the display name (if the display name is not available, the empty string SHOULD be used):

1. Replace all Unicode separator characters with spaces. Unicode separator characters are specified in [\[UNICODE5.1\]](#) section 6.2.
2. Separate name into base and extension parts. The extension is defined as the trailing part of a name that starts after the last appearance of a "." character (U+002E) in the name, or an empty string if name contains no such character.
3. Remove all leading and trailing spaces and leading and trailing "." (U+002E) characters from both the base and the extension.<148>

If the extension part of the display name is not empty and does not match the extension part of file name, it SHOULD be appended to the base part of display name.

If the file name base and/or file name extension is empty, the MIME reader SHOULD<149> generate an attachment file name base and/or extension. The filename can be created using any file name generation convention that conforms to the filename guidelines specified in this section.

After that, if the base part of the display name is empty, it SHOULD<150> be replaced with the base part of the file name. Finally, the file name base, file name extension, and display name SHOULD be reassembled from the base and extension parts and saved in the appropriate properties, as specified in the following table.

Property	Value
PidTagDisplayName ([MS-OXCFOld] section 2.2.2.2.2.5)	<display name base>.<file name extension>
PidTagAttachLongFilename ([MS-OXCMSG] section 2.2.2.11)	<file name base>.<file name extension>
PidTagAttachExtension ([MS-OXCMSG] section 2.2.2.12)	.<file name extension>

The value saved to the **PidTagAttachLongFilename** property SHOULD<151> be further processed to form a valid **8.3 name**, and then written to the **PidTagAttachFilename** property, as follows:

1. The value SHOULD be first separated into name and extension parts, using the last "." character (U+002E) as a separator. If no such character is present, or the only appearance of this character is in the beginning of the file name, the name part is considered to be empty but the extension is not empty; the separator character itself is not included into the name or extension.
2. MIME readers SHOULD replace the following characters with an underscore (U+005F) but MAY<152> remove them: plus sign "+" (U+002B), comma "," (U+002C), equal sign "=" (U+003D), left square bracket "[" (U+005B), right square bracket "]" (U+005D), semicolon ";" (U+003B). MIME readers MAY<153> replace the question mark "?" (U+003F) and the asterisk "*" (U+002A) with the underscore "_" (U+005F).
3. MIME readers SHOULD<154> remove the following characters: space (U+0020), period "." (U+002E), apostrophe "'" (U+0027), asterisk "*" (U+002A), question mark "?" (U+003F), as well as characters with a UTF8 code greater than 127.
4. If the name is empty after removing such characters, MIME readers SHOULD<155> generate a non-empty value.
5. MIME readers SHOULD trim the name part of the file name to 8 characters. MIME readers also SHOULD trim the extension part to 3 characters. Alternatively, MIME readers MAY<156> delete the extension portion of the file name if the extension is greater than three characters in length.
6. If the name was shortened, MIME readers SHOULD<157> additionally trim the name part to 6 characters, and "~1" SHOULD be added to its end.
7. MIME readers then recombine the file name and extension<158>, separated by a single "." (U+002E).

2.2.3.4.1.2 Content Type

MIME readers SHOULD save the value of the **Content-Type** header in the **PidTagAttachMimeTag** property ([MS-OXCMSG] section 2.2.2.26) during MIME analysis. The following notes apply for specific values of this header:

- The "application/ms-tnef" value SHOULD<159> be replaced with "application/octet-stream". This is in the rare case when a TNEF body part is corrupt and cannot be completely processed. Ordinarily, a TNEF body part SHOULD NOT be written to an attachment, but analyzed into Message object properties and discarded.
- The values "application/x-pkcs7-MIME" and "application/pkcs7-MIME": the entire **Content-Type** header value, including all parameter names and values, SHOULD be written to the **PidNameContentType** property ([MS-OXPROPS] section 2.408).
- For **Content-Type** header values that start with "text/", if a *charset* parameter is present, the parameter value SHOULD be written to the **PidTagTextAttachmentCharset** property ([MS-OXCMSG] section 2.2.2.25).

2.2.3.4.1.3 Attachment Creation and Modification Date

If a **Content-Disposition** header is present on the attachment MIME part, MIME readers SHOULD<160> use its parameters to set creation and modification dates on the Attachment object. If a parameter is missing or its value is not a valid date, the corresponding property value SHOULD NOT be set; however, a MIME reader MAY<161> use current system time to set the **PidTagCreationTime** property ([MS-OXCMSG] section 2.2.2.3) and the **PidTagLastModificationTime** property ([MS-

OXCMSG] section 2.2.2.2) when the **Content-Disposition** header values are missing or invalid. Date and time values MUST be translated to UTC.

Content-Disposition parameter name	Property name
<i>creation-date</i>	PidTagCreationTime ([MS-OXCMSG] section 2.2.2.3)
<i>modification-date</i>	PidTagLastModificationTime ([MS-OXCMSG] section 2.2.2.2)<162>

2.2.3.4.1.4 Attachment Content-Id, Content-Base, and Content-Location

If a **Content-Id** header is present on the attachment MIME part, MIME readers SHOULD copy its value to the **PidTagAttachContentId** ([MS-OXCMSG] section 2.2.2.26) property. If this value starts with "<" (U+003C) and/or ends with ">" (U+003E), these characters SHOULD be removed. Attachment MIME parts with a **Content-Id** header SHOULD be evaluated to determine whether it is an inline attachment, as specified in section 2.2.3.4.1.4.1.

If a **Content-Location** header is present on the attachment MIME part, then its value SHOULD be saved in the **PidTagAttachContentLocation** property ([MS-OXCMSG] section 2.2.2.26). The attachment MIME part SHOULD be evaluated to determine whether it is an inline attachment, as specified in section 2.2.3.4.1.4.1.

If a **Content-Base** header is present on the attachment MIME part, then MIME readers SHOULD<163> copy its value to the **PidTagAttachContentBase** property ([MS-OXCMSG] section 2.2.2.26).

2.2.3.4.1.4.1 Inline Attachments

MIME readers SHOULD mark an attachment MIME part as an inline attachment, as specified in section 2.1.3.4.1.2, when they meet the following criteria:

- The attachment MIME part MUST be a child of a "multipart/related" MIME entity or the child of a "multipart/mixed" MIME entity that is being used to create an aggregate body, as specified in section 2.2.3.3.2.2.
- The attachment MIME part MUST have a **Content-Id** or a **Content-Location** header.
- The attachment MIME part SHOULD<164> be referenced from an HTML element in the HTML message body.
- The attachment MIME part SHOULD<165> have one of the following **Content-Type** header values: "image/jpeg", "image/jpg", "image/pjpeg", "image/gif", "image/bmp", "image/png", or "image/x-png".

Marking attachment MIME parts that do not meet all of these criteria as inline can result in the attachment being hidden from recipients of the message.

A MIME reader MAY<166> set the **attInvisibleInHtml** flag in the **PidTagAttachFlags** property ([MS-OXCMSG] section 2.2.2.18) for inline attachments.

2.2.3.4.1.5 Attachment Content-Transfer-Encoding and MIME Part Body

As specified in [RFC2045], a **Content-Transfer-Encoding** header might be present on the attachment MIME part. MIME readers SHOULD support the following values for this header:

- Base64. See [RFC2045]
- Quoted-printable. See [RFC2045]
- 7bit. See [RFC2045]
- 8bit. See [\[RFC3516\]](#)
- Binary. See [\[RFC3030\]](#)
- Mac-binhex40
- X-uuencode
- X-uue

As specified in [RFC2045], if the **Content-Transfer-Encoding** header is missing, MIME readers MUST behave as if it were set to "7bit".

The attachment's content SHOULD be decoded by using the appropriate decoding procedure and saved as the value of the **PidTagAttachDataBinary** property ([\[MS-OXCMSG\]](#) section 2.2.2.7). MIME readers SHOULD, as a rule, use the **RopOpenStream** ROP ([\[MS-OXCROPS\]](#) section 2.2.9.1) to create this property value.

The encoding values "mac-binhex40", "x-uuencode", and "x-uue" are non-standard. If a "mac-binhex40" **Content-Transfer-Encoding** header value is encountered, MIME readers SHOULD treat the MIME part body as if it had a **Content-Type** header value of "application/mac-binhex40" and process it as specified in section [2.2.3.4.2.3](#).

However, in the unlikely case of an actual "application/mac-binhex40" **Content-Type** header value, MIME readers SHOULD extract only the data fork from the MIME element content and use it as the value of the Attachment object's **PidTagAttachDataBinary** property. For "x-uuencode" and "x-uue" values, MIME readers SHOULD treat the attachment content as encoded with uuencode, as specified in [\[IEEE1003.1\]](#). The decoded value SHOULD be written to the Attachment object's **PidTagAttachDataBinary** property value.

2.2.3.4.2 Apple File Formats

[\[RFC1740\]](#) and [\[RFC1741\]](#) specify the use of the MIME **Content-Types** "multipart/appledouble", "application/applefile", and "application/mac-binhex40" to encode files that originate from a Macintosh operating system, to preserve additional data that might be available for these files in that operating system. MIME readers SHOULD preserve this additional data for attached files to enable full support of Macintosh-based client applications.

In particular, the Attachment object content that is stored in the **PidTagAttachDataBinary** property ([\[MS-OXCMSG\]](#) section 2.2.2.7) MUST contain a MacBinary stream (1). This stream (1) format incorporates both the resource and data forks, as well as certain metadata.

Note that MIME analysis of "application/applefile" attachments is specified differently, depending on whether the "application/applefile" MIME entity is a subpart of "multipart/appledouble".

2.2.3.4.2.1 Multipart/Appledouble

A MIME element with a **Content-Type** header value of "multipart/appledouble", as specified in [\[RFC1740\]](#), has two child MIME elements. The "header part" has a **Content-Type** header value of "application/applefile"; the "data part" can have any MIME content-type except "application/applefile" or another multipart MIME content-type.

As a MIME reader copies data from a "multipart/appledouble" MIME entity to an Attachment object, it analyzes the three parts in the following sequence:

1. The header part (typically the first child of "multipart/appledouble").
2. The data part (typically the second child of "multipart/appledouble").
3. The "multipart/appledouble" envelope itself.

Property values that are set as a result of MIME header analysis of a particular MIME part, as specified in section [2.2.3.4.1](#), overwrite property values that are set as a result of previous MIME part analysis.

The procedure of header analysis for any part of a "multipart/appledouble" MIME part is similar to the procedure for ordinary file attachments specified in section 2.2.3.4.1, with the following additions:

MIME readers set the value of the **PidTagAttachMimeTag** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) to "multipart/appledouble".

MIME readers set the value of the **PidTagAttachEncoding** property to the following byte sequence (in hexadecimal): "%x2A.86.48.86.F7.14.03.0B.01".

MIME readers copy the value of the **Content-Type** header on the data part to the value of the **PidNameAttachmentMacContentType** property ([\[MS-OXCMSG\]](#) section 2.2.2.26).[<167>](#)

MIME readers SHOULD copy the entire MIME body of the header part to the value of the Attachment object's **PidNameAttachmentMacInfo** property ([\[MS-OXCMSG\]](#) section 2.2.2.26).[<168>](#) MIME readers SHOULD also parse this data and the MIME body from the data part to form a MacBinary structure, as specified in [\[RFC1740\]](#). The resulting MacBinary structure SHOULD then be written to the Attachment object's **PidTagAttachDataBinary** property ([\[MS-OXCMSG\]](#) section 2.2.2.7).

MIME readers copy file creator and file type information taken from the MacBinary representation of the attachment to the value of the **PidTagAttachAdditionalInformation** property ([\[MS-OXCMSG\]](#) section 2.2.2.21), with special formatting as follows; the file creator and type fields are both unsigned 32-bit integers in big-endian format:

- A single byte, value "0x3A" (colon character).
- The file creator, encoded by the rule that follows.
- A single byte, value "0x3A" (colon character).
- The file type, encoded by the rule that follows.
- A single byte, value "0x00".

Encoding is done from the highest-order byte to the lowest-order byte, by using the following scheme:

- Single bytes with values for "\" (%x5C), ":" (%x3A), and ";" (%x3B) are replaced with two-byte sequences: "\\\" (%x5C.5C), "\:" (%x5C.3A), and "\;" (%x5C.3B) respectively.
- Single bytes with values less than 32, greater than 251, or equal to 127 are encoded by a backslash (%x5C), followed by the byte value in octal, padded with zeroes to 3 digits. So, for example, a "0x01" byte is encoded as "\001", and "0xFF" is encoded as "\377".

If parsing of the header part fails, MIME readers SHOULD [<169>](#) reject the entire message as not MIME compliant.

If the AppleSingle structure from the header part contains a file name for this attachment, it SHOULD [<170>](#) be used as the file name only if no file name was found during processing of the MIME headers.

2.2.3.4.2.2 Application/Applefile

This section specifies MIME analysis for MIME parts with a **Content-Type** header value of "application/applefile" that are not subparts of a MIME part with a **Content-Type** header value of "multipart/appledouble".

The procedure of MIME header analysis for attachments with a MIME content-type of "application/applefile" is the same as for the procedure for ordinary file attachments specified in section [2.2.3.4.1](#), with one exception: MIME readers set the value of the **PidTagAttachMimeTag** property ([MS-OXCMSG] section 2.2.2.26) to "application/applefile".

Processing of MIME message content SHOULD include parsing the AppleSingle structure, as specified in [\[RFC1740\]](#). MIME readers SHOULD use the data from this structure to fill the **PidTagAttachDataBinary** property ([MS-OXCMSG] section 2.2.2.7) and the **PidNameAttachmentMacInfo** property ([MS-OXCMSG] section 2.2.2.26) with appropriate structures, as specified in section [2.2.3.4.2.1](#) and [\[MacBin\]](#).

If MIME body data does not match the definition of the AppleSingle structure, MIME readers can choose to try to interpret the body of this MIME part as a MacBinary structure. If this succeeds, MIME readers SHOULD copy the resulting MacBinary structure to the value of the **PidTagAttachDataBinary** property, and the **PidNameAttachmentMacInfo** property SHOULD be filled with appropriate data from the MacBinary structure. The value of the **PidNameAttachmentMacInfo** property SHOULD be "application/applefile" data that contains only the header and resource fork sections. But if the MIME reader fails to parse the MIME body, the entire message SHOULD [<171>](#) be rejected as not MIME compliant.

If the AppleSingle or MacBinary structure contains a file name for this attachment, it SHOULD [<172>](#) be used only if no file name was found during analysis of the attachment's MIME headers.

The remainder of this section specifies how MIME readers SHOULD [<173>](#) map elements from AppleSingle format, which can have a **Content-Type** header value of "multipart/appledouble" or "application/applefile", to MacBinary data in the value of the **PidTagAttachDataBinary** property.

The general structure of AppleSingle format is specified in [\[RFC1740\]](#). In short, this data structure contains a header part, followed by some number of entries. Each of these entries is identified by a number (AppleSingleEntryId, unsigned 32-bit integer), which defines the internal structure of its binary data. The value of each AppleSingleEntryId, along with the definition of the structure of each entry, is specified by [\[RFC1740\]](#). Custom entries are also allowed in this format.

The MacBinary structure consists of the following five parts; each part is padded to a 128-byte boundary, and all parts except the header are optional: [<174>](#)

1. Header
2. Additional header data
3. Actual file data (data fork)
4. Resource fork
5. Comment

The structure of the MacBinary header, with comments on usage of each field by MIME readers, is shown in the following table. All offsets and lengths are in bytes, and all integers use big-endian byte ordering.

Field offset	Field length	Description
0	1	Old version number, MUST be zero.
1	1	Length of file name, unsigned byte; MUST be less than 64.
2	63	File name, in ASCII; characters beyond the length specified in byte 1 MUST be ignored. <175>
65	4	File type data, normally expressed as four characters.
69	4	File creator data, normally expressed as four characters.
73	1	Finder flags , bits 15:8.
74	1	Pad, MUST be 0 (zero).
75	2	Icon vertical location, unsigned 16-bit integer.
77	2	Icon horizontal location, unsigned 16-bit integer.
79	2	File's folder ID.
81	1	File protected flag, low order bit.
82	1	Pad, MUST be 0.
83	4	Data fork length, signed 32-bit integer, zero if there is no data fork.
87	4	Resource fork length, signed 32-bit integer, zero if there is no resource fork.
91	4	File creation date, signed 32-bit integer representing a number of seconds since (or before, if negative) midnight, 01/01/2000, UTC.
95	4	File modification date, signed 32-bit integer representing a number of seconds since (or before, if negative) midnight, 01/01/2000, UTC.
99	2	Comment length, unsigned 16-bit integer, MUST be 0 (zero).
101	1	Finder flags, bits 7:0.
102	4	Signature. MIME reader SHOULD <176> set this value to "mBIN" (%x6D.42.49.4E).
106	1	File name script identifier. MIME reader SHOULD set to 0 (zero).
107	1	Extended finder flags, MIME reader SHOULD set to 0 (zero).
108	12	Zero fill.
120	2	Secondary header length, MUST be 0 (zero).
122	1	MacBinary version number. MUST be set to 130 (0x82), indicating MacBinary III, when the MIME reader creates the MacBinary structure.
123	1	Minimum MacBinary version supported by this structure. MUST be set to 129 (0x81), indicating MacBinary II, when the MIME reader creates the MacBinary structure.
124	2	CRC of the previous 124 bytes. MIME readers SHOULD calculate this value by applying a CRC algorithm on the first 124 bytes of the header. The CRC algorithm used by MacBinary is the CCITT algorithm, which uses the polynomial 0x1021. For more information on CRC-CCITT, see [X25] section 2.2.7.4. ([X25] refers to the CRC algorithm as a "frame check sequence".)

Field offset	Field length	Description
Bytes 126:127	2	Zero fill.

When processing AppleSingle data, MIME readers MUST map AppleSingle fields to MacBinary fields as specified in the following table.

AppleSingleEntryId and type	MacBinary field	Comment
1, Data fork	Bytes 83:86 – length; MacBinary data fork part	This mapping SHOULD only be used by MIME readers in MIME analysis of a standalone "application/applefile"; as specified in [RFC1740]), the data fork SHOULD be in a separate MIME part in "multipart/appledouble" case.
2, Resource fork	Bytes 87:90 – length; MacBinary resource fork part	None.
3, ASCII string	Byte 1 – length, Bytes 2:64 – ASCII string value (only length bytes used)	File name. Note that MacBinary limits this string to 63 bytes. Excess bytes MUST be truncated.
8, ASFileDates structure, create	Bytes 91:94	File creation date, MIME readers SHOULD map it for AppleSingle to MacBinary conversion.
8, ASFileDates structure, modify	Bytes 95:98	File modification date, MIME readers SHOULD map it for AppleSingle to MacBinary conversion.
8, ASFileDates structure, access	None	MIME writers SHOULD set to 0 (zero) on conversion to AppleSingle.
8, ASFileDates structure, backup	None	MIME writers SHOULD set to 0 (zero) on conversion to AppleSingle.
9, ASFinderInfo structure, ioFIFndrInfo.fdType	Bytes 65:68	File type information.
9, ASFinderInfo structure, ioFIFndrInfo.fdCreator	Bytes 69:72	File creator information.
9, ASFinderInfo structure, ioFIFndrInfo.fdFlags	Byte 73 – bits 15:8 Byte 101 – bits 7:0	File finder flags word. MIME readers SHOULD map this element for AppleSingle to MacBinary conversion.
9, ASFinderInfo structure ioFIFndrInfo.fdLocation.v	Bytes 75:76<177>	Icon vertical location. MIME readers SHOULD map this element for AppleSingle to MacBinary conversion.
9, ASFinderInfo structure, ioFIFndrInfo.fdLocation.h	Bytes 77:78<178>	Icon horizontal location. MIME readers SHOULD map this element for AppleSingle to MacBinary conversion.
9, ASFinderInfo structure, ioFIFndrInfo.fdFldr	Bytes 79:80	File folder ID. MIME readers SHOULD map this element for AppleSingle to MacBinary conversion.
9, ASFinderInfo structure,	None	MIME writers SHOULD fill with zeros on conversion to AppleSingle.

AppleSingleEntryId and type	MacBinary field	Comment
ioFIXFndrInfo		
10, ASMacInfo structure, filler	None	MIME writers SHOULD fill with zeros on conversion to AppleSingle.
10, ASMacInfo structure, ioFIAttrib, bit 1	Byte 81, low order bit	Protected flag. MIME readers SHOULD map this element for AppleSingle to MacBinary conversion. <179>

Conversion from a full AppleSingle structure, found in a standalone "application/applefile" MIME element that is not a child of a MIME part whose MIME content-type is "multipart/appledouble", to a reduced AppleSingle structure that SHOULD be used as a child of "multipart/appledouble", is done simply by removing the entry with AppleSingleEntryId equal to 1 (the data fork) and adjusting the AppleSingle header accordingly.

2.2.3.4.2.3 Application/Mac-binhex40

This section specifies MIME analysis for MIME parts with a **Content-Type** header value of "application/mac-binhex40", as specified in [RFC1741](#).

The procedure of MIME header analysis for "application/mac-binhex40" attachments is the same as for the procedure for ordinary file attachments that is specified in section [2.2.3.4.1](#), with the following exceptions:

1. MIME readers set the value of the **PidTagAttachMimeTag** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) to "application/mac-binhex40".
2. The value of the **Content-Transfer-Encoding** header SHOULD <180> be ignored. MIME readers use BinHex decoding, as specified in [\[RFC1741\]](#), instead.

Processing of the MIME body SHOULD include parsing a binary structure of the decoded content, as specified in [\[RFC1741\]](#). MIME readers SHOULD <181> use the header and resource fork data from this structure to fill the **PidNameAttachmentMacInfo** property ([\[MS-OXCMSG\]](#) section 2.2.2.26) with appropriate data, as specified in section [2.2.3.4.2.1](#). MIME readers SHOULD also use this data to fill the MacBinary structure, which SHOULD <182> be written to the value of the **PidTagAttachDataBinary** property ([\[MS-OXCMSG\]](#) section 2.2.2.7).

MIME readers SHOULD <183> copy the attachment file name that is extracted from the BinHex structure to the value of the **PidTagAttachFilename** property ([\[MS-OXCMSG\]](#) section 2.2.2.11), but only if no file name was found during analysis of the MIME headers. <184>

2.2.3.4.3 Attached Messages

If an attachment MIME part has its **Content-Type** header set to "message/rfc822" (or no **Content-Type** header is present, and this MIME part is a sub-part of the "multipart/digest" MIME part), MIME readers SHOULD treat this attachment as an embedded message attachment, and set the value of the Attachment object's **PidTagAttachMethod** property ([\[MS-OXCMSG\]](#) section 2.2.2.9) to "5".

MIME analysis for MIME headers SHOULD be performed by the server in the same way as for ordinary file attachments, with the exception that the procedure for extracting the display name and file name for the attachment is different.

The display name for embedded message attachments is extracted from MIME part headers in the following order:

1. If a **Content-Type** header is available on the attachment MIME part, and a non-empty name parameter is available on this header, its value SHOULD be used.
2. Otherwise, if a **Content-Disposition** header is available on the attachment MIME part, and a non-empty *filename* parameter is available on this header, its value SHOULD be used.
3. Otherwise, if a **Content-Description** header is available on the attachment, and its value is non-empty, it SHOULD be used.
4. Otherwise, if a **Subject** header is available on the attachments, and its value is non-empty, it SHOULD be used. A MIME reader MAY [<185>](#) use this header in preference to the **Content-Description** header value.
5. If none of these conditions apply, the MIME reader SHOULD generate a name at random, or use a name derived from some other text value related to the Attachment object.

The resulting value SHOULD be written to the **PidTagDisplayName** property ([\[MS-OXCFOOLD\]](#) section 2.2.2.2.5) on the attachment, and then processed further to obtain a valid file name, as follows:

1. All Unicode separator characters in the file name SHOULD be replaced with the "?" character (U+003F).
2. All trailing and starting space and "." characters SHOULD be removed.

The file name is then separated into base and extension parts. To do this, the server SHOULD look for the last occurrence of any of the following characters:

- backslash, "\", U+005C
- forward slash, "/", U+002F
- colon, ":", U+003A
- period, ".", U+002E

If a "." (U+002E) character is the last one found, the part of the file name that precedes this character is considered to be base, and the rest is considered to be extension. In all other cases, extension is considered to be an empty string, and base part is considered to be the same as whole file name.

The resulting file name value SHOULD [<186>](#) be written to the **PidTagAttachLongFilename** property ([\[MS-OXCMSG\]](#) section 2.2.2.10), and the resulting extension value SHOULD [<187>](#) be saved in the **PidTagAttachExtension** property ([\[MS-OXCMSG\]](#) section 2.2.2.12). The file name SHOULD then be processed further to obtain a valid 8.3 name, as follows:

1. The value SHOULD be first separated into base and extension parts, by using the last "." character as a separator (if no such character is present, the extension is considered to be empty; the separator character itself is not included in the name or extension).
2. "+", ",", "=", "[", "]", and ";" characters SHOULD be replaced with the "_" (underscore) character.
3. Space, ".", "\", "/", "*", "<", ">", "?", ":", and "|" characters, as well as characters with UTF8 code greater than 127, SHOULD be removed.
4. If the base becomes an empty string, a non-empty string, such as a random string or other auto-generated value, SHOULD be used.
5. The base part of the file name SHOULD be trimmed to 8 characters; the extension part to 3 characters.

If either the name or the extension changed, the base part SHOULD additionally be trimmed to 6 characters, and "~1" SHOULD be added to its end.

The file name is saved in the <base>.<extension> format.

The resulting file name SHOULD [<188>](#) be written to the **PidTagAttachFilename** property ([MS-OXCMMSG] section 2.2.2.11).

The MIME part body for this attachment SHOULD be used for further MIME analysis that SHOULD result in assigning values to the properties of the embedded message from this attachment. This MIME analysis is performed in a way that is similar to that for ordinary MIME messages, with the following exceptions:

1. The **X-MS-Exchange-Organization-Original-Sender** header value SHOULD [<189>](#) be saved in the **PidNameQuarantineOriginalSender** property ([MS-OXPROPS] section 2.463), if the header value is present.
2. Unknown MIME headers, starting with "X-MS-Exchange-Organization-" or "X-MS-Exchange-Forest-", SHOULD NOT [<190>](#) be excluded from analysis.

After MIME analysis is done for the embedded message, the **PidTagMessageFlags** property ([MS-OXCMMSG] section 2.2.1.6) SHOULD be modified; the mfUnsent flag ("0x8") SHOULD be removed, and the mfRead flag SHOULD be reset to 0x0.

2.2.3.4.4 Inbound vCard Conversion

Incoming vCard attachments SHOULD [<191>](#) be converted to a Contact object when received by the messaging server as specified in [\[MS-OXVCARD\]](#). This section describes additional information about converting an incoming vCard MIME part to its Contact object counterpart.

vCard can appear in any multipart MIME part as well as the root content-type, as specified in [\[RFC2426\]](#): The vCard format can be sent as the primary body or as the alternative body of a message. In Message object format, however, "vCard/contact" information is only exposed as an attachment.

- When the vCard format is encountered at the root level of a MIME message, the MIME reader promotes the vCard as a **contact attachment** to a Message object with no body.
- When the vCard format is encountered within a multipart/alternative MIME part, the MIME reader promotes the vCard content as a contact attachment and disregards it as a message body.

2.2.3.4.4.1 Content-Type

The vCard MIME part is characterized by a **Content-Type** of "text/directory" with a profile of "vCard" for vCard format Version 3.0, as specified in [\[MS-OXVCARD\]](#). If the **Content-Type** is "text/x-vCard", then the vCard MIME part uses vCard format Version 2.1, as specified in [\[RFC2426\]](#). [<192>](#)

2.2.3.4.4.2 General Parsing Guidelines

A MIME reader SHOULD follow these general parsing guidelines when processing the vCard format.

- If there are multiple instances of a property value that can be promoted to one single-valued **Messaging Application Programming Interface (MAPI)** property, then the last instance found is used, and others are dropped.
- vCard v2.1 allows parameters with the "type=" tag omitted. vCard v3.0 requires the "type=" tag.
- <grouping>.<property> is treated as <property>.

- This algorithm handles both base64 encoding and quoted-printable encoding.
- Unknown properties will be dropped.

2.2.3.5 External Body Attachments

Attachment MIME parts with a **Content-Type** header set to "message/external-body" SHOULD be analyzed in the same way as ordinary file attachments, with the exceptions specified in this section. <193>

If the **Content-Type** header has no access-type parameter, or if the value of that parameter is not "anon-ftp", MIME readers SHOULD save the entire MIME part in the **PidTagAttachDataBinary** property ([MS-OXCMSG] section 2.2.2.7) on the Attachment object.

Otherwise, the following differences in MIME analysis apply:

- Different file name extraction logic SHOULD be applied, as specified later in this section.
- MIME readers SHOULD ignore the MIME part body. Instead, a specially formatted URL data string is saved in the **PidTagAttachDataBinary** property in ASCII format, as specified in this section.
- In this case, MIME readers expect the name, site, directory, and mode parameters to be present in the **Content-Type** header. Clients SHOULD NOT create MIME that does not include these parameters.

The **URL** data string to save in the **PidTagAttachDataBinary** property is constructed as follows. The values xchar and lowalpha used in this definition are specified in [RFC1738].

```
"[InternetShortcut]" CR LF "URL=ftp://" site "/" directory "/" name [mode]
;contains header "site" parameter value
site=1*xchar

;contains header "directory" parameter value
directory=1*xchar
;contains header "name" parameter value
name=1*xchar
;if header "mode" parameter is "ascii", contains ";type=a"
;if header "mode" parameter is "image", contains ";type=i"
;otherwise not present
mode=";type=" 1*lowalpha CR LF
```

The file name extraction logic is similar to that for ordinary file attachments, with the following exceptions:

1. MIME readers use the name parameter value from the **Content-Type** header as a value of the attachment file name. The file extension (a part of the file name after the last appearance of the "." character) SHOULD be replaced with "URL"; if the original file name has no extension, "URL" SHOULD be added at the end of the file name string. <194>
2. The sanitizing logic specified in section 2.2.3.4.1.1 SHOULD NOT be applied in this case. MIME readers MAY <195> sanitize the external body attachment file name by replacing the following characters with the underscore "_" (U+005F): double quote "" (U+0022), forward slash "/" (U+002F), colon ":" (U+003A), left angle bracket "<" (U+003C), right angle bracket ">" (U+003E), pipe "|" (U+007C), and backslash "\" (U+005C).
3. The file name value constructed in Step 1 SHOULD be used as an attachment display name as well.
4. All additional filtering logic specified in section 2.2.3.4.1.1 still applies in this case.

5. The procedure for calculating a value for the **PidTagAttachFilename** property ([MS-OXCMSG] section 2.2.2.11) is the same as for embedded message attachments specified in section [2.2.3.4.3](#). If the base part of the external body attachment file name is empty, MIME readers MAY<196> set the value of the **PidTagAttachFilename** property ([MS-OXCMSG] section 2.2.2.11) to "NONAME.URL" and the value of the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10) to "URL".

2.2.3.6 Reading Pure MIME Messages

The MIME reader is responsible for reading pure MIME messages and saving them in a way that minimizes changes to the original MIME content. This process is specified in section [2.3](#).

After the MIME reader converts the MIME message to a Message object as specified in section [2.2](#), the reader SHOULD<197> copy the structure of the MIME message plus the contents of any message parts that are not converted, into the **PidTagMimeSkeleton** property ([MS-OXCMSG] section 2.2.1.28), specified in section 2.3. The process for copying message parts into the structure of **PidTagMimeSkeleton** is specified in section [2.4.3](#).

2.2.3.7 Reading Report Messages

When reading report messages, additional MIME analysis is required, as specified in the following sections.

- Delivery status notifications are specified in section [2.2.3.7.1](#).
- Message disposition notifications are specified in section [2.2.3.7.2](#).

2.2.3.7.1 Reading Delivery Status Notification Messages

When reading delivery status notifications, as specified in [\[RFC3464\]](#), the contents of the message/delivery-status body part are processed as follows:

1. The MIME reader finds the **Action** field with the highest severity value and sets the value of the **PidTagMessageClass** properties ([MS-OXCMSG] section 2.2.1.3) as specified in section [2.2.3.7.1.1](#).
2. The MIME reader writes the value of the **Reporting-MTA** field to the **PidTagReportingMessageTransferAgent** property ([MS-OXOMSG] section 2.2.2.35) on the message.
3. For each set of per-recipient delivery status notification fields, as specified in [\[RFC3464\]](#) section 2.3:
 1. If the value of the **Action** field does not match the highest severity value found in step 1, the MIME reader SHOULD<198> skip processing of this set of per-recipient delivery status notification fields and continue processing the next set.
 2. The MIME reader adds an entry to the Message object's recipient table.
 3. If the **Remote-MTA** field is present, the MIME reader writes the portion of its value that corresponds to the mta-name ABNF rule specified in [\[RFC3464\]](#) section 2.3.5 to the **PidTagRemoteMessageTransferAgent** property ([MS-OXOMSG] section 2.2.2.34) on the recipient.
 4. If the **X-Supplementary-Info** field is present, the MIME reader writes its value to the **PidTagSupplementaryInfo** property ([MS-OXOMSG] section 2.2.2.36) on the recipient. If the **X-Supplementary-Info** field is not present, the MIME reader calculates a value as

specified in section [2.2.3.7.1.2](#) and writes the value to the **PidTagSupplementaryInfo** property.

5. The MIME reader sets the value of the **PidTagReportTime** property ([\[MS-OXCSPAM\]](#) section 2.2.2.6) on the recipient to the sent time of the message.
6. If the value of the **Action** field is not "failed", the MIME reader sets the value of the **PidTagDeliverTime** property ([\[MS-OXOMSG\]](#) section 2.2.2.29) on the recipient to the sent time of the message.
7. If the value of the **Action** field is "failed", the MIME reader processes the value of the **Status** field as specified in section [2.2.3.7.1.3](#).

2.2.3.7.1.1 Determining the Value of the PidTagMessageClass Property

The value of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) is determined by checking the **Action** field for each set of per-recipient delivery status notification fields, as specified in [\[RFC3464\]](#) section 2.3, to find the **Action** field value with the highest severity, as specified in the following table.

Action field value	Severity
delivered	0
expanded	1
relayed	2
delayed	3
failed	4

Once the **Action** field value with the highest severity has been found, the value of the **PidTagMessageClass** property is found by prepending "REPORT." and appending the appropriate suffix from the following table to the value of the **PidTagMessageClass** property that was set by the MIME reader before processing the message/delivery-status body part.

Action field value	PidTagMessageClass suffix
delivered	*.DR
expanded	*.Expanded.DR
relayed	*.Relayed.DR
delayed	*.Delayed.DR
failed	*.NDR

2.2.3.7.1.2 Calculating a Value for the PidTagSupplementaryInfo Property

If a set of per-recipient delivery status notification fields, as specified in [\[RFC3464\]](#) section 2.3, does not have an **X-Supplementary-Info** field, the MIME reader generates a value for the **PidTagSupplementaryInfo** property ([\[MS-OXOMSG\]](#) section 2.2.2.36) by using the following process.

The format of the value is expressed by the following ABNF notation:

```

supplementary-info = "<" {remote-mta} SP "#" status {SP diagnostic-code} ">"
remote-mta         = 1*VCHAR
status             = DIGIT "." 1*3DIGIT "." 1*3DIGIT
diagnostic-code    = 1*VCHAR

```

The value of the *remote_mta* parameter is equal to the value for the **PidTagRemoteMessageTransferAgent** property ([MS-OXOMSG] section 2.2.2.34) determined in step 3 in section [2.2.3.7.1](#). If the value is null or empty, the *remote_mta* parameter is omitted.

The value of the *status* parameter is equal to the portion of the **Status** field that corresponds to the status-code ABNF rule as specified in [RFC3464] section 2.3.4.

The value of the *diagnostic-code* parameter is equal to the value of the **Diagnostic-Code** field. If the **Diagnostic-Code** field is missing or empty, the parameter and the space character before it are omitted.

2.2.3.7.1.3 Processing the Status Field

MIME readers process the **Status** field using the following process:

1. The portion of the value of the **Status** field that corresponds to the status-code ABNF rule is split into three separate integer values: **Kind**, **Subject**, and **Detail**. The number before the first period ('.') becomes the **Kind** value, the number after the first period and before the second period becomes the **Subject** value, and the number after the second period becomes the **Detail** value. For example, if the value of the **Status** field is "5.7.0", the resulting values are: **Kind** = 5, **Subject** = 7, **Detail** = 0.
2. If the values for **Subject** or **Detail** are less than 0 or more than 9, the values are set to 0.
3. The value of the **PidTagNonDeliveryReportStatusCode** property ([MS-OXOMSG] section 2.2.2.32) on the recipient is set to $(\mathbf{Kind} * 100) + (\mathbf{Subject} * 10) + \mathbf{Detail}$.
4. Two temporary values are created: **DiagnosticCode** and **ReasonCode**. **DiagnosticCode** is initialized to -1, and **ReasonCode** is initialized to 0.
5. The values of **DiagnosticCode** and **ReasonCode** are modified as specified in the table that follows.
6. The value of the **PidTagNonDeliveryReportDiagCode** property ([MS-OXOMSG] section 2.2.2.30) on the recipient is set to the value of **DiagnosticCode**.
7. The value of the **PidTagNonDeliveryReportReasonCode** property ([MS-OXOMSG] section 2.2.2.31) on the recipient is set to the value of **ReasonCode**.

The following table specifies the modifications made to the values of **DiagnosticCode** and **ReasonCode** based on the values of **Subject** and **Detail**. For more information about possible values of the **Status** field, see [\[RFC3463\]](#).

Value of Subject	Value of Detail	Action taken
1	1	DiagnosticCode = 35 ReasonCode = 1
1	2	DiagnosticCode = 48
1	3	DiagnosticCode = 32
1	4	DiagnosticCode = 1

Value of Subject	Value of Detail	Action taken
1	6	DiagnosticCode = 40
1	Any other value	DiagnosticCode = 0
2	1	DiagnosticCode = 38
2	2	DiagnosticCode = 13
2	3	DiagnosticCode = 13
2	4	DiagnosticCode = 30
2	Any other value	DiagnosticCode = 38
3	1	DiagnosticCode = 38
3	2	No action taken
3	3	DiagnosticCode = 18
3	4	DiagnosticCode = 13
3	5	DiagnosticCode = 18
3	Any other value	DiagnosticCode = 38
4	0	No action taken
4	3	ReasonCode = 6
4	4	No action taken
4	6	DiagnosticCode = 3
4	7	DiagnosticCode = 5
4	8	DiagnosticCode = 3
4	Any other value	DiagnosticCode = 2
5	3	DiagnosticCode = 16
5	4	DiagnosticCode = 11
5	Any other value	DiagnosticCode = 17
6	2	DiagnosticCode = 9
6	3	DiagnosticCode = 8
6	4	DiagnosticCode = 25
6	5	ReasonCode = 2
6	Any other value	DiagnosticCode = 15
7	1	DiagnosticCode = 29
7	2	DiagnosticCode = 28
7	3	DiagnosticCode = 26

Value of Subject	Value of Detail	Action taken
7	Any other value	DiagnosticCode = 46

2.2.3.7.2 Reading Message Disposition Notification Messages

When reading message disposition notifications, as specified in [\[RFC3798\]](#), the contents of the message/disposition-notification body part are processed as follows:

1. If the **Original-Recipient** field is present and has a value, the MIME reader sets the value of the **PidTagOriginalDisplayTo** property ([\[MS-OXOMSG\]](#) section 2.2.2.3) on the Message object to the display name portion of the **Original-Recipient** field. If this field has no display name portion, the MIME reader sets the value of the **PidTagOriginalDisplayTo** property to the email address from the field.
2. If the **Original-Recipient** field is not present or has no value, the MIME reader sets the value of the **PidTagOriginalDisplayTo** property on the Message object to the display name portion of the **Final-Recipient** field. If this field has no display name portion, the MIME reader sets the value of the **PidTagOriginalDisplayTo** property to the email address from the field.
3. If both the **From** and **Sender** fields are not present or have no value, the value of the **Original-Recipient** field (or the **Final-Recipient** field if the **Original-Recipient** field is not present) is used in their place and processed as specified in section [2.2.3.1.3](#) and section [2.2.3.1.4](#).
4. The MIME reader writes the sent time of the message to the **PidTagOriginalDeliveryTime** property ([\[MS-OXOMSG\]](#) section 2.2.2.2), the **PidTagReceiptTime** property ([\[MS-OXOMSG\]](#) section 2.2.2.33), and the **PidTagReportTime** property ([\[MS-OXCSPAM\]](#) section 2.2.2.6) on the Message object.
5. The MIME reader writes the value of the **Disposition** field to the **PidTagReportText** property ([\[MS-OXOMSG\]](#) section 2.2.2.23) on the Message object.
6. If the **X-MSEch-Correlation-Key** field is present, the MIME reader decodes the value of this field using base64 encoding. If the value is successfully decoded, the MIME reader writes the resulting value to the **PidTagParentKey** property ([\[MS-OXOMSG\]](#) section 2.2.2.18) on the Message object.
7. If the **Original-Message-ID** field is present, the MIME reader writes the value to the **PidTagOriginalMessageId** property ([\[MS-OXOMSG\]](#) section 2.2.1.85) and the **PidTagInternetReferences** property ([\[MS-OXCMSG\]](#) section 2.2.1.26) on the Message object.
8. The MIME reader modifies the existing value of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) on the Message object by prepending "REPORT." and appending a suffix based on the value of the portion of the **Disposition** field represented by the disposition-type ABNF rule, as specified in [\[RFC3798\]](#). The following table lists the possible values and the corresponding message class suffixes.

Value of disposition-type	Message class suffix
displayed	*.IPNRN
dispatched	*.IPNRN
processed	*.IPNRN

Value of disposition-type	Message class suffix
deleted	*.IPNNRN
denied	*.IPNNRN
failed	*.IPNNRN

2.2.3.8 Reading TNEF Messages

When reading TNEF messages, the MIME reader uses the following process:

1. The MIME reader processes all headers in the root MIME part as specified in section [2.2.3.1](#) and section [2.2.3.2](#) (and their respective subsections).
2. The MIME reader processes the content of the application/ms-tnef body part as specified in [\[MS-OXTNEF\]](#) and sets the corresponding properties on the Message object, with the following additional constraints:
 1. The MIME reader ignores any attributes in the application/ms-tnef body part that would result in setting a property on the Message object that was already set by the processing of the headers in the root MIME part in step 1.
 2. The MIME reader SHOULD resolve any recipients contained within embedded message attachments contained within the application/ms-tnef body part with an address type of "SMTP" against the address book. If the recipients are contained in the address book and have an address of type "EX", the "EX" type address should be used in place of the SMTP address.
3. The MIME reader SHOULD ignore the TNEF stream (2) if it contains the byte sequence 0x0D 0x0A at the end of the stream (2), outside of any attribute data or checksum values.
4. If the data in the application/ms-tnef body part does not conform to the requirements specified in [\[MS-OXTNEF\]](#), or if the value of the **PidTagTnefCorrelationKey** property ([\[MS-OXCMSG\]](#) section 2.2.1.29) does not match the value of the **X-MS-Tnef-Correlation** header, or if the MIME reader fails to parse the application/ms-tnef body part, the MIME reader treats the message as a pure MIME message and imports the text/plain body part as the body of the message. The contents of the application/ms-tnef body part are preserved as an attachment with a **Content-Type** header value of application/octet-stream.

2.2.3.9 Additional Content Types

2.2.3.9.1 Analysis of Non-MIME Content

Internet message content that lacks a **MIME-Version** header can still be supported by MIME readers. The absence of a **MIME-Version** header makes the payload of SMTP or elsewhere non-MIME, with different behavior for inline attachments; headers such as **Content-Type**, **Content-Disposition**, and such have no special meaning. MIME readers can, nevertheless, assume the presence of a **MIME-Version** header and treat headers such as **Content-Type**, as specified in [\[RFC2045\]](#) and elsewhere in this document. An example of such a message is provided in section [3.1.5](#).

MIME writers SHOULD NOT generate messages in this format; MIME SHOULD be generated instead. MIME readers SHOULD analyze messages in this format into headers, plain text body, and attached files (possibly including a TNEF attachment).

2.2.3.9.2 Message/Partial

The "message/partial" content type is not supported. MIME readers MUST NOT reassemble the individual messages of a "message/partial" message into a single message. MIME readers SHOULD [<199>](#) additionally reject messages that contain MIME entities with a **Content-Type** header of "message/partial". This is to prevent virus scanning from being defeated by splitting up attachment content.

2.2.3.9.3 Multipart/Digest

The "multipart/digest" **Content-Type** header value is treated exactly as "multipart/mixed", except that the assumed **Content-Type** for body parts with no **Content-Type** header SHOULD be set to a value of "message/rfc822" rather than "text/plain".

2.3 Unconverted MIME Part Generation Algorithm Details

The following section specifies when the **PidTagMimeSkeleton** property ([\[MS-OXCMSG\]](#) section 2.2.1.28) is re-generated, and specifies when the MIME writer should use the values of individual properties on the Message object as opposed to the values stored in the **PidTagMimeSkeleton** property.

2.3.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this algorithm. The described organization is provided to facilitate the explanation of how the algorithm 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

2.3.1.1 Global

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

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

2.3.1.2 Per Mailbox

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

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

2.3.1.3 Per Message Object

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

Mailbox.MessageObject.MIMESkeleton: The original MIME structure of the message.

2.3.2 Initialization

None.

2.3.3 Processing Rules

2.3.3.1 Impact of Message Changes on the MIME Skeleton

When a stored message is modified via remote operations (ROPs), as specified in [\[MS-OXCROPS\]](#), the value of the **PidTagMimeSkeleton** property ([\[MS-OXCMSG\]](#) section 2.2.1.28) is deleted or left unchanged, depending on the changes made to the message:

1. Changes made to the following Message object properties SHOULD leave the **PidTagMimeSkeleton** property unchanged:
 - **PidTagIconIndex** ([\[MS-OXOCAL\]](#) section 2.2.1.49).
 - **Reminders** on messages of type "IPM.NOTE".
 - **PidNameKeywords** ([\[MS-OXPROPS\]](#) section 2.444) that are mapped to the **Keywords** header in a MIME message.
2. Changes made to the following Message object properties leave the value of the **PidTagMimeSkeleton** property unchanged. However, MIME writers SHOULD prefer the values of these properties over data from the MIME skeleton when generating MIME messages.
 - Follow-up flags: These are stored in the **PidLidFlagRequest** property ([\[MS-OXOFLAG\]](#) section 2.2.1.9), which maps to the **X-Message-Flag** header in a MIME message. If this header already exists in the **PidTagMimeSkeleton** property, its value SHOULD be updated when generating a message. If the **X-Message-Flag** header is updated, it SHOULD be added to the top-level headers of a generated MIME message as well as to the **PidTagMimeSkeleton** property.
 - **PidTagImportance** ([\[MS-OXCMSG\]](#) section 2.2.1.11): The importance property maps directly to the **Importance** header. If the client allows the value of this property on a message to be modified, then the **Importance** header SHOULD be included in a generated MIME message and the change included in the stored **PidTagMimeSkeleton** property.
3. Changes to any other Message object properties that affect the MIME structure or content of a message SHOULD result in deletion of the **PidTagMimeSkeleton** property associated with the message. The following list specifies the other Message object properties that fall into this category.
 - **PidLidClassification** ([\[MS-OXCMSG\]](#) section 2.2.1.23)
 - **PidLidClassificationDescription** ([\[MS-OXCMSG\]](#) section 2.2.1.24)
 - **PidLidClassificationGuid** ([\[MS-OXPROPS\]](#) section 2.54)
 - **PidLidClassificationKeep** ([\[MS-OXPROPS\]](#) section 2.55)
 - **PidLidClassified** ([\[MS-OXCMSG\]](#) section 2.2.1.25)
 - **PidLidInboundICalStream** ([\[MS-OXPROPS\]](#) section 2.148)
 - **PidLidInfoPathFormName** ([\[MS-OXCMSG\]](#) section 2.2.1.27)

- **PidLidPercentComplete** ([MS-OXOFLAG] section 2.2.2.3)
- **PidLidSingleBodyICal** ([MS-OXPROPS] section 2.300)
- **PidLidTaskComplete** ([\[MS-OXOTASK\]](#) section 2.2.2.2.20)
- **PidLidTaskDateCompleted** ([MS-OXOTASK] section 2.2.2.2.9)
- **PidLidTaskDueDate** ([MS-OXOTASK] section 2.2.2.2.5)
- **PidLidTaskStartDate** ([MS-OXOTASK] section 2.2.2.2.4)
- **PidLidTaskStatus** ([MS-OXOTASK] section 2.2.2.2.2)
- **PidLidToDoTitle** ([MS-OXOFLAG] section 2.2.1.12)
- **PidNameAcceptLanguage**([MS-OXCMSG] section 2.2.1.42)
- **PidNameAttachmentMacContentType** ([MS-OXCMSG] section 2.2.2.26)
- **PidNameAttachmentMacInfo** ([MS-OXCMSG] section 2.2.2.26)
- **PidNameContentBase** ([MS-OXCMSG] section 2.2.1.41)
- **PidNameContentClass** ([MS-OXCMSG] section 2.2.1.48)
- **PidNameContentType** ([MS-OXPROPS] section 2.408)
- **PidNameCrossReference** (section [2.5.3](#))
- **PidNameLocationUrl** ([MS-OXPROPS] section 2.450)
- **PidNameQuarantineOriginalSender** ([MS-OXPROPS] section 2.463)
- **PidNameXCallId** ([MS-OXPROPS] section 2.474)
- **PidNameXFaxNumberOfPages** ([MS-OXPROPS] section 2.475)
- **PidNameXSenderTelephoneNumber** ([MS-OXPROPS] section 2.477)
- **PidNameXVoiceMessageAttachmentOrder** ([MS-OXPROPS] section 2.493)
- **PidNameXVoiceMessageDuration** ([MS-OXPROPS] section 2.494)
- **PidNameXVoiceMessageSenderName** ([MS-OXPROPS] section 2.495)
- **PidTagAddressBookProxyAddresses** ([\[MS-OXOABK\]](#) section 2.2.3.23)
- **PidTagAddressType** ([MS-OXOABK] section 2.2.3.13)
- **PidTagAttachAdditionalInformation** ([MS-OXCMSG] section 2.2.2.21)
- **PidTagAttachContentBase** ([MS-OXCMSG] section 2.2.2.26)
- **PidTagAttachContentId** ([MS-OXCMSG] section 2.2.2.26)
- **PidTagAttachContentLocation** ([MS-OXCMSG] section 2.2.2.26)
- **PidTagAttachDataBinary** ([MS-OXCMSG] section 2.2.2.7)
- **PidTagAttachDataObject** ([MS-OXCMSG] section 2.2.2.8)

- **PidTagAttachEncoding** ([MS-OXCMSG] section 2.2.2.20)
- **PidTagAttachExtension** ([MS-OXCMSG] section 2.2.2.12)
- **PidTagAttachFilename** ([MS-OXCMSG] section 2.2.2.11)
- **PidTagAttachFlags** ([MS-OXCMSG] section 2.2.2.18)
- **PidTagAttachLongFilename** ([MS-OXCMSG] section 2.2.2.10)
- **PidTagAttachMethod** ([MS-OXCMSG] section 2.2.2.9)
- **PidTagAttachMimeType** ([MS-OXCMSG] section 2.2.2.26)
- **PidTagAttachPayloadClass** ([MS-OXCMSG] section 2.2.2.26)
- **PidTagAttachPayloadProviderGuidString** ([MS-OXCMSG] section 2.2.2.26)
- **PidTagAttachSize** ([MS-OXCMSG] section 2.2.2.5)
- **PidTagAutoForwarded** ([MS-OXCMSG] section 2.2.1.20)
- **PidTagAutoResponseSuppress** (section [2.1.3.2.20](#))
- **PidTagBody** ([MS-OXCMSG] section 2.2.1.56.1)
- **PidTagBodyContentId** ([MS-OXCMSG] section 2.2.1.56.7)
- **PidTagBodyContentLocation** ([MS-OXCMSG] section 2.2.1.56.8)
- **PidTagCallId** ([MS-OXPROPS] section 2.619)
- **PidTagClientSubmitTime** ([\[MS-OXOMSG\]](#) section 2.2.3.11)
- **PidTagContentFilterSpamConfidenceLevel** ([\[MS-OXCSPAM\]](#) section 2.2.1.3)
- **PidTagConversationIndex** ([\[MS-OXOCFG\]](#) section 2.2.8.8)
- **PidTagConversationTopic** ([MS-OXOMSG] section 2.2.1.5)
- **PidTagCreationTime** ([MS-OXCMSG] section 2.2.2.3)
- **PidTagDisplayName** ([\[MS-OXCFOOLD\]](#) section 2.2.2.2.2.5)
- **PidTagEmailAddress** ([MS-OXOABK] section 2.2.3.14)
- **PidTagEntryId** ([\[MS-OXCPerm\]](#) section 2.2.4)
- **PidTagExpiryTime** ([MS-OXOMSG] section 2.2.3.7)
- **PidTagFaxNumberOfPages** ([MS-OXPROPS] section 2.686)
- **PidTagFlagCompleteTime** ([MS-OXOFLAG] section 2.2.1.3)
- **PidTagFlagStatus** ([MS-OXOFLAG] section 2.2.1.1)
- **PidTagHtml** ([MS-OXCMSG] section 2.2.1.56.9)
- **PidTagInReplyToId** ([MS-OXOMSG] section 2.2.1.13)
- **PidTagInternetCodepage** ([MS-OXCMSG] section 2.2.1.56.6)

- **PidTagInternetMessageId** ([MS-OXOMSG] section 2.2.1.12)
- **PidTagInternetReferences** ([MS-OXCMSG] section 2.2.1.26)
- **PidTagLastModificationTime** ([MS-OXCMSG] section 2.2.2.2)
- **PidTagListHelp** ([MS-OXPROPS] section 2.758)
- **PidTagListSubscribe** ([MS-OXPROPS] section 2.759)
- **PidTagListUnsubscribe** ([MS-OXPROPS] section 2.760)
- **PidTagMessageClass** ([MS-OXCMSG] section 2.2.1.3)
- **PidTagMessageCodepage** ([MS-OXCMSG] section 2.2.1.4)
- **PidTagMessageDeliveryTime** ([MS-OXOMSG] section 2.2.3.9)
- **PidTagMessageFlags** ([MS-OXCMSG] section 2.2.1.6)
- **PidTagMessageLocaleId** ([MS-OXCMSG] section 2.2.1.5)
- **PidTagNormalizedSubject** ([MS-OXCMSG] section 2.2.1.10)
- **PidTagObjectType** ([\[MS-OXCPRPT\]](#) section 2.2.1.7)
- **PidTagOriginatorDeliveryReportRequested** ([MS-OXOMSG] section 2.2.1.20)
- **PidTagPriority** ([MS-OXCMSG] section 2.2.1.12)
- **PidTagPurportedSenderDomain** ([MS-OXCMSG] section 2.2.1.43)
- **PidTagReadReceiptAddressType** ([MS-OXPROPS] section 2.868)
- **PidTagReadReceiptEmailAddress** ([MS-OXPROPS] section 2.869)
- **PidTagReadReceiptName** ([MS-OXPROPS] section 2.871)
- **PidTagReadReceiptRequested** ([MS-OXOMSG] section 2.2.1.29)
- **PidTagRecipientType** [MS-OXOMSG] section 2.2.3.1
- **PidTagRenderingPosition** ([MS-OXCMSG] section 2.2.2.16)
- **PidTagReplyRecipientEntries** ([MS-OXOMSG] section 2.2.1.43)
- **PidTagReplyRecipientNames** ([MS-OXOMSG] section 2.2.1.44)
- **PidTagReplyTime** ([MS-OXOFLAG] section 2.2.3.1)
- **PidTagRtfCompressed** ([MS-OXCMSG] section 2.2.1.56.4)
- **PidTagSenderIdStatus** ([MS-OXPROPS] section 2.992)
- **PidTagSenderName** ([MS-OXOMSG] section 2.2.1.51)
- **PidTagSenderSmtpAddress** ([MS-OXPROPS] section 2.995)
- **PidTagSenderTelephoneNumber** ([MS-OXPROPS] section 2.996)
- **PidTagSendInternetEncoding** ([MS-OXOABK] section 2.2.3.19)

- **PidTagSendRichInfo** ([MS-OXOABK] section 2.2.3.18)
- **PidTagSensitivity** ([MS-OXCMSG] section 2.2.1.13)
- **PidTagSentRepresentingAddressType** ([MS-OXOMSG] section 2.2.1.54)
- **PidTagSentRepresentingEmailAddress** ([MS-OXOMSG] section 2.2.1.55)
- **PidTagSentRepresentingEntryId** ([MS-OXOMSG] section 2.2.1.56)
- **PidTagSentRepresentingName** ([MS-OXOMSG] section 2.2.1.57)
- **PidTagSmtpAddress** ([MS-OXOABK] section 2.2.3.21)
- **PidTagSubject** ([MS-OXPROPS] section 2.1021)
- **PidTagSubjectPrefix** ([MS-OXOMSG] section 2.2.1.60)
- **PidTagTextAttachmentCharset** ([MS-OXCMSG] section 2.2.2.25)
- **PidTagTnefCorrelationKey** ([MS-OXCMSG] section 2.2.1.29)
- **PidTagToDoItemFlags** ([MS-OXOFLAG] section 2.2.1.6)
- **PidTagTransportMessageHeaders** ([MS-OXOMSG] section 2.2.1.61)
- **PidTagVoiceMessageAttachmentOrder** ([MS-OXPROPS] section 2.1047)
- **PidTagVoiceMessageDuration** ([MS-OXPROPS] section 2.1048)
- **PidTagVoiceMessageSenderName** ([MS-OXPROPS] section 2.1049)

2.4 Unconverted MIME Part Analysis Algorithm Details

The scenario in which a server receives a pure MIME message using a MIME protocol (such as SMTP), and then must send it as a pure MIME message using a different MIME protocol (such as POP3 or IMAP4) requires special functionality. The following diagram describes this process of "round-tripping".

SMTP => MIME reader => storage => MIME writer => POP3 or IMAP4

A fixed number of the headers allowed in a valid MIME message are mapped to properties of a Message object, as specified in section 2.2. The rest are saved in their raw, unconverted format to facilitate round-tripping <200>. The structure used in this algorithm for preserving unconverted MIME headers is the **PidTagMimeSkeleton** property ([MS-OXCMSG] section 2.2.1.28), as specified in section 2.4.1, which is generated by the MIME reader during the MIME analysis process. <201>

After the MIME reader converts the MIME message to a Message object, as specified in section 2.2, the reader copies the structure of the MIME message plus the contents of any message parts that are not converted into the **PidTagMimeSkeleton** property on the Message object.

2.4.1 Abstract Data Model

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

The following ADM types are defined in this section:

Global

Mailbox

MessageObject

2.4.1.1 Global

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

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

2.4.1.2 Per Mailbox

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

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

2.4.1.3 Per Message Object

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

Mailbox.MessageObject.MIMESkeleton: The original MIME structure of the message.

2.4.2 Initialization

None.

2.4.3 Processing Rules

2.4.3.1 MIME Conversion

During MIME conversion, the MIME reader keeps track of which MIME parts are saved as attachments and which MIME part is promoted as the message body. If any of these MIME parts are missing a **Content-Id** header, the MIME reader generates an **X-Exchange-MIME-Skeleton-Content-Id** header and promotes it to the **PidTagBodyContentId** property ([MS-OXCMSG] section 2.2.1.56.7) or the **PidTagAttachContentId** property ([MS-OXCMSG] section 2.2.2.26) as appropriate. If the message did not have a TNEF MIME part, or it was not promoted, after the message is converted, the whole MIME message SHOULD be saved in the **PidTagMimeSkeleton** property ([MS-OXCMSG] section 2.2.1.28), with the MIME part content filtered out for MIME parts that were promoted as a message attachment or message body.

If the message body was an aggregate body, as specified in section [2.2.3.3.2.2](#), MIME readers SHOULD NOT set the **PidTagMimeSkeleton** property.

Body part contents for the following content types are included in the **PidTagMimeSkeleton** property even when the MIME reader has copied their contents to the Message object.

1. MIME part content for vCard attachments is not removed from the **PidTagMimeSkeleton** property, even though the content is converted to a Contact object.
2. iCalendar MIME parts are not filtered out of the **PidTagMimeSkeleton** property.

3. For **S/MIME** messages, the **PidTagMimeSkeleton** property contains only root part headers since the rest of the data can be retrieved from the attachment.
4. For DSN messages, an original message attachment is filtered out only if it was promoted as an attachment. Human-readable body content is always filtered. Report parts, which are specified in [RFC3464](#) as message parts with a **Content-Type** header value of "message/delivery-status", are not filtered out.

2.5 Message Object Properties

This algorithm specifies the following additional properties for Message objects.

- **PidLidClassificationGuid** (section [2.5.1](#))
- **PidLidClassificationKeep** (section [2.5.2](#))
- **PidNameQuarantineOriginalSender** (section [2.5.4](#))

2.5.1 PidLidClassificationGuid

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

The **PidLidClassificationGuid** property ([\[MS-OXPROPS\]](#) section 2.54) is an optional property on Message objects. Its value is the value of the **X-Microsoft-ClassID** header.

2.5.2 PidLidClassificationKeep

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

The **PidLidClassificationKeep** property ([\[MS-OXPROPS\]](#) section 2.55) is an optional property on Message objects. Its value is the value of the **X-Microsoft-ClassKeep** header.

2.5.3 PidNameCrossReference

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

The **PidNameCrossReference** property ([\[MS-OXPROPS\]](#) section 2.410) is an optional property on Message objects. Its value is the value of the **XRef** header.

2.5.4 PidNameQuarantineOriginalSender

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

The **PidNameQuarantineOriginalSender** property ([\[MS-OXPROPS\]](#) section 2.463) is an optional property on Message objects. Its value is the value of the **X-MS-Exchange-Organization-Original-Sender** header.

2.6 Recipient Property Groups

Address elements other than recipients (2), such as From and Sender, are represented in a Message object by a recipient property group. Each recipient property group consists of four common attributes: a display name, an address type, an e-mail address, and an EntryID. These attributes correspond to specific properties on the Message object.

This algorithm defines the following property groups.

- **PidTagReadReceipt** (section [2.6.1](#))
- **PidTagReceivedBy** (section [2.6.2](#))
- **PidTagReceivedRepresenting** (section [2.6.3](#))
- **PidTagSender** (section [2.6.4](#))
- **PidTagSentRepresenting** (section [2.6.5](#))
- **Recipient Table** (section [2.6.6](#))

2.6.1 PidTagReadReceipt Property Group

The **PidTagReadReceipt** property group contains properties that represent the recipient to which any read receipts should be sent.

Attribute	Property
Display name	PidTagReadReceiptName ([MS-OXOMSG] section 2.2.2.27)
EntryID	PidTagReadReceiptEntryId ([MS-OXOMSG] section 2.2.2.26)
Address type	PidTagReadReceiptAddressType ([MS-OXOMSG] section 2.2.2.24)
Email address	PidTagReadReceiptEmailAddress ([MS-OXOMSG] section 2.2.2.25)
SMTP address	PidTagReadReceiptSmtpAddress ([MS-OXOMSG] section 2.2.1.30)

2.6.2 PidTagReceivedBy Property Group

The **PidTagReceivedBy** property group contains properties that represent the recipient that received the Message object.

Attribute	Property
Display name	PidTagReceivedByName ([MS-OXOMSG] section 2.2.1.39)
EntryID	PidTagReceivedByEntryId ([MS-OXOMSG] section 2.2.1.38)
Address type	PidTagReceivedByAddressType ([MS-OXOMSG] section 2.2.1.36)
Email address	PidTagReceivedByEmailAddress ([MS-OXOMSG] section 2.2.1.37)
SMTP address	PidTagReceivedBySmtpAddress ([MS-OXOMSG] section 2.2.1.41)

2.6.3 PidTagReceivedRepresenting Property Group

The **PidTagReceivedRepresenting** property group contains properties that represent the receiving mailbox owner.

Attribute	Property
Display name	PidTagReceivedRepresentingName ([MS-OXOMSG] section 2.2.1.26)
EntryID	PidTagReceivedRepresentingEntryId ([MS-OXOMSG] section 2.2.1.25)
Address type	PidTagReceivedRepresentingAddressType ([MS-OXOMSG] section 2.2.1.23)
Email address	PidTagReceivedRepresentingEmailAddress ([MS-OXOMSG] section 2.2.1.24)
SMTP address	PidTagReceivedRepresentingSmtpAddress ([MS-OXOMSG] section 2.2.1.28)

2.6.4 PidTagSender Property Group

The **PidTagSender** property group contains properties that represent the sender of the Message object.

Attribute	Property
Display name	PidTagSenderName ([MS-OXOMSG] section 2.2.1.51)
EntryID	PidTagSenderEntryId ([MS-OXOMSG] section 2.2.1.50)
Address type	PidTagSenderAddressType ([MS-OXOMSG] section 2.2.1.48)
Email address	PidTagSenderEmailAddress ([MS-OXOMSG] section 2.2.1.49)
SMTP address	PidTagSenderSmtpAddress ([MS-OXOMSG] section 2.2.1.53)

2.6.5 PidTagSentRepresenting Property Group

The **PidTagSentRepresenting** property group contains properties that represent the user that the sender of the Message object is representing.

Attribute	Property
Display name	PidTagSentRepresentingName ([MS-OXOMSG] section 2.2.1.57)
EntryID	PidTagSentRepresentingEntryId ([MS-OXOMSG] section 2.2.1.56)

Attribute	Property
Address type	PidTagSentRepresentingAddressType ([MS-OXOMSG] section 2.2.1.54)
Email address	PidTagSentRepresentingEmailAddress ([MS-OXOMSG] section 2.2.1.55)
SMTP address	PidTagSentRepresentingSmtAddress ([MS-OXOMSG] section 2.2.1.59)

2.6.6 Recipient Table Property Group

The **Recipient Table** property group contains properties that represent an entry in the recipient table of a Message object.

Attribute	Property
Display name	PidTagDisplayName ([MS-EXOABK] section 2.2.3.1)
EntryID	PidTagEntryId ([MS-EXOABK] section 2.2.3.2)
Address type	PidTagAddressType ([MS-EXOABK] section 2.2.3.13)
Email address	PidTagEmailAddress ([MS-EXOABK] section 2.2.3.14)
SMTP address	PidTagSmtAddress ([MS-EXOABK] section 2.2.3.21)

3 Algorithm Examples

3.1 MIME Examples

3.1.1 Simple MIME Message

This example shows a very simple e-mail message in both MIME and Message object formats. The following is the message in MIME format:

```
Received: from mailer01.example.com by mailer02.contoso.com
with Microsoft SMTP Server; Mon, 11 Feb 2008 14:45:44 -0800
From: <user1@contoso.com>
To: <user2@contoso.com>; <user3@contoso.com>
Subject: test message
Date: Mon, 11 Feb 2008 14:45:32 -0800
Message-ID: <000001c86cff$cf0dd670$ae62379d@mail.contoso.com>
MIME-Version: 1.0
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
Importance: normal
Priority: normal
```

```
this is a test message
```

The following table shows this simple message represented as a Message object. The message has several properties, and it contains recipients (2), each with several properties of its own. In this example, matching entries were found in the address book for user2@contoso.com and user3@contoso.com. A matching entry was not found for user1@contoso.com. The recipients (2) are shown in the table that follows. All of the data types are described in [\[MS-OXCDATA\]](#) section 2.11.1.

Message property name	Type	Value
PidTagMessageDeliveryTime ([MS-OXOMSG] section 2.2.3.9)	PtypTime	%xF9.2D.82.D6.FF.6C.C8.01
PidTagSentRepresentingName ([MS-OXOMSG] section 2.2.1.57)	PtypString	Test user 1
PidTagSentRepresentingAddress Type ([MS-OXOMSG] section 2.2.1.54)	PtypString	SMTP
PidTagSentRepresentingEmailAddress ([MS-OXOMSG] section 2.2.1.55)	PtypString	user1@contoso.com
PidTagSubject ([MS-OXPROPS] section 2.1021)	PtypString	test message
PidTagClientSubmitTime ([MS-OXOMSG] section 2.2.3.11)	PtypTime	%x00.8E.AD.CE.FF.6C.C8.01
PidTagInternetMessageId ([MS-OXOMSG] section 2.2.1.12)	PtypString	000001c86cff\$cf0dd670\$ae62379d@mail.contoso.com
PidTagImportance ([MS-OXCMSG] section 2.2.1.11)	PtypInteger32	1

Message property name	Type	Value
PidTagPriority ([MS-OXCMSG] section 2.2.1.12)	PtypInteger32	0
PidTagBody ([MS-OXCMSG] section 2.2.1.56.1)	PtypString	this is a test message
PidTagInternetCodepage ([MS-OXCMSG] section 2.2.1.56.6)	PtypInteger32	28591
PidTagObjectType ([MS-OXCPRPT] section 2.2.1.7)	PtypInteger32	5
PidTagMessageFlags ([MS-OXCMSG] section 2.2.1.6)	PtypInteger32	0x23

Row ID	Recipient property name	Type	Value
38714304	PidTagDisplayName ([MS-OXCFOLD] section 2.2.2.2.2.5)	PtypString	Test user 2
38714304	PidTagAddressType ([MS-OXOABK] section 2.2.3.13)	PtypString	EX
38714304	PidTagEmailAddress ([MS-OXOABK] section 2.2.3.14)	PtypString	/O=Example1/OU= Administrative Group/CN=recipients/CN=user2
38714304	PidTagSmtptAddress ([MS-OXOABK] section 2.2.3.21)	PtypString	user2@contoso.com
38714304	PidTagRecipientType ([MS-OXOMSG] section 2.2.3.1)	PtypInteger32	1
38714304	PidTagObjectType ([MS-OXCPRPT] section 2.2.1.7)	PtypInteger32	6
38714304	PidTagEntryId ([MS-OXPROPS] section 2.674)	PtypBinary	%x00.00.00.00.DC.A7.40.C8.C0.42.10.1A.B4.B9.08.00.2B.2F.E1.82.01.00.00.00.06.00.00.00.2F.4F.3D.45.78.61.6D.70.6C.65.31.2F.4F.55.3D.20.41.64.6D.69.6E.69.73.74.72.61.74.69.76.65.20.47.72.6F.75.70.2F.43.4E.3D.72.65.63.69.70.69.65.6E.74.73.2F.43.4E.3D.75.73.65.72.32.00
38714305	PidTagDisplayName	PtypString	Test user 3
38714305	PidTagAddressType	PtypString	EX
38714305	PidTagEmailAddress	PtypString	/O=Example1/OU= Administrative Group/CN=recipients/CN=user3

Row ID	Recipient property name	Type	Value
38714305	PidTagSmtAddress	PtypString	user3@contoso.com
38714305	PidTagRecipientType	PtypInteger32	1
38714305	PidTagObjectType	PtypInteger32	6
38714305	PidTagEntryId	PtypBinary	%x00.00.00.00.DC.A7. 40.C8.C0.42.10.1A.B4.B9.08.00.2B.2F.E1. 82.01.00.00.00.06.00.00.00.2F.4F.3D. 45.78.61.6D.70.6C.65.31.2F.4F.55.3D. 20.41.64.6D.69.6E.69.73.74.72.61.74. 69.76.65.20.47.72.6F.75.70.2F.43.4E. 3D.72.65.63.69.70.69.65.6E.74.73.2F. 43.4E.3D.75.73.65.72.33.00

While obviously less compact than the MIME format, the Message object format makes strongly typed data available. Both client and server code can sort, find, and process messages according to specific criteria such as "all unread messages", "all messages tagged as Personal", or "all calendar items occurring in the week of 2/12/2008, sorted by start time".

3.1.2 MIME Message Containing Inline and Non-Inline Attachments

The following example demonstrates a MIME-formatted message that contains both inline and non-inline attachments, as described in section [2.1.3.4](#).

```

From: <john@contoso.com>
To: <imtiaz@contoso.com>
Subject: Example with inline and non-inline attachments.
Date: Mon, 10 Mar 2008 14:36:46 -0700
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="simple boundary 1"

--simple boundary 1
Content-Type: multipart/related; boundary="simple boundary 2"

--simple boundary 2
Content-Type: multipart/alternative; boundary="simple boundary 3"

--simple boundary 3
Content-Type: text/plain

...Text without inline reference...
--simple boundary 3
Content-Type: text/html

...Text with inline reference...
--simple boundary 3--
--simple boundary 2
Content-Type: image/png; name="inline.PNG"
Content-Transfer-Encoding: base64
Content-ID: <6583CF49B56F42FEA6A4A118F46F96FB@example.com>
Content-Disposition: inline; filename="Inline.png"

...Attachment data encoded with base64...
--simple boundary 2--

--simple boundary 1
Content-Type: image/png; name=" Attachment "
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="Attachment.png"

```

```
...Attachment data encoded with base64...
--simple boundary 1--
```

3.1.3 MIME Message Containing Only Inline Attachments

The following example demonstrates a MIME-formatted message that contains only inline attachments, as described in section [2.1.3.4](#).

```
From: <john@contoso.com>
To: <imtiaaz@contoso.com>
Subject: Example with inline attachment.
Date: Mon, 10 Mar 2008 14:36:46 -0700
MIME-Version: 1.0
Content-Type: multipart/related; boundary="simple boundary"

--simple boundary
Content-Type: text/html;

...Text with reference...

--simple boundary
Content-Type: image/png; name="inline.PNG"
Content-Transfer-Encoding: base64
Content-ID: <6583CF49B56F42FEA6A4A118F46F96FB@example.com>
Content-Disposition: inline; filename=" inline.png"

...Attachment data encoded with base64...
--simple boundary--
```

3.1.4 MIME Message Containing Only Non-Inline Attachments

The following example demonstrates a MIME-formatted message that contains only non-inline attachments, as described in section [2.1.3.4](#).

```
From: <john@contoso.com>
To: <imtiaaz@contoso.com>
Subject: Example with non-inline attachment.
Date: Mon, 10 Mar 2008 14:36:46 -0700
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="simple boundary"

--simple boundary
Content-Type: text/plain;

...Text without reference...

--simple boundary
Content-Type: image/png; name=" Attachment"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="Attachment.png"

...Attachment data encoded with base64...
--simple boundary--
```

3.1.5 E-Mail Message Without a MIME-Version Header

The following is an example of an e-mail message lacking a **MIME-Version** header. Such messages can be treated as MIME messages, as described in section [2.2.3.9.1](#).

```
From: <user1@contoso.com>
To: <user2@contoso.com>
Subject: Example Legacy 822 message with attachment.
Date: Mon, 10 Mar 2008 14:36:46 -0700

this is a test message

begin 664 Flag.png
MB5!.1PT*&@H`-24A$4@`!0`-`"8`"I4$Y>`!F)+1T0`_P#_
M`/^@O:>3`"7!(67,`L3`+`$P$`FIP8`!W1)344'V`,+`!8G&XK)
MCP`1M)1$%4.,NMDTUJPE`4A;\7\Z,ET8$-07`@BE,1.Q)T#2*X!-V+6^@J
M7(2.Q14X<J+6@29$DZBOHTJA)GW0GO&Y!^=>SA6\O4O^49J*R;4#7#OX.W#2
MGU,I'6EZ6YK>EDKIR*0_SP2*M)6=_ (6"D1!$%F%L`O!BQMA6Q#DQ\"Y]82M
MZH919\G.=QXP@#`VV?D.H\Z25G6CGM#(W4ANN<S5TCP_$FI",AW.L/1K*LS2
MKTR',S0AU1+FM#NW>W8!TCQ/IQKNGG%OD0H;]Q8TW+WZ#0M&@A`2VXKX"&SN
M4CS.\6H'!)&%E()S8J@E/"<&86PR:*^HE0]TZVNZ]36U\H%!>T48FT]AF3W\
F+J]X`F![*O[Z*;K*.ZF`OO0)9G1H4-$`H@T`245.1*Y"8((`
`
end

begin 664 Flag.png
MB5!.1PT*&@H`-24A$4@`!0`-`"8`"I4$Y>`!F)+1T0`_P#_
M`/^@O:>3`"7!(67,`L3`+`$P$`FIP8`!W1)344'V`,+`!8G&XK)
MCP`1M)1$%4.,NMDTUJPE`4A;\7\Z,ET8$-07`@BE,1.Q)T#2*X!-V+6^@J
M7(2.Q14X<J+6@29$DZBOHTJA)GW0GO&Y!^=>SA6\O4O^49J*R;4#7#OX.W#2
MGU,I'6EZ6YK>EDKIR*0_SP2*M)6=_ (6"D1!$%F%L`O!BQMA6Q#DQ\"Y]82M
MZH919\G.=QXP@#`VV?D.H\Z25G6CGM#(W4ANN<S5TCP_$FI",AW.L/1K*LS2
MKTR',S0AU1+FM#NW>W8!TCQ/IQKNGG%OD0H;]Q8TW+WZ#0M&@A`2VXKX"&SN
M4CS.\6H'!)&%E()S8J@E/"<&86PR:*^HE0]TZVNZ]36U\H%!>T48FT]AF3W\
F+J]X`F![*O[Z*;K*.ZF`OO0)9G1H4-$`H@T`245.1*Y"8((`
`
end
```

Preli

4 Security

4.1 Security Considerations for Implementers

4.1.1 Unsolicited Commercial E-Mail (Spam)

A significant business has evolved around the sending of unsolicited commercial e-mail (colloquially referred as spam). Unlike physical bulk mail, with its built-in restrictions on labeling and cost, the general structure of SMTP allows anonymous sources to send e-mail messages virtually without restriction. Attempts are being made to reduce the volume of spam that makes it to a person's mailbox, but care has to be taken to not affect legitimate senders.

Part of the success of this industry is the fact that people impute importance to unverifiable things. For example, the purported sender of an e-mail message (considering most e-mail messages are not digitally signed) is commonly used by people to attach importance and priority. If the e-mail message appears to come from a person's boss, there is a higher probability that the employee would act on the message. In this case, care needs to be taken when receiving e-mail over unauthenticated transports. Even if the routing address of the sender matches a valid employee or contact, it is better if clients and servers preserve the external routing address on the Message object, because replacing it with its address book equivalent could impute elevated importance to the content.

4.1.2 Information Disclosure

Content that is sent can contain hints about the source network's topology and structure. In MIME, this can be discerned from the **Received** headers (every SMTP server and potentially the client's computer are listed by its network address). In addition, if the optional algorithm described in [\[RFC2822\]](#) section 3.6.4 to generate a unique **Message-Id** header value is implemented, the client's network address and internal domain name is exposed. Alternately, a GUID can be used to satisfy the unique identifier, circumventing the first data exposure. The aforementioned problem also exists for **Content-Id** header and boundary parameter values. It is suggested that a GUID be used here as well.

Additionally, when sending recipient (2) data other than the properties mentioned previously, implementations need to be aware that internal data can be exposed. For example, office numbers and phone numbers could be cached on each recipient (2). This is an issue on embedded messages that are transported via TNEF, as described in [\[MS-OXTNEF\]](#), because TNEF has the ability to carry more recipient information than is available with MIME headers.

When receiving e-mail, care also needs to be taken to deal with some information disclosure issues. If the e-mail message leverages any feature that requires the client to "fetch" additional **resources** when displaying it, the act of fetching can expose the fact that the recipient (2) is an actual employee of the sending organization, and the date and time that the message was read. Examples of this are external bodies, HTML style sheets, and images.

4.1.3 Content-Type Versus File Extension Mismatch

Various clients accept the **Content-Type** header received from the server but then verify the content. This can include checking the file extension or looking for a thumbprint at the start of the file, and then mapping this data back to a verified **Content-Type** header value. If the file extension or thumbprint does not match the stated **Content-Type** header value, a **Content-Type** header value derived from the file extension or thumbprint is used instead. This behavior is actually described in [\[RFC2045\]](#), which allows the sender to set the **Content-Type** header value to "application/octet-stream" (or not set it at all). The recipient (1) is then responsible for correctly determining the type of content via alternate means.

In addition, it was found that various clients incorrectly set the **Content-Type** header either by mistake or intentionally. Support to address the former has existed for quite some time but has opened a path to potentially thwart policy scanning and protection applications running on the server.

Therefore MIME readers can correct mislabeled **Content-Type** header values so that server Policy Agents and clients can trust the header value. Clients need to offer a mechanism to do one or more of the following:

- Suppress correcting the value of the **Content-Type** header.
- Block attachments by type or extension.
- Offer some sort of security barrier before running any script or binary.

These steps are particularly important if the sender is unauthenticated.

4.1.4 Do Not Support Message/Partial

A **Content-Type** header value of "message/partial" allows large messages to be sent in pieces and re-assembled by the client. It was originally designed to work around transmission failures during slow delivery causing the complete message to be resent from scratch, and to work around message size restrictions of implementations of protocols like SMTP. With increased bandwidth speeds and greater connectivity, the long transmission times are more a thing of the past. Continued support for this **Content-Type** header value allows an avenue for content that is inappropriate to reach (or leave) the e-mail client's computer. This content could include things such as "Information disclosure" of proprietary information, unsolicited commercial e-mail (spam), and computer virus attachments.

E-mail servers attempt to protect their users from inappropriate content by implementing policy applications that run as part of the algorithm. For them to work efficiently, the complete content is incorporated into one message. For this reason, servers need to prohibit sending or receiving messages with a **Content-Type** header value of "message/partial".

4.1.5 Considerations for Message/External-Body

The original MIME RFC, [\[RFC1521\]](#), allowed the body of an entity to be referenced externally rather than requiring it to be inline. The current MIME RFC, [\[RFC2046\]](#), describes the form of this construct; the security implications are as follows:

1. The blind retrieval of the content by the client can disclose information about the recipient (1).
2. The authentication mechanism tied to the retrieval (*access-type* parameter) can result in a pop-up dialog box, leading the user to expose credential information.
3. The server (policy or delivery application) that is attempting to check the content opens up a denial of service vector for the remote host to tie up server resources.

4.1.6 Preventing Denial of Service Attacks

4.1.6.1 Submission Limits

Servers can limit the size of received messages to limit resource consumption. Such limits can be different for authenticated versus anonymous senders.

4.1.6.2 Complexity of Nested Entities

It is possible to represent very complex hierarchies with the MIME format and to add superfluous entity layers (multipart **Content-Type** headers). Servers and clients need to protect themselves from stack overflows or heap starvation. This can involve limiting the nesting depth of attachments and body parts within a single message.

4.1.6.3 Number of Embedded Messages

A server that implements this algorithm converts MIME content into a Message object representation. This causes each embedded message to be mapped individually and all attachments to be included therein. One implementation of this is to recursively handle each attached embedded message, but care needs to be taken not to encounter a stack overflow by doing so.

4.1.6.4 Compressed Attachments

Analyzing each attachment on the server is a concern when decompression is required. It is possible to encounter compressed content that requires large volumes of disk space, memory, or other resources, leading to a denial of service.

4.2 Index of Security Parameters

None.

Preliminary

5 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 2.1](#): Office Outlook 2003 and Office Outlook 2007 do not use the HTML only setting.

<2> [Section 2.1.3.1](#): Microsoft Exchange Server 2010 Service Pack 2 (SP2), Exchange 2013, and Exchange 2016 Preview support this method of display name generation.

<3> [Section 2.1.3.1](#): Exchange 2003 supports this method of generating display names for recipients.

<4> [Section 2.1.3.1](#): Office Outlook 2003 and Office Outlook 2007 do not encapsulate addresses.

<5> [Section 2.1.3.1.1](#): Exchange 2007 does not ignore recipient types other than To, Cc, or Bcc.

<6> [Section 2.1.3.1.1](#): Exchange 2007 performs a bitwise **AND** of the value of the **PidTagRecipientType** property ([\[MS-OXOMSG\]](#) section 2.2.3.1) and 0x00000003.

<7> [Section 2.1.3.1.1](#): Office Outlook 2003 and Office Outlook 2007 do not generate MIME recipients (2) for Bcc recipients when generating a message to send via SMTP.

<8> [Section 2.1.3.1.1](#): Office Outlook 2003 and Office Outlook 2007 use the listed steps in the following order: 4, 5, 1, 2, 3, 7. Office Outlook 2003 and Office Outlook 2007 do not use step 6.

<9> [Section 2.1.3.1.1.1](#): Exchange 2003 generates the **attRecipTable** attribute according to this procedure.

- <10> [Section 2.1.3.1.1.2](#): Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not copy Bcc recipients to the MIME **Bcc** header when generating messages for POP3 or IMAP4.
- <11> [Section 2.1.3.1.1.2](#): Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not copy Bcc recipients to the TNEF recipient table for attached messages when generating messages for POP3 or IMAP4.
- <12> [Section 2.1.3.1.2](#): Office Outlook 2003 and Office Outlook 2007 do not copy the values of the **PidTagReplyRecipientEntries** property ([MS-OXOMSG] section 2.2.1.43) and the **PidTagReplyRecipientNames** property ([MS-OXOMSG] section 2.2.1.44) to the TNEF body part.
- <13> [Section 2.1.3.1.3](#): Office Outlook 2003 and Office Outlook 2007 do not copy the values of the **PidTagSentRepresenting** property group (section [2.6.5](#)) to the TNEF body part.
- <14> [Section 2.1.3.1.4](#): Office Outlook 2003 and Office Outlook 2007 do not copy the values of the **PidTagSender** property group (section [2.6.4](#)) to the TNEF body part.
- <15> [Section 2.1.3.1.5](#): Exchange 2007 copies the values in the **PidTagSentRepresenting** property group (section 2.6.5) as described in this procedure.
- <16> [Section 2.1.3.1.5](#): Office Outlook 2003 and Office Outlook 2007 do not copy the values of the **PidTagOriginatorDeliveryReportRequested** property ([MS-OXOMSG] section 2.2.1.20) or the specified e-mail properties to the TNEF body part.
- <17> [Section 2.1.3.1.6](#): Office Outlook 2003 and Office Outlook 2007 do not copy the values of the **PidTagReadReceipt** property group (section [2.6.1](#)) and the **PidTagSentRepresenting** property group (section 2.6.5). Note that they do copy the **PidTagReadReceiptRequested** property ([MS-OXOMSG] section 2.2.1.29).
- <18> [Section 2.1.3.1.8](#): Office Outlook 2003 and Office Outlook 2007 do not encapsulate addresses.
- <19> [Section 2.1.3.1.8](#): Exchange 2007 uses this procedure for encoding and decoding encapsulated addresses. Exchange 2007 has no limitation on the length of the **address-type** part. Exchange 2007 does not properly de-encapsulate if the **address-type** part contains an ASCII hyphen ("-").
- <20> [Section 2.1.3.1.8](#): Exchange 2003 uses this procedure for encoding and decoding encapsulated addresses. Exchange 2003 does not properly de-encapsulate if the **address-type** part contains an ASCII hyphen ("-").
- <21> [Section 2.1.3.2](#): Office Outlook 2003 and Office Outlook 2007 do not copy properties to the TNEF body part if there is a corresponding MIME header.
- <22> [Section 2.1.3.2.1](#): Office Outlook 2003 and Office Outlook 2007 do not convert appointment items to MIME.
- <23> [Section 2.1.3.2.2](#): Exchange 2003 does not generate the **Content-class** header.
- <24> [Section 2.1.3.2.2](#): Exchange 2007 writes the **Content-class** header value as "fax".
- <25> [Section 2.1.3.2.2](#): Exchange 2007 writes the **Content-class** header value as "voice".
- <26> [Section 2.1.3.2.3](#): Office Outlook 2003 does not support Unified Messaging.
- <27> [Section 2.1.3.2.3](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-CallingTelephoneNumber** header.
- <28> [Section 2.1.3.2.3](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-VoiceMessageDuration** header.

<29> [Section 2.1.3.2.3](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-VoiceMessageSenderName** header.

<30> [Section 2.1.3.2.3](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-FaxNumberOfPages** header.

<31> [Section 2.1.3.2.3](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-AttachmentOrder** header.

<32> [Section 2.1.3.2.3](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-CallID** header.

<33> [Section 2.1.3.2.4](#): Exchange 2003 only implements encoding for three specific headers: **Subject**, **Thread-Topic**, and **X-Message-Flag**, which is not fully compliant with [RFC2047](#).

<34> [Section 2.1.3.2.4](#): Exchange 2003 does not check to prevent the use of the reserved name headers **X-Microsoft-Exchange-Organization** and **X-Microsoft-Exchange-Forest**.

<35> [Section 2.1.3.2.7](#): Office Outlook 2003 and Office Outlook 2007 do not copy the value of the **PidTagClientSubmitTime** property ([MS-OXOMSG] section 2.2.3.11) to the TNEF body part.

<36> [Section 2.1.3.2.8](#): Office Outlook 2003 and Office Outlook 2007 instead copy the value of the **PidTagSubject** property ([\[MS-OXCMSG\]](#) section 2.2.1.46) to the **Subject** header.

<37> [Section 2.1.3.2.8](#): Office Outlook 2003 and Office Outlook 2007 do not copy the message subject to the TNEF body part.

<38> [Section 2.1.3.2.11](#): Office Outlook 2003 and Office Outlook 2007 generate new **Message-ID** header values in this instance.

<39> [Section 2.1.3.2.14](#): Exchange 2003 does not copy this value. Microsoft Exchange Server 2003 Service Pack 2 (SP2) with the hotfix for Exchange 2003 SP2 described in [\[MSKB908027\]](#) applied copies this value.

<40> [Section 2.1.3.2.16](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **Accept-Language** header.

<41> [Section 2.1.3.2.16](#): Exchange 2007 generates a value for the **Accept-Language** header when the **PidNameAcceptLanguage** property ([MS-OXCMSG] section 2.2.1.42) is missing and the **mfSubmitted** flag is not set in the **PidTagMessageFlags** property ([MS-OXCMSG] section 2.2.1.6).

<42> [Section 2.1.3.2.16](#): Exchange 2003 does not generate the **Content-Language** header.

<43> [Section 2.1.3.2.17](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate any classification headers.

<44> [Section 2.1.3.2.18](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not support retrieving the value of the **PidTagAttachPayloadProviderGuidIdString** property ([MS-OXCMSG] section 2.2.2.26).

<45> [Section 2.1.3.2.18](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview only generate **X-Payload-Provider-GUID** and **X-Payload-Class** headers when they are properties on a message, which includes message attachments. These headers are not generated when they are properties of file attachments.

<46> [Section 2.1.3.2.18](#): Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not generate the **X-Payload-Provider-GUID** header. Exchange 2003 does not copy the value of the **PidTagAttachPayloadProviderGuidIdString** property ([MS-OXCMSG] section 2.2.2.26) to the **X-Payload-Provider-Guid** header.

<47> [Section 2.1.3.2.18](#): Outlook 2010, Outlook 2013, and Outlook 2016 Preview set the **PidTagAttachPayloadProviderGuidString** property ([MS-OXCMSG] section 2.2.2.26) as a property of the attachment.

<48> [Section 2.1.3.2.18](#): Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not generate the **X-Payload-Class** header. Exchange 2003 does not copy the value of the **PidTagAttachPayloadClass** property to the **X-Payload-Class** header.

<49> [Section 2.1.3.2.19](#): Office Outlook 2003 generates the **X-MS-Has-Attach** header with no output.

<50> [Section 2.1.3.2.19](#): Exchange 2010, Exchange 2013, and Exchange 2016 Preview will not generate an **X-MS-Has-Attach** header if one was not included in the original MIME message.

<51> [Section 2.1.3.2.20](#): Exchange 2003, Exchange 2007, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-Auto-Response-Suppress** header.

<52> [Section 2.1.3.2.21](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-MS-Exchange-Organization-AutoForwarded** header.

<53> [Section 2.1.3.2.22](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-MS-Exchange-Organization-SenderIdResult** header.

<54> [Section 2.1.3.2.23](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-MS-Exchange-Organization-PRD** header.

<55> [Section 2.1.3.2.24](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the **X-MS-Exchange-Organization-SCL** header.

<56> [Section 2.1.3.2.26](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not write a value to the **X-MS-TNEF-Correlator** header or the **PidTagTnefCorrelationKey** property ([MS-OXCMSG] section 2.2.1.29) in the **attMsgProps** attribute of the TNEF body part when generating a MIME message for download via IMAP4.

<57> [Section 2.1.3.2.28](#): Exchange 2003 does not write the **Reply-By** header.

<58> [Section 2.1.3.2.29](#): Office Outlook 2003 and Office Outlook 2007 do not copy the value of the **PidTagBodyContentId** property ([MS-OXCMSG] section 2.2.1.56.7) to the **Content-ID** header.

<59> [Section 2.1.3.2.30](#): Exchange 2007 does not copy the value of the **PidTagBodyContentLocation** property ([MS-OXCMSG] section 2.2.1.56.8) to the **Content-Location** header.

<60> [Section 2.1.3.2.31](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not generate an **XRef** header.

<61> [Section 2.1.3.3.1](#): Exchange 2003 defaults to ISO-8859-1 when the **PidTagInternetCodepage** property ([MS-OXCMSG] section 2.2.1.56.6) is not set by the client for plain text message bodies.

<62> [Section 2.1.3.3.1](#): Office Outlook 2003 and Office Outlook 2007 either convert the RTF text to HTML or generate a TNEF attachment that contains the RTF body.

<63> [Section 2.1.3.3.1](#): Exchange 2007 re-encodes the message body using this procedure. Exchange 2010, Exchange 2013, and Exchange 2016 Preview preserve the character set of the original MIME message body.

<64> [Section 2.1.3.3.1](#): Exchange 2010, Exchange 2013, and Exchange 2016 Preview preserve the character set as described.

<65> [Section 2.1.3.3.1](#): Exchange 2007 uses this procedure for body formats.

[<66> Section 2.1.3.3.2](#): Exchange 2003 loses body text when reading TNEF if only a plain text body is encoded in TNEF.

[<67> Section 2.1.3.3.4](#): Exchange 2003 uses this procedure to copy the body.

[<68> Section 2.1.3.3.6](#): Exchange 2003 and Exchange 2007 do not check to determine whether an apparently inline attachment is, in fact, referenced from the message body.

[<69> Section 2.1.3.3.8](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview support a configuration option to select a character set encoding other than UTF-8.

[<70> Section 2.1.3.3.8.2](#): Exchange 2003 copies the value of the **PidTagHtml** property ([MS-OXCMSG] section 2.2.1.56.9).

[<71> Section 2.1.3.3.8.2](#): Exchange 2003 uses a value of "8bit" for the **Content-Transfer-Encoding** header in this circumstance.

[<72> Section 2.1.3.3.8.2](#): Exchange 2007 uses a value of "base64" for the **Content-Transfer-Encoding** header in this circumstance.

[<73> Section 2.1.3.3.8.2](#): Exchange 2003 uses a value of "quoted-printable" for the **Content-Transfer-Encoding** header.

[<74> Section 2.1.3.3.8.2](#): Exchange 2007 uses a value of "base64" for the **Content-Transfer-Encoding** header in this circumstance.

[<75> Section 2.1.3.3.8.3](#): Exchange 2003 uses a value of "multipart/mixed" for the **Content-Type** header in this circumstance.

[<76> Section 2.1.3.3.9](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview generate MIME entities with **Content-Type** header values of "text/enriched" if configured by an administrator to do so.

[<77> Section 2.1.3.4.1](#): Exchange 2007 does not exclude attached Message objects or attachments to plain text messages from being considered inline.

[<78> Section 2.1.3.4.1.1](#): Exchange 2007 does not exclude non-OLE attachments in an RTF message from being considered inline.

[<79> Section 2.1.3.4.1.2](#): Exchange 2003 does not generate the cid URI.

[<80> Section 2.1.3.4.1.2](#): Exchange 2007 checks only condition 1, not conditions 2 or 3, when classifying attachments as inline.

[<81> Section 2.1.3.4.2.1](#): Exchange 2003, Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, Outlook 2016 Preview use an empty string if values for the **PidTagAttachLongFilename** ([MS-OXCMSG] section 2.2.2.10) and **PidTagAttachFilename** ([MS-OXCMSG] section 2.2.2.11) properties are unavailable.

[<82> Section 2.1.3.4.2.2](#): Exchange 2003 generates the value of the **PidTagAttachMimeTag** property ([MS-OXCMSG] section 2.2.2.26) only in this circumstance.

[<83> Section 2.1.3.4.2.2](#): Office Outlook 2003 and Office Outlook 2007 do not generate the **Content-Description** header.

[<84> Section 2.1.3.4.2.2](#): Office Outlook 2003 uses this procedure for the **Content-Description** header.

[<85> Section 2.1.3.4.2.2](#): Exchange 2003 and Office Outlook 2003 do not generate the **Content-Disposition** header for inline attachments.

- [<86> Section 2.1.3.4.2.2](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the *size* parameter for the **Content-Disposition** header.
- [<87> Section 2.1.3.4.2.2](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the *creation-date* parameter for the **Content-Disposition** header.
- [<88> Section 2.1.3.4.2.2](#): Exchange 2007 appends the string "GMT" to the *creation-date* parameter for the **Content-Disposition** header without adjusting the time from the server time zone to GMT.
- [<89> Section 2.1.3.4.2.2](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not generate the *modification-date* parameter for the **Content-Disposition** header. Exchange 2007 appends the string "GMT" to the *modification-date* parameter for the **Content-Disposition** header without adjusting the time from the server time zone to GMT.
- [<90> Section 2.1.3.4.2.2](#): Exchange 2007 appends the string "GMT" to the *modification-date* parameter for the **Content-Disposition** header without adjusting the time from the server time zone to GMT.
- [<91> Section 2.1.3.4.2.2](#): Exchange 2010, Exchange 2013, and Exchange 2016 Preview copy the value as described.
- [<92> Section 2.1.3.4.3](#): Office Outlook 2003 and Office Outlook 2007 do not encode attachments in the MacBinary format.
- [<93> Section 2.1.3.4.3](#): Exchange 2007 does not ignore the presence of secondary header data in a MacBinary stream; the behavior of Exchange 2007 is undefined if secondary header data is present.
- [<94> Section 2.1.3.4.3](#): Exchange 2003 uses this procedure for generating the attachment MIME content-type.
- [<95> Section 2.1.3.4.3](#): Exchange 2007 does not ignore the presence of additional data in a MacBinary stream; the behavior of Exchange 2007 is undefined if additional data is present.
- [<96> Section 2.1.3.4.3](#): Exchange 2003 and Exchange 2007 use this procedure for generating the attachment MIME content-type.
- [<97> Section 2.1.3.4.3](#): Exchange 2003 uses a maximum field length of 62 bytes for the file name, rather than 63 bytes.
- [<98> Section 2.1.3.4.4](#): Exchange 2003 sets the **Content-Type** header to "image/BMP".
- [<99> Section 2.1.3.4.4](#): Exchange 2003 uses "ole#.BMP" as the file name, where "#" is a digit representing the index number of the attached file.
- [<100> Section 2.1.3.4.4](#): Exchange 2003 does not produce a **Content-Disposition** header for OLE attachments.
- [<101> Section 2.1.3.4.4](#): Exchange 2003 keeps OLE renderings in their bitmap type. Office Outlook 2003 and Office Outlook 2007 do not convert OLE attachments. OLE attachments are omitted from the MIME version of the message.
- [<102> Section 2.1.3.4.6](#): Exchange 2003 and Exchange 2007 do not support generating vCard attachments from MIME.
- [<103> Section 2.1.3.4.6](#): Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not support vCard version 2.1 on outbound MIME messages.
- [<104> Section 2.1.3.5](#): Exchange 2003, Exchange 2007, Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not support the functionality specified in this section.

<105> [Section 2.1.3.5.1](#): Exchange 2010, Exchange 2013, and Exchange 2016 Preview set the **MIME-Version** header to "1.0".

<106> [Section 2.2](#): Outlook 2010, Outlook 2013, and Outlook 2016 Preview prefer TNEF body part data but will discard information from the TNEF message (such as recipients (2)) that it does not need.

<107> [Section 2.2.3.1.1](#): Office Outlook 2003 and Office Outlook 2007 do not check for IMCEA encapsulation and do not perform de-encapsulation.

<108> [Section 2.2.3.1.2](#): Office Outlook 2003 and Office Outlook 2007 do not check for IMCEA encapsulation and do not perform de-encapsulation.

<109> [Section 2.2.3.1.3](#): Office Outlook 2003 and Office Outlook 2007 will not use the **attSentFor** attribute, as described in [\[MS-OXTNEF\]](#) section 2.1.3.3.17, or the values from the **PidTagSentRepresenting** property group (section 2.6.5) if the **From** header is absent.

<110> [Section 2.2.3.2](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview use the last instance of a header to set the value of the corresponding property.

<111> [Section 2.2.3.2.5](#): Exchange 2003 only generates the **Importance** header, not the **X-Priority** header.

<112> [Section 2.2.3.2.5](#): Exchange 2003 maps a **Priority** header value of "Urgent" to the **PidTagImportance** property ([MS-OXCMSG] section 2.2.1.11) with the value 0x00000001.

<113> [Section 2.2.3.2.7](#): Exchange 2007 uses this procedure to set the **PidTagConversationTopic** property ([MS-OXCMSG] section 2.2.1.10).

<114> [Section 2.2.3.2.11](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not copy the value of the **Accept-Language** or **X-Accept-Language** header.

<115> [Section 2.2.3.2.13](#): Exchange 2007 selects between the two headers by using this procedure.

<116> [Section 2.2.3.2.14](#): Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not read the **X-Auto-Response-Suppress** header.

<117> [Section 2.2.3.2.14](#): Exchange 2003 uses this value for the **PidTagAutoResponseSuppress** property ([MS-OXOMSG] section 2.2.1.77).

<118> [Section 2.2.3.2.14](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 ignore the **X-Auto-Response-Suppress** and **Precedence** headers.

<119> [Section 2.2.3.2.15](#): Exchange 2003 does not generate the **Content-Class** header.

<120> [Section 2.2.3.2.15](#): Exchange 2003 sets the value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) for all **Content-Class** header values specified in this section to "IPM.Note".

<121> [Section 2.2.3.2.15](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do no special processing for "urn:content-class:custom."

<122> [Section 2.2.3.2.16](#): Exchange 2003 does not support the id-mapped properties used in this section: **PidLidToDoTitle** ([\[MS-OXOFLAG\]](#) section 2.2.1.12), **PidLidTaskStatus** ([\[MS-OXOTASK\]](#) section 2.2.2.2.2), **PidLidTaskDueDate** ([\[MS-OXOTASK\]](#) section 2.2.2.2.5), **PidLidTaskStartDate** ([\[MS-OXOTASK\]](#) section 2.2.2.2.4), **PidLidTaskDateCompleted** ([\[MS-OXOTASK\]](#) section 2.2.2.2.9), **PidLidTaskComplete** ([\[MS-OXOTASK\]](#) section 2.2.2.2.20), and **PidLidPercentComplete** ([\[MS-OXOFLAG\]](#) section 2.2.2.3).

<123> [Section 2.2.3.2.17](#): Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not read the **X-List-Help**, **X-List-Subscribe**, or **X-List-Unsubscribe** headers.

<124> [Section 2.2.3.2.18](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not support retrieving the value of the **PidTagAttachPayloadProviderGuidString** property ([MS-OXCMSG] section 2.2.2.26).

<125> [Section 2.2.3.2.18](#): Exchange 2003 does not write either an **X-Payload-class** or an **X-Payload-Provider-GUID** header.

<126> [Section 2.2.3.2.18](#): Office Outlook 2003 does not generate the **PidTagAttachPayloadClass** property ([MS-OXCMSG] section 2.2.2.26) on incoming messages. Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not read or write the **X-Payload-Class** and **X-Payload-Provider-GUID** headers.

<127> [Section 2.2.3.2.18](#): Outlook 2010, Outlook 2013, and Outlook 2016 Preview set the **PidTagAttachPayloadClass** property ([MS-OXCMSG] section 2.2.2.26) and the **PidTagAttachPayloadProviderGuidString** property ([MS-OXCMSG] section 2.2.2.26), but sets them on the attachment instead of the message.

<128> [Section 2.2.3.2.18](#): Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not read the **X-Payload-Class** and **X-Payload-Provider-GUID** headers when they appear on a MIME entity that is analyzed as a message or message body, rather than as an attachment.

<129> [Section 2.2.3.2.19](#): Office Outlook 2003 and Office Outlook 2007 neither read nor write the **PidTagPurportedSenderDomain** property ([MS-OXCMSG] section 2.2.1.43).

<130> [Section 2.2.3.2.20](#): Office Outlook 2003 and Office Outlook 2007 neither read nor write the **PidTagSenderIdStatus** property ([MS-OXPROPS] section 2.992).

<131> [Section 2.2.3.2.21](#): Office Outlook 2003 and Office Outlook 2007 neither read nor write the **PidTagContentFilterSpamConfidenceLevel** property ([MS-OXCSPAM] section 2.2.1.3).

<132> [Section 2.2.3.2.22](#): Office Outlook 2007, Outlook 2010, Outlook 2013, and Outlook 2016 Preview neither read nor write the **X-Microsoft-Classified** header.

<133> [Section 2.2.3.2.22](#): Exchange 2003, Office Outlook 2003 and Office Outlook 2007 do not read or set any of the classification headers.

<134> [Section 2.2.3.2.23](#): Exchange 2003, Office Outlook 2003 and Office Outlook 2007 do not read or set any of the Unified Messaging headers.

<135> [Section 2.2.3.2.24](#): Office Outlook 2003 and Office Outlook 2007 do not copy the value of the **Content-ID** header to the **PidTagBodyContentId** property ([MS-OXCMSG] section 2.2.1.56.7).

<136> [Section 2.2.3.2.25](#): Office Outlook 2003 and Office Outlook 2007 do not copy the value of a **Content-Base** header to the **PidNameContentBase** property.

<137> [Section 2.2.3.2.26](#): Exchange 2003, Office Outlook 2003 and Office Outlook 2007 do not copy the value of a **Content-Location** header to the **PidTagBodyContentLocation** property ([MS-OXCMSG] section 2.2.1.56.8).

<138> [Section 2.2.3.2.27](#): Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not read the **XRef** header.

<139> [Section 2.2.3.2.27](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not support copying the value of the **PidNameCrossReference** property (section [2.5.3](#)).

<140> [Section 2.2.3.2.28](#): Office Outlook 2003 and Office Outlook 2007 set the **PidTagTransportMessageHeaders** property ([MS-OXOMSG] section 2.2.1.61) under this circumstance.

<141> [Section 2.2.3.2.29](#): Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not create the named property corresponding to a generic header if (1) the mailbox database named property quota has been exceeded or (2) measures adopted to prevent exhausting the named property quota interfere with creating the named property. Office Outlook 2003 and Office Outlook 2007 do not create named properties for headers.

<142> [Section 2.2.3.3.2](#): Exchange 2010 SP2, Exchange 2013, and Exchange 2016 Preview create aggregate bodies for "multipart/mixed" MIME entities

<143> [Section 2.2.3.3.2.2](#): Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not create aggregate bodies. Exchange 2010 SP2 creates aggregate bodies if the specified values are found in the **MIME-Version** or **X-User-Agent** headers only. Update Rollup 1 for Exchange Server 2010 Service Pack 2 (SP2), Exchange 2013, and Exchange 2016 Preview create aggregate bodies as specified.

<144> [Section 2.2.3.4.1.1](#): Exchange 2003 replaces all of the illegal characters with the underscore "_" (U+005F), except for the backslash "\" (U+005C), which is removed.

<145> [Section 2.2.3.4.1.1](#): Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not sanitize the display name of attachments.

<146> [Section 2.2.3.4.1.1](#): Exchange 2007 does not strip control characters U+0001 through U+0004 from the attachment file name.

<147> [Section 2.2.3.4.1.1](#): Exchange 2003 does not perform the steps when it creates the value of the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.11) and the value of the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.5).

<148> [Section 2.2.3.4.1.1](#): Exchange 2007 removes leading spaces from the start of the base. It removes trailing spaces from the end of the extension. It does not remove "." (U+002E) characters from the base.

<149> [Section 2.2.3.4.1.1](#): Exchange 2003 copies a sanitized version of the file name to the **PidTagAttachLongFilename** property; it generates a base and extension only when there is no MIME file name. Exchange 2007 does not generate an extension for the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.11) when only the name portion is in a MIME message. Exchange 2010, Exchange 2013, and Exchange 2016 Preview do not generate an extension when only the name portion is in a MIME message.

<150> [Section 2.2.3.4.1.1](#): Exchange 2003 does not replace an empty display name with the base part of the file name. The display name contains only the file extension.

<151> [Section 2.2.3.4.1.1](#): Exchange 2003 copies the value of the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.11) from the MIME header value after removing invalid file-name characters. Only step 1 of the included list of seven steps is completed before the MIME header value is copied.

<152> [Section 2.2.3.4.1.1](#): Exchange 2003 removes plus "+" (U+002B), equal "=" (U+003D), left square bracket "[" (U+005B), right square bracket "]" (U+005D), and semicolon ";" (U+003B).

<153> [Section 2.2.3.4.1.1](#): Exchange 2003 replaces the question mark "?" (U+003F) and the asterisk "*" (U+002A) with the underscore "_" (U+005F).

<154> [Section 2.2.3.4.1.1](#): Exchange 2003 does not remove the apostrophe "'" (U+0027).

[<155> Section 2.2.3.4.1.1](#): Exchange 2003 sets the **PidTagAttachFilename** property ([MS-OXCMSG] section 2.2.2.11) to "NONAME" when the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10) is empty.

[<156> Section 2.2.3.4.1.1](#): Exchange 2003 deletes the extension portion of the file name copied to the **PidTagAttachFilename** property ([MS-OXCMSG] section 2.2.2.11) when the original extension is greater than three characters in length.

[<157> Section 2.2.3.4.1.1](#): Exchange 2003 truncates the name part of the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10) to eight characters and does not add "~1".

[<158> Section 2.2.3.4.1.1](#): Exchange 2003 does not generate the extension part of the **PidTagAttachFilename** property ([MS-OXCMSG] section 2.2.2.11).

[<159> Section 2.2.3.4.1.2](#): Exchange 2003 only replaces the "application/ms-tnef " value with "application/octet-stream" when it is the **Content-Type** header for the root body part.

[<160> Section 2.2.3.4.1.3](#): Office Outlook 2003 and Office Outlook 2007 do not use the parameters of the **Content-Disposition** header to set creation or modification dates.

[<161> Section 2.2.3.4.1.3](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 Preview set the **PidTagCreationTime** property ([MS-OXCMSG] section 2.2.2.3) and the **PidTagLastModificationTime** property ([MS-OXCMSG] section 2.2.2.2) by using this procedure.

[<162> Section 2.2.3.4.1.3](#): Exchange 2003 uses current system time to set the **PidTagLastModificationTime** property ([MS-OXCMSG] section 2.2.2.2).

[<163> Section 2.2.3.4.1.4](#): Exchange 2003 does not map the **Content-Base** header to the **PidTagAttachContentBase** property ([MS-OXCMSG] section 2.2.2.26) on attachments.

[<164> Section 2.2.3.4.1.4.1](#): Exchange 2007 does not verify that inline attachment candidates are referenced from an HTML element in the HTML message body before marking them as inline.

[<165> Section 2.2.3.4.1.4.1](#): The initial release version of Exchange 2010 and Microsoft Exchange Server 2010 Service Pack 1 (SP1) do not check the **Content-Type** header of inline attachment candidates. Exchange 2010 SP2, Exchange 2013, and Exchange 2016 Preview check the **Content-Type** header.

[<166> Section 2.2.3.4.1.4.1](#): Microsoft Office Outlook 2007 QFE 2276479, as described in [\[MSKB2276479\]](#), sets both the **attInvisibleInHtml** and the **attRenderedInBody** flag in the **PidTagAttachFlags** property ([MS-OXCMSG] section 2.2.2.18).

[<167> Section 2.2.3.4.2.1](#): Exchange 2003 does not write the **PidNameAttachmentMacContentType** property ([MS-OXCMSG] section 2.2.2.26).

[<168> Section 2.2.3.4.2.1](#): Exchange 2003 does not write the **PidNameAttachmentMacInfo** property ([MS-OXCMSG] section 2.2.2.26).

[<169> Section 2.2.3.4.2.1](#): Exchange 2003 does not reject the message as not MIME-compliant when parsing of the header part fails.

[<170> Section 2.2.3.4.2.1](#): Exchange 2007 gets the attachment file name from AppleSingle data, instead of preferring a file name found in MIME headers.

[<171> Section 2.2.3.4.2.2](#): Exchange 2003 does not reject the message when the MIME reader fails to parse the MIME body.

[<172> Section 2.2.3.4.2.2](#): Exchange 2007 gets the attachment file name from AppleSingle or MacBinary data, instead of preferring a file name found in MIME headers.

<173> [Section 2.2.3.4.2.2](#): Exchange 2003 does not map the MacBinary version number (MacBinary header offset 122) or the minimum MacBinary version number (MacBinary header offset 123) into the **PidTagAttachDataBinary** property ([MS-OXCMSG] section 2.2.2.7).

<174> [Section 2.2.3.4.2.2](#): Exchange 2007 does not support a MacBinary structure with additional header data or comment part present.

<175> [Section 2.2.3.4.2.2](#): Exchange 2003 uses a maximum field length of 62 bytes for file name, not 63 bytes.

<176> [Section 2.2.3.4.2.2](#): Exchange 2003 does not set the signature.

<177> [Section 2.2.3.4.2.2](#): Exchange 2003 uses bytes 75:76 to map the horizontal location of the icon.

<178> [Section 2.2.3.4.2.2](#): Exchange 2003 uses bytes 77:78 to map the vertical location of the icon.

<179> [Section 2.2.3.4.2.2](#): Exchange 2003 and Exchange 2007 conversion handle only EntryID values of 1, 2, 3, 8, and 9. Other EntryID values are ignored.

<180> [Section 2.2.3.4.2.3](#): Exchange 2007 does not ignore the **Content-Transfer-Encoding** header when dealing with BinHex60 attachments. If the **Content-Transfer-Encoding** header is present, then Exchange 2007 uses the encoding method specified in the header to process the data.

<181> [Section 2.2.3.4.2.3](#): Exchange 2003 does not set the value of the **PidNameAttachmentMacInfo** property ([MS-OXCMSG] section 2.2.2.26).

<182> [Section 2.2.3.4.2.3](#): Exchange 2003 does not decode the attachment data and sets the **PidTagAttachDataBinary** property ([MS-OXCMSG] section 2.2.2.7) to the encoded binhex40 attachment data.

<183> [Section 2.2.3.4.2.3](#): Exchange 2007 gets the attachment file name from BinHex data instead of preferring a file name found in MIME headers.

<184> [Section 2.2.3.4.2.3](#): Exchange 2003 generates the values of the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.5) and the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10) from BinHex data, instead of preferring a file name found in MIME headers. The **PidTagAttachFilename** property ([MS-OXCMSG] section 2.2.2.11) is generated from the MIME headers.

<185> [Section 2.2.3.4.3](#): Exchange 2003 uses this procedure when generating the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.5).

<186> [Section 2.2.3.4.3](#): Exchange 2003 does not write the file name value of an embedded attachment to the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10).

<187> [Section 2.2.3.4.3](#): Exchange 2003 does not save the resulting extension value in the **PidTagAttachExtension** property ([MS-OXCMSG] section 2.2.2.12).

<188> [Section 2.2.3.4.3](#): Exchange 2003 does not generate the value of the **PidTagAttachFilename** property ([MS-OXCMSG] section 2.2.2.11) for embedded message attachments.

<189> [Section 2.2.3.4.3](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 do not save the **X-MS-Exchange-Organization-Original-Sender** header value.

<190> [Section 2.2.3.4.3](#): Exchange 2003, Office Outlook 2003, and Office Outlook 2007 exclude unknown MIME headers that start with "X-MS-Exchange-Organization-" or "X-MS-Exchange-Forest-" from analysis.

<191> [Section 2.2.3.4.4](#): Exchange 2003 and Exchange 2007 do not support converting vCard attachments to Contact objects.

<192> [Section 2.2.3.4.4.1](#): Exchange 2003 and Exchange 2007 do not support vCard Version 2.1.

<193> [Section 2.2.3.5](#): Office Outlook 2003 and Office Outlook 2007 analyze attachment MIME parts with the **Content-Type** header set to "message/external-body" the same as they do ordinary file attachments. No special analysis is performed.

<194> [Section 2.2.3.5](#): Exchange 2007 does not append the "URL" extension when the name is empty.

<195> [Section 2.2.3.5](#): Exchange 2003 sanitizes the external body attachment file name using this procedure.

<196> [Section 2.2.3.5](#): Exchange 2003 uses this procedure when the base part of the external body attachment file name is empty.

<197> [Section 2.2.3.6](#): Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not copy data into the **PidTagMimeSkeleton** property ([MS-OXCMSG] section 2.2.1.28).

<198> [Section 2.2.3.7.1](#): Exchange 2010, Exchange 2010 SP1, and Exchange 2010 SP2 do not skip processing of sets of per-recipient delivery status notification fields that contain an **Action** header value that does not match the highest severity value. Update Rollup 5 for Exchange Server 2010 Service Pack 2 (SP2) does skip processing of these sets of per-recipient delivery status notification fields.

<199> [Section 2.2.3.9.2](#): Exchange 2003, Exchange 2007, Microsoft Exchange Server 2007 Service Pack 1 (SP1), and Microsoft Exchange Server 2007 Service Pack 2 (SP2) do not reject a message containing the **Content-Type** header of "message/partial".

<200> [Section 2.4](#): Exchange 2007 does not have this functionality. Exchange 2003 does not have this functionality but does save the entire original MIME message.

<201> [Section 2.4](#): Outlook 2010, Outlook 2013, and Outlook 2016 Preview do not create the **PidTagMimeSkeleton** property ([MS-OXCMSG] section 2.2.1.28).

6 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 Algorithm Details	Updated product behavior notes for the "Algorithm Details" section to include behavior of Exchange 2016 and Outlook 2016.	Y	Product behavior note updated.
5 Appendix A: Product Behavior	Added Exchange 2016 and Outlook 2016 to the list of applicable products.	Y	Content update.

Preliminary

7 Index

A

Abstract data model
[MIME analysis](#) 58
[MIME generation](#) 23
[unconverted MIME part analysis](#) 101
[unconverted MIME part generation](#) 96
[Applicability](#) 21

C

[Change tracking](#) 127
[Considerations for message/external-body security](#)
113
[Content-type versus file extension mismatch security](#)
[considerations](#) 112

D

Data model - abstract
[MIME analysis](#) 58
[MIME generation](#) 23
[unconverted MIME part analysis](#) 101
[unconverted MIME part generation](#) 96
[Data models](#) 19
[Do not support message/partial security](#)
[considerations](#) 113

G

Global ADM type
[MIME analysis](#) 58
[MIME generation](#) 23
[unconverted MIME part analysis](#) 102
[unconverted MIME part generation](#) 96
[Glossary](#) 10

I

[Index of security parameters](#) 114
[Information disclosure security considerations](#) 112
[Informative references](#) 18
Initialization
[MIME analysis](#) 58
[MIME generation](#) 23
[unconverted MIME part analysis](#) 102
[unconverted MIME part generation](#) 97
[Introduction](#) 10

M

Message object properties
[overview](#) 103
[PidLidClassificationGuid](#) 103
[PidLidClassificationKeep](#) 103
[PidNameCrossReference](#) 103
[PidNameQuarantineOriginalSender](#) 103
MIME analysis
[abstract data model](#) 58

[initialization](#) 58
[overview](#) 58
MIME analysis - processing rules
[address elements](#) 59
[attachments](#) 77
[body text](#) 74
[envelope elements](#) 62
[external body attachments](#) 89
[overview](#) 59
[reading pure MIME messages](#) 90
MIME analysis ADM types
[global](#) 58
[per mailbox](#) 58
[per Message object](#) 58
MIME examples
[MIME message containing inline and non-inline](#)
[attachments](#) 109
[MIME message containing only inline attachments](#)
110
[MIME message containing only non-inline](#)
[attachments](#) 110
[MIME message without a MIME-version header](#) 111
[simple MIME message](#) 107
MIME Generation
[abstract data model](#) 23
[initialization](#) 23
[overview](#) 22
MIME generation - processing rules
[address elements](#) 24
[attachments](#) 44
[body text](#) 39
[envelope elements](#) 30
[generating pure MIME messages](#) 52
[overview](#) 23
MIME generation ADM types
[global](#) 23
[per mailbox](#) 23
[per Message object](#) 23
[MIME message containing inline and non-inline](#)
[attachments example](#) 109
[MIME message containing only inline attachments](#)
[example](#) 110
[MIME message containing only non-inline](#)
[attachments example](#) 110
[MIME message without a MIME-version header](#)
[example](#) 111

N

[Normative references](#) 16

O

[Overview - data models](#) 19
[Overview \(synopsis\)](#) 19

P

[Parameters - security index](#) 114
Per mailbox ADM type

[MIME analysis](#) 58
[MIME generation](#) 23
[unconverted MIME part analysis](#) 102
[unconverted MIME part generation](#) 96

Per Message object ADM type
[MIME analysis](#) 58
[MIME generation](#) 23
[unconverted MIME part analysis](#) 102
[unconverted MIME part generation](#) 96

[PidLidClassificationGuid Message object property](#) 103
[PidLidClassificationKeep Message object property](#) 103
[PidNameCrossReference Message object property](#) 103
[PidNameQuarantineOriginalSender Message object property](#) 103
[PidTagReadReceipt property group](#) 104
[PidTagReceivedBy property group](#) 104
[PidTagReceivedRepresenting property group](#) 105
[PidTagSender property group](#) 105
[PidTagSentRepresenting property group](#) 105

Preventing denial of service attacks security considerations
[complexity of nested entities](#) 114
[compressed attachments](#) 114
[number of embedded messages](#) 114
[submission limits](#) 113

Processing rules - MIME analysis
[address elements](#) 59
[attachments](#) 77
[body text](#) 74
[envelope elements](#) 62
[external body attachments](#) 89
[overview](#) 59
[reading pure MIME messages](#) 90

Processing rules - MIME generation
[address elements](#) 24
[attachments](#) 44
[body text](#) 39
[envelope elements](#) 30
[generating pure MIME messages](#) 52
[overview](#) 23

Processing rules - unconverted MIME part analysis
[MIME conversion](#) 102

Processing rules - unconverted MIME part generation
[impact of message changes on the MIME skeleton](#) 97

[Product behavior](#) 115

R

Recipient property groups
[overview](#) 103
[PidTagReadReceipt property group](#) 104
[PidTagReceivedBy property group](#) 104
[PidTagReceivedRepresenting property group](#) 105
[PidTagSender property group](#) 105
[PidTagSentRepresenting property group](#) 105
[Recipient Table property group](#) 106
[Recipient Table property group](#) 106

References
[informative](#) 18
[normative](#) 16

Relationship to
[other algorithms](#) 20

[other protocols](#) 20

S

Security
[parameter index](#) 114

Security considerations
[considerations for message/external-body](#) 113
[content-type versus file extension mismatch](#) 112
[do not support message/partial](#) 113
[information disclosure](#) 112
[preventing denial of service attacks – complexity of nested entities](#) 114
[preventing denial of service attacks – compressed attachments](#) 114
[preventing denial of service attacks – number of embedded messages](#) 114
[preventing denial of service attacks – submission limits](#) 113
[unsolicited commercial e-mail \(spam\)](#) 112

[Simple MIME message example](#) 107

[Standards assignments](#) 21

T

[Tracking changes](#) 127

U

Unconverted MIME part analysis
[abstract data model](#) 101
[initialization](#) 102
[overview](#) 101

Unconverted MIME part analysis ADM types
[global](#) 102
[per mailbox](#) 102
[per Message object](#) 102
[Unconverted MIME part analysis processing rules - MIME conversion](#) 102

Unconverted MIME part generation
[abstract data model](#) 96
[initialization](#) 97
[overview](#) 96

Unconverted MIME part generation - processing rules
[impact of message changes on the MIME skeleton](#) 97

Unconverted MIME part generation ADM types
[global](#) 96
[per mailbox](#) 96
[per Message object](#) 96
[Unsolicited commercial e-mail \(spam\) security considerations](#) 112