[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'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.
- **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.

Copyright © 2010 Microsoft Corporation.

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.

Copyright © 2010 Microsoft Corporation.

Table of Contents

1 Introduction 5
1.1 Glossary 5
1.2 References 5
1.2.1 Normative References 5
1.2.2 Informative References 6
1.3 Overview 6
1.4 Relationship to Other Protocols 6
1.5 Prerequisites/Preconditions 6
1.6 Applicability Statement 6
1.7 Versioning and Capability Negotiation 6
1.8 Vendor-Extensible Fields 6
1.9 Standards Assignments 6

2 Messages 7
2.1 Transport 7
2.2 Message Syntax 7
2.2.1 Post Object Properties 7
2.2.2 Additional Property Constraints 7
2.2.2.1 PidTagConversationIndex 7
2.2.2.2 PidTagConversationTopic 7
2.2.2.3 PidTagIconIndex 8
2.2.2.4 PidTagMessageClass 8
2.2.2.5 Sender properties 8
2.2.2.6 Recipients 8

3 Protocol Details 9
3.1 Common Details 9
3.1.1 Abstract Data Model 9
3.1.1.1 Post Objects 9
3.1.2 Timers 9
3.1.3 Initialization 9
3.1.4 Higher-Layer Triggered Events 9
3.1.4.1 Creation of a Post Object 9
3.1.4.2 Modification of a Post Object 9
3.1.4.3 Deletion of a Post Object 9
3.1.4.4 Reply to Folder 9
3.1.5 Message Processing Events and Sequencing Rules 10
3.1.6 Timer Events 10
3.1.7 Other Local Events 10

4 Protocol Examples 11
4.1 Sample Post Object 11

5 Security 13
5.1 Security Considerations for Implementers 13
5.2 Index of Security Parameters 13

6 Appendix A: Product Behavior 14

7 Change Tracking 15

Copyright © 2010 Microsoft Corporation.

8 Index 17

Copyright © 2010 Microsoft Corporation.

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]:

address book
conversation
entry ID
Folder object
handle
HTML
Message object
property (1)
property ID
property type
recipient
remote operation (ROP)
store

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

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-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, "Exchange Server 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.

Copyright © 2010 Microsoft Corporation.

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

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

1.2.2 Informative References

None.

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 object. 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.

Copyright © 2010 Microsoft Corporation.

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 2.12.1:

- 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

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 as specified 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

Copyright © 2010 Microsoft Corporation.

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: **PtypString**, 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**.

Copyright © 2010 Microsoft Corporation.

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 as the original, and an incremented PidTagConversationIndex. For more details, see [MS-OXOMSG].

Copyright © 2010 Microsoft Corporation.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

Copyright © 2010 Microsoft Corporation.

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 **ROPs**, see [MS-OXCPRPT] and [MS-OXCMSG].

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	0x001f (PtypString)	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
PidTagSenderAddressType	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
PidTagSentRepresentingAddressType	0x0064	0x001f	EX

Copyright © 2010 Microsoft Corporation.

Property	Property ID	Property type	Value
		(PtypString)	
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 1 according to [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
```

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 specified 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 specified in [MS-OXOMSG], and are used as Joe's search key, PidTagSearchKey.

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.

Copyright © 2010 Microsoft Corporation.

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.

Copyright © 2010 Microsoft Corporation.

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, PidTagInternetReferences

<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

Copyright © 2010 Microsoft Corporation.

7 Change Tracking

This section identifies changes made to [MS-OXOPOST] protocol documentation between February 2010 and May 2010 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.

Copyright © 2010 Microsoft Corporation.

- 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
1.3 Overview	Updated the section title.	N	Content updated for template compliance.

Copyright © 2010 Microsoft Corporation.

8 Index

A

Applicability 6

C

Capability negotiation 6
Change tracking 15
Client
  overview 9

E

Examples
  overview 11

F

Fields – vendor-extensible 6

G

Glossary 5

I

Implementer – security considerations 13
Index of security parameters 13
Informative references 6
Introduction 5

M

Messages
  overview 7
Messaging
  transport 7

N

Normative references 5

O

Overview 6

P

Parameters – security index 13
Preconditions 6
Prerequisites 6
Product behavior 14

R

References
  informative 6
  normative 5
Relationship to other protocols 6

S

Security
  implementer considerations 13
  overview 13
  parameter index 13
Standards Assignments 6

T

Tracking changes 15
Transport 7

V

Vendor-extensible fields 6
Versioning 6

Copyright © 2010 Microsoft Corporation.