[MS-OXONOTE]:
Note Object Protocol Specification

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's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). 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.

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.

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.
04/10/2009	2.0		Updated 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.

Table of Contents

1 Introduction

This document specifies the **Note object** protocol, which defines **properties** of objects that function as electronic equivalents of paper sticky notes. A Note object presents the user with a very simple interface for keeping brief notes.

1.1 Glossary

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

Attachment object
best body
Folder object
handle
Mail User Agent (MUA)
Message object
named property
plain text
plain text message body
property
property ID
property type
recipient
remote operation (ROP)
special folder

The following terms are specific to this document:

Note object: A Message object that represents a simple text note in a messaging store and that adheres to the property specifications in this document. A Note object functions as the electronic equivalent of a paper sticky note.

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-OXCDATA] Microsoft Corporation, "Data Structures", June 2008.

[MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol Specification", June 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-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", June 2008.

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol Specification", June 2008.

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List", June 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-OXRTFEX] Microsoft Corporation, "Rich Text Format (RTF) Extensions Specification", June 2008.

1.3 Protocol Overview

The Note object protocol enables the representation of a simple text note in a messaging **store**. The Note object protocol extends the **Message** and **Attachment object** protocol in that it defines new properties and adds **restrictions** to the properties that are defined in [MS-OXCMSG].

A Note object represents a "sticky note." The properties that are specific to a Note object facilitate retaining information about the background color, window location, and size of the note. A Note object contains simple text (that is, text with minimal formatting other than line breaks), and is stored in a **Folder Object**. The Note object protocol also specifies how a Note object is created and manipulated.

1.4 Relationship to Other Protocols

The Note object protocol has the same dependencies as the Message and Attachment object Protocol, which it extends. For details about the Message and Attachment object protocol, see [MS-OXCMSG].

1.5 Prerequisites/Preconditions

The Note object protocol has the same prerequisites and preconditions as the Message and Attachment object protocol, as specified in [MS-OXCMSG].

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

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

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Note object protocol uses the protocols defined in [MS-OXCPRPT] and [MS-OXCMSG] as its primary transport mechanism.

2.2 Message Syntax

A Note object can be created and modified by clients and servers. Except where noted, this section defines constraints under which both clients and servers operate.

Clients operate on Note objects by using the Message and Attachment object protocol, as specified in [MS-OXCMSG]. How a server operates on Note objects is implementation-dependent. The results of any such operation are exposed to clients in a manner that is consistent with the Note object protocol.

Unless otherwise specified, a Note object adheres to all property constraints specified in [MS-OXPROPS] and all property constraints specified in [MS-OXCMSG]. A Note object can also contain other properties <1><2>, which are defined in [MS-OXPROPS], but these properties have no impact on the Note object protocol.

2.2.1 Note Object Properties

The following properties are specific to Note objects. All property value types (**PtypInteger32**, **PtypString**, etc.) are defined in [MS-OXCDATA] section 2.12.1.

2.2.1.1 PidLidNoteColor

Type: **PtypInteger32**

Specifies the suggested background color of the Note object; MUST be one of the entries in the following table <3>.

Value	Color
0x00000000	Blue
0x00000001	Green
0x00000002	Pink
0x00000003	Yellow
0x00000004	White

2.2.1.2 PidLidNoteWidth

Type: **PtypInteger32**, signed.

Specifies the width of the visible Message window in pixels; the value MUST be greater than zero.

2.2.1.3 PidLidNoteHeight

Type: **PtypInteger32**, signed.

Specifies the height of the visible Message window in pixels; the value MUST be greater than zero.

2.2.1.4 PidLidNoteX

Type: **PtypInteger32**, signed.

Specifies the distance, in pixels, from the left edge of the screen that a user interface displays a Note object.

2.2.1.5 PidLidNoteY

Type: **PtypInteger32**, signed.

Specifies the distance, in pixels, from the top edge of the screen that a user interface displays a Note object.

2.2.2 Additional Property Constraints

This protocol specifies additional constraints on the following properties beyond what is specified in [MS-OXCMSG]. All property value types, (**PtypInteger32**, **PtypString**, etc.) are defined in [MS-OXCDATA] section 2.12.1.

2.2.2.1 Best Body Properties

Specifies the content of the Note object. The content is a **plain text message body** stored in PidTagBody, as specified in [MS-OXCMSG] section 2.2.1.20.1 <4>.

2.2.2.2 PidTagIconIndex

Type: **PtypInteger32**.

Specifies which icon is to be used by a user interface when displaying a group of Note objects. The value MUST be 0x00000300 added to the value of PidLidNoteColor.

2.2.2.3 PidTagMessageClass

Type: **PtypString8**, case-insensitive.

Specifies the type of the Message item. The value MUST be "IPM.StickyNote" or begin with "IPM.StickyNote.", in addition to meeting the criteria specified in [MS-OXCMSG] section 2.2.1.3.

2.2.2.4 PidTagNormalizedSubject

Type: **PtypString**.

Specifies an abbreviated version of the contents of the note that is suitable for displaying groups of Note objects to a user. <5>

2.2.2.5 Recipients

A Note object MUST NOT have **recipients**.

2.2.2.6 Attachment Objects

A Note object MUST NOT have Attachment objects.

3 Protocol Details

General protocol details, as specified in [MS-OXPROPS] and [MS-OXCMSG], apply.

3.1 Common Details

The client and server roles are to create and manipulate electronic sticky notes, and otherwise operate in their roles as specified in [MS-OXCMSG].

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 specification 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 Note Objects

A Note object extends the **Message object** as defined in [MS-OXCMSG].

3.1.1.2 Note Special Folder

A Note object is created in the Notes **special folder** as defined in [MS-OXOSFLD] unless the **Mail User Agent (MUA)** explicitly specifies another Folder Object.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Creation of a Note Object

To create a Note object, the server or client creates a Message object as specified in [MS-OXCMSG], sets properties in accordance with the requirements in section 2 and [MS-OXCPRPT], and saves the resulting Message object as specified in [MS-OXCMSG].

3.1.4.2 Modification of a Note Object

When modifying a Note object, the client or server opens a Message object, as specified in [MS-OXCMSG], modifies any of the properties in accordance with the requirements in section 2 and [MS-OXCPRPT], and saves the Message object as specified in [MS-OXCMSG].

3.1.4.3 Deletion of a Note Object

Note objects have no special semantics in relation to deletion beyond what is defined in [MS-OXCFOLD].

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Sample Note Object

Joe creates a Note object, types in his grocery list, and saves it. The following is a description of what a client might do to accomplish Joe's intentions and the responses a server might return. For information about **remote operations (ROPs)**, see [MS-OXCPRPT] and [MS-OXCMSG].

Before manipulating Note objects, the client needs to ask the server to perform a mapping from **named properties** to **property IDs**, using RopGetPropertyIdsFromNames:

Property	Property set GUID	NameID
PidLidNoteColor	{0006200E-0000-0000-C000-000000000046}	0x00008B00
PidLidNoteWidth	{0006200E-0000-0000-C000-000000000046}	0x00008B02
PidLidNoteHeight	{0006200E-0000-0000-C000-000000000046}	0x00008B03
PidLidNoteX	{0006200E-0000-0000-C000-000000000046}	0x00008B04
PidLidNoteY	{0006200E-0000-0000-C000-000000000046}	0x00008B05

The server might respond with the following identifiers, which will be used in the example that follows. (The actual identifiers are at the discretion of the server.)

Property	Property ID
PidLidNoteColor	0x8046
PidLidNoteWidth	0x8047
PidLidNoteHeight	0x8048
PidLidNoteX	0x8049
PidLidNoteY	0x804A

To create a Note object, the client uses RopCreateMessage. The server returns a success code and a **handle** to a Message object.

After Joe has input his content for the Note object, the client uses RopSetProperties to transmit his data to the server. For information about **property types**, see [MS-OXCDATA].

Property	Property ID	Data Type	Value
PidLidNoteColor	0x8046	0x0003 (PtypInteger32)	0x00000003
PidLidNoteWidth	0x8047	0x0003 (PtypInteger32)	0x000000C8
PidLidNoteHeight	0x8048	0x0003 (PtypInteger32)	0x000000A6
PidLidNoteX	0x8049	0x0003 (PtypInteger32)	0x0000006E
PidLidNoteY	0x804A	0x0003 (PtypInteger32)	0x0000006E

Property	Property ID	Data Type	Value
PidTagIconIndex	0x1080	0x0003 (PtypInteger32)	0x00000303
PidTagMessageClass	0x001A	0x001F (PtypString)	IPM.StickyNote
PidTagNormalizedSubject	0x0E1D	0x001F (PtypString)	Grocery List
PidTagSubjectPrefix	0x003D	0x001F (PtypString)	(null)
PidTagBody	0x1000	0x001F (PtypString)	Grocery List: Celery Broccoli

When Joe is ready to save his changes, the client uses RopSaveChangesMessage to commit the properties on the server, and then RopRelease to release the Note object.

The values of some properties will change during the execution of RopSaveChangesMessage, but the properties specified in this document will not change.

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to the Note object protocol. General security considerations pertaining to the underlying transport apply, as specified in [MS-OXCMSG] and [MS-OXCPRPT].

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following product versions. References to product versions include released service packs.

- Microsoft Office Outlook 2003
- Microsoft Exchange Server 2003
- Microsoft Office Outlook 2007
- Microsoft Exchange Server 2007
- Microsoft Outlook 2010
- Microsoft Exchange Server 2010

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

Unless otherwise specified, any statement of optional behavior in this specification 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 product does not follow the prescription.

<1> Section 2.2: Outlook 2003 SP3 and Outlook 2007 SP1 set the following properties regardless of user input; their values have no meaning in the context of this protocol.PidLidAgingDontAgeMe, PidLidCurrentVersion, PidLidCurrentVersionName, PidLidPrivate, PidLidSideEffects, PidTagAlternateRecipientAllowed, PidTagClientSubmitTime, PidTagDeleteAfterSubmit, PidTagImportance, PidTagMessageDeliveryTime, PidTagPriority, PidTagReadReceiptRequested, PidTagSensitivity, PidLidReminderDelta, PidLidReminderSet, PidLidReminderTimeTime, PidLidTaskMode

<2> Section 2.2: Outlook 2007 SP1 sets the following properties regardless of user input; their values have no meaning in the context of this protocol.PidLidPercentComplete, PidLidTaskActualEffort, PidLidTaskComplete, PidLidTaskAssigner, PidLidTaskAcceptanceState, PidLidTaskEstimatedEffort, PidLidTaskFFixOffline, PidLidTaskFRecurring, PidLidTaskNoCompute, PidLidTaskOrdinal, PidLidTaskOwnership, PidLidTaskRole, PidLidTaskState, PidLidTaskStatus, PidLidTaskVersion, PidLidTeamTask, PidLidValidFlagStringProof

<3> Section 2.2.1.1: Outlook 2003 SP3 will always use PidLidNoteColor to determine the background color, regardless of the existence or value of PidNameKeywords. Outlook 2007 SP1 ignores PidLidNoteColor if the item has PidNameKeywords set also. In that case, the background color will be the color associated with the first keyword listed, as specified in [MS-OXOCFG].

<4> Section 2.2.2.1: Outlook 2003 SP3 and Office Outlook 2007 SP1 set encapsulated **plain text** as a Rich Text Body.For more information, see [MS-OXRTFEX] and [MS-OXCMSG].

<5> Section 2.2.2.4: Microsoft Outlook 2007 always sets this property to the first line of the message body.

7 Change Tracking

This section identifies changes made to [MS-OXONOTE] protocol documentation between July 2009 and November 2009 releases. Changes are classed as major, minor, or editorial.

Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

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

- New content added.
- Content update.
- 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.
- New content added for template compliance.
- Content updated for template compliance.

- Content removed for template compliance.
- Obsolete document removed.

Editorial changes always have the revision type "Editorially updated."

Some important terms used in revision 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.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
1.2.1 Normative References	49453 Added reference for property type values.	Y	Content update.
1.2.2 Informative References	49452 Added [MS-OXRTFEX] reference.	N	Content update.
2.2 Message Syntax	49597 Clarified behavior for setting certain properties.	N	Product behavior note updated.
2.2.1 Note Object Properties	49453 Added reference for property type values.	Y	Content update.
2.2.2 Additional Property Constraints	49453 Added reference for property type values.	Y	Content update.
2.2.2.1 Best Body Properties	49452 Re-worded for informative reference.	N	Product behavior note updated.
2.2.2.1 Best Body Properties	49458 Clarified description.	N	Content update.
2.2.2.4 PidTagNormalizedSubject	49450 Added note about Microsoft Outlook 2007 behavior.	N	New content added.
4.1 Sample Note Object	49459 Revised data type of PidTagMessageClass in sample.	Y	Content update.
6 Appendix A: Product Behavior	49460 Added 2003 products.	N	Content update.

8 Index