[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 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		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.
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	4.3	Minor	Clarified the meaning of the technical content.
01/20/2012	5.0	Major	Significantly changed the technical content.
04/27/2012	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	5.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Journal Object Protocol is used to track activity related to a meeting, task, **contact (3)**, or application file. This protocol extends the Message and Attachment Object Protocol, which is described in [MS-OXCMSG].

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

Coordinated Universal Time (UTC)
handle

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

Attachment object
contact
EntryID
Folder object
Journal object
Mail User Agent (MUA)
Message object
metafile
named property
property ID
recipient
remote operation (ROP)
Rich Text Format (RTF)
ROP request
ROP response
special folder
storage
stream

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 Specifications documentation do not include a publishing year because links are to the latest version of the technical 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-OXOSFLD] Microsoft Corporation, "Special Folders Protocol Specification".

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

[MS-OXRTFCP] Microsoft Corporation, "Rich Text Format (RTF) Compression Algorithm".

[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".

1.3 Overview

The Journal Object Protocol allows a user to track activity on a meeting, task, or contact (3). Activity on a file such as a spreadsheet or word-processing document can also be tracked. This protocol defines a **Journal object** to represent a journal entry that contains a log of the activity. The properties of a Journal object specify the name of the activity, the duration, any contacts (3) that are associated with the activity, and other details. The Journal object is stored in a **Folder object**.

The Journal Object Protocol extends the Message and Attachment Object Protocol, described in [MS-OXCMSG], in that it defines new properties on the **Message object** and adds constraints to the existing properties of the Message object.

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 information 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, as specified in [MS-OXCMSG].

1.6 Applicability Statement

A client can use this protocol to record the user's activities on various items.

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

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 a Journal object by using the Message and Attachment Object Protocol, as specified in [MS-OXCMSG]. How a server operates on a Journal object is implementation-dependent, but the results of any such operations MUST be exposed to clients in a manner that is consistent with the Journal Object Protocol.

Unless otherwise specified, a Journal object adheres to all property constraints specified in [MS-OXPROPS] and [MS-OXCMSG].

2.2.1 Journal Object Properties

2.2.1.1 PidLidLogType Property

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

The **PidLidLogType** property ([MS-OXPROPS] section 2.168) specifies the name of the activity that is being recorded.

2.2.1.2 PidLidLogTypeDesc Property

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

The **PidLidLogTypeDesc** property ([MS-OXPROPS] section 2.169) describes the activity that is being recorded.

2.2.1.3 PidLidLogStart Property

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

The **PidLidLogStart** property ([MS-OXPROPS] section 2.167) specifies the time, in **Coordinated Universal Time (UTC)**, at which the activity began. The value of this property MUST be equal to the value of the **PidLidCommonStart** property ([MS-OXCMSG] section 2.2.1.18).

2.2.1.4 PidLidLogEnd Property

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

The **PidLidLogEnd** property ([MS-OXPROPS] section 2.165) specifies the time, in **UTC**, at which the activity ended. The value of this property MUST be equal to the value of the **PidLidCommonEnd** property ([MS-OXCMSG] section 2.2.1.19) and therefore greater than or equal to the **PidLidLogStart** property (section 2.2.1.3).

2.2.1.5 PidLidLogDuration Property

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

The **PidLidLogDuration** property ([MS-OXPROPS] section 2.164) specifies the duration, in minutes, of the activity. The value of this property is the difference between the values of the **PidLidLogEnd** (section 2.2.1.4) and **PidLidLogStart** (section 2.2.1.3) properties.

2.2.1.6 PidLidLogFlags Property

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

The **PidLidLogFlags** property ([MS-OXPROPS] section 2.166) contains bits that specify whether the Journal object has a journal-associated attachment. This property MUST be set to one of the values in the following table.

Bit value	Meaning
0x00000000	This Journal object has no journal-associated attachment (section 2.2.2.6).
0x40000000	This Journal object has a journal-associated attachment (section 2.2.2.6).

2.2.1.7 PidLidLogDocumentPrinted Property

Note *Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.*

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

The **PidLidLogDocumentPrinted** property ([MS-OXPROPS] section 2.161) indicates whether the application document was printed during the tracking of the activity on the document.<1> If printing occurred, this property is set to 0x01.

2.2.1.8 PidLidLogDocumentSaved Property

Note *Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.*

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

The **PidLidLogDocumentSaved** property ([MS-OXPROPS] section 2.163) indicates whether the application document was saved during the tracking of the activity on the document.<2> If the document was saved, this property is set to 0x01.

2.2.1.9 PidLidLogDocumentRouted Property

Note *Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.*

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

The **PidLidLogDocumentRouted** property ([MS-OXPROPS] section 2.162) indicates whether the application document was sent to a **recipient (1)** during the tracking of the activity on the document.<3> If the document was sent, this property is set to 0x01.

2.2.1.10 PidLidLogDocumentPosted Property

Note *Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.*

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

The **PidLidLogDocumentPosted** property ([MS-OXPROPS] section 2.160) indicates whether the application document was sent by e-mail or posted to a server's folder during the tracking of the activity on the document.<4> If the document was posted, this property is set to 0x01. If the document was sent by e-mail, it is set to 0x00.

2.2.2 Additional Property Constraints

This protocol places additional constraints on properties beyond what is specified in [MS-OXCMSG]. These constraints are specified in section 2.2.2.1 through section 2.2.2.6.

2.2.2.1 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 of this property MUST be "IPM.Activity" or MUST 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 the **PidTagRtfCompressed** property ([MS-OXCMSG] section 2.2.1.49.4), as specified in [MS-OXRTFCP].

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 the Journal object. The valid values for this property are specified in the following table.

Value	Meaning
0x00000601	Conversation
0x00000612	Document
0x00000602	E-mail message
0x00000609	Fax
0x0000060C	Letter
0x00000613	Meeting

Value	Meaning
0x00000614	Meeting cancellation
0x00000603	Meeting request
0x00000604	Meeting response
0x00000610	A database application file
0x0000060E	A spreadsheet application file
0x0000060F	A slide-show presentation application file
0x0000060D	A word processing application file
0x00000608	Note
0x0000060A	Phone call
0x00000615	Remote session
0x0000060B	Task
0x00000606	Task request
0x00000607	Task response
0x00000003	Other

2.2.2.4 PidLidCcompanies Property

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

The **PidLidCompanies** property ([MS-OXPROPS] section 2.64) contains a list of company names, each of which is associated with a contact (3) that is specified in the **PidLidContacts** property ([MS-OXCMSG] section 2.2.1.50.2).

2.2.2.5 Recipients

A Journal object MUST NOT have recipients (1).

2.2.2.6 Journal-Associated Attachments

A journal-associated attachment links a Journal object with another object, such as a document. A Journal object MUST NOT have more than one journal-associated attachment.

A journal-associated attachment follows the requirements specified in [MS-OXCMSG] for **Attachment objects**, except that certain properties on the Attachment object MUST be set as specified in the following table.

Property	Value
PidTagAttachmentLinkId ([MS-OXCMSG] section 2.2.2.22)	0x00000004
PidTagAttachMethod ([MS-OXCMSG] section 2.2.2.9)	0x00000006
PidTagRenderingPosition ([MS-OXCMSG] section 2.2.2.16)	0xFFFFFFFF

Property	Value
PidTagAttachmentFlags ([MS-OXCMSG] section 2.2.2.23)	0x00000000
PidTagAttachmentHidden ([MS-OXCMSG] section 2.2.2.24)	0x00
PidTagAccess ([MS-OXCPRPT] section 2.2.1.1)	0x00000002

The content of the **PidTagAttachDataBinary** property ([MS-OXCMSG] section 2.2.2.7) is formatted as a structured **storage** that comprises eight **streams (1)**, the names and contents of which are detailed in the following table.

Stream 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 the **MailMsgAttSrchKey** stream is empty.
MailMsgAttSrchFld	The object EntryID of the Sent Items **special folder** of the linked Message object. For details about special folders, see [MS-OXOSFLD].
MailMsgAttSrchKey	The value of the **PidTagSearchKey** property ([MS-OXCPRPT] section 2.2.1.9), as specified in [MS-OXCMSG], of the linked Message object; required only if the **MailMsgAttMsg** stream is empty.
MailMsgAttSubject	The value of the **PidTagSubject** property ([MS-OXCMSG] section 2.2.1.46) of the linked Message object.

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

3 Protocol Details

3.1 Client Details

The client creates and manipulates a Journal 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 the external behavior of the implementation is consistent with the behavior 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 Journal object is an extension of the Message object.
- A Journal object is created in the Journal special folder unless the end user or **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 Journal Object

When the user creates a journal entry, 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, and saves the resulting Journal object, as specified in [MS-OXCMSG] section 3.1.4.3.

3.1.4.2 Modifying a Journal Object

When the user modifies a journal entry, the client opens the Journal object as specified in [MS-OXCMSG] section 3.1.4.1, modifies any of the properties in accordance with the requirements in section 2, and saves the Journal object as specified in [MS-OXCMSG] section 3.1.4.3.

3.1.4.3 Deleting a Journal Object

When a user deletes a journal entry, the client deletes the Journal 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 Journal 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 the external behavior of the implementation is consistent with the behavior 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 Journal object is an extension of the Message object.
- A Journal object is created in the Journal special folder unless the end-user or 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 Journal Object for a Telephone Call Example

A user creates a journal entry to track a telephone call. The user records the start and end times, puts notes in the body, and links a contact (3) and company. 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 a Journal object, the client asks the server to map **named properties** to **property IDs**, by sending a **RopGetPropertyIDsFromNames ROP request** ([MS-OXCROPS] section 2.2.8.1).

Property	Property set GUID	LID
PidLidCommonStart ([MS-OXCMSG] section 2.2.1.18)	{00062008-0000-0000-C000-000000000046}	0x00008516
PidLidCommonEnd ([MS-OXCMSG] section 2.2.1.19)	{00062008-0000-0000-C000-000000000046}	0x00008517
PidLidCompanies (section 2.2.2.4)	{00062008-0000-0000-C000-000000000046}	0x00008539
PidLidContacts ([MS-OXCMSG] section 2.2.1.50.2)	{00062008-0000-0000-C000-000000000046}	0x0000853A
PidLidContactLinkName ([MS-OXCMSG] section 2.2.1.50.3)	{00062008-0000-0000-C000-000000000046}	0x00008586
PidLidContactLinkEntry ([MS-OXCMSG] section 2.2.1.50.1)	{00062008-0000-0000-C000-000000000046}	0x00008585
PidLidContactLinkSearchKey ([MS-OXCMSG] section 2.2.1.50.4)	{00062008-0000-0000-C000-000000000046}	0x00008584
PidLidLogTypeDesc (section 2.2.1.2)	{0006200A-0000-0000-C000-000000000046}	0x00008712
PidLidLogType (section 2.2.1.1)	{0006200A-0000-0000-C000-000000000046}	0x00008700
PidLidLogStart (section 2.2.1.3)	{0006200A-0000-0000-C000-000000000046}	0x00008706
PidLidLogEnd (section 2.2.1.4)	{0006200A-0000-0000-C000-000000000046}	0x00008708
PidLidLogDuration (section 2.2.1.5)	{0006200A-0000-0000-C000-000000000046}	0x00008707
PidLidLogFlags (section 2.2.1.6)	{0006200A-0000-0000-C000-000000000046}	0x0000870C
PidLidLogDocumentPrinted (section 2.2.1.7)	{0006200A-0000-0000-C000-000000000046}	0x0000870E
PidLidLogDocumentSaved (section 2.2.1.8)	{0006200A-0000-0000-C000-000000000046}	0x0000870F

Property	Property set GUID	LID
PidLidLogDocumentRouted (section 2.2.1.9)	{0006200A-0000-0000-C000-000000000046}	0x00008710
PidLidLogDocumentPosted (section 2.2.1.10)	{0006200A-0000-0000-C000-000000000046}	0x00008711

The server sends a **RopGetPropertyIDsFromNames** **ROP response** with the following property IDs, which will be used in the example that follows. (The actual property IDs are at the discretion of the server.)

Property	Property ID
PidLidCommonStart	0X81BD
PidLidCommonEnd	0X81BC
PidLidCompanies	0X800C
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 the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2). The server returns a **handle** to a Message object.

After the user has input his content for the Journal object, the client transmits the data to the server by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6). For information about the property types in the following table, see [MS-OXCDATA] section 2.11.1.

Property	Property ID	Property type	Value
PidLidCommonStart	0x81bd	0x0040 (**PtypTime**)	2008/02/20 23:02:00.000
PidLidCommonEnd	0x81bc	0x0040	2008/02/20 23:12:00.000
PidLidCompanies	0x800c	0X101F (**PtypMultipleString** ([MS-OXCDATA] section 2.11.1))	[1 entry] "Contoso Pharmaceuticals"
PidLidContacts	0x8019	0X101F	[1 entry] "Adam Barr"
PidLidContactLinkName	0x802b	0X001F (**PtypString**)	"Adam Barr"
PidLidContactLinkEntry	0x82f6	0x0102 (**PtypBinary**)	*(See note following table)
PidLidContactLinkSearchKey	0x82f7	0x0102	**(See note following table)
PidLidLogTypeDesc	0x8230	0X001F	"Phone call"
PidLidLogType	0x801a	0X001F	"Phone call"
PidLidLogStart	0x8233	0x0040	2008/02/20 23:02:00.000
PidLidLogEnd	0x8234	0x0040	2008/02/20 23:12:00.000
PidLidLogDuration	0x8235	0x0003 (**PtypInteger32**)	0x0000000A
PidLidLogFlags	0x8236	0x0003	0x00000000
PidLidLogDocumentPrinted	0x8238	0X000B (**PtypBoolean**)	0x00
PidLidLogDocumentSaved	0x8239	0X000B	0x00
PidLidLogDocumentRouted	0x823a	0X000B	0x00
PidLidLogDocumentPosted	0x823b	0X000B	0x00
PidTagRtfCompressed ([MS-OXCMSG] section 2.2.1.49.4)	0x1009	0x0102	***(See note following table)
PidTagIconIndex ([MS-OXOMSG] section 2.2.1.10)	0x1080	0x0003	0x0000060A

* The **PidLidContactLinkEntry** property contains a representation of the contact (3) link, as described in [MS-OXCMSG] section 2.2.1.50.1.

** The **PidLidContactLinkSearchKey** property contains a representation of the contact (3) link, as described in [MS-OXCMSG] section 2.2.1.50.4.

*** The **PidTagRtfCompressed** property ([MS-OXCMSG] section 2.2.1.49.4) contains the compressed **Rich Text Format (RTF)** representation of the body, as described in [MS-OXRTFCP].

When the user saves the 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 Journal object by using the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3).

The values of some properties will change during the processing 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 Journal Object Protocol. General security considerations pertaining to the underlying transport apply, as described in [MS-OXCMSG].

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 2013 Preview
- Microsoft® Office Outlook® 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Outlook® 2010
- Microsoft® Outlook® 2013 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 2.2.1.7: Outlook 2013 Preview does not support the **PidLidLogDocumentPrinted** property (section 2.2.1.7).

<2> Section 2.2.1.8: Outlook 2013 Preview does not support the **PidLidLogDocumentSaved** property (section 2.2.1.8).

<3> Section 2.2.1.9: Outlook 2013 Preview does not support the **PidLidLogDocumentRouted** property (section 2.2.1.9).

<4> Section 2.2.1.10: Outlook 2013 Preview does not support the **PidLidLogDocumentPosted** property (section 2.2.1.10).

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

R

S

T

V