# [MS-OXOPOST]: Post 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 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.
- **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 (beta) 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 (beta) 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 |
|---|---|---|---|
| 04/04/2008 | 0.1 | | Initial Availability. |
| 04/25/2008 | 0.2 | | Revised and updated property names and other technical content. |
| 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. |
| 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 | 4.2 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 03/18/2011 | 4.2 | No change | No changes to the meaning, language, and formatting of the technical content. |
| 08/05/2011 | 4.2 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 10/07/2011 | 4.2 | No change | No changes to the meaning, language, or formatting of the technical content. |
| 01/20/2012 | 5.0 | Major | Significantly changed the technical content. |

# Table of Contents

# 1 Introduction

This document specifies the Post Object Protocol, which defines properties of an object that models the electronic equivalent of a bulletin board post.

Sections 1.8, 2, and 3 of this specification are normative and contain RFC 2119 language. Sections 1.5 and 1.9 are also normative but cannot contain RFC 2119 language. All other sections and examples in this specification are informative.

## 1.1 Glossary

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

**handle**

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

**address book**
**conversation**
**entry ID**
**Folder object**
**Hypertext Markup Language (HTML)**
**Message object**
**Post object**
**recipient**
**remote operation (ROP)**
**store**

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

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 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".

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

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

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

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

[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

### 1.2.2 Informative References

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

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

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

[MS-OXOABK] Microsoft Corporation, "Address Book Object Protocol Specification".

## 1.3 Overview

The Post Object Protocol allows the representation of a bulletin board post in a messaging **store**. The Post Object Protocol extends the Message and Attachment Object Protocol in that it adds restrictions to the properties that are specified in [MS-OXCMSG].

A **Post object** represents a bulletin board post. There are no properties specific to Post objects. A Post object is stored in a **Folder object**. The Post Object Protocol also specifies how a Post object is created and manipulated.

## 1.4 Relationship to Other Protocols

The Post 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].

The Post Object Protocol is a peer of the E-mail Object Protocol, and uses a subset of the properties specified in [MS-OXOMSG].

## 1.5 Prerequisites/Preconditions

The Post Object Protocol has the same prerequisites and preconditions as the Message and Attachment Object Protocol. For details about the Message and Attachment Object Protocol, see [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 Post Object Protocol uses protocols specified in [MS-OXCPRPT] and [MS-OXCMSG] as its primary transport mechanism.

## 2.2 Message Syntax

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

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

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

Post object properties can be of several different data types. The following data types used by Post object properties are specified in [MS-OXCDATA] section 2.11.1:

- PtypBinary
- PtypInteger32
- PtypString
- PtypString8

### 2.2.1 Post Object Properties

There are no properties in addition to those listed in [MS-OXCMSG] and [MS-OXOMSG] that are specific to a Post object.

### 2.2.2 Additional Property Constraints

This protocol specifies additional constraints on the following properties beyond what is specified in [MS-OXCMSG] and [MS-OXOMSG].

#### 2.2.2.1 PidTagConversationIndex Property

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

The **PidTagConversationIndex** property ([MS-OXOMSG] section 2.2.1.3) specifies the depth of the reply in a hierarchical representation of Post objects in one **conversation**. This value MUST be set on a Post object as specified in [MS-OXOMSG].

#### 2.2.2.2 PidTagConversationTopic Property

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

The **PidTagConversationTopic** property ([MS-OXOMSG] section 2.2.1.5) contains an unchanging copy of the original subject. This value MUST be set to the same value as the **PidTagNormalizedSubject** property ([MS-OXCMSG] section 2.2.1.10) on a new Post object when it is first saved. When creating a Post object as a reply, the **PidTagConversationTopic** property on the new Post object MUST be copied from the original Post object.

### 2.2.2.3 PidTagIconIndex Property

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

The **PidTagIconIndex** property ([MS-OXOMSG] section 2.2.1.10) specifies which icon is to be used by a user interface when displaying a group of Post objects. This value MUST be "0x00000001".

### 2.2.2.4 PidTagMessageClass Property

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

The **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) specifies the type of the **Message object**. This value MUST be "IPM.Post" or begin with "IPM.Post", in addition to meeting the criteria specified in [MS-OXCMSG].

### 2.2.2.5 Sender Properties

The following properties are specified in [MS-OXOMSG] to represent the sender of an e-mail Message object. They are used in this protocol to represent the creator of a Post object:

- **PidTagSenderAddressType** ([MS-OXOMSG] section 2.2.2.6)
- **PidTagSenderEntryId** ([MS-OXOMSG] section 2.2.1.42)
- **PidTagSenderName** ([MS-OXOMSG] section 2.2.1.43)
- **PidTagSenderSearchKey** ([MS-OXOMSG] section 2.2.1.44)
- **PidTagSentRepresentingAddressType** ([MS-OXOMSG] section 2.2.1.46)
- **PidTagSentRepresentingEntryId** ([MS-OXOMSG] section 2.2.1.48)
- **PidTagSentRepresentingName** ([MS-OXOMSG] section 2.2.1.49)
- **PidTagSentRepresentingSearchKey** ([MS-OXOMSG] section 2.2.1.50)

### 2.2.2.6 Recipients

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

# 3 Protocol Details

General protocol details, as specified in [MS-OXCPRPT] and [MS-OXCMSG], apply to Post objects.

## 3.1 Common Details

The client and server roles are to create and operate on electronic discussion items, and fulfill 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 document does not mandate that implementations adhere to this model as long as the external behavior of the implementation is consistent with that specified in this document.

#### 3.1.1.1 Post Objects

A Post object extends the Message object as specified in [MS-OXCMSG].

### 3.1.2 Timers

None.

### 3.1.3 Initialization

None.

### 3.1.4 Higher-Layer Triggered Events

#### 3.1.4.1 Creation of a Post Object

To create a Post 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 Post Object

When modifying a Post object, the client or server opens a Message object (as specified in [MS-OXCMSG]), modifies any 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 Post Object

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

#### 3.1.4.4 Reply to Folder

To create a reply to a Post object, the protocol client creates a new Post object in the same Folder object as the original. The new Post object has the same **PidTagConversationTopic** ([MS-OXOMSG] section 2.2.1.5) as the original, and an incremented **PidTagConversationIndex** ([MS-OXOMSG] section 2.2.1.3). For more details, see [MS-OXOMSG].

### 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 Post Object

Joe wants to record his grocery list of celery and broccoli, so he creates a Post object, adds a subject and body, and posts it in a Folder object. The following is a description of what a client might do to accomplish Joe's intentions, and the responses a server might return. For more details about **remote operations (ROPs)**, see [MS-OXCPRPT] and [MS-OXCMSG].

To create a Post object, the client uses **RopCreateMessage** ([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 Post object, the client uses **RopSetProperties** ([MS-OXCROPS] section 2.2.8.6) to transmit his data to the server.

| Property | Property ID | Property type | Value |
|---|---|---|---|
| **PidTagIconIndex** ([MS-OXOMSG] section 2.2.1.10) | 0x1080 | 0x0003 (PtypInteger32) | 0x00000001 |
| **PidTagMessageClass** ([MS-OXCMSG] section 2.2.1.3) | 0x001a | 0x001f (PtypString) | IPM.Post |
| **PidTagNormalizedSubject** ([MS-OXCMSG] section 2.2.1.10) | 0x0e1d | 0x001f (PtypString) | Grocery List |
| **PidTagSubjectPrefix** ([MS-OXCMSG] section 2.2.1.9) | 0x003d | 0x001f (PtypString) | (null) |
| **PidTagConversationTopic** ([MS-OXOMSG] section 2.2.1.5) | 0x0070 | 0x001f (PtypString) | Grocery List |
| **PidTagConversationIndex** ([MS-OXOMSG] section 2.2.1.3) | 0x0071 | 0x0102 (PtypBinary) | See Note 1, following this table. |
| **PidTagHtml** ([MS-OXCMSG] section 2.2.1.48.9) | 0x1013 | 0x0102 (PtypBinary) | See Note 2, following this table. |
| **PidTagSenderName** ([MS-OXOMSG] section 2.2.1.43) | 0x0c1a | 0x001f (PtypString) | Joe Healy |
| **PidTagSenderAddressType** ([MS-OXOMSG] section 2.2.2.6) | 0x0c1e | 0x001f (PtypString) | EX |
| **PidTagSenderEntryId** ([MS-OXOMSG] section 2.2.1.42) | 0x0c19 | 0x0102 (PtypBinary) | See Note 3, following this table. |
| **PidTagSenderSearchKey** ([MS-OXOMSG] section 2.2.1.44) | 0x0c1d | 0x0102 (PtypBinary) | See Note 4, following this table. |
| **PidTagSentRepresentingName** ([MS-OXOMSG] section 2.2.1.49) | 0x0042 | 0x001f | Joe Healy |

| Property | Property ID | Property type | Value |
|---|---|---|---|
| | | (PtypString) | |
| **PidTagSentRepresentingAddressType** ([MS-OXOMSG] section 2.2.1.46) | 0x0064 | 0x001f<br>(PtypString) | EX |
| **PidTagSentRepresentingEntryId** ([MS-OXOMSG] section 2.2.1.48) | 0x0041 | 0x0102<br>(PtypBinary) | See Note 3, following this table. |
| **PidTagSentRepresentingSearchKey** ([MS-OXOMSG] section 2.2.1.50) | 0x003b | 0x0102<br>(PtypBinary) | See Note 4, following this table. |

**Notes**

1. The **PidTagConversationIndex** property is set with a depth of 1, as described in [MS-OXOMSG] section 2.2.1.3, and has the following binary contents:

```
0000: 01 c8 73 2d a1 0a 3e b3-ee 24 90 f4 45 be 97 10
0010: 90 b2 a5 07 7a 13
```

2. The **PidTagHtml** property is set to the **Hypertext Markup Language (HTML)** representation of "Celery\r\nBroccoli", which is as follows:

```
<html>
<head>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=us-ascii">
</head>
<body lang=EN-US>
<p>Celery</p>
<p>Broccoli</p>
</body>
</html>
```

3. The **PidTagSenderEntryId** and **PidTagSentRepresentingEntryId** properties are identical because Joe isn't posting this on behalf of another user. The contents of these properties are Joe's **address book entry ID** as described in [MS-OXOABK].

4. The **PidTagSenderSearchKey** and **PidTagSentRepresentingSearchKey** properties are identical because Joe isn't posting this on behalf of another user. The contents of these properties are described in [MS-OXOMSG], and are used as Joe's search key, **PidTagSearchKey** ([MS-OXCPRPT] section 2.2.1.9).

When Joe is ready to save his changes, the client uses **RopSaveChangesMessage** ([MS-OXCROPS] section 2.2.6.3) to commit the properties on the server, and then uses **RopRelease** ([MS-OXCROPS] section 2.2.15.3) to release the handle to the 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 Post Object Protocol. General security considerations pertaining to the underlying transport apply, as described 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 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 15 Technical Preview
- Microsoft® Office Outlook® 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Outlook® 2010
- Microsoft® Outlook® 15 Technical 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 2.2: Office Outlook 2003 and Office Outlook 2007 set the following properties regardless of user input, but 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**, **PidTagInternetReferences**. For details about these properties, see [MS-OXPROPS].

<2> Section 2.2: Office Outlook 2007 sets the following properties regardless of user input, but 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**. For details about these properties, see [MS-OXPROPS].

# 7 Change Tracking

This section identifies changes that were made to the [MS-OXOPOST] protocol document between the October 2011 and January 2012 releases. 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.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 language and 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 or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- 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 protocol@microsoft.com.

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Change type |
|---|---|---|---|
| 6 Appendix A: Product Behavior | Added Exchange 15 Technical Preview and Outlook 15 Technical Preview to the list of applicable product versions. | Y | Content updated. |

# 8 Index