[MS-OXODOC]:
Document Object Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Revised and edited technical content.
09/03/2008	1.02		Updated references.
12/03/2008	1.03		Updated IP notice.
03/04/2009	1.04		Revised and edited technical content.
04/10/2009	2.0		Updated technical content and applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	4.1.0	Minor	Updated the technical content.
05/05/2010	4.1.1	Editorial	Revised and edited the technical content.
08/04/2010	4.2	Minor	Clarified the meaning of the technical content.
11/03/2010	4.3	Minor	Clarified the meaning of the technical content.
03/18/2011	4.3	No change	No changes to the meaning, language, and formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Document Object protocol, which is an extension to the Message and Attachment Object Protocol. The Document Object protocol defines properties that allow a messaging client to store and display ordinary files. For more information about the Message and Attachment Object Protocol, see [MS-OXCMSG].

1.1 Glossary

The following terms are defined in [MS-GLOS]:

handle

The following terms are defined in [MS-OXGLOS]:

Attachment object
message class
Message object
remote operation (ROP)

The following terms are specific to this document:

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

1.2 References

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. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-OSHARED] Microsoft Corporation, "Office Common Data Types and Objects Structure Specification", June 2008.

[MS-OXCDATA] Microsoft Corporation, "Data Structures", April 2008.

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol Specification", June 2008.

[MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol Specification", June 2008.

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List", April 2008.

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

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-OXCROPS] Microsoft Corporation, "Remote Operations (ROP) List and Encoding Protocol Specification", June 2008.

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", April 2008.

[MS-OXOMSG] Microsoft Corporation, "E-Mail Object Protocol Specification", June 2008.

1.3 Overview

The Document Object protocol extends the Message and Attachment Object Protocol by defining the document object, which is a **Message object** with additional properties. For more information about the Message and Attachment Object Protocol, see [MS-OXCMSG]. For more information about Message objects, see [MS-OXOMSG].

The Document object is used to represent a file, such as a document generated by a word-processing application. The file is embedded within the Document object; the embedded file is referred to as an attachment. Specifically, the Document object is a Message object that:

▪ Contains one file.

▪ Includes additional properties to describe the file.

A file represented by a Document object can be stored in a mail folder and retrieved from the mail folder. For example, a user might choose to store a few files in his or her mail folders so that the files can be accessed not just on one computer, but on any computer that provides access to the user's e-mail.

1.4 Relationship to Other Protocols

The Document Object protocol relies on the same protocols as the Message and Attachment Object Protocol, which the Document Object protocol extends. For more information about the Message and Attachment object protocol, see [MS-OXCMSG].

1.5 Prerequisites/Preconditions

The Document Object protocol assumes that the messaging client has a mail folder open. Before sending requests to the server, the client obtains a **handle** to the Message object that is used in property operations.

1.6 Applicability Statement

The client can utilize this protocol to store ordinary files in a user's mail folders and to expose the files that are stored in the mail folders.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

This protocol provides no extensibility beyond what is already described in [MS-OXCMSG].

1.9 Standards Assignments

None.

2   Messages

2.1   Transport

The Document Object protocol uses the same underlying transport as the Message and Attachment Object Protocol, as specified in [MS-OXCMSG].

2.2   Message Syntax

A Document object can be created and modified by both clients and servers. This section defines the constraints under which both clients and servers operate.

Clients operate on a Document object by using the Message and Attachment Object protocol, as specified in [MS-OXCMSG], and by using the Property and Stream Object protocol, as specified in [MS-OXCPRPT]. The manner in which servers operate on a Document object is implementation-dependent, but the results of any such operations MUST be exposed to clients as specified by the Document Object protocol.

Unless otherwise stated in sections 2.2.1 and 2.2.2, a Document object MUST adhere to all property constraints specified in both [MS-OXPROPS] and [MS-OXCMSG]. In addition to the properties specified in sections 2.2.1 and 2.2.2, the client can set the following properties on a Document object,<1> but these properties do not have any impact on the Document Object protocol: **PidLidAgingDontAgeMe** ([MS-OXPROPS] section 2.4), **PidLidCurrentVersion** ([MS-OXPROPS] section 2.83), **PidLidCurrentVersionName** ([MS-OXPROPS] section 2.84), **PidLidPrivate** ([MS-OXPROPS] section 2.208), **PidLidSideEffects** ([MS-OXPROPS] section 2.297), **PidTagAlternateRecipientAllowed** ([MS-OXPROPS] section 2.641), **PidTagClientSubmitTime** ([MS-OXPROPS] section 2.701), **PidTagDeleteAfterSubmit** ([MS-OXPROPS] section 2.740), **PidTagImportance** ([MS-OXPROPS] section 2.821), **PidTagMessageDeliveryTime** ([MS-OXPROPS] section 2.886), **PidTagPriority** ([MS-OXPROPS] section 2.965), **PidTagReadReceiptRequested** ([MS-OXPROPS] section 2.980), **PidTagSensitivity** ([MS-OXPROPS] section 2.1115), **PidLidReminderDelta** ([MS-OXPROPS] section 2.216), **PidLidReminderSet** ([MS-OXPROPS] section 2.220), **PidLidReminderSignalTime** ([MS-OXPROPS] section 2.221), **PidLidTaskMode** ([MS-OXPROPS] section 2.322), **PidLidPercentComplete** ([MS-OXPROPS] section 2.199), **PidLidTaskActualEffort** ([MS-OXPROPS] section 2.305), **PidLidTaskComplete** ([MS-OXPROPS] section 2.308), **PidLidTaskAssigner** ([MS-OXPROPS] section 2.306), **PidLidTaskAcceptanceState** ([MS-OXPROPS] section 2.303), **PidLidTaskEstimatedEffort** ([MS-OXPROPS] section 2.313), **PidLidTaskFFixOffline** ([MS-OXPROPS] section 2.315), **PidLidTaskFRecurring** ([MS-OXPROPS] section 2.316), **PidLidTaskNoCompute** ([MS-OXPROPS] section 2.324), **PidLidTaskOrdinal** ([MS-OXPROPS] section 2.325), **PidLidTaskOwnership** ([MS-OXPROPS] section 2.327), **PidLidTaskRole** ([MS-OXPROPS] section 2.330), **PidLidTaskState** ([MS-OXPROPS] section 2.332), **PidLidTaskStatus** ([MS-OXPROPS] section 2.333), **PidLidTaskVersion** ([MS-OXPROPS] section 2.336), **PidLidTeamTask** ([MS-OXPROPS] section 2.337), and **PidLidValidFlagStringProof** ([MS-OXPROPS] section 2.347).

The property data types are specified in [MS-OXCDATA] section 2.11.1.

2.2.1   Document-Specific Properties

A Document object encapsulates the behavior of the attached file. As such, many properties on a file SHOULD be promoted as properties on the message itself if those properties exist on the file. The following is a list of such properties that SHOULD be set on the message if they exist on the attached file. For more details about these properties, see [MS-OXPROPS].

2.2.1.1   PidNameTitle

Type: **PtypString**

Specifies the title of the file attached to the Document object. This property corresponds to the **GKPIDSI_TITLE** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.2   PidNameSubject

Type: **PtypString**

Specifies the subject of the file attached to the Document object. This property corresponds to the **GKPIDSI_SUBJECT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.3   PidNameAuthor

Type: **PtypString**

Represents the author of the file attached to the Document object. This property corresponds to the **GKPIDSI_AUTHOR** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.4   PidNameKeywords

Type: **PtypMultipleString**

Specifies the categories of the file attached to the Document object. This property corresponds to the **GKPIDSI_KEYWORDS** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.5   PidNameComments

Type: **PtypString**

Specifies the comments of the file attached to the Document object. This property corresponds to the **GKPIDSI_COMMENTS** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.6   PidNameTemplate

Type: **PtypString**

Specifies the template of the file attached to the Document object. This property corresponds to the **GKPIDSI_TEMPLATE** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.7   PidNameLastAuthor

Type: **PtypString**

Specifies the most recent author of the file attached to the Document object. This property corresponds to the **GKPIDSI_LASTAUTHOR** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.8   PidNameRevisionNumber

Type: **PtypString**

Specifies the revision number of the file attached to the Document object. This property corresponds to the **GKPIDSI_REVNUMBER** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.9   PidNameApplicationName

Type: **PtypString**

Specifies the application that might open the file attached to the document object. This property corresponds to the **GKPIDSI_APPNAME** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.10   PidNameEditTime

Type: **PtypString**

Specifies the time that the file was last edited. This property corresponds to the **GKPIDSI_EDITTIME** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.11   PidNameLastPrinted

Type: **PtypTime**

Specifies the time that the file was last printed. This property corresponds to the **GKPIDSI_LASTPRINTED** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.12   PidNameCreateDateTimeReadOnly

Type: **PtypTime**

Specifies the time that the file was first created. This property corresponds to the **GKPIDSI_CREATE_DTM** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.13   PidNameLastSaveDateTime

Type: **PtypTime**

Specifies the time that the file was last saved. This property corresponds to the **GKPIDSI_LASTSAVE_DTM** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.14   PidNamePageCount

Type: **PtypInteger32**

Specifies the page count of the file attached to the Document object. This property corresponds to the **GKPIDSI_PAGECOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.15   PidNameWordCount

Type: **PtypInteger32**

Specifies the number of words in the file attached to the Document object. This property corresponds to the **GKPIDSI_WORDCOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.16   PidNameCharacterCount

Type: **PtypInteger32**

Specifies the number of characters in the file attached to the Document object. This property corresponds to the **GKPIDSI_CHARCOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.17   PidNameSecurity

Type: **PtypInteger32**

Specifies the security level of the file attached to the Document object. This property corresponds to the **GKPIDSI_DOC_SECURITY** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.18   PidNameCategory

Type: **PtypString**

Specifies the category of the file attached to the Document object. This property corresponds to the **GKPIDDSI_CATEGORY** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.19   PidNamePresentationFormat

Type: **PtypString**

Specifies the presentation format of the file attached to the Document object. This property corresponds to the **GKPIDDSI_PRESFORMAT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.20   PidNameManager

Type: **PtypString**

Specifies the manager of the file attached to the Document object. This property corresponds to the **GKPIDDSI_MANAGER** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.21   PidNameCompany

Type: **PtypString**

Specifies the company for which the file was created. This property corresponds to the **GKPIDDSI_COMPANY** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.22   PidNameByteCount

Type: **PtypInteger32**

Specifies the size, in bytes, of the file attached to the Document object. This property corresponds to the **GKPIDDSI_BYTECOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.23   PidNameLineCount

Type: **PtypInteger32**

Specifies the number of lines in the file attached to the Document object. This property corresponds to the **GKPIDDSI_LINECOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.24 PidNameParagraphCount

Type: **PtypInteger32**

Specifies the number of paragraphs in the file attached to the Document object. This property corresponds to the **GKPIDDSI_PARACOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.25 PidNameSlideCount

Type: **PtypInteger32**

Specifies the number of slides in the file attached to the Document object. This property corresponds to the **GKPIDDSI_SLIDECOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.26 PidNameNoteCount

Type: **PtypInteger32**

Specifies the number of notes in the file attached to the Document object. This property corresponds to the **GKPIDDSI_NOTECOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.27 PidNameHiddenCount

Type: **PtypInteger32**

Specifies the hidden value of the file attached to the Document object. This property corresponds to the **GKPIDDSI_HIDDENCOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.28 PidNameMultimediaClipCount

Type: **PtypInteger32**

Specifies the number of multimedia clips in the file attached to the Document object. This property corresponds to the **GKPIDDSI_MMCLIPCOUNT** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.29 PidNameDocumentParts

Type: **PtypMultipleString**

Specifies the title of each part of the document. This property corresponds to the **GKPIDDSI_DOCPARTS** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.30 PidNameHeadingPairs

Type: **PtypBinary**

Specifies which group of headings are indented in the document. This property corresponds to the **GKPIDDSI_HEADINGPAIR** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.31 PidNameLinksDirty

Type: **PtypBoolean**

Indicates whether the links in the document are up-to-date. The value TRUE indicates that the links are up-to-date; FALSE indicates otherwise. This property corresponds to the **GKPIDDSI_LINKSDIRTY** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.32 PidNameScale

Type: **PtypBoolean**

Indicates whether the image is to be scaled or cropped. The value TRUE indicates thumbnail scaling; FALSE indicates cropping. This property corresponds to the **GKPIDDSI_SCALE** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.1.33 PidNameThumbnail

Type: **PtypBinary**

Specifies the data representing the thumbnail image of the document. This property corresponds to the **GKPIDSI_THUMBNAIL** property, which is specified in [MS-OSHARED] section 2.3.3.2.1.1.

2.2.2 Additional Property Constraints

The following sections specify additional property constraints beyond what is specified in [MS-OXCMSG].

2.2.2.1 PidTagMessageClass Property

Type: **PtypString**

Specifies the type of the Message object. For a message to be treated like a Document object by a messaging client, the value of this property MUST begin with "IPM.document.". The value of the sub-string that follows "IPM.document." depends on the type of the attached file. For more details about the **PidTagMessageClass** property, see [MS-OXCMSG] section 2.2.1.3.

2.2.2.2 PidTagDisplayName Property

Type: **PtypString**

Specifies the name of the attachment. A Document object SHOULD have the **PidTagDisplayName** ([MS-OXPROPS] section 2.746) property defined.

2.2.2.3 Attachment to the Message Object

A Document object SHOULD have only one attachment and MUST have at least one attachment. For more details about how attachments are stored within a message, see [MS-OXCMSG]. If there is more than one attachment, then a messaging client can choose to display an error or to pick one of the attachments to use as the main file. However, a Document object only makes sense if there is only one attachment. If there are zero attachments, then the messaging client SHOULD cause an error when the item is invoked.

3   Protocol Details

3.1   Client Details

3.1.1   Abstract Data Model

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

3.1.1.1   Managing Document Objects

A messaging client user can choose to create a Document object either programmatically or as a result of user interaction. Choosing either of these routes will result in the creation of a message that has certain properties that make the message a Document object. For instance, a user could drag a file from his or her desktop into a folder in the messaging client. The end result of this user interaction is a Document object in that folder. How a user interacts with this Document object is entirely up to the messaging client. One behavior can be as follows: When a user invokes (double-clicks) this message, the file is directly opened instead of first opening the message and then necessitating another click on the attachment.

3.1.2   Timers

None.

3.1.3   Initialization

A Document object is initialized or created either programmatically or as a result of user interaction. In either case, a Document object is created with the correct attachment and properties being set on the message. For more details about the properties, see section 2.2.

3.1.4   Higher-Layer Triggered Events

3.1.4.1   Creating a Document Object

The messaging client creates a Document object when the user drags a file from the desktop (or from any file folder) into a mail folder. The Document object SHOULD have one attachment and SHOULD have the **PidTagMessageClass** ([MS-OXPROPS] section 2.884) and **PidTagDisplayName** ([MS-OXPROPS] section 2.746) properties correctly set. The value of the **PidTagMessageClass** property MUST be "IPM.document.<FileType>", where the "<FileType>" sub-string indicates the type of the attached file. The method that the messaging client uses to determine the "<FileType>" sub-string is implementation-dependent.

For details about the **PidTagMessageClass** and **PidTagDisplayName** properties, see sections 2.2.2.1 and 2.2.2.2, respectively. For details about the **ROPs** involved in creating a Document object, see [MS-OXCMSG] section 3.1.4.2.

3.1.4.2   Invoking a Document Object

When a message is invoked, the messaging client first needs to determine the message type. This involves examining the **PidTagMessageClass** ([MS-OXPROPS] section 2.884) property. For a Document object, the value of **PidTagMessageClass** is "IPM.document.<FileType>", where the

"IPM.document." sub-string indicates that the message is a Document object and the "<FileType>" sub-string identifies the type of the attached file. The method that the client uses to interpret the "<FileType>" sub-string is implementation-dependent.

If the value of **PidTagMessageClass** does not begin with "IPM.document.", then the message is not a Document object, and the messaging client will handle it in a different way. If the value of **PidTagMessageClass** does begin with "IPM.document.", then the message is a Document object, and the messaging client will proceed to retrieve the attachment from the message's attachment collection and open that attachment. A Document object SHOULD have only one attachment. When a Document object is invoked, the client can open the message's underlying attachment directly, thereby behaving in the most optimal fashion from a user's perspective.

For details about the **PidTagMessageClass** property, see section 2.2.2.1. For details about the ROPs involved in invoking a Document object, see [MS-OXCMSG] section 3.1.4.1.

3.1.4.3   Other Tasks

Other tasks related to a Document Object include the following:

- Creating an attachment — see [MS-OXCMSG] section 3.1.4.12

- Opening an attachment — see [MS-OXCMSG] section 3.1.4.11

- Setting content in an attachment — see [MS-OXCMSG] section 3.1.4.13

- Saving an attachment — see [MS-OXCMSG] section 3.1.4.14

- Setting properties on a Document object — see [MS-OXCPRPT] section 3.1.4.2

- Saving a Document object — see [MS-OXCMSG] section 3.1.4.3

3.1.5   Message Processing Events and Sequencing Rules

None.

3.1.6   Timer Events

None.

3.1.7   Other Local Events

None.

3.2   Server Details

The server role for this protocol is as specified in [MS-OXCMSG] and [MS-OXCPRPT].

4   Protocol Examples

4.1   Example PidTagMessageClass Values for Different File Types

The following table shows example **message class** values for different file types.

File extension	PidTagMessageClass value
.doc	IPM.document.Word.document.8
.docx	IPM.document.Word.document.12
.xls	IPM.document.Excel.Sheet.8
.xlsx	IPM.document.Excel.Sheet.12
.ppt	IPM.document.PowerPoint.Show.8
.pptx	IPM.document.PowerPoint.Show.12
.txt	IPM.document.txtfile

4.2   Example for Creating a New Document Object Item

Joe drags a file (for example, testDocObj.txt) from his desktop into one of his mail folders. The following sub-sections provide descriptions of what a client might do to accomplish Joe's intentions, and the responses that a server might return.

4.2.1   Creating the Object

To create a Document object, the protocol client uses the **RopCreateMessage** ([MS-OXCROPS] section 2.2.6.2) operation. The protocol server returns a success code and a handle to a Message object.

4.2.2   Attachment Details

The client uses the **RopCreateAttachment** ([MS-OXCROPS] section 2.2.6.13) operation to create the **Attachment object**. Then, the protocol client uses the **RopOpenStream** ([MS-OXCROPS] section 2.2.9.1) and **RopSetStreamSize** ([MS-OXCROPS] section 2.2.9.6) operations followed by **RopWriteStream** ([MS-OXCROPS] section 2.2.9.3) to write out the contents of the file into the attachment.

The client then uses **RopSetProperties** ([MS-OXCROPS] section 2.2.8.6) to set various properties on the attachment. The following table shows just some of the properties that would be set on the attachment.

Property	Property ID	Type	Value
PidTagAttachLongFilename ([MS-OXPROPS] section 2.659)	0x3707	0x001f (string)	"testDocObj.txt"
PidTagAttachExtension ([MS-OXPROPS] section 2.656)	0x3703	0x001f (string)	".txt"
PidTagCreationTime ([MS-OXPROPS] section	0x3007	0x0040 (date	2008/02/15

Property	Property ID	Type	Value
2.725)		and time)	19:57:52.557

Now the protocol client uses **RopSaveChangesAttachment** ([MS-OXCROPS] section 2.2.6.15) to save the attachment.

4.2.3   Setting Properties on the Document Object

The protocol client uses the **RopSetProperties** ([MS-OXCROPS] section 2.2.8.6) operation to transmit his data to the protocol server. The following table shows some of the relevant properties that need to be set for a Document object.

Property	Property ID	Type	Value
PidTagDisplayName ([MS-OXPROPS] section 2.746)	0x3001	0x001f (PT_UNICODE)	"testDocObj.txt"
PidTagMessageClass ([MS-OXPROPS] section 2.884)	0x001a	0x001f	"IPM.document.txtfile"

4.2.4   Final Save

The protocol client uses the **RopSaveChangesMessage** ([MS-OXCROPS] section 2.2.6.3) operation to commit the properties on the protocol server and then uses **RopRelease** ([MS-OXCROPS] section 2.2.15.3) to release the object. The values of some properties will change during the execution of **RopSaveChangesMessage** ([MS-OXCROPS] section 2.2.6.3), but none of the properties specified in this protocol will change.

5   Security

5.1   Security Considerations for Implementers

Document objects store files as attachments. These files can be any files on the hard drive. When a user invokes a Document object, one behavior is to open up the attached file directly. There is a security implication here in that this file could do harmful things when invoked. While this is less of an issue for a user's personal mail folders, it becomes much more of an issue for public mail folders. It is up to the messaging client to choose what kind of behavior to follow when a user clicks on one of these Document objects.

5.2   Index of Security Parameters

Security Parameter	Section
PidNameSecurity ([MS-OXPROPS] section 2.539) property	Section 2.2.1.17

6 Appendix A: Product Behavior

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

- Microsoft® Exchange Server 2003

- Microsoft® Exchange Server 2007

- Microsoft® Exchange Server 2010

- Microsoft® Office Outlook® 2003

- Microsoft® Office Outlook® 2007

- Microsoft® Outlook® 2010

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

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

<1> Section 2.2: Office Outlook 2003 does not set the following properties: **PidLidPercentComplete**, **PidLidTaskActualEffort**, **PidLidTaskComplete**, **PidLidTaskAssigner**, **PidLidTaskAcceptanceState**, **PidLidTaskEstimatedEffort**, **PidLidTaskFFixOffline**, **PidLidTaskFRecurring**, **PidLidTaskNoCompute**, **PidLidTaskOrdinal**, **PidLidTaskOwnership**, **PidLidTaskRole**, **PidLidTaskState**, **PidLidTaskStatus**, **PidLidTaskVersion**, **PidLidTeamTask**, and **PidLidValidFlagStringProof**. For details about these properties, see [MS-OXPROPS].

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index