[MS-OXTNEF]:

Transport Neutral Encapsulation Format (TNEF) Data 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.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1		Initial Availability.
4/25/2008	0.2		Revised and updated property names and other technical content.
6/27/2008	1.0		Initial Release.
8/6/2008	1.01		Revised and edited technical content.
9/3/2008	1.02		Revised and edited technical content.
12/3/2008	1.03		Revised and edited technical content.
3/4/2009	1.04		Revised and edited technical content.
4/10/2009	2.0		Updated technical content and applicable product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	3.1.0	Minor	Updated the technical content.
2/10/2010	4.0.0	Major	Updated and revised the technical content.
5/5/2010	5.0.0	Major	Updated and revised the technical content.
8/4/2010	5.1	Minor	Clarified the meaning of the technical content.
11/3/2010	5.2	Minor	Clarified the meaning of the technical content.
3/18/2011	5.2	No change	No changes to the meaning, language, and formatting of the technical content.
8/5/2011	5.2	No Change	No changes to the meaning, language, or formatting of the technical content.
10/7/2011	6.0	Major	Significantly changed the technical content.
1/20/2012	7.0	Major	Significantly changed the technical content.
4/27/2012	7.1	Minor	Clarified the meaning of the technical content.
7/16/2012	7.1	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	8.0	Major	Significantly changed the technical content.
2/11/2013	8.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	8.1	Minor	Clarified the meaning of the technical content.
11/18/2013	8.1	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	8.1	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	8.2	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
7/31/2014	8.2	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	8.3	Minor	Clarified the meaning of the technical content.
3/16/2015	9.0	Major	Significantly changed the technical content.
5/26/2015	9.0	No Change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Transport Neutral Encapsulation Format (TNEF) Data Algorithm enables the **encoding** of rich properties in electronic mail messages over a serial data stream. The result can be transported as a stream, as a file attachment in an arbitrary transport, or as a **MIME** entity body on an Internet transport.

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:

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

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].

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

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

character set: (1) A mapping between the characters of a written language and the values that are used to represent those characters to a computer.

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

checksum: A value that is the summation of a byte stream. By comparing the checksums computed from a data item at two different times, one can quickly assess whether the data items are identical.

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 (1)** 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.

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

email address: A string that identifies a user and enables the user to receive Internet messages.

encoding: A process that specifies a Content-Transfer-Encoding for transforming character data from one form to another.

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

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].

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

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.

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.

plain text message body: A message body (2) for which the Content-Type value of the Email Text Body header field is "text/plain". A plain text message body can be identified explicitly in the content, or implicitly if it is in a message that is as described in [RFC822] or a message that does not contain a Content-Type header field.

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

recipient: An entity that can receive email messages.

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].

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) Reader: An entity that decodes a **Transport Neutral Encapsulation Format (TNEF)** structure after receiving a message, for the purpose of reconstructing the rich properties that are contained in the stream.

Transport Neutral Encapsulation Format (TNEF) Writer: An entity that encodes or builds a **Transport Neutral Encapsulation Format (TNEF)** structure for the purpose of transporting rich properties.

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).

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents

in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

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

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

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

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

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

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

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

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

[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

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

1.2.2 Informative References

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

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

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

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

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

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

[MSDN-UAF] Microsoft Corporation, "UUENCODE Attachment Format", http://msdn.microsoft.com/en-us/library/aa579638(v=EXCHG.80).aspx

[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

1.3 Overview

This algorithm organizes a hierarchy of rich message properties into a flattened structure that can be represented as a serial data stream. The typical format of a particular property within the stream is: identifier (which usually also includes type information), size (where not exactly determined by type),

and data. In some cases, as described in this document, groups of properties or multiple-value properties include counts. Others might include padding to enforce a particular alignment of the data.

A typical scenario for using this algorithm is as follows. A **TNEF Writer** encodes rich properties into a serial data stream in order to transmit the properties through a messaging system that does not support those properties directly. By encoding the properties in **TNEF**, the properties that do not have direct representations in the underlying messaging system can be encapsulated during transport and then decoded by a **TNEF Reader** in order to make all the properties included in the original message available to the client application.

1.4 Relationship to Protocols and Other Algorithms

This algorithm is intended to permit the transmission of rich message property information over transports that have no mechanism for representing that information natively.

The output of the algorithm can be included in any of the following:

- A file attachment (winmail.dat).
- A MIME **body part**, as described in [RFC2045], using the "application/ms-TNEF" media type.
- As an addition to the transmitted **plain text message body** using UUENCODE, as described in [MSDN-UAF], or a similar method to be decoded at the **recipient** end.
- A transmission from the sender to the recipient using whatever means are provided by the protocol employed in transmitting message information between them.

Specifically, this algorithm transmits message data over **Simple Mail Transfer Protocol (SMTP)**, **Post Office Protocol - Version 3 (POP3)**, **Internet Message Access Protocol - Version 4 (IMAP4)**, or other Internet protocols that incorporate MIME, as described in [RFC2045].

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

1.5 Applicability Statement

The original application of the algorithm was to permit the creation and representation of **message classes** other than simple email messages, and some additional attributes that were not natively supported by the transport protocol.

This application was further extended to allow the transport of the rich set of properties required by more modern messaging clients, including **named properties**. For backward compatibility with the original implementation, a special attribute is used to encapsulate the new message properties, and those properties with analogues to the original implementation are usually represented using the original attribute syntax.

1.6 Standards Assignments

None.

2 Algorithm Details

2.1 Common Algorithm Details

This section specifies details that are common to both the TNEF Writer and TNEF Reader roles. For details specific to the TNEF Writer role, see section 2.2. For details specific to the TNEF Reader role, see section 2.3.

All numeric data types in this algorithm that are greater than one byte in size are little-endian, and any handling of these values on platforms that are not **little-endian** are required to take this into account and perform the appropriate transformations to get correct numbers, counts, values, and so on.

String examples in this document are shown in **Augmented Backus-Naur Form (ABNF)** format, as specified in [RFC5234]. When the string has a terminating null character, the terminating null character is included as well; for example, "user1@example.com" %x00. For the purpose of string examples, the different terminating null character size in a **Unicode character set (1)** is not illustrated.

2.1.1 Abstract Data Model

None.

2.1.2 Initialization

None.

2.1.3 Processing Rules

In this specification, attributes using the original syntax described in section 1.5 are referred to as "attributes" and the rich set of properties described in section 1.5 will be referred to as "properties".

The TNEF stream starts with a signature, a legacy key value, an attribute containing a legacy version number, and an attribute containing the **code page** used by the encoder for ANSI attributes and properties. After that, the stream is a series of attributes laid out, one after the other – message attributes followed by attachment attributes. The special attributes **attMsgProps**, **attRecipTable**, and **attAttachment** contain the various message and attachment properties. **attMsgProps** and **attAttachment** are counted lists; **attRecipTable** is a counted list of counted lists.

Each attribute is laid out as follows: the level at which it applies (message or attachment), ID, length of the contained data, the data itself, and a simple 16-bit **checksum** of the bytes comprising the data.

The **attMsgProps** attribute SHOULD be encoded after all other message attributes, and the **attAttachment** attribute SHOULD be encoded after all other attachment attributes. Values of encapsulated properties SHOULD be used instead of any conflicting mapped attribute values. Message attributes and properties SHOULD be encoded before attachment attributes and properties.

Each set of attachment attributes MUST begin with the **attAttachRendData** attribute, as specified in section 2.1.3.3.15, followed by any other attributes; attachment properties encoded in the **attAttachment** attribute, as specified in section 2.1.3.3.23, SHOULD be last.

2.1.3.1 Conventions

This algorithm uses the conventions and common definitions defined in this section for address representations and ABNF rule definitions.

2.1.3.1.1 Address Representations

Address elements other than recipients, such as From and Sender, are represented in a **Message object** by a group of four properties: **display name**, **address type**, **email address**, **EntryID**. In subsequent sections, these groups are referred to as described here.

PidTagReceivedRepresenting_XXX: Refers to the **PidTagReceivedRepresentingName** ([MS-OXOMSG] section 2.2.1.26), **PidTagReceivedRepresentingAddressType** ([MS-OXOMSG] section 2.2.1.23), and **PidTagReceivedRepresentingEmailAddress** ([MS-OXOMSG] section 2.2.1.24), or **PidTagReceivedRepresentingEntryId** ([MS-OXOMSG] section 2.2.1.25) properties, where either would suffice to fully represent the display name, email transport type, and email address of a particular recipient or person on behalf of whom a message was received.

PidTagSender_XXX: Refers to **PidTagSenderName** ([MS-OXOMSG] section 2.2.1.51), **PidTagSenderAddressType** ([MS-OXOMSG] section 2.2.1.48), and **PidTagSenderEmailAddress** ([MS-OXOMSG] section 2.2.1.49), or **PidTagSenderEntryId** ([MS-OXOMSG] section 2.2.1.50) properties, where either would suffice to fully represent the display name, email transport type, and email address of a sender.

PidTagSentRepresenting_XXX: Refers to **PidTagSentRepresentingName** ([MS-OXOMSG] section 2.2.1.57), **PidTagSentRepresentingAddressType** ([MS-OXOMSG] section 2.2.1.54), and **PidTagSentRepresentingEmailAddress** ([MS-OXOMSG] section 2.2.1.55), or **PidTagSentRepresentingEntryId** ([MS-OXOMSG] section 2.2.1.56) properties, where either would suffice to fully represent the display name, email transport type, and email address of a sender or person on behalf of whom a message was sent.

2.1.3.1.2 ABNF Rules

This section specifies ABNF rules for data types common throughout section 2. For more details about these data types, see [MS-DTYP].

Data type	ABNF rules
INT16	**INT16** = ["-"] 1*5DIGIT
INT32	**INT32** = ["-"] 1*10DIGIT
INT64	**INT64** = ["-"] 1*19DIGIT
UINT16	**UINT16** = 1*5DIGIT
UINT32	**UINT32** = 1*10DIGIT
OCTET	**%x0-FF**
CHAR	**%x0-FF**

2.1.3.2 ABNF Description

```
TNEFStream = TNEFHeader TNEFVersion OEMCodePage MessageData *AttachData

TNEFHeader = TNEFSignature LegacyKey
TNEFSignature = %x78.9F.3E.22

; Any number will suffice here. This is now legacy.
LegacyKey = UINT16
```

```
TNEFVersion = attrLevelMessage idTnefVersion Length TNEFVersionData Checksum

OEMCodePage = attrLevelMessage idOEMCodePage Length OEMCodePageData Checksum

MessageData = *MessageAttribute [MessageProps]
MessageAttribute = attrLevelMessage idMessageAttr Length Data Checksum
MessageProps = attrLevelMessage idMsgProps Length Data Checksum

; An attachment is determined/delimited by attAttachRendData, followed by
; other encoded attributes, if any, and ending with attAttachment if there are
; any encoded properties.
AttachData = AttachRendData [*AttachAttribute] [AttachProps]
AttachRendData = attrLevelAttachment idAttachRendData Length Data Checksum
AttachAttribute = attrLevelAttachment idAttachAttr Length Data Checksum
AttachProps = attrLevelAttachment idAttachment Length Data Checksum

; TNEF Version. Any number will suffice here. This is now legacy.
TNEFVersionData = 4*4OCTET

; This is the code page of attribute strings. The TNEF stream MUST contain a
; code page in which all characters are 8 bits
; in length, for compatibility with legacy applications that cannot handle
; strings with embedded zero characters.
OEMCodePageData = PrimaryCodePage SecondaryCodePage
PrimaryCodePage=UINT32

; Secondary CodePage is unused. It SHOULD contain zero.
SecondaryCodePage=%x00.00.00.00

; The length of the following data field in bytes. All attribute lengths are
; 32-bit integers, including any terminating null characters.
Length = INT32

; Data of the attribute itself, flattened out based on the particular attribute
; according to the rules that follow.
Data = 0*OCTET

; 16-bit unsigned integer that is the sum, modulo 65536, of the data bytes for
; the attribute value data, calculated over the entire length of the attribute data.
; In the case where the attribute contains enhanced properties with padding, the
; pad bytes MUST be included in the calculation.
Checksum = UINT16

; Level where attribute applies, either to the message itself or to
; an attachment.
attrLevelMessage = %x01
attrLevelAttachment = %x02

; Attribute ID Tags

; TNEF Version
idTnefVersion = %x06.90.08.00

; OEM Codepage. See attOemCodepage handling in section 5 and Appendix A.
idOemCodepage = %x07.90.06.00

; Message-level attributes. SHOULD all be at attrLevelMessage.
idMessageAttr = idMessageClass / idFrom / idSubject / idDateSent /
 idDateRecd / idMessageStatus / idMessageID /
 idBody / idPriority / idDateModified /
 idRecipTable / idOriginalMessageClass / idOwner / idSentFor /
 idDelegate / idDateStart / idDateEnd / idAidOwner / idRequestRes

;  PidTagMessageClass property ([MS-OXOMSG] section 2.2.2.1)
idMessageClass = %x08.80.07.00

;  PidTagSender_XXX properties, as specified in section 2.1.3.1.1
idFrom = %x00.80.00.00
```

```
;  PidTagSubject property ([MS-OXCMSG] section 2.2.1.46)
idSubject = %x04.80.01.00

; PidTagClientSubmitTime property ([MS-OXOMSG] section 2.2.3.11)
idDateSent = %x05.80.03.00

; PidTagMessageDeliveryTime property ([MS-OXOMSG] section 2.2.3.9)
idDateRecd = %x06.80.03.00

; PidTagMessageFlags property ([MS-OXCMSG] section 2.2.1.6)
idMessageStatus = %x07.80.06.00

; PidTagSearchKey property ([MS-OXCPRPT] section 2.2.1.9)
idMessageID = %x09.80.01.00

; PidTagBody property ([MS-OXCMSG] section 2.2.1.56.1)
idBody = %x0C.80.02.00

; PidTagImportance property ([MS-OXCMSG] section 2.2.1.11)
idPriority = %x0D.80.04.00

; PidTagLastModificationTime property ([MS-OXCMSG] section 2.2.2.2)
; for the message
idDateModified = %x20.80.03.00

; Message Property Encapsulation
idMsgProps = %x03.90.06.00

; PidTagMessageRecipients propety ([MS-OXCMSG] section 2.2.1.47).
; For more details, see section 2.1.3.5.2.2.
idRecipTable = %x04.90.06.00

; PidTagOriginalMessageClass property ([MS-OXPROPS] section 2.819)
idOriginalMessageClass = %x00.06.07.00

; PidTagReceivedRepresenting_XXX or
PidTagSentRepresenting_XXX properties, as specified in section 2.1.3.1.1
idOwner = %x00.00.06.00

 ; PidTagSentRepresenting_XXX properties, as specified in section 2.1.3.1.1
idSentFor = %x01.00.06.00

; PidTagReceivedRepresentingEntryId property ([MS-OXOMSG] section 2.2.1.25)
idDelegate = %x02.00.06.00

; PidTagStartDate property ([MS-OXOCAL] section 2.2.1.30)
idDateStart = %x06.00.03.00

; PidTagEndDate property ([MS-OXOCAL] section 2.2.1.31)
idDateEnd = %x07.00.03.00

; PidTagOwnerAppointmentId property ([MS-OXOCAL] section 2.2.1.29)
idAidOwner = %x08.00.05.00

; PidTagResponseRequested property ([MS-OXOCAL] section 2.2.1.36)
idRequestRes = %x09.00.04.00

; Attachment-level attributes. All MUST be at attrLevelAttachment.
idAttachAttr = idAttachData / idAttachTitle / idAttachMetaFile /
idAttachCreateDate / idAttachModifyDate / idAttachTransportFilename

; PidTagAttachDataBinary property ([MS-OXCMSG] section 2.2.2.7)
idAttachData = %x0F.80.06.00

; PidTagAttachFilename property ([MS-OXCMSG] section 2.2.2.11)
idAttachTitle = %x10.80.01.00
```

```
; PidTagAttachRendering property ([MS-OXCMSG] section 2.2.2.17)
idAttachMetaFile = %x11.80.06.00

;  PidTagCreationTime property ([MS-OXCMSG] section 2.2.2.3)
idAttachCreateDate = %x12.80.03.00

; PidTagLastModificationTime property ([MS-OXCMSG] section 2.2.2.2)
; for the attachment.
 idAttachModifyDate = %x13.80.03.00

; PidTagAttachTransportName property ([MS-OXCMSG] section 2.2.2.19)
idAttachTransportFilename = %x01.90.06.00

; Attachment RendData, as specified in section 2.1.3.3.15
idAttachRendData = %x02.90.06.00

; Attachment table row
idAttachment = %x05.90.06.00
```

2.1.3.3 Attributes

This section specifies the attributes that appear in the TNEF stream, including the structure of the attributes, the message properties they map to, and any required conversions between them and message properties.

Each attribute is described by a level at which it applies (message or attachment), the attribute ID, the length of the attribute data, the attribute data, and a simple checksum. They are documented in the following sections in the form "**attXXXX**". For example, **attSubject** refers to the following:

attrLevelMessage idSubject Length 1*CHAR Checksum

2.1.3.3.1 attTnefVersion Attribute

For details about how the TNEF Writer handles this attribute, see section 2.2.3.1. For details about the TNEF Reader handles this attribute, see section 2.3.3.1.

2.1.3.3.2 attOemCodepage Attribute

The **attOemCodepage** attribute contains the code page used by the TNEF Writer for all attribute string values, and for any ANSI strings in the encapsulated message properties. For more information about code page handling, see section 2.1.3.5.1.

2.1.3.3.3 attFrom Attribute

The **attFrom** attribute is a message-level attribute that maps to and from the **PidTagSender_XXX** properties, as specified in section 2.1.3.1.1.

The data for this attribute encodes as follows:

```
TRP-structure = TRP-header sender-display-name sender-email empty
TRP-header = trpidOneOff structure-length sender-name-length sender-email-length
; Structure type
trpidOneOff = %x04.00

; The length of the entire structure. See the following description.
structure-length = UINT16

; Length of sender display name string, including the terminating null character.
sender-name-length = UINT16
```

```
; Length of sender email string, including the terminating null character.
sender-email-length = UINT16
sender-display-name = 1*OCTET %x00
sender-email = sender-email-type ":" sender-email-address %x00
sender-email-type = 1*CHAR
sender-email-address = 1*CHAR


;Empty 8-byte structure
empty = 8*%x00
```

The **structure-length** field is calculated as eight (the size of TRP-header in OCTETs) plus the length of sender-display-name (including the terminating null character), plus the length of sender-email (including the terminating null character), plus eight (the size of the empty eight bytes).

The **sender-name-length** field is the length of sender-display-name in OCTETs (including the terminating null character).

The **sender-email-length** field is calculated as the length of sender-email (including the terminating null character).

The **sender-email** field is composed of four parts, the address-type (for example, the literal sequence "SMTP" for Internet addresses), a literal colon (":"), the address itself, and a terminating null character. For example, the string "SMTP:user2@example.com" %x00 is a legal sender-email value.

The **empty** field is a zero-filled 8-byte structure.

For details about how the TNEF Writer handles the **sender-email** field, see section 2.2.3.3. For details about the TNEF Reader handles the **sender-email** field, see section 2.3.3.3.

2.1.3.3.4 Date Attributes

The date attributes, which are of type **PtypTime** ([MS-OXCDATA] section 2.11.1), are encoded into a **Date Time Record** structure, as follows. The encoding process used is implementation-dependent.

```
DTR = wYear wMonth wDay wHour wMinute wSecond wDayOfWeek
wYear = UINT16
wMonth = UINT16
wDay = UINT16
wHour = UINT16
wMinute = UINT16
wSecond = UINT16
wDayOfWeek = UINT16
```

The **wYear** field contains the year (for example, 2008 is %xD8.07); the **wMonth** field is 1 for January, and so on; The **WDay** field is 1 for the first day of the month; the **wHour**, **wMinute**, and **wSecond** fields contain the time; the **wDayOfWeek** field is 1 for Monday.

Attribute (in TNEF)	Level	Message property
attDateSent	Message	**PidTagClientSubmitTime** ([MS-OXOMSG] section 2.2.3.11)
attDateRecd	Message	**PidTagMessageDeliveryTime** ([MS-OXOMSG] section 2.2.3.9)
attAttachCreateDate	Attachment	**PidTagCreationTime** ([MS-OXCMSG] section 2.2.2.3)
attAttachModifyDate	Attachment	**PidTagLastModificationTime** ([MS-OXCMSG] section 2.2.2.2)
attDateModified	Message	**PidTagLastModificationTime**

Attribute (in TNEF)	Level	Message property
attDateStart	Message	**PidTagStartDate** ([MS-OXOCAL] section 2.2.1.30)
attDateEnd	Message	**PidTagEndDate** ([MS-OXOCAL] section 2.2.1.31)

2.1.3.3.5 Message Class Attributes

The message class attribute value is stored as a zero-terminated string that holds the name specified by the client.

Message class attributes in TNEF are as follows:

Attribute in TNEF	Level	Message Property
attMessageClass	Message	**PidTagMessageClass** ([MS-OXPROPS] section 2.776)
attOriginalMessageClass	Message	**PidTagOriginalMessageClass** ([MS-OXPROPS] section 2.819)

For details about how the TNEF Writer handles these attributes, see section 2.2.3.4. For details about the TNEF Reader handles this field, see section 2.3.3.4.

2.1.3.3.6 attMessageID Attribute

The **attMessageID** attribute is a message-level attribute that maps to the **PidTagSearchKey** property ([MS-OXCPRPT] section 2.2.1.9). The **PidTagSearchKey** property that maps to this attribute is stored in binary, but the attribute is represented in text format.

For details about how the TNEF Writer handles this attribute, see section 2.2.3.7. For details about how the TNEF Reader handles this attribute, see section 2.3.3.5.

2.1.3.3.7 attSubject Attribute

The **attSubject** attribute is a message-level attribute that maps to and from the **PidTagSubject** property ([MS-OXCMSG] section 2.2.1.46).

The data for this attribute is stored as a string with a terminating null character.

2.1.3.3.8 attMessageStatus Attribute

The **attMessageStatus** attribute is a message-level attribute that maps to and from the **PidTagMessageFlags** property ([MS-OXCMSG] section 2.2.1.6).

The data for this attribute is stored as an unsigned 32-bit integer, with the correct bit value set to indicate the message status. For compatibility, these MUST be mapped to and from the following values:

Status	Attribute Flag (in TNEF)	Bit Value	Message Property Flag	Bit Value
Read	**fmsRead**	0x20	**MSGFLAG_READ**	0x01
Unmodified	**fmsModified**	0x01	**MSGFLAG_UNMODIFIED**	0x02
Submitted	**fmsSubmitted**	0x04	**MSGFLAG_SUBMIT**	0x04

Status	Attribute Flag (in TNEF)	Bit Value	Message Property Flag	Bit Value
Unsent	**fmsLocal**	0x02	**MSGFLAG_UNSENT**	0x08
Has Attachments	**fmsHasAttach**	0x80	**MSGFLAG_HASATTACH**	0x10

If the **fmsModified** attribute flag is set, then the corresponding message property flag (**MSGFLAG_UNMODIFIED**) is clear (that is, not set). If the **fmsModified** attribute flag is clear, then the corresponding message property flag (**MSGFLAG_UNMODIFIED**) is set. For all other flags, either both the attribute flag and corresponding message property flag are set or both the attribute flag and corresponding message property flag are clear.

2.1.3.3.9 attBody Attribute

The **attBody** attribute MAY<1> be supported.

The **attBody** attribute is a message-level attribute that maps to and from the **PidTagBody** property ([MS-OXCMSG] section 2.2.1.56.1).

The data for this attribute is stored as a string with a terminating null character.

For more details about how the TNEF Writer handles this attribute, see section 2.2.3.6.

2.1.3.3.10 attPriority Attribute

The **attPriority** attribute is a message-level attribute that maps to and from the **PidTagImportance** property ([MS-OXCMSG] section 2.2.1.11).

The data for this attribute is stored as an unsigned 16-bit integer. For compatibility, these MUST be mapped to and from the following values:

Priority	Value (in TNEF)	Message Property Value
Low	3	0
Normal	2	1
High	1	2

2.1.3.3.11 attAttachData Attribute

The **attAttachData** attribute is an attachment-level attribute that maps to and from the **PidTagAttachDataBinary** property ([MS-OXCMSG] section 2.2.2.7).

The data for this attribute is stored as a binary stream.

For details about how the TNEF Writer handles this attribute, see section 2.2.3.8. For details about how the TNEF Reader handles this attribute, see section 2.3.3.7.

2.1.3.3.12 attAttachTitle Attribute

The **attAttachTitle** attribute is an attachment-level attribute that maps to and from the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10).

The data for this attribute is stored as a string with a terminating null character.

For details about how the TNEF Writer handles this attribute, see section 2.2.3.9. For details about how the TNEF Reader handles this attribute, see section 2.3.3.8.

2.1.3.3.13 attAttachMetaFile Attribute

The **attAttachMetaFile** attribute is an attachment-level attribute that maps to and from the **PidTagAttachRendering** property ([MS-OXCMSG] section 2.2.2.17).

The data for this attribute is stored as a binary stream. For a description of the content of the binary stream, see [MS-WMF].

2.1.3.3.14 attAttachTransportFilename Attribute

The **attAttachTransportFilename** attribute is an attachment-level attribute that maps to and from the **PidTagAttachTransportName** property ([MS-OXCMSG] section 2.2.2.19).

The data for this attribute is stored as a string with a terminating null character.

2.1.3.3.15 attAttachRendData Attribute

The **attAttachRendData** attribute is an attachment-level attribute that contains a structure of information that can be used for rendering the attachment in the body.

Each set of attachment attributes MUST begin with the **attAttachRendData** attribute, followed by any other attributes; attachment properties encoded in the **attAttachment** attribute SHOULD be last.

The structure is formatted as follows:

```
attAttachRendData = AttachType AttachPosition RenderWidth RenderHeight DataFlags
AttachType = AttachTypeFile / AttachTypeOle
AttachTypeFile=%x01.00
AttachTypeOle=%x02.00
AttachPosition= INT32
RenderWidth=INT16
RenderHeight=INT16
DataFlags = FileDataDefault / FileDataMacBinary
FileDataDefault= %x00.00.00.00
FileDataMacBinary=%x01.00.00.00
```

The **AttachPosition** field maps to and from the **PidTagRenderingPosition** property ([MS-OXCMSG] section 2.2.2.16).

For details about how the TNEF Writer handles this attribute, see section 2.2.3.10. For details about how the TNEF Reader handles this attribute, see section 2.3.3.9.

2.1.3.3.16 attOwner Attribute

The **attOwner** attribute is a message-level attribute that is also a meeting attribute that maps to and from the **PidTagReceivedRepresenting_XXX** or **PidTagSentRepresenting_XXX** properties, as specified in section 2.1.3.1.1.

The **attOwner** attribute is encoded as counted strings laid end-to-end. The format for **attOwner** attribute is as follows:

```
attOwner=display-name-length display-name address-length address

; Length of "display-name," including terminating null character.
display-name-length=UINT16
```

```
display-name=*CHAR %x00

; Length of "address", including terminating null character.
address-length=UINT16
address= 1*CHAR ":" 1*CHAR %x00
```

The display-name-length and address-length are unsigned 16-bit values, including the terminating null characters for the strings. The type and address strings in the email-address entry are separated by a colon (:) character; for example, "SMTP:user1@example.com" %x00.

The mapping of message properties to the **attOwner** attribute is dependent on the message class of the message being encoded.

For details about how the TNEF Writer handles this attribute, see section 2.2.3.11. For details about how the TNEF Reader handles this attribute, see section 2.3.3.10.

2.1.3.3.17 attSentFor Attribute

The **attSentFor** attribute is a message-level attribute that is also a meeting attribute that maps to and from the **PidTagSentRepresenting_XXX** properties, as specified in section 2.1.3.1.1.

For details about how the TNEF Writer handles this attribute, see section 2.2.3.12. For details about how the TNEF Reader handles this attribute, see section 2.3.3.11.

2.1.3.3.18 attDelegate Attribute

The **attDelegate** attribute MAY<2> be supported.

The **attDelegate** attribute is a message-level attribute that is a meeting attribute that maps to and from **PidTagReceivedRepresentingEntryId** property ([MS-OXOMSG] section 2.2.1.25).

The content can be transported as a **binary large object (BLOB)**.

For details about how the TNEF Writer handles this attribute, see section 2.2.3.13. For details about how the TNEF Reader handles this attribute, see section 2.3.3.12.

2.1.3.3.19 attAidOwner Attribute

The **attAidOwner** attribute is a message-level attribute that is a meeting attribute that maps to and from the **PidTagOwnerAppointmentId** property ([MS-OXOCAL] section 2.2.1.29).

The content can be transported as a BLOB.

2.1.3.3.20 attRequestRes Attribute

The **attRequestRes** attribute is a message-level attribute that is a meeting attribute that maps to and from the **PidTagResponseRequested** property ([MS-OXOCAL] section 2.2.1.36).

The data type of this value is a **BOOLEAN** ([MS-DTYP]); however, it SHOULD be transported as an **INT16** value ([MS-DTYP]). The 2-byte **INT16** value contains a null byte as the high-order 8 bits and the value of the **PidTagResponseRequested** property as the low-order 8 bits.

2.1.3.3.21 attMsgProps Attribute

The **attMsgProps** attribute is a message-level attribute that maps to and from an arbitrary set of message properties.

For a description of the encoding for this attribute, see section 2.1.3.4.

For details about how the TNEF Writer handles this attribute, see section 2.2.3.14.

2.1.3.3.22 attRecipTable Attribute

The **attRecipTable** attribute is a message-level attribute that maps to and from the **PidTagMessageRecipients** property ([MS-OXCMSG] section 2.2.1.47).

For a description of the encoding for this attribute, see section 2.1.3.4.

2.1.3.3.23 attAttachment Attribute

The **attAttachment** attribute is an attachment-level attribute that maps to and from an arbitrary set of properties for a single attachment.

For a description of the encoding for this attribute, see section 2.1.3.4.

For details about how the TNEF Writer handles this attribute, see section 2.2.3.15.

2.1.3.4 Encapsulated Message and Attachment Properties

The **attMsgProps** and **attAttachment** attributes, as specified in sections section 2.1.3.3.21 and section 2.1.3.3.23 respectively, are special in that they are used to encode any message or attachment property that does not have a counterpart in the set of existing TNEF-defined attributes. The attribute data is a counted set of message or attachment properties laid end-to-end. The format of this encoding, which allows for any set of message or attachment properties, is as follows.

```
; Attributes containing encapsulated properties:

; attMsgProps - Maps to and from an arbitrary set of message properties.
MsgPropsData = MsgPropertyList

; attRecipTable - Maps to and from PidTagMessageRecipients,
; as specified in [MS-OXCMSG] section 2.2.1.47.
; Number of table rows followed by the rows.
RecipTableData = NumRows *RecipRow

NumRows = UINT32
RecipRow = MsgPropertyList

; attAttachment – Maps to/from an arbitrary set of properties for a single attachment.
AttachPropsData = MsgPropertyList

MsgPropertyList = MsgPropertyCount *MsgPropertyValue
MsgPropertyCount = UINT32

MsgPropertyValue = MsgPropertyTag MsgPropertyData
MsgPropertyTag = MsgPropertyType MsgPropertyId [NamedPropSpec]

; This property MUST be present when MsgPropertyId is >= 0x8000 (the minimum value
; of property ID for named properties, as specified in
; [MS-OXPROPS] section 1.3.3).
NamedPropSpec = PropNameSpace PropIDType PropMap

; Contains a GUID, as specified in [MS-OXCDATA], to specify the namespace.
; The TNEF Writer obtains this value by using the RopGetNamesFromPropertyIDs ROP,
; as specified in [MS-OXCROPS] section 2.2.8.2.
PropNameSpace = GUID
PropIDType = IDTypeNumber / IDTypeString
PropMap = PropMapID / PropMapString
```

```
IDTypeNumber = %x00.00.00.00
IDTypeString = %x01.00.00.00

; Used if PropIDType is IDTypeNumber. Contains a number, as specified in
; [MS-OXPROPS] section 1.3.4.2, that is used to identify the property within the namespace.
PropMapID = UINT32

; Used if PropIDType is IDTypeString. Contains a length, then a
; UTF-16LE encoded Unicode string.
; The length is the length of the UTF-16LE encoded Unicode string
; (including the terminating 2-byte null character).
; Optional padding to UINT32 boundary.
; The TNEF Writer obtains this value by using the RopGetNamesFromPropertyIds
; ROP, as specified in [MS-OXCROPS] section 2.2.8.2.
PropMapString = UINT32 *UINT16 %x00.00 [PropMapPad]

; Padding for a UTF-16LE encoded Unicode string to achieve a
; multiple of 4 bytes in length.
; The TNEF Writer MUST use zero bytes and the TNEF Reader
; MUST permit non-zero bytes.
; SHOULD be either 0 or 2 bytes.
PropMapPad=*1UINT16

MsgPropertyType = TypeUnspecified / TypeNull / TypeInt16 / TypeInt32 /
 TypeFlt32 / TypeFlt64 / TypeCurrency / TypeAppTime / TypeError /
 TypeBoolean / TypeObject / TypeInt64 / TypeString8 / TypeUnicode /
 TypeSystime / TypeCLSID / TypeBinary / TypeMVInt16 / TypeMVInt32 /
 TypeMVFlt32 / TypeMVFlt64 / TypeMVCurrency / TypeMVAppTime /
 TypeMVSystime / TypeMVString8 / TypeMVBinary / TypeMVUnicode /
 TypeMVCLSID / TypeMVInt64

; An arbitrary value that corresponds to the property. For named properties, the value is >=
0x8000 (as specified in [MS-OXPROPS] section 1.3.3).
MsgPropertyId = UINT16

TypeUnspecified = %x00.00

TypeNull = %x01.00

; Signed 16-bit value = INT16.
TypeInt16 = %x02.00
TypeMVInt16 = %x02.10

; Signed 32-bit value = INT32.
TypeInt32 = %x03.00
TypeMVInt32 = %x03.10

; Signed 32-bit floating point= FLOAT.
TypeFlt32 = %x04.00
TypeMVFlt32 = %x04.10

; 64-bit floating point= DOUBLE.
TypeFlt64 = %x05.00
TypeMVFlt64 = %x05.10

; Signed 64-bit int = OLE CURRENCY type.
TypeCurrency = %x06.00
TypeMVCurrency = %x06.10

; Application time= OLE DATE type.
TypeAppTime = %x07.00
TypeMVAppTime = %x07.10

TypeError = %x0A.00

; 16-bit Boolean (non-zero = TRUE)
TypeBoolean = %x0B.00
```

```
; Embedded object on a property.
TypeObject = %x0D.00

; 8-byte signed integer= INT64.
TypeInt64 = %x14.00
TypeMVInt64 = %x14.10

; 8-bit character string with terminating null character.
TypeString8 = %x1E.00
TypeMVString8 = %x1E.10

; UTF-16LE or variant character string with terminating 2-byte null character.
TypeUnicode = %x1F.00
TypeMVUnicode = %x1F.10

; FILETIME (a PtypTime value, as specified in [MS-OXCDATA] section 2.11.1)
TypeSystime = %x40.00
TypeMVSystime = %x40.10

; OLE GUID
TypeCLSID = %x48.00
TypeMVCLSID = %x48.10

; Uninterpreted BLOB.
TypeBinary = %x02.01
TypeMVBinary = %x02.11

; MsgPropertyData varies by property type. Individual property values
; are formatted as specified in [MS-OXCDATA] section 2.11, with
; padding as specified in this document.
 MsgPropertyData = PropertyScalarContent / PropertyMultiScalarContent /
 PropertyMultiVariableContent

; Scalars – Types Int16, Int32, Flt32, Flt64, Currency, AppTime,
; Bool, Int64, Systime, CLSID
; The data for the particular property is written to the stream and if necessary,
; padded with bytes (which SHOULD be zero) to achieve a multiple of 4-bytes in length.
PropertyScalarContent = MsgPropertyContent [PropertyPad]

; PropertyPad – between 0 and 3 zero-filled bytes, added to the streamed
; property values to achieve 4-byte boundary. The TNEF Writer MUST use zero bytes
; and the TNEF Reader MUST permit non-zero bytes. These pad bytes MUST be counted
; in the checksum of the containing attribute.
PropertyPad=*3ZERO
ZERO= %x00

; Multi-value Scalars – Types MVInt16, MVInt32, MVFlt32, MVFlt64, MVCurrency,
; MVAppTime, MVInt64, MVSystime, MVCLSID
; The number of values for the property is written to the stream as a 4-byte
; value, then the data for each value is written to the stream and if need
; be, padded with bytes (which SHOULD be zero) to achieve a multiple of 4 bytes in length.
PropertyMultiScalarContent = PropertyContentCount *PropertyScalarContent
PropertyContentCount = UINT32

; Variable-length – Types Unicode, String8, Object, Binary.
; These are handled as a special case of Multi-Variable-Length with the number of values=1.

; Multi-Variable-length – Types MVUnicode, MVString8, MVBinary
; The number of values for the property is written to the stream as a 4-byte value. ; Then,
for each value, the size of the property is written to the stream as a
; 4-byte value, then the data for the property is written to the stream and, if
; necessary, padded with zero bytes to achieve a multiple of 4 bytes in length.

PropertyMultiVariableContent = MsgPropertyCount 1*PropertyVariableContent
MsgPropertyCount = UINT32

; The size of the property is written to the stream as a 4-byte value, then the data
```

```
; for the property is written to the stream and if necessary, padded with zero bytes
; to achieve a multiple of 4 bytes in length. The size includes the Interface
; Identifier at the beginning of the value stream for an object but does not include
; the padding bytes.
;
; If the property is of type PtypObject, as specified in
; [MS-OXCDATA] section 2.11.1,
; the value-size is followed by the Interface Identifier of the object, then the serialized
; stream of the data of the object. Only the Interface Identifier values
; %x0B.00.00.00.00.00.00.00.C0.00.00.00.00.00.00.46,
; %x0C.00.00.00.00.00.00.00.C0.00.00.00.00.00.00.46, and the Interface Identifiers of
; Message objects are supported. The size of the Interface Identifier is
; included in the calculation of value-size.
;
; If the object is an attached message (that is, it has a property type of
; PtypObject (as specified in [MS-OXCDATA] section 2.11.1) and an
; Interface Identifier %x07.03.02.00.00.00.00.00.C0.00.00.00.00.00.00.46,
; (that of a Message object), the value data is encoded as an embedded TNEF
; stream (that is, a complete TNEF stream as defined by this specification).

PropertyVariableContent = MsgPropertySize MsgPropertyContent [PropertyPad]
MsgPropertySize = UINT32
```

2.1.3.5 Other Compatibility Issues

The preceding sections addressed issues of compatibility between the original implementation of TNEF and the richer implementation that is now used. This section discusses some compatibility issues within the richer implementation.

2.1.3.5.1 attOemCodepage Attribute Handling

The **attOemCodepage** attribute, as specified in section 2.1.3.3.2, has been handled inconsistently between multiple implementations of TNEF. This can be particularly troublesome when forwarding messages with attached messages between systems that have different default code pages. To minimize problems caused by this inconsistent behavior, section 2.2.3.2 specifies the recommended way to handle string encoding and section 2.3.3.2 specifies the way to handle string decoding.

2.1.3.5.2 TNEF Encapsulation Versus Outer Wrapper Attributes

It is possible to encapsulate properties in the TNEF stream that represents data that is already conveyed as part of an outer wrapper such as MIME. Because in many cases the outer wrapper properties are subject to substantial validation during transport (reverse DNs lookups and other validation), they are considered more trustworthy than information inside an attached file that otherwise could overwrite correct information with incorrect information.

As a general rule, the TNEF Writer does not duplicate any data that is already being reliably transported outside the stream and the TNEF Reader does not override data from outside the stream with data from inside the stream. Individual implementations determine how this rule is applied. For more details, see [MS-OXCMAIL]. Some highlights of this behavior are specified in the following sections.

2.1.3.5.2.1 attBody Attribute Handling

When TNEF is being sent as a MIME body part, the **attBody** attribute is not written to TNEF or read from TNEF either by the client or server. The plain text message body is instead transmitted in a MIME body part. For more details about **attBody** attribute handling by the TNEF Writer, see section 2.2.3.6. For more details about MIME body parts, see [MS-OXCMAIL].

2.1.3.5.2.2 attRecipTable Attribute Handling

When TNEF is being sent as a MIME body part, the **attRecipTable** attribute is not written to TNEF or read from TNEF either by the client or server, except when sending or receiving a legacy **non-delivery report** message (the value of the **attMessageClass** attribute begins with "REPORT." and ends in ".DR" or ".NDR"). The requisite message recipient information is transmitted in the MIME structure instead.

When TNEF is being sent as a MIME body part, when sending or receiving a legacy non-delivery report message (the value of the **attMessageClass** attribute begins with "REPORT." and ends in ".DR" or ".NDR"), recipient information from the **attRecipTable** attribute is used to populate the received message recipient table.

For more details about **attRecipTable** attribute handling by the TNEF Writer, see section 2.2.3.16.

2.1.3.5.2.3 attFrom Attribute Handling

The **attFrom** attribute is not written to TNEF or read from TNEF either by the client or the server for MIME messages. The requisite message recipient information is transmitted in the MIME stream instead. For more details about how the TNEF Writer encodes sender information, see section 2.2.3.3.

2.1.3.5.2.4 attSubject Attribute Handling

The **attSubject** attribute, if present for a MIME message, is overwritten by the subject value in the MIME stream. For details about how the TNEF Writer handles this attribute, see section 2.2.3.5. For details about how the TNEF Reader handles this attribute, see section 2.3.3.6.

2.2 TNEF Writer Algorithm Details

The TNEF Writer encodes or builds a TNEF stream for the purpose of transporting rich properties. This section specifies details specific to the TNEF Writer role.

For details common to both the TNEF Reader and TNEF Writer roles, see section 2.1.

2.2.1 Abstract Data Model

None.

2.2.2 Initialization

None.

2.2.3 Processing Rules

This section specifies the processing rules applied by the TNEF Writer to specific TNEF attributes. For processing rules common to the TNEF Writer and the TNEF Reader, see section 2.1.3.

2.2.3.1 attTnefVersion Attribute Handling by the TNEF Writer

The **attTnefVersion** attribute, as specified in section 2.1.3.3.1 MUST be written as %x00.00.01.00 by the TNEF Writer.

2.2.3.2 attOemCodepage Attribute Handling by TNEF Writer

The TNEF Writer writes all ANSI strings, both the attribute strings and the encapsulated message property strings, with a consistent ANSI code page and puts its value into the **attOemCodepage** property, as specified in section 2.1.3.3.2. Choose a code page that does not allow embedded zero bytes other than the zero terminator.

If **PidTagInternetCodepage** ([MS-OXCMSG] section 2.2.1.56.6) is being stored in the encapsulated properties, set it to the same value as the **attOemCodepage** attribute. This code page also determines the character set (1) used in MIME if this TNEF is transmitted as a MIME entity body.

Do not write to an attribute any string that would suffer data loss if written using the consistent code page. Write to encapsulated properties with a Unicode data type all strings that would suffer data loss when using the consistent code page.

For compatibility purposes, the **attMessageClass** attribute MUST be written using the consistent code page.

2.2.3.3 attFrom Attribute Handling by the TNEF Writer

When encoding the **sender-email** field of the **attFrom** attribute, as specified in section 2.1.3.3.3, the TNEF Writer can use the **PidTagSenderName** ([MS-OXOMSG] section 2.2.1.51), **PidTagSenderAddressType** ([MS-OXOMSG] section 2.2.1.48), and **PidTagSenderEmailAddress** ([MS-OXOMSG] section 2.2.1.49) properties if present, or access their values using the **PidTagSenderEntryId** property ([MS-OXOMSG] section 2.2.1.50).

When TNEF is being used to encode an attached message, the sender information is carried in encapsulated message properties in the **attFrom** attribute.

2.2.3.4 Message Class Attribute Handling by the TNEF Writer

The TNEF Writer can represent any message class, as specified in section 2.1.3.3.5, whose value is shown in the Message Property Value column of the following table, using the value in the Attribute value column.

Attribute value (in TNEF)	Message property value
IPM.Microsoft Mail.Note	IPM.Note
IPM.Microsoft Mail.Read Receipt	Report.IPM.Note.IPNRN
IPM.Microsoft Mail.Non-Delivery	Report.IPM.Note.NDR
IPM.Microsoft Schedule.MtgRespP	IPM.Schedule.Meeting.Resp.Pos
IPM.Microsoft Schedule.MtgRespN	IPM.Schedule.Meeting.Resp.Neg
IPM.Microsoft Schedule.MtgRespA	IPM.Schedule.Meeting.Resp.Tent
IPM.Microsoft Schedule.MtgReq	IPM.Schedule.Meeting.Request
IPM.Microsoft Schedule.MtgCncl	IPM.Schedule.Meeting.Canceled

Values not included in the table SHOULD be written in their original form.

2.2.3.5 attSubject Attribute Handling by the TNEF Writer

When TNEF is being used to encode an attached message, the message subject information is carried in encapsulated message properties in the **attMsgProps** attribute, not the **attSubject** attribute.

2.2.3.6 attBody Attribute Handling by the TNEF Writer

When TNEF is being used to encode an attached message, the **attBody** attribute, as specified in section 2.1.3.3.9, MAY<3> be written and read.

2.2.3.7 attMessageID Attribute Handling by the TNEF Writer

For compatibility, when generating a TNEF stream, the TNEF Writer MUST translate the binary representation of the **attMessageID** attribute, as specified in section 2.1.3.3.6, into a textual one by emitting the two CHAR hexadecimal equivalent per OCTET (i.e. %x01 becomes "01", %x2D becomes "2D", and so on).

2.2.3.8 attAttachData Attribute Handling by the TNEF Writer

The TNEF Writer can use the **attAttachData** attribute, as specified in section 2.1.3.3.11, for the attachment data if the **PidTagAttachMethod** property ([MS-OXCMSG] section 2.2.2.9) of the original attachment contains the value 0x0001 (ATTACH_BY_VALUE). If the **PidTagAttachMethod** of the original attachment contains the value 0x0005 (ATTACH_EMBEDDED_MSG) or the value 0x0006 (ATTACH_OLE), then the TNEF Writer MUST encode the data in **attAttachment** (section 2.1.3.4).

2.2.3.9 attAttachTitle Attribute Handling by the TNEF Writer

The TNEF Writer obtains the value of the **attAttachTitle** attribute, as specified in section 2.1.3.3.12, from the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10) if it is available; otherwise, the TNEF Writer obtains the value of the **attAttachTitle** attribute from the **PidTagAttachFilename** property.

2.2.3.10 attAttachRendData Attribute Handling by the TNEF Writer

The **AttachType** field of the **attAttachRendData** attribute, as specified in section 2.1.3.3.15, MUST be set by the TNEF Writer to the value of the **AttachTypeFile** attribute when the **PidTagAttachMethod** property ([MS-OXCMSG] section 2.2.2.9) for the attachment is set to ATTACH_BY_VALUE or ATTACH_EMBEDDED_MSG, and to the value of the **AttachTypeOle** attribute when the **PidTagAttachMethod** property is set to ATTACH_OLE.

The **RenderWidth** and **RenderHeight** fields of the **attAttachRendData** attribute SHOULD be set by the TNEF Writer to the size of the system icons if the attachment is a simple file attachment, or to {-1,-1} if the attachment is not a simple file attachment.

The **DataFlags** field SHOULD<4> be set to **FileDataDefault** (%x00.00.00.00), regardless of the attachment's **Error! Hyperlink reference not valid.** property ([MS-OXCMSG] section 2.2.2.20) value. However, if the **PidTagAttachEncoding** property ([MS-OXCMSG] section 2.2.2.20) for the attachment consists of bytes %x2A.86.48.86.F7.14.03.0B.01, then the **FileDataMacBinary** bit MAY<5> be set by the TNEF Writer; otherwise the **FileDataDefault** field SHOULD be set by the TNEF Writer.

2.2.3.11 attOwner Attribute Handling by the TNEF Writer

If the message is either a meeting request or meeting cancellation (from the organizer, who is creating or deleting a meeting), the TNEF Writer SHOULD encode the

PidTagSentRepresenting_XXX properties, as specified in section 2.1.3.1.1, in the **attOwner** attribute, as specified in section 2.1.3.3.16.

If the message is a meeting response of any type (from attendee, accepting, declining, and so on), the TNEF Writer SHOULD encode the **PidTagReceivedRepresenting_XXX** properties, as specified in section 2.1.3.1.1, in the **attOwner** attribute. The encoding used is implementation-dependent. For details about how to construct the **EntryId** property, see [MS-OXOMSG] section 2.2.1.25.

The TNEF Writer can use the discrete **PidTag{Sent | Rcvd}Representing Name**, **AddressType**, and **EmailAddress** properties if present, or access their values by using the **PidTag{Sent | Rcvd}EntryId** property.

2.2.3.12 attSentFor Attribute Handling by the TNEF Writer

The TNEF Writer SHOULD encode the **PidTagSentRepresenting_XXX** properties, as specified in section 2.1.3.1.1, in the **attSentFor** attribute, as specified in section 2.1.3.3.17.

The TNEF Writer can use the discrete **PidTagSentRepresentingName** ([MS-OXOMSG] section 2.2.1.57), **PidTagSentRepresentingAddressType** ([MS-OXOMSG] section 2.2.1.54), and **PidTagSentRepresentingEmailAddress** ([MS-OXOMSG] section 2.2.1.55) properties if present, or access their values by using the **PidTagSentRepresentingEntryId** property ([MS-OXOMSG] section 2.2.1.56).

2.2.3.13 attDelegate Attribute Handling by the TNEF Writer

The TNEF Writer SHOULD encode the **PidTagReceivedRepresentingEntryId** properties, as specified in [MS-OXOMSG] section 2.2.1.25, in the **attDelegate** attribute, as specified in section 2.1.3.3.18.

2.2.3.14 attMsgProps Attribute Handling by the TNEF Writer

The TNEF Writer SHOULD write the **attMsgProps** attribute, as specified in section 2.1.3.3.21, after all the other message attributes and immediately before all the attachment attributes. For a description of the encoding for this attribute, see section 2.1.3.4.

2.2.3.15 attAttachment Attribute Handling by the TNEF Writer

The TNEF Writer SHOULD write the encapsulated **attAttachment** attribute, as specified in section 2.1.3.3.23, after all the attributes for an attachment.

2.2.3.16 attRecipTable Attribute Handling by the TNEF Writer

When TNEF is being used to encode an attached message, recipient information from the **attRecipTable** attribute, as specified in section 2.1.3.3.22, is used to populate the received message recipient table.

2.3 TNEF Reader Algorithm Details

The TNEF Reader decodes a TNEF stream after receiving a message, for the purpose of reconstructing the rich properties that are contained in the stream. This section specifies details specific to the TNEF Reader role.

For details common to both the TNEF Reader and TNEF Writer roles, see section 2.1.

2.3.1 Abstract Data Model

None.

2.3.2 Initialization

None.

2.3.3 Processing Rules

This section specifies the processing rules applied by the TNEF Reader to specific TNEF attributes. For processing rules common to the TNEF Writer and the TNEF Reader, see section 2.1.3.

2.3.3.1 attTnefVersion Attribute Handling by the TNEF Reader

The TNEF Reader MUST reject values other than %x00.00.01.00 for the **attTnefVersion** attribute, as specified in section 2.1.3.3.1.

2.3.3.2 attOemCodepage Attribute Handling by the TNEF Reader

When performing ANSI string decoding from the TNEF stream, the TNEF Reader uses the first code page determined from the following sources:

1. Any non-US-ASCII MIME character set (2), if the stream was transmitted as a MIME body part and the character set (2) name is available in MIME, mapped to the most appropriate code page that does not allow embedded zero bytes other than a zero terminator.
2. The **attOemCodepage** attribute, as specified in section 2.1.3.3.2, if available and nonzero.
3. The **PidTagInternetCodepage** property ([MS-OXCMSG] section 2.2.1.56.6), if available in the encapsulated properties, mapped to a code page that does not allow embedded zero bytes other than a null terminator.
4. The most appropriate default code page for the processing environment.

2.3.3.3 attFrom Attribute Handling by the TNEF Reader

When reading the **sender-email** field of the **attFrom** attribute, as specified in section 2.1.3.3.3, the TNEF Reader MUST set the **PidTagSenderEntryId** property ([MS-OXOMSG] section 2.2.1.50) and SHOULD decode the **attFrom** attribute into the other **PidTagSender_XXX** properties, as specified in section 2.1.3.1.1.

2.3.3.4 attMessageClass Attribute Handling by the TNEF Reader

If the **attMessageClass** or **attOriginalMessageClass** attribute, as specified in section 2.1.3.3.5, contains a value that is shown in the Attribute value column of the following table, the value that MUST be returned is that of the value in the Message property value column.

Attribute value (in TNEF)	Message property value
IPM.Microsoft Mail.Note	IPM.Note
IPM.Microsoft Mail.Read Receipt	Report.IPM.Note.IPNRN
IPM.Microsoft Mail.Non-Delivery	Report.IPM.Note.NDR

Attribute value (in TNEF)	Message property value
IPM.Microsoft Schedule.MtgRespP	IPM.Schedule.Meeting.Resp.Pos
IPM.Microsoft Schedule.MtgRespN	IPM.Schedule.Meeting.Resp.Neg
IPM.Microsoft Schedule.MtgRespA	IPM.Schedule.Meeting.Resp.Tent
IPM.Microsoft Schedule.MtgReq	IPM.Schedule.Meeting.Request
IPM.Microsoft Schedule.MtgCncl	IPM.Schedule.Meeting.Canceled

Values not included in the table SHOULD be read in their original form.

If the value of the **attMessageClass** or **attOriginalMessageClass** attribute begins with the string "Microsoft Mail v3.0 ", the TNEF Reader MUST ignore the "Microsoft Mail v3.0 " prefix when attempting to match the value of the **attMessageClass** or **attOriginalMessageClass** attribute to a value in the Attribute Value column. For example, the TNEF Reader considers the value of the **attMessageClass** or **attOriginalMessageClass** attribute Microsoft Mail v3.0 IPM.Microsoft.Mail.Note to be the same as IPM.Microsoft.Mail.Note, which corresponds to the Message Property Value IPM.Note.

The TNEF Reader SHOULD ignore the checksum value for message class attributes because some legacy TNEF Writers generated non-valid checksum values.

2.3.3.5 attMessageID Attribute Handling by the TNEF Reader

When reading the **attMessageID** attribute, as specified in section 2.1.3.3.6, from TNEF, the TNEF Reader SHOULD attempt to decode the text format back into binary format.

2.3.3.6 attSubject Attribute Handling by the TNEF Reader

When the TNEF stream is being decoded, if the **attSubject** attribute, as specified in section 2.1.3.3.7, is present, it is used for the message if there is no MIME structure, if a MIME subject is not available, or if an attached message is being decoded.

2.3.3.7 attAttachData Attribute Handling by the TNEF Reader

If the **PidTagAttachMethod** property ([MS-OXCMSG] section 2.2.2.9) of the original attachment contains the value 0x0005 (ATTACH_EMBEDDED_MSG) or the value 0x0006 (ATTACH_OLE), then the TNEF Reader SHOULD ignore the **attAttachData** attribute, as specified in section 2.1.3.3.11.

2.3.3.8 attAttachTitle Attribute Handling by the TNEF Reader

The TNEF Reader MUST set the value of the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10), which maps to the **attAttachTitle** attribute, as specified in section 2.1.3.3.12.

2.3.3.9 attAttachRendData Attribute Handling by the TNEF Reader

The TNEF Reader SHOULD set the **PidTagAttachTag** property ([MS-OXCMSG] section 2.2.2.15) for the attachment to %x2A.86.48.86.F7.14.03.0A.03.02.01 when the **AttachTypeOle** field of the **attAttachRendData** attribute, as specified in section 2.1.3.3.15, is set. The TNEF Reader SHOULD NOT set the **PidTagAttachTag** property for the attachment when **AttachTypeFile** field of the **attAttachRendData** attribute is set.

The TNEF Reader SHOULD set the value of the **PidTagAttachEncoding** property ([MS-OXCMSG] section 2.2.2.20) for the attachment to the same bytes when the **DataFlags** field of the

attAttachRendData attribute has the **FileDataMacBinary** bit set. If the **DataFlags** field contains the value of FileDataDefault, then the TNEF Reader SHOULD NOT set the value of the **PidTagAttachEncoding** property for the attachment.

2.3.3.10 attOwner Attribute Handling by the TNEF Reader

If the message is either a meeting request or meeting cancellation (from the organizer, who is creating or deleting a meeting), the TNEF Reader SHOULD decode the **attOwner** attribute, as specified in section 2.1.3.3.16, into the **PidTagSentRepresenting_XXX** properties, as specified in section 2.1.3.1.1.

If the message is a meeting response of any type (from attendee, accepting, declining, and so on), the TNEF Reader SHOULD decode the **attOwner** attribute into the **PidTagReceivedRepresenting_XXX** properties, as specified in section 2.1.3.1.1. The decoding process used is implementation-dependent, as long as the decoded property values match those that were originally encoded. For details about how to construct the **PidTagReceivedRepresentingEntryId** property, see [MS-OXOMSG] section 2.2.1.25.

The TNEF Reader MUST set the **PidTag{Sent | Rcvd} EntryId** property and SHOULD decode the **attOwner** attribute into the other **PidTag{Sent | Rcvd} Representing** properties.

2.3.3.11 attSentFor Attribute Handling by the TNEF Reader

The TNEF Reader SHOULD decode the **attSentFor** attribute, as specified in section 2.1.3.3.17, into the **PidTagSentRepresenting_XXX** properties, as specified in section 2.1.3.1.1.

The TNEF Reader SHOULD set the **PidTagSentRepresentingEntryId** property ([MS-OXOMSG] section 2.2.1.56) and SHOULD decode the **attOwner** attribute, as specified in section 2.1.3.3.16, into the other **PidTagSentRepresenting_XXX** properties.

2.3.3.12 attDelegate Attribute Handling by the TNEF Reader

The TNEF Reader SHOULD decode the **attDelegate** attribute, as specified in section 2.1.3.3.18, into the **PidTagReceivedRepresentingEntryId** properties ([MS-OXOMSG] section 2.2.1.25).

3 Algorithm Examples

3.1 Sample Message

Explanation of content	Byte stream
TNEF Signature, Value=0x223E9F78	78 9f 3e 22
Attach Key, Value=0x0001	01 00
Message attribute, **attTnefVersion** Length=4 bytes	01 06 90 08 00 04 00 00 00
Value=0x00010000	00 00 01 00
Checksum=0x0001	01 00
Message attribute, **attOemCodepage** Length=8 bytes	01 07 90 06 00 08 00 00 00
Value= (Primary=1252, Secondary=0)	e4 04 00 00 00 00 00 00
Checksum=0x00E8	e8 00
Message attribute, **attMessageClass** Length=24 bytes	01 08 80 07 00 18 00 00 00
Value="IPM.Microsoft Mail.Note" %x00	49 50 4d 2e 4d 69 63 72 6f 73 6f 66 74 20 4d 61 69 6c 2e 4e 6f 74 65 00
Checksum=0x0831	31 08
Message attribute, **attPriority** Length=2 bytes	01 0d 80 04 00

Explanation of content	Byte stream
	02 00 00 00
Value=0x0002 (Normal Priority)	02 00
Checksum=0x0002	02 00
Message attribute, **attSubject** Length=15 bytes	01 04 80 01 00 0f 00 00 00
Value="Simple subject" %x00	53 69 6d 70 6c 65 20 73 75 62 6a 65 63 74 00
Checksum=0x057A	7a 05
Message attribute, **attDateSent** Length=14 bytes	01 05 80 03 00 0e 00 00 00
Value= Tuesday, 02/17/2004, 11:25:35	d4 07 02 00 11 00 0b 00 19 00 23 00 02 00
Checksum=0x0137	37 01
Message attribute, **attDateModified** Length=14 bytes	01 20 80 03 00 0e 00 00 00
Value= Tuesday, 02/17/2004, 12:26:09	D4 07 02 00 11 00 0C 00 -- 1A 00 09 00 02 00
Checksum=0x011F	1F 01
Message attribute, **attMessageID** Length=33 bytes	01 09 80 01 00

Explanation of content	Byte stream
	21 00 00 00
Value= "757BB19CDE936A4087D90BB784C58E3B"	37 35 37 42 42 31 39 43 44 45 39 33 36 41 34 30 38 37 44 39 30 42 42 37 38 34 43 35 38 45 33 42 00
Checksum=0x0751	51 07
Message attribute, Encapsulation, **attMsgProps** Length=2256 bytes Number of properties=70	01 03 90 06 00 d0 08 00 00 46 00 00 00
Message property, **PidTagAlternateRecipientAllowed** ([MS-OXCMSG] section 2.2.1.36) Value=1 (TRUE)	0b 00 02 00 01 00 00 00
Message property, **PidTagPriority** ([MS-OXCMSG] section 2.2.1.12) Value=0 (LOW)	03 00 26 00 00 00 00 00
Message property, **PidTagSensitivity** ([MS-OXCMSG] section 2.2.1.13) Value=0 (LOW)	03 00 36 00 00 00 00 00
Message property, **PidTagClientSubmitTime** ([MS-OXOMSG] section 2.2.3.11) Value=Tuesday, 02/17/2004, 19:25:35	40 00 39 00 AA D6 F3 D0 8B F5 C3 01
Message property, **PidTagSubjectPrefix** ([MS-OXOMSG] section 2.2.1.60) Number of property values=1 Size of first property=1	1e 00 3d 00 01 00 00 00 01 00 00 00
Value = %x00	00
Pad=3 bytes	00 00 00
Message property, **PidTagReceivedByEntryId** ([MS-OXOMSG] section 2.2.1.38) Number of property values=1	02 01 3f 00 01 00 00 00

Explanation of content	Byte stream
Size of first property=107	6b 00 00 00
Value	00 00 00 00 DC A7 40 C8 C0 42 10 1A B4 B9 08 00 2B 2F E1 82 01 00 00 00 00 00 00 2F 4F 3D 46 49 52 53 54 20 4F 52 47 41 4E 49 5A 41 54 49 4F 4E 2F 4F 55 3D 46 49 52 53 54 20 41 44 4D 49 4E 49 53 54 52 41 54 49 56 45 20 47 52 4F 55 50 2F 43 4E 3D 52 45 43 49 50 49 45 4E 54 53 2F 43 4E 3D 54 45 53 54 32 31 55 57 32 00
Pad=1 byte	00
Message property, **PidTagReceivedByName** ([MS-OXOMSG] section 2.2.1.39) Number of property values=1 Size of first property=10	1e 00 40 00 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00
Message property, **PidTagReceivedRepresentingEntryId** ([MS-OXOMSG] section 2.2.1.25) Number of property values=1 Size of first property=107	02 01 43 00 01 00 00 00 6b 00 00 00
Value	00 00 00 00 DC A7 40 C8 C0 42 10 1A B4 B9 08 00 2B 2F E1 82 01 00 00 00 00 00 00 00 2F 4F 3D 46 49 52 53 54 20 4F 52 47 41 4E 49 5A 41 54 49 4F 4E 2F 4F 55 3D 46 49 52 53 54 20 41 44 4D 49 4E 49 53 54 52 41 54 49 56 45 20 47 52 4F 55 50 2F 43 4E 3D 52 45 43 49 50 49 45 4E 54 53 2F 43 4E 3D 54 45 53 54 32 31 55 57 32 00
Pad=1 byte	00

Explanation of content	Byte stream
Message property, **PidTagReceivedRepresentingName** ([MS-OXOMSG] section 2.2.1.26) Number of property values=1 Size of first property=10	1e 00 44 00 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00
Message property, **PidTagMessageSubmissionId** ([MS-OXOMSG] section 2.2.1.79) Number of property values=1 Size of first property=57	02 01 47 00 01 00 00 00 39 00 00 00
Value	63 3D 55 53 3B 61 3D 20 3B 70 3D 46 69 72 73 74 20 4F 72 67 61 6E 69 7A 61 74 69 3B 6C 3D 4A 45 53 45 4F 47 50 55 57 32 2D 30 34 30 32 31 37 31 39 32 35 33 35 5A 2D 32 00
Pad=3 bytes	00 00 00
Message property, **PidTagReceivedBySearchKey** ([MS-OXOMSG] section 2.2.1.40) Number of property values=1 Size of first property=82	02 01 51 00 01 00 00 00 52 00 00 00
Value	45 58 3A 2F 4F 3D 46 49 52 53 54 20 4F 52 47 41 4E 49 5A 41 54 49 4F 4E 2F 4F 55 3D 46 49 52 53 54 20 41 44 4D 49 4E 49 53 54 52 41 54 49 56 45 20 47 52 4F 55 50 2F 43 4E 3D 52 45 43 49 50 49 45 4E 54 53 2F 43 4E 3D 54 45 53 54 32 31 55 57 32 00
Pad=2 bytes	00 00
Message property, **PidTagReceivedRepresentingSearchKey** ([MS-OXOMSG] section 2.2.1.27) Number of property values=1 Size of first property=82	02 01 52 00 01 00 00 00 52 00 00 00
Value	45 58 3A 2F 4F 3D 46 49 52 53 54 20 4F 52 47 41 4E 49 5A 41 54 49 4F 4E 2F 4F

Explanation of content	Byte stream
	55 3D 46 49 52 53 54 20 41 44 4D 49 4E 49 53 54 52 41 54 49 56 45 20 47 52 4F 55 50 2F 43 4E 3D 52 45 43 49 50 49 45 4E 54 53 2F 43 4E 3D 54 45 53 54 32 31 55 57 32 00
Pad=2 bytes	00 00
Message property, **PidTagMessageToMe** ([MS-OXOMSG] section 2.2.1.17) Value=1 (TRUE)	0b 00 00 57 01 00 00 00
Message property, **PidTagMessageCcMe** ([MS-OXOMSG] section 2.2.1.18) Value=0 (FALSE)	0b 00 00 58 00 00 00 00
Message property **PidTagMessageRecipientMe** ([MS-OXOMSG] section 2.2.1.19) Value=1 (TRUE)	0b 00 00 59 01 00 00 00
Message property, **PidTagConversationTopic** ([MS-OXOMSG] section 2.2.1.5) Number of property values=1 Size of first property=15	1e 00 70 00 01 00 00 00 0f 00 00 00
Value="Simple subject" %x00	53 69 6D 70 6C 65 20 73 75 62 6A 65 63 74 00
Pad=1 byte	00
Message property, **PidTagConversationIndex** ([MS-OXOMSG] section 2.2.1.3) Number of property values=1 Size of first property=22	02 01 71 00 01 00 00 00 16 00 00 00
Value	01 C3 F5 8B 07 63 76 8D 89 1B D0 79 47 C9 8F 66 4E 21 9A D2 4A F2
Pad=2 bytes	00 00
Message property, **PidTagReceivedByAddressType** ([MS-OXOMSG] section 2.2.1.36) Number of property values=1 Size of first property=5	1e 00 75 00 01 00 00 00 05 00 00 00

Explanation of content	Byte stream
Value="SMTP" %x00	53 4d 54 50 00
Pad=3 bytes	00 00 00
Message property, **PidTagReceivedByEmailAddress** ([MS-OXOMSG] section 2.2.1.37) Number of property values=1 Size of first property=40	1e 00 76 00 01 00 00 00 28 00 00 00
Value="Test21uw2@mydomuw2.extest.microsoft.com" %x00	54 65 73 74 32 31 75 77 32 40 6D 79 64 6F 6D 75 77 32 2E 65 78 74 65 73 74 2E 6D 69 63 72 6F 73 6F 66 74 2E 63 6F 6D 00
Message property, **PidTagReceivedRepresentingAddressType** ([MS-OXOMSG] section 2.2.1.23) Number of property values=1 Size of first property=5	1e 00 77 00 01 00 00 00 05 00 00 00
Value="SMTP" %x00	53 4d 54 50 00
Pad=3 bytes	00 00 00
Message property, **PidTagReceivedRepresentingEmailAddress** ([MS-OXOMSG] section 2.2.1.24) Number of property values=1 Size of first property=40	1e 00 78 00 01 00 00 00 28 00 00 00
Value="Test21uw2@mydomuw2.extest.microsoft.com" %x00	54 65 73 74 32 31 75 77 32 40 6D 79 64 6F 6D 75 77 32 2E 65 78 74 65 73 74 2E 6D 69 63 72 6F 73 6F 66 74 2E 63 6F 6D 00
Message property, **PidTagSenderName** ([MS-OXOMSG] section 2.2.1.51) Number of property values=1 Size of first property=10	1e 00 1A 0C 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00

Explanation of content	Byte stream
Pad=2 bytes	00 00
Message property, **PidTagNormalizedSubject** ([MS-OXCMSG] section 2.2.1.10) Number of property values=1 Size of first property=15	1e 00 1D 0E 01 00 00 00 0f 00 00 00
Value="Simple subject" %x00	53 69 6D 70 6C 65 20 73 75 62 6A 65 63 74 00
Pad=1 byte	00
Message property, **PidTagRtfCompressed** ([MS-OXCMSG] section 2.2.1.56.4) Number of property values=1 Size of first property=150	02 01 09 10 01 00 00 00 96 00 00 00
Value	92 00 00 00 AB 00 00 00 4C 5A 46 75 A8 14 2F 17 03 00 0A 00 72 63 70 67 31 32 35 16 32 00 F8 0B 60 6E 0E 10 30 33 33 4F 01 F7 02 A4 03 E3 02 00 63 68 0A C0 73 B0 65 74 30 20 07 13 02 80 7D 0A 80 9D 00 00 2A 09 B0 09 F0 04 90 61 74 05 B1 1A 52 0D E0 68 09 80 01 D0 20 35 2E 30 35 30 2E 33 13 B0 01 D0 30 32 49 02 80 5C 76 08 90 77 6B 0B 80 64 FA 34 0C 60 63 00 50 0B 03 0B B5 06 00 07 70 C9 0B 50 65 20 07 81 73 61 12 50 0A A2 0B 0A 80 11 E1 00 17 A0
Pad=2 bytes	00 00
Message property, **PidTagInternetMessageId** ([MS-OXOMSG] section 2.2.1.12) Number of property values=1 Size of first property=80	1e 00 35 10 01 00 00 00 50 00 00 00
Value="<2896107D7E52DF4DB5D10536D BFEFAD07E37@jeseogpuw2.mydomuw2. extest.microsoft.com>" %x00	3C 32 38 39 36 31 30 37 44 37 45 35 32 44 46 34 44 42 35 44 31 30 35 33 36 44 42 46 45 46 41 44 30 37 45 33 37 40 6A 65 73 65 6F 67 70 75 77 32 2E 6D 79 64 6F 6D 75 77 32 2E 65 78 74 65 73 74 2E 6D 69 63 72 6F 73 6F 66 74 2E 63 6F 6D 3E 00

Explanation of content	Byte stream
Message property, **PidTagIconIndex** ([MS-OXOMSG] section 2.2.1.10) Value=0xFFFFFFFF	03 00 80 10 ff ff ff ff
Unspecified **PtypBinary** property Number of property values=1 Size of first property=40	02 01 f0 10 01 00 00 00 28 00 00 00
Value	03 00 00 00 01 00 00 00 01 00 00 00 01 00 00 00 01 00 00 00 08 00 00 00 DB 11 00 00 49 53 4F 2D 38 38 35 39 2D 31 00 00
Unspecified property Number of property values=1 Size of first property=38	1f 00 f3 10 01 00 00 00 26 00 00 00
Value="Simple subject.EML" %x00.00 (Unicode)	53 00 69 00 6D 00 70 00 6C 00 65 00 20 00 73 00 75 00 62 00 6A 00 65 00 63 00 74 00 2E 00 45 00 4D 00 4C 00 00 00
Pad=2 bytes	00 00
Message property, **PidTagAttributeHidden** ([MS-OXCFOLD] section 2.2.2.2.2.1) Value=0 (FALSE)	0b 00 f4 10 00 00 00 00
Unspecified **PtypBoolean** ([MS-OXCDATA] section 2.11.1) property Value=0 (FALSE)	0b 00 f5 10 00 00 00 00
Message property, **PidTagAttributeReadOnly** ([MS-OXPROPS] section 2.603) Value=0 (FALSE)	0b 00 f6 10 00 00 00 00
Message property, **PidTagCreationTime** ([MS-OXCMSG] section 2.2.2.3) Value=Tuesday, 02/17/2004, 20:26:09	40 00 07 30 90 24 46 47 94 f5 c3 01
Message property, **PidTagLastModificationTime** ([MS-OXCMSG] section 2.2.2.2) Value=Tuesday, 02/17/2004, 20:26:09	40 00 08 30 90 24 46 47 94 f5 c3 01
Message property, **PidTagInternetCodepage** ([MS-OXCMSG] section 2.2.1.56.6) Value=(Primary=1252, Secondary=0)	03 00 de 3f e4 04 00 00
Message property, **PidTagMessageLocaleId** ([MS-OXCMSG] section 2.2.1.5) Value=(Primary=1033, Secondary=0)	03 00 f1 3f 09 04 00 00

Explanation of content	Byte stream
Message property, **PidTagCreatorName** ([MS-OXPROPS] section 2.647) Number of property values=1 Size of first property=10	1e 00 f8 3f 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00
Message property, **PidTagCreatorEntryId** ([MS-OXCMSG] section 2.2.1.31) Number of property values=1 Size of first property=107	02 01 F9 3F 01 00 00 00 6b 00 00 00
Value	00 00 00 00 DC A7 40 C8 C0 42 10 1A B4 B9 08 00 2B 2F E1 82 01 00 00 00 00 00 00 00 2F 4F 3D 46 49 52 53 54 20 4F 52 47 41 4E 49 5A 41 54 49 4F 4E 2F 4F 55 3D 46 49 52 53 54 20 41 44 4D 49 4E 49 53 54 52 41 54 49 56 45 20 47 52 4F 55 50 2F 43 4E 3D 52 45 43 49 50 49 45 4E 54 53 2F 43 4E 3D 54 45 53 54 32 31 55 57 32 00
Pad=1 byte	00
Message property, **PidTagLastModifierName** ([MS-OXCPRPT] section 2.2.1.5) Number of property values=1 Size of first property=10	1e 00 fa 3f 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00
Message property, **PidTagLastModifierEntryId** ([MS-OXCMSG] section 2.2.1.32) Number of property values=1 Size of first property=107	02 01 FB 3F 01 00 00 006b 00 00 00
Value	00 00 00 00 DC A7 40 C8 C0 42 10 1A B4 B9 08 00 2B 2F E1 82 01 00 00 00 00 00 00 00 2F 4F 3D 46 49 52 53 54 20 4F 52 47 41 4E 49 5A 41 54 49 4F 4E 2F 4F 55 3D 46 49 52 53 54 20 41 44 4D 49 4E 49

Explanation of content	Byte stream
	53 54 52 41 54 49 56 45 20 47 52 4F 55 50 2F 43 4E 3D 52 45 43 49 50 49 45 4E 54 53 2F 43 4E 3D 54 45 53 54 32 31 55 57 32 00
Pad=1 byte	00
Message property, **PidTagMessageCodepage** ([MS-OXCMSG] section 2.2.1.4) Value=(Primary=1252, Secondary=0)	03 00 fd 3f e4 04 00 00
Unspecified message property Value=0	03 00 19 40 00 00 00 00
Message property, **PidTagSentRepresentingFlags** ([MS-OXPROPS] section 2.1004) Value=0	03 00 1A 40 00 00 00 00
Unspecified message property Value=0	03 00 1B 40 00 00 00 00
Unspecified message property, Value=0	03 00 1C 40 00 00 00 00
Unspecified **PtypString** ([MS-OXCDATA] section 2.11.1) property Number of property values=1 Size of first property=10	1e 00 30 40 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00
Unspecified **PtypString** property Number of property values=1 Size of first property=10	1e 00 31 40 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00

Explanation of content	Byte stream
Unspecified **PtypString** property Number of property values=1 Size of first property=10	1e 00 34 40 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00
Unspecified **PtypString** property Number of property values=1 Size of first property=10	1e 00 35 40 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00
Unspecified **PtypString** property Number of property values=1 Size of first property=10	1e 00 38 40 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00
Unspecified **PtypString** property Number of property values=1 Size of first property=10	1e 00 39 40 01 00 00 00 0a 00 00 00
Value="Test21uw2" %x00	54 65 73 74 32 31 75 77 32 00
Pad=2 bytes	00 00
Message property, **PidTagContentFilterSpamConfidenceLevel** ([MS-OXCSPAM] section 2.2.1.3) Value=0xFFFFFFFF	03 00 76 40 ff ff ff ff

Explanation of content	Byte stream
Message property, **PidTagMessageEditorFormat** ([MS-OXOMSG] section 2.2.1.78) Value=0x00000003	03 00 09 59 03 00 00 00
Message property, **PidTagTnefCorrelationKey** ([MS-OXCMSG] section 2.2.1.29) Number of property values=1 Size of first property=12	02 01 7f 00 01 00 00 00 0c 00 00 00
Value	38 71 6b 6a 30 30 73 67 6d 34 66 00
Message property, **PidLidAgingDontAgeMe** ([MS-OXCMSG] section 2.2.1.33) PSETID_Common, MNID_ID, Id=0x850E Value=0	0b 00 87 81 08 20 06 00 00 00 00 00 c0 00 00 00 00 00 00 46 00 00 00 00 0e 85 00 00 00 00 00 00
Message property, **PidLidCurrentVersion** ([MS-OXCMSG] section 2.2.1.34) PSETID_Common, MNID_ID, Id=0x8552 Value=115608	03 00 9f 81 08 20 06 00 00 00 00 00 c0 00 00 00 00 00 00 46 00 00 00 00 52 85 00 00 98 c3 01 00
Message property, **PidLidCurrentVersionName** ([MS-OXCMSG] section 2.2.1.35) PSETID_Common, MNID_ID, Id=0x8554 Number of property values=1 Size of first property=5	1E 00 A0 81 08 20 06 00 00 00 00 00 c0 00 00 00 00 00 00 46 00 00 00 00 54 85 00 00 01 00 00 00 05 00 00 00
Value="11.0" %x00	31 31 2E 30 00
Pad=3 bytes	00 00 00
Message property, **PidLidReminderDelta** ([MS-OXORMDR] section 2.2.1.3) PSETID_Common, MNID_ID, Id=0x8501 Value=0	03 00 eb 81 08 20 06 00 00 00 00 00 c0 00 00 00 00 00 00 46 00 00 00 00 01 85 00 00 00 00 00 00
Message property, **PidLidReminderSet** ([MS-OXORMDR] section 2.2.1.1) PSETID_Common, MNID_ID, Id=0x8503 Value=0 (FALSE)	0b 00 f0 81 08 20 06 00 00 00 00 00 c0 00 00 00 00 00 00 46 00 00 00 00 03 85 00 00 00 00 00 00
Message property, **PidLidSideEffects** ([MS-OXCMSG] section 2.2.1.16)	03 00 fa 81 08 20 06 00 00 00 00 00 c0 00 00 00 00 00 00 46 00 00 00 00 10 85

Explanation of content	Byte stream
PSETID_Common, MNID_ID, Id=0x8510 Value=0	00 00 00 00 00 00
Message property, **PidLidTaskMode** ([MS-OXOTASK] section 2.2.2.2.1) PSETID_Common, MNID_ID, Id=0x8518 Value=0	03 00 01 82 08 20 06 00 00 00 00 00 c0 00 00 00 00 00 00 46 00 00 00 00 18 85 00 00 00 00 00 00
Message property, **PidLidPrivate** ([MS-OXCMSG] section 2.2.1.15) PSETID_Common, MNID_ID, Id=0x8506 Value=0 (FALSE)	0b 00 43 82 08 20 06 00 00 00 00 00 c0 00 00 00 00 00 00 46 00 00 00 00 06 85 00 00 00 00 00 00
Message property, **PidLidUseTnef** ([MS-OXOMSG] section 2.2.1.66) PSETID_Common, MNID_ID, Id=0x8582 Value=0 (FALSE)	0b 00 44 82 08 20 06 00 00 00 00 00 c0 00 00 00 00 00 00 46 00 00 00 00 82 85 00 00 00 00 00 00
Message property, **PidTagReadReceiptRequested** ([MS-OXOMSG] section 2.2.1.29) Value=0 (FALSE)	0b 00 29 00 00 00 00 00
Message property, **PidTagOriginatorDeliveryReportRequested** ([MS-OXOMSG] section 2.2.1.20) Value=0 (FALSE)	0b 00 23 00 00 00 00 00
Unspecified **PtypInteger32** ([MS-OXCDATA] section 2.11.1) property Value=0xF00D29FF	03 00 06 10 ff 29 0d f0
Unspecified **PtypInteger32** property Value=13	03 00 07 10 0d 00 00 00
Unspecified **PtypInteger32** property Value=0	03 00 10 10 00 00 00 00
Unspecified **PtypInteger32** property Value=0	03 00 11 10 00 00 00 00
Unspecified **PtypString** property Number of property values=1 Size of first property=14	1e 00 08 10 01 00 00 00 0e 00 00 00
Value="SIMPLEMESSAGE" %x00	53 49 4d 50 4c 45 4d 45 53 53 41 47 45 00

Explanation of content	Byte stream
Pad=2 bytes	00 00
Message property, **PidTagTnefCorrelationKey** ([MS-OXCMSG] section 2.2.1.29) Number of property values=1 Size of first property=56	02 01 7f 00 01 00 00 00 38 00 00 00
Value	3C 32 38 39 36 31 30 37 44 37 45 35 32 44 46 34 44 42 35 44 31 30 35 33 36 44 42 46 45 46 41 44 30 37 45 33 37 40 75 73 65 72 2E 65 78 61 6D 70 6C 65 2E 63 6F 6D 3E 00
Checksum for **AttMsgProps**	45 C1

3.2 Sample Meeting Response

This example is particularly short, as it was sent by a mobile application.

Explanation of content	Byte stream
TNEF Signature, Value=0x223E9F78	78 9f 3e 22
Attach Key, Value=0x0001	01 00
Message attribute, **attTnefVersion** Length=4 bytes	01 06 90 08 00 04 00 00 00
Value=0x00010000	00 00 01 00
Checksum=0x0001	01 00
Message attribute, **attOemCodepage** Length=8 bytes	01 07 90 06 00 08 00 00 00

Explanation of content	Byte stream
Value= (Primary=1252, Secondary=0)	e4 04 00 00 00 00 00 00
Checksum=0x00E8	e8 00
Message attribute, **attMessageClass** Length=32 bytes	01 08 80 07 00 20 00 00 00
Value="IPM.Microsoft Schedule.MtgRespN" %x00	49 50 4d 2e 4d 69 63 72 6f 73 6f 66 74 20 53 63 68 65 64 75 6c 65 2e 4d 74 67 52 65 73 70 4e 00
Checksum=0x0B55	55 0b
Message attribute, **attPriority** Length=2 bytes	01 0d 80 04 00 02 00 00 00
Value=0x0002 (Normal Priority)	02 00
Checksum=0x0002	02 00
Message attribute, **attDateSent** Length=14 bytes	01 05 80 03 00 0e 00 00 00
Value= Wednesday, 01/16/2008, 23:28:08	d8 07 01 00 10 00 17 00 1c 00 08 00 03 00
Checksum=0x012E	2e 01
Message attribute, **attDateModified** Length=14 bytes	01 20 80 03 000e 00 00 00
Value= Wednesday, 01/16/2008, 23:28:08	d8 07 01 00 10 00 17 00 1c 00 08 00 03 00
Checksum=0x012E	2e 01

Explanation of content	Byte stream
Message attribute Encapsulation, **attMsgProps** Length=136 bytes Number of properties=2	01 03 90 06 00 88 00 00 00 02 00 00 00
Message property, **PidTagTnefCorrelationKey** ([MS-OXCMSG] section 2.2.1.29) Number of property values=1 Size of first property=12	02 01 7f 00 01 00 00 00 0c 00 00 00
Value	38 71 6b 6a 30 30 73 67 6d 34 66 00
Message property, **PidTagRtfCompressed** ([MS-OXCMSG] section 2.2.1.56.4) Number of property values=1 Size of first property=93	02 01 09 10 01 00 00 00 5d 00 00 00
Value	59 00 00 00 b3 00 00 00 4c 5a 46 75 a9 be bb ed 87 00 0a 01 0d 03 43 74 65 78 74 01 f7 ff 02 a4 03 e4 05 eb 02 83 00 50 02 f3 06 b4 02 83 26 32 03 c5 02 00 63 68 0a c0 73 65 d8 74 30 20 07 13 02 80 7d 0a 80 08 cf 3f 09 d9 02 80 0a 84 0b 37 12 c2 01 d0 20 46 10 59 49 00 7d 18 20
Pad=3 bytes	00 00 00
Checksum for **attMsgProps**	f7 21

4 Security

4.1 Security Considerations for Implementers

Extreme care is recommended in the use of the **attRecipTable** attribute and the **attFrom** attribute, particularly by the TNEF Reader, to avoid address or identity spoofing. More information about this can be found in [MS-OXCMSG] and section 2.1.3.5.2 of this specification.

4.2 Index of Security Parameters

None.

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 2.1.3.3.9: Exchange 2003 supports the use of the **attBody** attribute.

<2> Section 2.1.3.3.18: Exchange 2003 supports the use of the **attDelegate** attribute.

<3> Section 2.2.3.6: In Exchange 2003, the TNEF Writer writes the **attBody** attribute when TNEF is used to encode an attached message.

<4> Section 2.2.3.10: In Exchange 2003, the TNEF Writer does not set the **DataFlags** field to **FileDataDefault**.

<5> Section 2.2.3.10: In Exchange 2003, the TNEF Writer sets the **FileDataMacBinary** bit if the **PidTagAttachEncoding** property ([MS-OXCMSG] section 2.2.2.20) for the attachment consists of bytes %x2A.86.48.86.F7.14.03.0B.01.

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
5 Appendix A: Product Behavior	Added Exchange 2016 and Outlook 2016 to the list of applicable products.	Y	Content update.

7 Index