[MS-OXOJRNL]:
Journal 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.

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		Updated references to reflect date of initial release.
09/03/2008	1.02		Updated references.
12/03/2008	1.03		Revised and edited technical content.
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.

Table of Contents

1 Introduction

This document specifies the **Journal object** protocol, which defines **properties** of an object that models an entry in a **journal** or log.

1.1 Glossary

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

Attachment object
contact
Coordinated Universal Time (UTC)
EntryID
Folder object
GUID
handle
journal
Journal object
Message object
metafile
named property
property
property ID
recipient
remote operation (ROP)
restriction
Rich Text Format (RTF)
special folder
store
stream (1)
Task request

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

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", June 2008.

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

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

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", June 2008.

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

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

[MS-OXRTFCP] Microsoft Corporation, "Rich Text Format (RTF) Compression Protocol Specification", June 2008.

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

1.2.2 Informative References

None.

1.3 Protocol Overview

The Journal Object protocol allows the representation of journal entries in a messaging **store**. The Journal Object protocol extends the Message and Attachment Object protocol in that it defines new properties and adds **restrictions** to the properties that are defined in [MS-OXCMSG].

A Journal object represents a journal entry. A Journal object is characterized by the name of the activity, the duration, and any **contacts** or businesses that are associated with the activity, and is stored in a **Folder object**. This document specifies the properties that are unique to Journal objects and how such Journal objects are created, stored, and manipulated.

1.4 Relationship to Other Protocols

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

1.5 Prerequisites/Preconditions

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

2.2 Message Syntax

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

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

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

2.2.1 Journal Object Properties

2.2.1.1 PidLidLogType

Type: **PtypString**

Briefly describes the activity that is being recorded.

2.2.1.2 PidLidLogTypeDesc

Type: **PtypString**

Describes the activity that is being recorded.

2.2.1.3 PidLidLogStart

Type: **PtypTime**, in **Coordinated Universal Time (UTC)**

The time at which the activity began; MUST be equal to PidLidCommonStart, as specified in [MS-OXCMSG].

2.2.1.4 PidLidLogEnd

Type: **PtypTime**, in UTC

The time at which the activity ended; MUST be equal to PidLidCommonEnd, and therefore greater than or equal to PidLidLogStart, as specified in [MS-OXCMSG].

2.2.1.5 PidLidLogDuration

Type: **PtypInteger32**, signed

The duration in minutes of the activity; MUST be the difference between PidLidLogEnd and PidLidLogStart.

2.2.1.6 PidLidLogFlags

Type: **PtypInteger32**

A bit field that contains metadata about the Journal object; MUST be either zero or the following value.

Value	Meaning
0x00000000	This Journal object has no journal-associated **attachment** (section 2.2.2.5).
0x40000000	This Journal object has a journal-associated attachment (section 2.2.2.5).

2.2.1.7 PidLidLogDocumentPrinted

Type: **PtypBoolean**

Indicates whether the document was printed during journaling.

2.2.1.8 PidLidLogDocumentSaved

Type: **PtypBoolean**

Indicates whether the document was saved during journaling.

2.2.1.9 PidLidLogDocumentRouted

Type: **PtypBoolean**

Indicates whether the document was sent to a routing **recipient** during journaling.

2.2.1.10 PidLidLogDocumentPosted

Type: **PtypBoolean**

Indicates whether the document was sent by e-mail or posted to a server **folder** during journaling.

2.2.2 Additional Property Constraints

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

2.2.2.1 PidTagMessageClass

Type: **PtypString**

Specifies the type of the Message item; MUST be "IPM.Activity" or begin with "IPM.Activity", in addition to **meeting** the criteria specified in [MS-OXCMSG].

2.2.2.2 Best Body Properties

The main text of the Journal object; MUST be stored in PidTagRtfCompressed, as specified in [MS-OXRTFCP].

2.2.2.3 PidTagIconIndex

Type: **PtypInteger32**

Specifies which icon is to be used by a user interface when displaying a group of Journal objects; MUST be one of the values listed in the following table.

Value	Meaning
0x00000601	**Conversation**
0x00000612	Document
0x00000602	E-mail message
0x00000609	Fax
0x0000060C	Letter
0x00000613	Meeting
0x00000614	Meeting cancellation
0x00000603	**Meeting request**
0x00000604	Meeting response
0x00000610	Microsoft Office Access
0x0000060E	Microsoft Office Excel
0x0000060F	Microsoft Office PowerPoint
0x0000060D	Microsoft Office Word
0x00000608	Note
0x0000060A	Phone call
0x00000615	Remote session
0x0000060B	Task
0x00000606	**Task request**
0x00000607	Task response
0x00000003	Other

2.2.2.4 Recipients

A Journal object MUST NOT have recipients.

2.2.2.5 Journal-Associated Attachments

A journal-associated attachment links a Journal object with another object, such as a document. A journal-associated attachment follows the specifications for structured storage Attachment objects in [MS-OXCMSG], except that certain properties on the Attachment object MUST be set as listed in the following table.

Property	Value
PidTagAttachmentLinkId	0x00000004
PidTagAttachMethod	0x00000006
PidTagRenderingPosition	0xFFFFFFFF
PidTagAttachmentFlags	0x00000000
PidTagAttachmentHidden	0x00
PidTagAccess	0x00000002

The contents of the structured storage are written to PidTagAttachDataBinary. The structured storage contains eight **streams**, the names and contents of which are detailed in the following table.

Name	Contents
IOlePres000	A **metafile** that contains the icon to be used when rendering the attachment.
\3MailStream*	Binary contents: 04 00 00 00 00 00 00 00 00 00 00 00
MailMsgAttFld	The **EntryID** of the folder of the linked **Message object**.
MailMsgAttMdb	The EntryID of the store of the linked Message object.
MailMsgAttMsg	The EntryID of the linked Message object; required only if **MailMsgAttSrchKey** is empty.
MailMsgAttSrchFld	The object EntryID of the Sent Items **special folder** [MS-OXOSFLD] of the linked Message object.
MailMsgAttSrchKey	PidTagSearchKey, as specified in [MS-OXCMSG], of the linked Message object; required only if **MailMsgAttMsg** is empty.
MailMsgAttSubject	PidTagSubject, as specified in [MS-OXPROPS], of the linked Message object.

* The \3 in \3MailStream represents the byte 0x03.

A Journal object MUST NOT have more than one journal-associated attachment.

3 Protocol Details

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

3.1 Common Details

The client and server roles are to create and operate on electronic journal entries, and otherwise operate in their roles, as specified in [MS-OXCMSG].

3.1.1 Abstract Data Model

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

3.1.1.1 Journal Objects

A Journal object extends the Message object, as defined in [MS-OXCMSG].

3.1.1.2 Journal Object Folders

A Journal object is created in the journal special Folder, as defined in [MS-OXOSFLD], unless the end user or user agent explicitly specifies another Folder Object.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Creation of a Journal Object

To create a Journal 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 Journal Object

When modifying a Journal object, the client or server creates 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 Journal Object

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

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Journal Object for a Telephone Call Example

Joe creates a Journal object for a telephone call, records the start and end times, puts notes in the body, and links a contact and company with it. The following is a description of what a client might do to accomplish Joe's intentions and the responses a server might return. For information about **remote operations (ROPs)**, see [MS-OXCPRPT] and [MS-OXCMSG].

Before manipulating Journal objects, the client has to ask the server to perform a mapping from **named properties** to **property IDs**, by using RopGetPropertyIDsFromNames.

Property	Property set GUID	NameID
PidLidCommonStart	{00062008-0000-0000-C000-000000000046}	0x8516
PidLidCommonEnd	{00062008-0000-0000-C000-000000000046}	0x8517
PidLidCompanies	{00062008-0000-0000-C000-000000000046}	0x8539
PidLidContacts	{00062008-0000-0000-C000-000000000046}	0x853A
PidLidContactLinkName	{00062008-0000-0000-C000-000000000046}	0x8586
PidLidContactLinkEntry	{00062008-0000-0000-C000-000000000046}	0x8585
PidLidContactLinkSearchKey	{00062008-0000-0000-C000-000000000046}	0x8584
PidLidLogTypeDesc	{0006200A-0000-0000-C000-000000000046}	0x8712
PidLidLogType	{0006200A-0000-0000-C000-000000000046}	0x8700
PidLidLogStart	{0006200A-0000-0000-C000-000000000046}	0x8706
PidLidLogEnd	{0006200A-0000-0000-C000-000000000046}	0x8708
PidLidLogDuration	{0006200A-0000-0000-C000-000000000046}	0x8707
PidLidLogFlags	{0006200A-0000-0000-C000-000000000046}	0x870C
PidLidLogDocumentPrinted	{0006200A-0000-0000-C000-000000000046}	0x870E
PidLidLogDocumentSaved	{0006200A-0000-0000-C000-000000000046}	0x870F
PidLidLogDocumentRouted	{0006200A-0000-0000-C000-000000000046}	0x8710
PidLidLogDocumentPosted	{0006200A-0000-0000-C000-000000000046}	0x8711

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

Property	Property ID
PidLidCommonStart	0x81bd
PidLidCommonEnd	0x81bc
PidLidCompanies	0x800c

Property	Property ID
PidLidContacts	0x8019
PidLidContactLinkName	0x802b
PidLidContactLinkEntry	0x82f6
PidLidContactLinkSearchKey	0x82f7
PidLidLogTypeDesc	0x8230
PidLidLogType	0x801a
PidLidLogStart	0x8233
PidLidLogEnd	0x8234
PidLidLogDuration	0x8235
PidLidLogFlags	0x8236
PidLidLogDocumentPrinted	0x8238
PidLidLogDocumentSaved	0x8239
PidLidLogDocumentRouted	0x823a
PidLidLogDocumentPosted	0x823b

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

Property	Property ID	Data type	Value
PidLidCommonStart	0x81bd	0x0040 (PtypTime)	2008/02/20 23:02:00.000
PidLidCommonEnd	0x81bc	0x0040 (PtypTime)	2008/02/20 23:12:00.000
PidLidCompanies	0x800c	0x101f (PtypMultipleString)	[1 entry] "Contoso Pharmaceuticals"
PidLidContacts	0x8019	0x101f (PtypMultipleString)	[1 entry] "Adam Barr"
PidLidContactLinkName	0x802b	0x001f (PtypString)	"Adam Barr"
PidLidContactLinkEntry	0x82f6	0x0102 (PtypBinary)	See Note 1.
PidLidContactLinkSearchKey	0x82f7	0x0102 (PtypBinary)	See Note 2.
PidLidLogTypeDesc	0x8230	0x001f (PtypString)	"Phone call"
PidLidLogType	0x801a	0x001f (PtypString)	"Phone call"

Property	Property ID	Data type	Value
PidLidLogStart	0x8233	0x0040 (PtypTime)	2008/02/20 23:02:00.000
PidLidLogEnd	0x8234	0x0040 (PtypTime)	2008/02/20 23:12:00.000
PidLidLogDuration	0x8235	0x0003 (PtypInteger32)	0x0000000A
PidLidLogFlags	0x8236	0x0003 (PtypInteger32)	0x00000000
PidLidLogDocumentPrinted	0x8238	0x000b (PtypBoolean)	0x00
PidLidLogDocumentSaved	0x8239	0x000b (PtypBoolean)	0x00
PidLidLogDocumentRouted	0x823a	0x000b (PtypBoolean)	0x00
PidLidLogDocumentPosted	0x823b	0x000b (PtypBoolean)	0x00
PidTagRtfCompressed	0x1009	0x0102 (PtypBinary)	See Note 3, below.
PidTagIconIndex	0x1080	0x0003 (PtypInteger32)	0x0000060A

Note 1: PidLidContactLinkEntry contains a representation of the contact link, as specified in [MS-OXCMSG].

Note 2: PidLidContactLinkSearchKey contains a representation of the contact link, as specified in [MS-OXCMSG].

Note 3: PidTagRtfCompressed contains the compressed **Rich Text Format (RTF)** representation of the body, as specified in [MS-OXRTFCP].

When Joe is ready to save his changes, the client uses RopSaveChangesMessage to commit the properties on the server, and then RopRelease to release the Journal 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 Journal object protocol. General security considerations pertaining to the underlying transport apply, as specified in [MS-OXCMSG] and [MS-OXCPRPT].

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

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

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

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

Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that product does not follow the prescription.

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

<2> Section 2.2: Outlook 2007 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

7 Change Tracking

This section identifies changes made to [MS-OXOJRNL] protocol documentation between November 2009 and February 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.

- 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
1.1 Glossary	54364 Clarified that the protocol's definition of "stream" is "stream (1)."	N	Content update.
2.2.1.6 PidLidLogFlags	52061 Added value 0x00000000 to the table of allowable values.	N	New content added.

8 Index