[MS-OXOPOST]: Post Object Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights**. This protocol 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 protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.

- **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 protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp/default.mspx). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@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.

Preliminary Documentation. This documentation is preliminary documentation for these protocols. Since the documentation may change between this preliminary version and the final version, 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.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and networking programming art, and assumes that the reader is either familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for a Licensee to develop an implementation. Licensees who have access to Microsoft programming tools and environments are free to take advantage of them.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability

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.

1.1 Glossary

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

folder object
message object
property
property ID
recipient
search key

The following terms are specific to this document:

post object: A **message object** that represents an entry in a discussion thread stored in a messaging store that adheres to the specifications in 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

[MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol Specification", April 2008.

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

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

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

[MS-OXOABK] Microsoft Corporation, "Address Book Object Protocol Specification", April 2008.

[MS-OXOMSG] Microsoft Corporation, "E-mail Object Protocol Specification", April 2008.

[MS-OXPROPS] Microsoft Corporation, "Office Exchange Protocols Master Property List Specification", 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

None.

1.3 Protocol Overview (Synopsis)

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 defined 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.

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

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].

2.2.2.1 PidTagConversationIndex

Type: PtypBinary.

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, following the specification in [MS-OXOMSG].

2.2.2.2 PidTagConversationTopic

Type: PtypString.

Contains an unchanging copy of the original subject. This value MUST be set to the same value as **PidTagNormalizedSubject** on a new post object when it is first saved. When creating a post object as a reply, **PidTagConversationTopic** on the new post object MUST be copied from the original post object.

2.2.2.3 PidTagIconIndex

Type: PtypInteger32.

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

Type: PtypString8, case-insensitive.

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**
- **PidTagSenderEntryId**
- **PidTagSenderName**
- **PidTagSenderSearchKey**
- **PidTagSentRepresentingAddressType**
- **PidTagSentRepresentingEntryId**
- **PidTagSentRepresentingName**
- **PidTagSentRepresentingSearchKey**

2.2.2.6 Recipients

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

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 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 their external behavior is consistent with that described in this document.

3.1.1.1 Post Items

A post object extends the **message object** as defined 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 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 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** as the original, and an incremented **PidTagConversationIndex** (see [MS-OXOMSG] for details).

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 Item

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. See [MS-OXCPRPT] and [MS-OXCMSG] for details on ROPs.

To create a post 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 post object, the client uses **RopSetProperties** to transmit his data to the server.

Property	**Property ID**	**Property Type**	**Value**
PidTagIconIndex	0x1080	0x0003 (PtypInteger32)	0x00000001
PidTagMessageClass	0x001a	0x001e (PtypString8)	IPM.Post
PidTagNormalizedSubject	0x0e1d	0x001f (PtypString)	Grocery List
PidTagSubjectPrefix	0x003d	0x001f (PtypString)	(null)
PidTagConversationTopic	0x0070	0x001f (PtypString)	Grocery List
PidTagConversationIndex	0x0071	0x0102 (PtypBinary)	See Note 1, below.
PidTagHtml	0x1013	0x0102 (PtypBinary)	See Note 2, below.
PidTagSenderName	0x0c1a	0x001f (PtypString)	"Joe Healy"
PidTagSenderAddrType	0x0c1e	0x001f (PtypString)	"EX"
PidTagSenderEntryid	0x0c19	0x0102 (PtypBinary)	See Note 3, below.
PidTagSenderSearchKey	0x0c1d	0x0102 (PtypBinary)	See Note 4, below.
PidTagSentRepresentingName	0x0042	0x001f (PtypString)	"Joe Healy"
PidTagSentRepresentingAddrtype	0x0064	0x001f (PtypString)	"EX"
PidTagSentRepresentingEntryId	0x0041	0x0102 (PtypBinary)	See Note 3, below.
PidTagSentRepresentingSearchKey	0x003b	0x0102 (PtypBinary)	See Note 4, below.

Note 1 **PidTagConversationIndex** is set with a depth of one according to [MS-OXCMSG], 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
```

Note 2 **PidTagHtml** is set to the 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>
```

Note 3 **PidTagSenderEntryid** and **PidTagSentRepresentingEntryid** 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 defined in [MS-OXOABK].

Note 4 **PidTagSenderSearchKey** and **PidTagSentRepresentingSearchKey** are identical, because Joe isn't posting this on behalf of another user. The contents of these properties are Joe's **search key** as defined in [MS-OXOMSG].

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

The values of some properties will change during the execution of **RopSaveChangesMessage**, but the properties specified in [MS-OXOPOST] 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 specified in [MS-OXCMSG] and [MS-OXCPRPT].

5.2 Index of Security Parameters

None.

6 Appendix A: Office/Exchange Behavior

The information in this specification is applicable to the following versions of Office/Exchange:

- Microsoft Office 2003 with Service Pack 3 applied
- Microsoft Exchange 2003 with Service Pack 2 applied
- Microsoft Office 2007 with Service Pack 1 applied

- Microsoft Exchange 2007 with Service Pack 1 applied

<1> Section 2.2: “Microsoft Office Outlook 2003” and “Microsoft Office Outlook 2007” sometimes set the following properties regardless of user input; their values have no meaning in the context of this protocol.

PidLidAgingDontAgeMe, **PidLidCurrentVersion**, **PidLidCurrentVersionName**, **PidLidPrivate**, **PidLidSideEffect**, **PidTagAlternateRecipientAllowed**, **PidTagClientSubmitTime**, **PidTagDeleteAfterSubmit**, **PidTagImportance**, **PidTagMessageDeliveryTime**, **PidTagPriority**, **PidTagReadReceiptRequested**, **PidTagSensitivity**, **PidLidReminderDelta**, **PidLidReminderSet**, **PidLidReminderNextTime**, **PidLidTaskMode, PidTagInternetReferences**

<2> Section 2.2: “Microsoft Office Outlook 2007” sometimes 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**

Index