Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.

- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.

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

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

- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.

- **Trademarks.** The names of companies and products contained in this documentation might 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, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/4/2008</td>
<td>0.1</td>
<td>New</td>
<td>Initial Availability.</td>
</tr>
<tr>
<td>6/27/2008</td>
<td>1.0</td>
<td>Major</td>
<td>Initial Release.</td>
</tr>
<tr>
<td>8/6/2008</td>
<td>1.01</td>
<td>Minor</td>
<td>Revised and edited technical content.</td>
</tr>
<tr>
<td>9/3/2008</td>
<td>1.02</td>
<td>Minor</td>
<td>Updated references.</td>
</tr>
<tr>
<td>12/3/2008</td>
<td>1.03</td>
<td>Minor</td>
<td>Updated IP notice.</td>
</tr>
<tr>
<td>4/10/2009</td>
<td>2.0</td>
<td>Major</td>
<td>Updated applicable product releases.</td>
</tr>
<tr>
<td>7/15/2009</td>
<td>3.0</td>
<td>Major</td>
<td>Revised and edited for technical content.</td>
</tr>
<tr>
<td>11/4/2009</td>
<td>4.0.0</td>
<td>Major</td>
<td>Updated and revised the technical content.</td>
</tr>
<tr>
<td>2/10/2010</td>
<td>4.0.0</td>
<td>None</td>
<td>Version 4.0.0 release</td>
</tr>
<tr>
<td>5/5/2010</td>
<td>4.0.1</td>
<td>Editorial</td>
<td>Revised and edited the technical content.</td>
</tr>
<tr>
<td>8/4/2010</td>
<td>4.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>11/3/2010</td>
<td>4.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>4.2</td>
<td>None</td>
<td>No changes to the meaning, language, and formatting of the technical content.</td>
</tr>
<tr>
<td>8/5/2011</td>
<td>4.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/7/2011</td>
<td>4.3</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/27/2012</td>
<td>5.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>5.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>5.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>5.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/26/2013</td>
<td>5.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>5.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>5.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>5.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>5.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>
# Table of Contents

1 Introduction ............................................................................................................. 6  
1.1 Glossary ........................................................................................................... 6  
1.2 References ....................................................................................................... 7  
1.2.1 Normative References ............................................................................. 7  
1.2.2 Informative References .......................................................................... 7  
1.3 Overview ......................................................................................................... 8  
1.4 Relationship to Other Protocols .................................................................. 8  
1.5 Prerequisites/Preconditions .......................................................................... 8  
1.6 Applicability Statement ................................................................................. 8  
1.7 Versioning and Capability Negotiation ....................................................... 8  
1.8 Vendor-Extensible Fields ............................................................................. 8  
1.9 Standards Assignments ................................................................................ 8  

2 Messages ............................................................................................................. 9  
2.1 Transport ....................................................................................................... 9  
2.2 Message Syntax ............................................................................................. 9  
2.2.1 Note Object Properties .......................................................................... 9  
2.2.1.1 PidLidNoteColor Property .......................................................... 9  
2.2.1.2 PidLidNoteWidth Property ......................................................... 9  
2.2.1.3 PidLidNoteHeight Property ....................................................... 9  
2.2.1.4 PidLidNoteX Property .............................................................. 10  
2.2.1.5 PidLidNoteY Property .............................................................. 10  
2.2.2 Additional Property Constraints .......................................................... 10  
2.2.2.1 Best Body Properties .................................................................. 10  
2.2.2.2 PidTagIconIndex Property ....................................................... 10  
2.2.2.3 PidTagMessageClass Property ............................................... 10  
2.2.2.4 PidTagNormalizedSubject Property ..................................... 10  
2.2.2.5 Recipients ..................................................................................... 10  
2.2.2.6 Attachment Objects .................................................................... 11  

3 Protocol Details .................................................................................................. 12  
3.1 Client Details ............................................................................................... 12  
3.1.1 Abstract Data Model ............................................................................. 12  
3.1.2 Timers ..................................................................................................... 12  
3.1.3 Initialization ............................................................................................ 12  
3.1.4 Higher-Layer Triggered Events .............................................................. 12  
3.1.4.1 Creating a Note Object ............................................................... 12  
3.1.4.2 Modifying a Note Object ............................................................. 12  
3.1.4.3 Deleting a Note Object ................................................................ 12  
3.1.5 Message Processing Events and Sequencing Rules .................................. 13  
3.1.6 Timer Events ............................................................................................ 13  
3.1.7 Other Local Events ................................................................................ 13  
3.2 Server Details ................................................................................................. 13  
3.2.1 Abstract Data Model ............................................................................. 13  
3.2.2 Timers ..................................................................................................... 13  
3.2.3 Initialization ............................................................................................ 13  
3.2.4 Higher-Layer Triggered Events .............................................................. 13  
3.2.5 Message Processing Events and Sequencing Rules .................................. 13  
3.2.6 Timer Events ............................................................................................ 13  
3.2.7 Other Local Events ................................................................................ 14  

4 Protocol Examples .............................................................................................. 15  
4.1 Sample Note Object ...................................................................................... 15  

5 Security .............................................................................................................. 17  
5.1 Security Considerations for Implementers ................................................. 17
1 Introduction

The Note Object Protocol enables the representation of a brief note that functions as the electronic equivalent of a paper sticky note. This protocol extends the Message and Attachment Object Protocol, which is described in [MS-OXCMSG].

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

- **Attachment object**: A set of properties that represents a file, **Message object**, or structured storage that is attached to a Message object and is visible through the attachments table for a Message object.

- **Folder object**: A messaging construct that is typically used to organize data into a hierarchy of objects containing Message objects and folder associated information (FAI) Message objects.

- **handle**: Any token that can be used to identify and access an object such as a device, file, or a window.

- **long ID (LID)**: A 32-bit quantity that, in combination with a GUID, defines a **named property**.

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

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

- **Message 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.

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

- **named property set**: A GUID that groups related named properties into a set.

- **Note object**: A **Message object** that represents a simple text note in a messaging store and that adheres to the property descriptions that are described in [MS-OXONOTE]. A Note object functions as an electronic equivalent of a paper sticky note.

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

- **plain text message body**: A message body 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.

- **property ID**: A 16-bit numeric identifier of a specific attribute. A property ID does not include any **property type** information.

- **property name**: A string that, in combination with a property set, identifies a **named property**.

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

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

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

**ROP request:** See ROP request buffer.

**ROP response:** See ROP response buffer.

**special folder:** One of a default set of Folder objects that can be used by an implementation to store and retrieve user data objects.

**tagged property:** A property that is defined by a 16-bit property ID and a 16-bit property type. The property ID for a tagged property is in the range 0x001 – 0x7FFF. Property IDs in the range 0x8000 – 0x8FFF are reserved for assignment to named properties.

**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-OXCDATA] Microsoft Corporation, "Data Structures".
- [MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol".
- [MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol".
- [MS-OXOMSG] Microsoft Corporation, "Email Object Protocol".
- [MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol".

#### 1.2.2 Informative References

- [MS-OXOCFG] Microsoft Corporation, "Configuration Information Protocol".
1.3 Overview

The Note Object Protocol allows a user to store in his mailbox a simple text note (that is, text with minimal formatting) that functions as the electronic equivalent of a paper sticky note. To represent the sticky note, this protocol defines a Note object. The properties that are specific to a Note object contain information about the background color, window location, and size of the note. A Note object is stored in a Folder object. The Note Object Protocol also specifies how a Note object is created and manipulated.

The Note Object Protocol extends the Message and Attachment Object Protocol, described in [MS-OXCMMSG], in that it defines new properties for a Message object and adds constraints to the existing properties of a Message object.

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 information about the Message and Attachment Object Protocol, see [MS-OXCMMSG].

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

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-OXCMMSG].

1.6 Applicability Statement

A client uses this protocol to create and maintain a user's sticky notes.

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-OXCMMSG].

1.9 Standards Assignments

None.
2 Messages

2.1 Transport

The Note Object Protocol uses the same underlying transport as that used by the Message and Attachment Object Protocol, which is specified in [MS-OXCMSG].

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.

A client operates on a Note object by using the Message and Attachment Object Protocol, as specified in [MS-OXCMSG]. How a server operates on a Note object is implementation-dependent, but the results of any such operation MUST be 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].

2.2.1 Note Object Properties

The properties specific to a Note object are defined in section 2.2.1.1 through section 2.2.1.5.

2.2.1.1 PidLidNoteColor Property

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

The PidLidNoteColor property ([MS-OXPROPS] section 2.183) specifies the suggested background color of the note. This property MUST be set to one of the values specified in the following table.<1>

<table>
<thead>
<tr>
<th>Value</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Blue</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Green</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Pink</td>
</tr>
<tr>
<td>0x00000003</td>
<td>Yellow</td>
</tr>
<tr>
<td>0x00000004</td>
<td>White</td>
</tr>
</tbody>
</table>

2.2.1.2 PidLidNoteWidth Property

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

The PidLidNoteWidth property ([MS-OXPROPS] section 2.185) specifies the width of the note's visible window in pixels. The value of this property MUST be greater than zero.

2.2.1.3 PidLidNoteHeight Property

Type: PtypInteger32 ([MS-OXCDATA] section 2.11.1)
The **PidLidNoteHeight** property ([MS-OXPROPS] section 2.184) specifies the height of the note’s visible window in pixels. The value of this property MUST be greater than zero.

### 2.2.1.4 PidLidNoteX Property

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

The **PidLidNoteX** property ([MS-OXPROPS] section 2.186) specifies the distance, in pixels, from the left edge of the screen that a user interface displays the note.

### 2.2.1.5 PidLidNoteY Property

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

The **PidLidNoteY** property ([MS-OXPROPS] section 2.187) specifies the distance, in pixels, from the top edge of the screen that a user interface displays the note.

### 2.2.2 Additional Property Constraints

This protocol specifies additional constraints on some Message object properties beyond what is specified in [MS-OXCMSG]. These constraints are specified in sections 2.2.2.1 through 2.2.2.6.

#### 2.2.2.1 Best Body Properties

Best body properties specify the content of the note. The content is a plain text message body stored in the **PidTagBody** property ([MS-OXCMSG] section 2.2.1.56.1).<2>

#### 2.2.2.2 PidTagIconIndex Property

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

The **PidTagIconIndex** property ([MS-OXOMSG] section 2.2.1.10) specifies which icon a user interface is to use when displaying a group of Note objects. The value of this property MUST be 0x00000300 added to the value of the **PidLidNoteColor** property (section 2.2.1.1).

#### 2.2.2.3 PidTagMessageClass Property

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

The **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) specifies the type of the Message object. 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 Property

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

The **PidTagNormalizedSubject** property ([MS-OXCMSG] section 2.2.1.10) specifies an abbreviated version of the contents of the note.<3>

#### 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

3.1 Client Details

The client creates and manipulates a Note object and in all other ways operates within the client role 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 document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

This protocol uses the abstract data model that is specified in [MS-OXCMSG] section 3.1.1 with the following adaptations:

- The Note object is an extension of the Message object.
- A Note object is created in the Notes folder, which is a special folder, unless the Mail User Agent (MUA) explicitly specifies another folder. For details about special folders, see [MS-OXOSFLD].

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Creating a Note Object

When a user creates a new note, the client creates a Message object as specified in [MS-OXCMSG] section 3.1.4.2, sets properties in accordance with the requirements in section 2.2, and saves the resulting Message object as specified in [MS-OXCMSG] section 3.1.4.3. For details about setting properties, see [MS-OXCPRPT].

3.1.4.2 Modifying a Note Object

When a user opens and modifies an existing note, the client opens the Note object in the same way that it opens any Message object, as specified in [MS-OXCMSG] section 3.1.4.1. The client then modifies any of the properties in accordance with the requirements in section 2.2 and saves the Note object as specified in [MS-OXCMSG] section 3.1.4.3. For details about setting properties, see [MS-OXCPRPT].

3.1.4.3 Deleting a Note Object

When a user deletes a note, the client deletes the Note object in the same way that it deletes any Message object, as specified in [MS-OXCFOLD] section 3.1.4.8.
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 processes a client's requests regarding a Note object and in all other ways operates within the server role as specified in [MS-OXCMSG].

3.2.1 Abstract Data Model

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

This protocol uses the abstract data model that is specified in [MS-OXCMSG] section 3.2.1 with the following adaptations:

- The Note object is an extension of the Message object.
- A Note object is created in the Notes folder, which is a special folder, unless the MUA explicitly specifies another folder. For details about special folders, see [MS-OXOSFLD].

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The server responds to client requests as specified in [MS-OXCMSG] section 3.2.5.

3.2.6 Timer Events

None.
3.2.7 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.

This example uses both **named properties** and **tagged properties** of a Note object. The **property ID** of a named property is provided by the server. Therefore, before setting or reading any properties of a Note object, the client asks the server to perform a mapping from property names or long IDs (LIDs) to property IDs. To request this mapping, the client sends a **RopGetPropertyIdsFromNames ROP request** ([MS-OXCROPS] section 2.2.8.1).

The following table lists each named property with its **named property set GUID** and its long ID (LID) or property name. The server's **RopGetPropertyIdsFromNames ROP response** provides the corresponding property IDs, as shown in the subsequent table.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property set GUID</th>
<th>LID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidNoteColor</td>
<td>{0006200E-0000-0000-C000-000000000046}</td>
<td>0x00008B00</td>
</tr>
<tr>
<td>PidLidNoteWidth</td>
<td>{0006200E-0000-0000-C000-000000000046}</td>
<td>0x00008B02</td>
</tr>
<tr>
<td>PidLidNoteHeight</td>
<td>{0006200E-0000-0000-C000-000000000046}</td>
<td>0x00008B03</td>
</tr>
<tr>
<td>PidLidNoteX</td>
<td>{0006200E-0000-0000-C000-000000000046}</td>
<td>0x00008B04</td>
</tr>
<tr>
<td>PidLidNoteY</td>
<td>{0006200E-0000-0000-C000-000000000046}</td>
<td>0x00008B05</td>
</tr>
</tbody>
</table>

The following table lists the property IDs that might be provided in the server's **RopGetPropertyIdsFromNames ROP** response. The actual property IDs returned are at the discretion of the server.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidNoteColor</td>
<td>0x8046</td>
</tr>
<tr>
<td>PidLidNoteWidth</td>
<td>0x8047</td>
</tr>
<tr>
<td>PidLidNoteHeight</td>
<td>0x8048</td>
</tr>
<tr>
<td>PidLidNoteX</td>
<td>0x8049</td>
</tr>
<tr>
<td>PidLidNoteY</td>
<td>0x804A</td>
</tr>
</tbody>
</table>

To create a Note object, the client uses the **RopCreateMessage ROP** ([MS-OXCROPS] section 2.2.6.2). 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 transmits the properties of the Note object to the server by using the **RopSetProperties ROP** ([MS-OXCROPS] section 2.2.8.6). The properties that are set on the Note object are shown in the following table. For information about **property types** in the following table, see [MS-OXCDATA] section 2.11.1.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidNoteColor</td>
<td>0x8046</td>
<td>PtypInteger32</td>
<td>0x00000003</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidNoteWidth</td>
<td>0x8047</td>
<td>0x0003</td>
<td>0x000000C8</td>
</tr>
<tr>
<td>PidLidNoteHeight</td>
<td>0x8048</td>
<td>0x0003</td>
<td>0x000000A6</td>
</tr>
<tr>
<td>PidLidNoteX</td>
<td>0x8049</td>
<td>0x0003</td>
<td>0x0000006E</td>
</tr>
<tr>
<td>PidLidNoteY</td>
<td>0x804A</td>
<td>0x0003</td>
<td>0x0000006E</td>
</tr>
<tr>
<td>PidTagIconIndex</td>
<td>0x1080</td>
<td>0x0003</td>
<td>0x00000303</td>
</tr>
<tr>
<td>PidTagMessageClass</td>
<td>0x001A</td>
<td>0x001F (PtypString)</td>
<td>&quot;IPM.StickyNote&quot;</td>
</tr>
<tr>
<td>PidTagNormalizedSubject</td>
<td>0x0E1D</td>
<td>0x001F</td>
<td>&quot;Grocery List&quot;</td>
</tr>
<tr>
<td>PidTagSubjectPrefix</td>
<td>0x003D</td>
<td>0x001F</td>
<td>&quot;&quot; (null)</td>
</tr>
<tr>
<td>PidTagBody</td>
<td>0x1000</td>
<td>0x001F</td>
<td>&quot;Grocery List: Celery Broccoli&quot;</td>
</tr>
</tbody>
</table>

When Joe is ready to save his changes, the client commits the properties on the server by using the `RopSaveChangesMessage` ROP ([MS-OXCROPS] section 2.2.6.3) and then releases the Note object by using the `RopRelease` ROP ([MS-OXCROPS] section 2.2.15.3).

The values of some properties of the Message object will change during the execution of the `RopSaveChangesMessage` ROP, 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 described in [MS-OXCMSG].

5.2 Index of Security Parameters

None.
Appendix A: Product Behavior

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

- Microsoft Exchange Server 2003
- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016
- Microsoft Exchange Server 2019
- Microsoft Outlook 2019
- Microsoft Outlook 2021

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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 2.2.1.1: Microsoft Office Outlook 2003 Service Pack 3 (SP3) will always use the PidLidNoteColor property to determine the background color, regardless of the existence or value of the PidNameKeywords property ([MS-OXCMSG] section 2.2.1.17). Microsoft Office Outlook 2007 Service Pack 1 ignores the PidLidNoteColor property if the item has the PidNameKeywords property set also. In that case, the background color is the color associated with the first keyword listed, as described in [MS-OXOCFG].

<2> Section 2.2.2.1: Office Outlook 2003 SP3 and Office Outlook 2007 SP1 set encapsulated plain text as a Rich Text Format (RTF) for the message body. For more information, see [MS-OXRTFEX] and [MS-OXCMSG].

<3> Section 2.2.2.4: Office Outlook 2007 always sets this property to the first line of the message body.
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

A
Abstract data model
  client 12
  server 13
Additional property constraints
  Attachment objects 11
  best body properties 10
  PidTagIconIndex property 10
  PidTagMessageClass property 10
  PidTagNormalizedSubject property 10
  recipients 10
  Additional Property Constraints message 10
Applicability 8
Attachment objects additional property constraints 11

B
Best body properties additional property constraints 10

C
Capability negotiation 8
Change tracking 19
Client
  abstract data model 12
  initialization 12
  message processing 13
  other local events 13
  overview 12
  sequencing rules 13
  timer events 13
  timers 12
Client - higher-layer triggered events
  creating a Note object 12
  deleting a Note object 12
  modifying a Note object 12

D
Data model - abstract
  client 12
  server 13

E
Examples
  sample Note object 15

F
Fields - vendor-extensible 8

G
Glossary 6

H
Higher-layer triggered events

server 13
Higher-layer triggered events - client
  creating a Note object 12
  deleting a Note object 12
  modifying a Note object 12

I
Implementer - security considerations 17
Index of security parameters 17
Informative references 7
Initialization
  client 12
  server 13
  Introduction 6

M
Message processing
  client 13
  server 13
Message syntax 9
Messages
  Additional Property Constraints 10
  message syntax 9
  Note Object Properties 9
  transport 9

N
Normative references 7
Note object properties
  PidLidNoteColor property 9
  PidLidNoteHeight property 9
  PidLidNoteWidth property 9
  PidLidNoteX property 10
  PidLidNoteY property 10
  Note Object Properties message 9

O
Other local events
  client 13
  server 14
  Overview (synopsis) 8

P
Parameters - security index 17
  PidLidNoteColor Note object property 9
  PidLidNoteHeight Note object property 9
  PidLidNoteWidth Note object property 9
  PidLidNoteX Note object property 10
  PidLidNoteY Note object property 10
  PidTagIconIndex property additional property constraints 10
  PidTagMessageClass property additional property constraints 10
  PidTagNormalizedSubject property additional property constraints 10
  Preconditions 8
  Prerequisites 8
Product behavior 18

R
Recipients additional property constraints 10
References 7
   informative 7
   normative 7
Relationship to other protocols 8

S
Sample Note object example 15
Security
   implementer considerations 17
   parameter index 17
Sequencing rules
   client 13
   server 13
Server
   abstract data model 13
   higher-layer triggered events 13
   initialization 13
   message processing 13
   other local events 14
   overview 13
   sequencing rules 13
   timer events 13
   timers 13
Standards assignments 8

T
Timer events
   client 13
   server 13
Timers
   client 12
   server 13
Tracking changes 19
Transport 9
Triggered events - client
   creating a Note object 12
   deleting a Note object 12
   modifying a Note object 12
Triggered events - higher-layer
   server 13

V
Vendor-extensible fields 8
Versioning 8