

[MS-OXOSFLD]: Special Folders 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.msp>). 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		Revised and edited technical content.
12/03/2008	1.03		Revised and edited technical content.
03/04/2009	1.04		Revised and edited technical content.
04/10/2009	2.0		Updated technical content and applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	3.1	Minor	Updated the technical content.
02/10/2010	3.2	Minor	Updated the technical content.
05/05/2010	3.3	Minor	Updated the technical content.
08/04/2010	3.4	Minor	Clarified the meaning of the technical content.
11/03/2010	3.4	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	3.4	No change	No changes to the meaning, language, and formatting of the technical content.
08/05/2011	4.0	Major	Significantly changed the technical content.
10/07/2011	4.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 Overview	7
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	8
1.7 Versioning and Capability Negotiation	8
1.8 Vendor-Extensible Fields	8
1.9 Standards Assignments	8
2 Messages	9
2.1 Transport	9
2.2 Message Syntax	9
2.2.1 List of Special Folders	9
2.2.2 Identifiers for Special Folders	11
2.2.3 Binary Identification Properties	11
2.2.4 PidTagAdditionalRenEntryIds Property	12
2.2.5 PidTagAdditionalRenEntryIdsEx Property	12
2.2.5.1 PersistData Structure	12
2.2.5.2 PersistElement Structure	13
2.2.6 Inbox Identification	14
2.2.7 PidTagContainerClass Property	14
3 Protocol Details	15
3.1 Common Details	15
3.1.1 Abstract Data Model	15
3.1.1.1 Folder Hierarchy	15
3.1.1.2 Search Criteria for Search Special Folders	16
3.1.2 Timers	18
3.1.3 Initialization	18
3.1.4 Higher-Layer Triggered Events	18
3.1.4.1 Folder Creation	18
3.1.4.1.1 Creating the Reminders Folder	18
3.1.4.1.2 Creating the To-Do Folder	20
3.1.4.1.3 Creating the Tracked Mail Processing Folder	22
3.1.4.1.4 Creating Other Special Folders	23
3.1.5 Message Processing Events and Sequencing Rules	24
3.1.6 Timer Events	24
3.1.7 Other Local Events	24
4 Protocol Examples	25
4.1 Opening a Special Folder	25
4.1.1 Client Request for Opening a Special Folder	25
4.1.2 Server Response for Opening a Special Folder	27
4.2 Creating a Special Folder	28
4.2.1 Client Request for Creating a Special Folder	28
4.2.2 Server Response for Creating a Special Folder	30

5 Security	33
5.1 Security Considerations for Implementers.....	33
5.2 Index of Security Parameters	33
6 Appendix A: Product Behavior	34
7 Change Tracking	35
8 Index	36

1 Introduction

The Special Folders Protocol is used to create and open a folder that is part of the set of default folders that store particular types of data within a **mailbox**. The Special Folders Protocol extends the Folder Object Protocol, which is described in [\[MS-OXCFOLD\]](#).

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

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

little-endian
remote procedure call (RPC)

The following terms are defined in [\[MS-OXGLOS\]](#):

calendar
Calendar folder
Calendar object
client-side rule
Common Views folder
contact
Contact object
Contacts folder
Container class
conversation action
Deferred Action Message (DAM)
Deferred Error Message (DEM)
delegate
Deleted Items folder
Drafts folder
entry ID
folder associated information (FAI)
Folder object
Inbox folder
Journal object
Junk E-mail folder
Junk E-mail rule
locale
mailbox
message class
Message object
Note object
Outbox folder
Really Simple Syndication (RSS)
Receive folder
remote operation (ROP)
restriction
Root folder
ROP request
ROP response
search criteria

search folder
Sent Items folder
Server object
Server object handle
special folder
store
Store object
Task object

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-DTYP] Microsoft Corporation, "[Windows Data Types](#)".

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

[MS-OXCSTOR] Microsoft Corporation, "[Store Object Protocol Specification](#)".

[MS-OXOCAL] Microsoft Corporation, "[Appointment and Meeting Object Protocol Specification](#)".

[MS-OXODLGT] Microsoft Corporation, "[Delegate Access Configuration Protocol Specification](#)".

[MS-OXOFLAG] Microsoft Corporation, "[Informational Flagging Protocol Specification](#)".

[MS-OXOPFFB] Microsoft Corporation, "[Public Folder-Based Free/Busy Protocol Specification](#)".

[MS-OXORMDR] Microsoft Corporation, "[Reminder Settings Protocol Specification](#)".

[MS-OXOTASK] Microsoft Corporation, "[Task-Related Objects Protocol Specification](#)".

[MS-OXPFOAB] Microsoft Corporation, "[Offline Address Book \(OAB\) Public Folder Retrieval Protocol Specification](#)".

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

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

1.2.2 Informative References

- [MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".
- [MS-OXCPRPT] Microsoft Corporation, "[Property and Stream Object Protocol Specification](#)".
- [MS-OXCRPC] Microsoft Corporation, "[Wire Format Protocol Specification](#)".
- [MS-OXCSPAM] Microsoft Corporation, "[Spam Confidence Level Protocol Specification](#)".
- [MS-OXCSYNC] Microsoft Corporation, "[Mailbox Synchronization Protocol Specification](#)".
- [MS-OXGLOS] Microsoft Corporation, "[Exchange Server Protocols Master Glossary](#)".
- [MS-OXOCFG] Microsoft Corporation, "[Configuration Information Protocol Specification](#)".
- [MS-OXOCNTC] Microsoft Corporation, "[Contact Object Protocol Specification](#)".
- [MS-OXOJRNL] Microsoft Corporation, "[Journal Object Protocol Specification](#)".
- [MS-OXOMSG] Microsoft Corporation, "[E-Mail Object Protocol Specification](#)".
- [MS-OXONOTE] Microsoft Corporation, "[Note Object Protocol Specification](#)".
- [MS-OXORSS] Microsoft Corporation, "[RSS Object Protocol Specification](#)".
- [MS-OXORULE] Microsoft Corporation, "[E-Mail Rules Protocol Specification](#)".
- [MS-OXOSRCH] Microsoft Corporation, "[Search Folder List Configuration Protocol Specification](#)".

1.3 Overview

The Special Folders Protocol is used to create and open a folder that is part of the set of default folders within a mailbox. Each of these **special folders** stores a particular type of **Message object**. For example, there is a special folder for e-mail messages, another for appointments, and another for **contacts (3)**. A complete list of the special folders, along with descriptions and the type of data contained in each special folder, is provided in section [2.2.1](#).

Each special folder has an identifier. The identifier ensures that the same special folder will continue to be used to store a particular type of Message object after the folder is created.

The Special Folders Protocol extends the Folder Object Protocol by defining additional constraints and adaptations for creating and accessing special folders. For information about the Folder Object Protocol, see [\[MS-OXCFOLD\]](#).

1.4 Relationship to Other Protocols

The Special Folders Protocol extends the Folder Object Protocol, and, therefore, has the same dependencies. For information about the Folder Object Protocol, see [\[MS-OXCFOLD\]](#).

1.5 Prerequisites/Preconditions

The Special Folders Protocol has the same prerequisites and preconditions as the Folder Object Protocol. For information about the Folder Object Protocol, see [\[MS-OXCFOLD\]](#).

1.6 Applicability Statement

A client can use the Special Folders Protocol to organize and store well-known object types in a user's mailbox.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Special Folders Protocol uses the same underlying transport as that used by the Folder Object Protocol, as specified in [\[MS-OXCFOLD\]](#).

2.2 Message Syntax

2.2.1 List of Special Folders

The set of folders that are special folders, along with the **Container class** for each folder where applicable and references for further information, are listed in the following table.

Special folder name	Description	Container class	More information
Root	The store hierarchy's top-level folder, which contains all other Folder objects in that store.	None	[MS-OXCSTOR]
Finder	Contains the default search folders (2) .	None	[MS-OXOSRCH]
Freebusy Data	Contains the free/busy data of the mailbox's owner.	None	[MS-OXOPFFB]
Top of Personal Folders	The top folder of the interpersonal message hierarchy, which contains user data folders, including most special folders (2), such as the Inbox folder .	None	[MS-OXCSTOR]
Deleted Items	The default location for objects that have been deleted.	"IPF.Note"	[MS-OXOMSG]
Outbox	Outgoing e-mail Message objects are placed in this folder when the Message object is sent.	"IPF.Note"	[MS-OXOMSG]
Sent Items	The default location in which copies of e-mail Message objects are placed after they have been submitted (sent).	"IPF.Note"	[MS-OXOMSG]
Inbox	The default location for incoming (received) e-mail Message objects.	"IPF.Note"	[MS-OXOMSG]
Common Views	Contains the data for default views that are standard for the message store and that can be used by any user of a client accessing the message store.	None	None
Personal Views	Contains the data for views defined by a particular user.	None	None
Deferred Action	Contains any Deferred Action Message (DAM) or Deferred Error	None	[MS-

Special folder name	Description	Container class	More information
	Message (DEM) that results from the execution of client-side rules .		OXORULE
Calendar	Contains Calendar objects , such as appointments.	"IPF.Appointment"	[MS- OXOCAL]
Contacts	Contains Contact objects .	"IPF.Contact"	[MS- OXOCNTC]
Journal	Contains Journal objects .	"IPF.Journal"	[MS- OXOJRNL]
Notes	Contains Note objects .	"IPF.StickyNote"	[MS- OXONOTE]
Tasks	Contains Task objects .	"IPF.Task"	[MS- OXOTASK]
Reminders	Search folder (2) that supports reminder functionality.	"Outlook.Reminder"	[MS- OXORMDR]
Drafts	The default location for composed e-mail Message objects that have been saved but not sent.	"IPF.Note"	[MS- OXOMSG]
Conflicts	Contains Message objects that indicate synchronization conflicts between client and server.	"IPF.Note"	[MS- OXCSYNC]
Sync Issues	Contains folders that contain messages that indicate particular issues encountered during synchronization between client and server.	"IPF.Note"	[MS- OXCSYNC]
Local Failures	Contains messages that indicate client-side synchronization failures.	"IPF.Note"	[MS- OXCSYNC]
Server Failure	Contains messages that indicate server-side synchronization failures.	"IPF.Note"	[MS- OXCSYNC]
Junk E-mail	Default location for e-mail Message objects determined to be junk e-mail by a Junk E-mail rule .	"IPF.Note"	[MS- OXCSPPAM]
RSS Feeds	Contains Really Simple Syndication (RSS) feed messages.	"IPF.Note.OutlookHomepage"	[MS- OXORSS]
Tracked Mail Processing	Search folder (2) that contains flagged objects.	"IPF.Note"	[MS- OXOFLAG]
To-Do	Search folder (2) used for tracking Task objects.	"IPF.Task"	[MS- OXOTASK]
Conversation Action Settings	Contains folder associated information (FAI) messages that are used for persisting conversation actions .	"IPF.Configuration"	[MS- OXOCFG]

2.2.2 Identifiers for Special Folders

The value that identifies a special folder is obtained from one of the following:

- The folder IDs (FIDs) ([\[MS-OXCDATA\]](#) section 2.2.1.1) returned in a **RopLogon remote operation (ROP)** response ([\[MS-OXCROPS\]](#) section 2.2.3.1), as specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.3. These FIDs identify the following folders: **Root folder**, Finder folder, Top of Personal Folders folder, **Deleted Items folder**, **Outbox folder**, **Sent Items folder**, Inbox folder, **Common Views folder**, Personal Views folder, and Deferred Action folder.
- One of several binary properties on the Root folder or the Inbox folder. Each property contains only a single **entry ID**. For more details, see section [2.2.3](#).
- The **PidTagAdditionalRenEntryIds** property (section [2.2.4](#)) on the Inbox folder. The property contains an array of entry IDs.
- The **PidTagAdditionalRenEntryIdsEx** property (section [2.2.5](#)) on the **Store object**. The property contains binary data in its own format, allowing for multiple entry IDs.
- The **PidTagFreeBusyEntryIds** property ([\[MS-OXOPFFB\]](#) section 2.2.2.1) on the Root folder or the Inbox folder. Indexed value 3 contains the entry ID for the Freebusy Data folder.
- The FID returned by the **RopGetReceiveFolder ROP response** ([\[MS-OXCROPS\]](#) section 2.2.3.2). This FID is for the **Receive folder**.

Unless otherwise noted, the entry IDs MUST be converted to FIDs by using the **RopIdFromLongTermId** ROP ([\[MS-OXCROPS\]](#) section 2.2.3.9) before they are used with the Folder Object Protocol, as specified in [\[MS-OXCFOLD\]](#), to open a special folder.

2.2.3 Binary Identification Properties

Each binary identification property contains the identifier of a special folder. These properties are read from or written to either the Inbox folder or the Root folder. The implementation MUST use the Inbox folder when the mailbox is that of the owner, and it MUST use the Root folder when the mailbox is that of a **delegate**. For details about delegates, see [\[MS-OXODLGT\]](#).

These properties along with their descriptions, are listed in the following table. Each property is a **PtypBinary** ([\[MS-OXCADATA\]](#) section 2.11.1).

Property	Description
PidTagIpmAppointmentEntryId ([MS-OXPROPS] section 2.807)	Contains the entry ID of the Calendar folder
PidTagIpmContactEntryId ([MS-OXPROPS] section 2.808)	Contains the entry ID of the Contacts folder
PidTagIpmJournalEntryId ([MS-OXPROPS] section 2.810)	Contains the entry ID of the Journal folder
PidTagIpmNoteEntryId ([MS-OXPROPS] section 2.811)	Contains the entry ID of the Notes folder
PidTagIpmTaskEntryId ([MS-OXPROPS] section 2.812)	Contains the entry ID of the Tasks folder
PidTagRemindersOnlineEntryId ([MS-OXPROPS] section 2.964)	Contains the entry ID of the Reminders folder
PidTagIpmDraftsEntryId ([MS-OXPROPS] section 2.809)	Contains the entry ID of the Drafts

Property	Description
	folder

2.2.4 PidTagAdditionalRenEntryIds Property

Type: **PtypMultipleBinary**

The **PidTagAdditionalRenEntryIds** property ([\[MS-OXPROPS\]](#) section 2.580) of the Inbox folder specifies the identifiers of several special folders. Each identifier is an entry ID.

The index into the array contained in the **PidTagAdditionalRenEntryIds** property for each of these special folders is listed in the following table. The implementation MUST ignore and MUST preserve data at other indexes.

Index	Folder identified
0x0000	Conflicts
0x0001	Sync Issues
0x0002	Local Failures
0x0003	Server Failures
0x0004	Junk E-mail

2.2.5 PidTagAdditionalRenEntryIdsEx Property

Type: **PtypBinary**

The **PidTagAdditionalRenEntryIdsEx** property ([\[MS-OXPROPS\]](#) section 2.581) of the Store object specifies the identifiers of several special folders. If present, the value of this property is an array of **PersistData** structures (section [2.2.5.1](#)), each of which contains the entry ID of a special folder.

2.2.5.1 PersistData Structure

The fields of the **PersistData** structure are specified in the following table.

Field name	Type	Size	Description
PersistID	WORD ([MS-DTYP])	2	Type identifier value for this PersistData structure. SHOULD be one of the values specified in the following table.
DataElementsSize	WORD	2	The size, in bytes, of the DataElements field.
DataElements	Array of PersistElement structures	variable	0 (zero) or more PersistElement structures (section 2.2.5.2).

The value of the **PersistID** field SHOULD be one of those listed in the following table. If a **PersistData** structure is encountered where the value of the **PersistID** field is not known to the implementation, the implementation MUST ignore that **PersistData** structure and continue processing until either a value of 0x0000 (PERSIST_SENTINEL) in the **PersistID** field is encountered or the end of the data stream is encountered. [<1>](#)

Value name	Value	Meaning
RSF_PID_RSS_SUBSCRIPTION	0x8001	Indicates that the structure contains data for the RSS Feeds folder.
RSF_PID_SEND_AND_TRACK	0x8002	Indicates that the structure contains data for the Tracked Mail Processing folder.
RSF_PID_TODO_SEARCH	0x8004	Indicates that the structure contains data for the To-Do folder.
RSF_PID_CONV_ACTIONS	0x8006	Indicates that the structure contains data for the Conversation Action Settings folder.
RSF_PID_COMBINED_ACTIONS	0x8007	This value is reserved.
RSF_PID_SUGGESTED_CONTACTS	0x8008	Indicates that the structure contains data for the Suggested Contacts folder.
PERSIST_SENTINEL	0x0000	Indicates that the implementation MUST stop processing further PersistData structures. PERSIST_SENTINEL is optional; if it is not included, processing continues to the end of the data stream.

2.2.5.2 PersistElement Structure

The fields of the **PersistElement** structure are specified in the following table.

Field name	Type	Size	Description
ElementID	WORD ([MS-DTYP])	2	Type identifier value for this PersistElement structure. SHOULD be one of the values given in the following table.
ElementDataSize	WORD	2	The size, in bytes, of the ElementData field.
ElementData	BYTE ([MS-DTYP]) array of binary data	variable	The data corresponding to the value of the PersistID field of the PersistData structure (section 2.2.5.1).

The value of the **ElementID** field SHOULD be one of those listed in the following table. If a **PersistElement** structure is encountered where the value of the **ElementID** field is not known to the implementation, the implementation MUST ignore that **PersistElement** structure and continue processing further **PersistElement** structures until either a value of 0x0000 (ELEMENT_SENTINEL) in the **ElementID** field is encountered or the end of the data stream is encountered. The implementation MUST then continue processing additional **PersistData** structures until either a value of 0x0000 (PERSIST_SENTINEL) in the **PersistID** field is encountered or the end of the data stream is encountered.

Value name	Value	Value of the ElementDataSize field	Meaning
RSF_ELID_HEADER	0x0002	0x0004	Indicates that the ElementData field contains a value of type DWORD ([MS-DTYP]). The interpretation of this value depends on the value of the PersistID field in the current PersistData structure. For all PersistID types

Value name	Value	Value of the ElementDataSize field	Meaning
			specified in this section, this value MUST be 0 (zero).
RSF_ELID_ENTRYID	0x0001	variable	Indicates that the PersistElement structure contains the entry ID of the folder indicated by the value of the PersistID field of the PersistData structure.
ELEMENT_SENTINEL	0x0000	0x0000	Indicates that the implementation MUST stop processing further PersistElement structures for the current PersistData structure.

2.2.6 Inbox Identification

To identify the Receive folder, an implementation MUST use the **RopGetReceiveFolder** ROP ([\[MS-OXCROPS\]](#) section 2.2.3.2) of the Store Object Protocol to get the FID ([\[MS-OXCADATA\]](#) section 2.2.1.1) for the default Receive folder for the Store object. For details about the Store Object Protocol, see [\[MS-OXCSTOR\]](#).

2.2.7 PidTagContainerClass Property

Type: **PtypString**

The **PidTagContainerClass** property ([\[MS-OXPROPS\]](#) section 2.701), located on the special folder, specifies the type of Message object that the folder contains. An implementation MUST set this property to the string that specifies the correct Container class for the special folder, as specified in section [2.2.1](#).

3 Protocol Details

3.1 Common Details

Special folders can be opened or created by clients and servers. Except where noted, this section defines constraints to which clients and servers adhere when interacting with special folders. In all other respects, clients and servers operate as specified in [\[MS-OXCFOLD\]](#).

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 Folder Hierarchy

The following outline depicts a hierarchy for special folders that are within the Root folder, which is the top-level folder, in a store:

1. Deferred Action Folder
2. Finder
3. Reminders
4. Tracked Mail Processing
5. To-Do
6. Common Views
7. Personal Views
8. Top of Personal Folders
 1. Deleted Items
 2. Outbox
 3. Sent Items
 4. Inbox
 5. Calendar
 6. Contacts
 7. Journal
 8. Notes
 9. Tasks
 10. Drafts

11. Sync Issues
 1. Conflicts
 2. Local Failures
 3. Server Failures
12. Junk E-mail
13. RSS Feeds
14. Conversation Action Settings

3.1.1.2 Search Criteria for Search Special Folders

The Reminders folder, the To-Do folder, and the Tracked Mail Processing folder are search folders (2), as specified in [\[MS-OXCFOLD\]](#), and the functionality supported by these folders depends on specific **search criteria**. The search criteria are set on each search folder (2) when the folder is created. Complete details about setting the search criteria are specified in section [3.1.4.1](#), and an abstracted outline is as follows.

For the Reminders folder, include all Message objects in any folder contained within the Top of Personal Folders folder, with the following **restrictions (2)**:

- The following special folders are excluded from the search:
 - Deleted Items folder
 - **Junk E-mail folder**
 - Drafts folder
 - Outbox folder
 - Conflicts folder
 - Local Failures folder
 - Server Failures folder
 - Sync Issues folder
- The value of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) does not contain a string with the prefix "IPM.Schedule."
- The value of the **PidTagMessageFlags** property ([\[MS-OXCMSG\]](#) section 2.2.1.6) does not have the mfSubmitted flag set (that is, submitted Message objects are excluded).
- The value of the **PidLidReminderSet** property ([\[MS-OXORMDR\]](#) section 2.2.1.1) is set to the Boolean value 1, or the value of the **PidLidRecurring** property ([\[MS-OXOCAL\]](#) section 2.2.1.12) is set to the value 1.

For the To-Do folder, include all Message objects in any folder contained within the Top of Personal Folders folder, with the following restrictions (2):

- The following special folders are excluded from the search:

- Deleted Items folder
- Junk E-mail folder
- Drafts folder
- Outbox folder
- Conflicts folder
- Local Failures folder
- Server Failures folder
- Sync Issues folder
- The **message class** does not start with "IPM.Appointment" or "IPM.Activity" or "IPM.StickyNote".
- Any one of the following is true:
 - The Message object is a Task object, as specified in [\[MS-OXOTASK\]](#), and the Task object is owned and not accepted and the Task object was sent to the currently logged-on user.
 - The value of the Message object's **PidTagFollowupIcon** property ([\[MS-OXOFLAG\]](#) section 2.2.1.2) is greater than 0.
 - The value of the Message object's **PidTagToDoItemFlags** property ([\[MS-OXOFLAG\]](#) section 2.2.1.6) includes the IsToDoItem flag.
 - The Message object is an object with the complete flag set to TRUE or a completed task.

For the Tracked Mail Processing folder, include all Message objects in any folder contained within the Top of Personal Folders folder, with the following restrictions (2):

- The following special folders are excluded from the search:
 - Deleted Items folder
 - Junk E-mail folder
 - Drafts folder
 - Outbox folder
 - Conflicts folder
 - Local Failures folder
 - Server Failures folder
 - Sync Issues folder
- The **PidTagSwappedToDoStore** property ([\[MS-OXOFLAG\]](#) section 2.2.1.8) exists on the object.
- The value of the **PidTagMessageFlags** property does not include the mfUnsent or the mfSubmitted flags.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

Before an implementation tries to read or write Message objects within a special folder, the implementation MUST obtain a handle to the special folder by using the following steps:

1. The implementation MUST try to open the special folder by using the ID of the special folder and the Folder Object Protocol, as specified in [\[MS-OXCFOLD\]](#). The ID of the special folder is obtained as specified in section [2.2.2](#).
2. If the ID cannot be retrieved, or the folder cannot be opened, or the special folder does not exist within the store, the implementation MUST create the special folder as specified in section [3.1.4.1](#).

3.1.4.1 Folder Creation

The folder hierarchy specified in section [3.1.1.1](#) also specifies the dependency chain for special folder creation. Before a special folder can be created, the Store object and the Root folder MUST exist and the parent folder in this hierarchy MUST exist. A handle to the Store object that contains the folder MUST be obtained by opening the Store object, as specified in [\[MS-OXCSTOR\]](#).

To create a special folder, an implementation MUST do the following:

1. Open the parent folder of the desired special folder, as specified in the folder hierarchy in section [3.1.1.1](#), by using the Folder Object protocol.
2. Create the folder in the opened parent folder, by using the Folder Object Protocol with the special folder name for the implementation's **locale**, reusing the existing folder if one already exists by that name. The FID that is returned MUST be converted to an entry ID by using a **RopLongTermIdFromId ROP request** ([\[MS-OXCROPS\]](#) section 2.2.3.8). The resulting entry ID of the new folder is used in the remaining steps.
3. Store the entry ID at the correct location, specified in section [2.2.2](#), for the particular special folder.
4. If applicable, set the **PidTagContainerClass** property (section [2.2.7](#)) to the appropriate value.
5. Perform any folder-specific initialization, as specified in sections [3.1.4.1.1](#) through [3.1.4.1.4](#).

3.1.4.1.1 Creating the Reminders Folder

To complete creation of the Reminders folder, an implementation SHOULD [<2>](#) set the search criteria for the Reminders folder. The search criteria are composed of restriction (2) structures, as specified in [\[MS-OXCDATA\]](#) section 2.12. The search criteria are set by using the Folder Object Protocol, as specified in [\[MS-OXCFOLD\]](#), such that the Top of Personal Folders folder is the only container included in the search and the search applies a **RES_AND** restriction (2) with the following two subclauses.

First, a **RES_AND** restriction (2) with any of the following subclauses. Each of these subclauses is included in the **RES_AND** restriction (2) only if the special folder exists within the store. For example, if only the Drafts folder exists, then only the **RES_PROPERTY** subclause for the Drafts folder is included in the **RES_AND** restriction (2).

- A **RES_PROPERTY** restriction (2) with a relational operator (specified by the **RelOp** field) value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property ([\[MS-OXPFOAB\]](#) section 2.2.1.8) with the FID ([\[MS-OXCDATA\]](#) section 2.2.1.1)/message ID (MID) ([\[MS-OXCDATA\]](#) section 2.2.1.2) pair of the Deleted Items folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Junk E-mail folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Drafts folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Outbox folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Conflicts folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Local Failures folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Server Failures folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Sync Issues folder.

Second, a **RES_AND** restriction (2) with the following three subclauses:

1. A **RES_NOT** restriction (2) that contains a **RES_AND** restriction (2) with the following two subclauses:
 - A **RES_EXIST** restriction (2) that specifies the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3).
 - A **RES_CONTENT** restriction (2) with the **FuzzyLevelLow** field set to **FL_PREFIX**, comparing the value of **PidTagMessageClass** property to the string value "IPM.Schedule".
2. A **RES_BITMASK** restriction (2) with the **BitMapRelOp** field set to **BMR_EQZ** to compare the value of the **PidTagMessageFlags** property ([\[MS-OXCMSG\]](#) section 2.2.1.6) to the **ULONG** ([\[MS-DTYP\]](#)) value **mfSubmitted**.
3. A **RES_OR** restriction (2) with the following two subclauses:
 1. A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_EQ**, comparing the value of the **PidLidReminderSet** property ([\[MS-OXORMDR\]](#) section 2.2.1.1) to a Boolean value of 1.
 2. A **RES_AND** restriction (2) with the following two subclauses:
 - A **RES_EXIST** restriction (2) that specifies the **PidLidRecurring** property ([\[MS-OXOCAL\]](#) section 2.2.1.12).

- A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_EQ, comparing the value of the **PidLidRecurring** property to the Boolean value of 1.

The search SHOULD provide the client with the information when requested, include child folders, and run without content indexing.

3.1.4.1.2 Creating the To-Do Folder

To complete creation of the To-Do folder, an implementation SHOULD<3> set the search criteria for the To-Do folder. The search criteria are composed of restriction (2) structures, as specified in [\[MS-OXCDATA\]](#) section 2.12. The search criteria are set by using the Folder Object Protocol, as specified in [\[MS-OXCFOAD\]](#), such that the Top of Personal Folders folder is the only container included in the search and the search applies a **RES_AND** restriction with the following two subclauses.

First, a **RES_AND** restriction with the following three subclauses:

1. A **RES_NOT** restriction (2) with a **RES_CONTENT** subclause that compares the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) to the string value "IPM.appointment". The **RES_CONTENT** subclause has the **FuzzyLevelLow** field set to FL_PREFIX and the **FuzzyLevelHigh** field set to FL_IGNORECASE.
2. A **RES_NOT** restriction (2) with a **RES_CONTENT** subclause that compares the **PidTagMessageClass** property to the string value "IPM.Activity". The **RES_CONTENT** subclause has the **FuzzyLevelLow** field set to FL_PREFIX and the **FuzzyLevelHigh** field set to FL_IGNORECASE.
3. A **RES_NOT** restriction (2) with a **RES_CONTENT** subclause that compares the **PidTagMessageClass** property to the string value "IPM.StickyNote". The **RES_CONTENT** subclause has the **FuzzyLevelLow** field set to FL_PREFIX and the **FuzzyLevelHigh** field set to FL_IGNORECASE.

Second, a **RES_AND** restriction (2) with the following two subclauses:

1. A **RES_AND** restriction (2) with any of the following subclauses. Each of these subclauses is included in the **RES_AND** restriction (2) only if the special folder exists within the store. For example, if only the Drafts folder exists, then only the **RES_PROPERTY** subclause for the Drafts folder is included in the **RES_AND** restriction:
 - A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property ([\[MS-OXPFOAB\]](#) section 2.2.1.8) with the FID ([\[MS-OXCDA\]](#) section 2.2.1.1)/MID ([\[MS-OXCDA\]](#) section 2.2.1.2) pair of the Deleted Items folder.
 - A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Junk E-mail folder.
 - A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Drafts folder.
 - A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Outbox folder.
 - A **RES_PROPERTY** restriction with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Conflicts folder.

- A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Local Failures folder.
 - A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Server Failures folder.
 - A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Sync Issues folder.
2. A **RES_OR** restriction (2) with the following four subclauses:
1. A **RES_AND** restriction (2) with the following two subclauses:
 1. A **RES_OR** restriction (2) with the following two subclauses:
 - A **RES_CONTENT** restriction (2) with the **FuzzyLevelLow** field set to FL_FULLSTRING and the **FuzzyLevelHigh** field set to FL_IGNORECASE comparing the value of the **PidTagMessageClass** property to the string value "IPM.Task".
 - A **RES_CONTENT** restriction (2) with the **FuzzyLevelLow** field set to FL_PREFIX and the **FuzzyLevelHigh** field set to FL_IGNORECASE comparing the value of the **PidTagMessageClass** property to the string value "IPM.Task".
 2. A **RES_NOT** restriction (2) with a **RES_AND** subclause. The **RES_AND** subclause has the following two subclauses:
 - A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_EQ, comparing the value of the **PidLidTaskState** ([\[MS-OXPROPS\]](#) section 2.328) property to the **LONG** ([\[MS-DTYP\]](#)) value 2.
 - A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidLidTaskAccepted** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.7) to the binary value 1.
 2. A **RES_AND** restriction (2) with the following two subclauses:
 - A **RES_EXIST** restriction (2) that specifies the **PidTagFollowupIcon** property ([\[MS-OXOFLAG\]](#) section 2.2.1.2).
 - A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_GT, comparing the value of the **PidTagFollowupIcon** property to the **LONG** value 0 (zero).
 3. A **RES_AND** restriction (2) with the following two subclauses:
 - A **RES_EXIST** restriction (2) that specifies the **PidTagToDoItemFlags** property ([\[MS-OXOFLAG\]](#) section 2.2.1.6).
 - A **RES_BITMASK** restriction (2) with the **BitMapRelOp** field set to BMR_NEZ to compare the value of the **PidTagToDoItemFlags** property to the **ULONG** ([\[MS-DTYP\]](#)) value 0x00000001.
 4. A **RES_OR** restriction (2) with the following two subclauses:
 1. A **RES_AND** restriction (2) with the following three subclauses:
 1. A **RES_OR** restriction (2) with the following two subclauses:

- A **RES_NOT** restriction (2) with a **RES_EXIST** subclause that specifies the **PidTagFollowupIcon** property.
 - A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_EQ**, comparing the value of the **PidTagFollowupIcon** property to the **LONG** value 0 (zero).
2. A **RES_EXIST** restriction (2) that specifies the **PidTagFlagStatus** property ([\[MS-OXOFLAG\]](#) section 2.2.1.1).
 3. A **RES_PROPERTY** restriction (2), with a relational operator value of **RELOP_EQ**, comparing the value of the **PidTagFlagStatus** property to the **LONG** value 1.
2. A **RES_AND** restriction (2) with the following two subclauses:
 - A **RES_EXIST** restriction (2) that specifies the **PidLidTaskStatus** property ([\[MS-OXOTASK\]](#) section 2.2.2.2).
 - A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_EQ**, comparing the value of the **PidLidTaskStatus** property to the **ULONG** value 2.

The search SHOULD provide the client with the information when requested, include child folders, and run without content indexing.

3.1.4.1.3 Creating the Tracked Mail Processing Folder

To complete creation of the Tracked Mail Processing folder, an implementation SHOULD<4> set the search criteria for the Tracked Mail Processing folder. The search criteria are composed of restriction (2) structures, as specified in [\[MS-OXCDATA\]](#) section 2.12. The search criteria are set by using the Folder Object Protocol, as specified in [\[MS-OXCFOLD\]](#), such that the Top of Personal Folders folder is the only container included in the search and the search applies a **RES_AND** restriction (2) with the following two subclauses.

First, a **RES_AND** restriction (2) with any of the following subclauses. Each of these subclauses is included in the **RES_AND** restriction (2) only if the special folder exists within the store. For example, if only the Drafts folder exists, then only the **RES_PROPERTY** subclause for the Drafts folder is included in the **RES_AND** restriction (2).

- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property ([\[MS-OXPROPS\]](#) section 2.908) with the FID ([\[MS-OXCDATA\]](#) section 2.2.1.1)/MID ([\[MS-OXCDATA\]](#) section 2.2.1.2) pair of the Deleted Items folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Junk E-mail folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Drafts folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Outbox folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Conflicts folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Local Failures folder.

- A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Server Failures folder.
- A **RES_PROPERTY** restriction (2) with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property with the FID/MID pair of the Sync Issues folder.

Second, a **RES_AND** restriction (2) with the following two subclauses:

- A **RES_EXIST** restriction (2) that specifies the **PidTagSwappedToDoStore** property ([\[MS-OXOFLAG\]](#) section 2.2.1.8).
- A **RES_BITMASK** restriction (2) with the **BitMapRelOp** field set to of BMR_EQZ to compare the value of the **PidTagMessageFlags** property ([\[MS-OXPROPS\]](#) section 2.847) to the **ULONG** ([\[MS-DTYP\]](#)) value that is the bitwise OR combination of mfUnsent and mfSubmitted.

The search SHOULD provide the client with the information when requested, include child folders, and run without content indexing.

3.1.4.1.4 Creating Other Special Folders

If the special folder that is being created is one of the following:

- Deleted Items folder
- Junk E-mail folder
- Outbox folder
- Conflicts folder
- Local Failures folder
- Server Failures folder
- Sync Issues folder
- Drafts folder

An implementation MUST take the following additional steps:

1. Open the Reminders folder by using the ID of the special folder and the Folder Object Protocol. If this succeeds, perform the steps in section [3.1.4.1.1](#). The ID of the special folder is obtained as specified in section [2.2.2](#).
2. Open the To-Do folder by using the ID of the special folder and the Folder Object Protocol. If this succeeds, perform the steps in section [3.1.4.1.2](#).
3. Open the Tracked Mail Processing folder by using the ID of the special folder and the Folder Object Protocol. If this succeeds, perform the steps in section [3.1.4.1.3](#).

If the special folder that is being created is a Conversation Action Settings folder, an implementation MUST set the **PidTagAttributeHidden** property ([\[MS-OXCFOLD\]](#) section 2.2.2.2.1) of the Folder object to TRUE.

3.1.5 Message Processing Events and Sequencing Rules

An implementation **MUST** treat any failure to open the Root folder as a failure of the entire Special Folders Protocol. For all other special folders, an implementation **SHOULD** create the special folder if an attempt to open the folder fails or if the ID of the folder cannot be retrieved.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

The following examples illustrate the byte order of ROPs in a buffer that is being prepared for transmission. Note that the examples in this section show only the relevant portions of the specified ROPs; this is not the final byte sequence that gets transmitted over the wire. Also note that the data format for a multibyte field appears in **little-endian** format, with the **bytes** in the field presented from least significant to most significant.

Frequently, these ROP requests are packed with other ROP requests, compressed, obfuscated, and then packed in one or more **remote procedure calls (RPCs)**. These examples assume that the client has already successfully logged on to the server and has obtained any **Server object handles** that are to be used as inputs in the ROPs. For more information about RPCs, see [\[MS-OXCRPC\]](#).

Examples in this section use the following format for byte sequences. Each byte is expressed as a two-digit hexadecimal number.

```
0080: 45 4D 53 4D 44 42 2E 44-4C 4C 00 00 00 00 00 00
```

The value 0080 at the far left is the byte sequence's offset from the beginning of the buffer. Following the offset is a colon and then a series of up to 16 bytes. Here, the first byte (45) in the series is located 0x80 bytes (128 bytes) from the beginning of the buffer. The seventh byte (2E) in the series is located 0x86 bytes (134 bytes) from the beginning of the buffer. The dash between the eighth byte (44) and the ninth byte (4C) has no semantic value; it serves only to distinguish the eight-byte boundary for readability.

This byte sequence is followed by one or more lines that interpret it. In larger examples, the byte sequence is shown once in its entirety and then repeated in smaller chunks, with each smaller chunk interpreted separately.

When explaining the values of the **InputHandleIndex** and **OutputHandleIndex** fields, the example text describes the **Server object** that is referenced by the handle index. For more information about Server object handles, see [\[MS-OXCROPS\]](#) section 1.3.1.

4.1 Opening a Special Folder

Using the Calendar folder as an example, opening a special folder involves the following procedure:

1. Open the Inbox folder by using the Inbox folder's FID ([\[MS-OXCADATA\]](#) section 2.2.1.1) and the Folder Object Protocol, as described in [\[MS-OXCFOLD\]](#). This operation returns a Folder object.
2. Get the value of the **PidTagIpmAppointmentEntryId** property (section [2.2.3](#)) from the Inbox folder object returned in step 1 by using the Property and Stream Object Protocol, as described in [\[MS-OXCPRPT\]](#). This operation returns an entry ID for the Calendar folder.
3. Convert the entry ID returned in step 2 to a FID, as specified in section [2.2.2](#), and then open this folder by using this FID and the Folder Object Protocol. This operation returns a Folder object for the Calendar folder.

4.1.1 Client Request for Opening a Special Folder

A complete set of ROP requests to open the Calendar folder would look like the following.

0000: 02 01 00 01 01 00 00 00-01 42 0E 44 00

RopOpenFolder ([\[MS-OXCROPS\]](#) section 2.2.4.1)

RopId: 0x02

LogonId: 0

InputHandleIndex: 0 (Server object handle = 0x00000160)

FolderId: 0001-000001420e44

Note This is the FID ([\[MS-OXCDATA\]](#) section 2.2.1.1) of the Inbox folder to open.

OpenModeFlags: 0x00 Open existing folder

0000: 07 00 02 00 00 01 00 1D-00 14 00 49 67 03 00 F4
0010: 0F 02 01 72 66 1F 00 E5-36 1F 00 E6 36 1F 00 01
0020: 30 03 00 01 36 03 00 02-36 03 00 03 36 0B 00 0A
0030: 36 1F 00 13 36 02 01 16-36 02 01 D0 36 02 01 D1
0040: 36 02 01 D2 36 02 01 D3-36 02 01 D4 36 02 01 D5
0050: 36 02 01 D6 36 02 01 D7-36 02 11 D8 36 02 01 D9
0060: 36 03 00 DE 36 02 01 DF-36 02 01 E0 36 03 00 E1
0070: 36 02 11 E4 36 02 01 EB-36 02 01 DA 36

RopGetPropertiesSpecific ([\[MS-OXCROPS\]](#) section 2.2.8.3)

RopId: 0x07

LogonId: 1

InputHandleIndex: 1 (Server object handle = 0x000000E2)

PropertySizeLimit: 0x0000

WantUnicode: 0x0001 (TRUE)

PropertyValueCount: 29 (0x001D)

...

PidTagIpmAppointmentEntryId (section [2.2.3](#))

...

0000: 02 01 00 01 01 00 00 00-01 50 4D F6 00

RopOpenFolder

RopId: 0x02

LogonId: 1

InputHandleIndex: 0 (Server object handle = 0x00000160)

FolderId: 0001-000001504DF6

Note This is the FID ([\[MS-OXCDATA\]](#) section 2.2.1.1) for the Calendar folder to open.

OpenModeFlags: 0x00 Open existing folder

4.1.2 Server Response for Opening a Special Folder

```
0000: 02 01 00 00 00 00 00 00
```

RopOpenFolder ([\[MS-OXCROPS\]](#) section 2.2.4.1)

RopId: 0x02

OutputHandleIndex: 1 (Server object handle = 0x000000E2)

ReturnValue: Success (0x00000000)

HasRules: 0x00 (FALSE)

IsGhosted: 0x00 (FALSE)

Only 256 bytes dumped.

```
0000: 07 01 00 00 00 00 01 00-01 00 00 00 01 42 0E 41
0010: 00 3F 00 00 00 0A 0F 01-04 80 0A 0F 01 04 80 0A
0020: 0F 01 04 80 00 49 00 6E-00 62 00 6F 00 78 00 00
0030: 00 00 01 00 00 00 00 00-00 00 00 00 00 00 00 00
0040: 00 00 00 49 00 50 00 46-00 2E 00 4E 00 6F 00 74
0050: 00 65 00 00 00 0A 0F 01-04 80 00 2E 00 00 00 00
0060: 00 6A 3C B8 FA 3B A9 F0-46 B4 F4 E4 B6 C7 74 45
0070: 09 01 00 02 27 39 56 14-8B EF 4F 98 14 81 7E 2C
0080: 82 BD C2 00 00 01 50 4D-F6 00 00 00 2E 00 00 00
0090: 00 00 6A 3C B8 FA 3B A9-F0 46 B4 F4 E4 B6 C7 74
00a0: 45 09 01 00 02 27 39 56-14 8B EF 4F 98 14 81 7E
00b0: 2C 82 BD C2 00 00 01 50-4D F7 00 00 00 2E 00 00
00c0: 00 00 00 6A 3C B8 FA 3B-A9 F0 46 B4 F4 E4 B6 C7
00d0: 74 45 09 01 00 02 27 39-56 14 8B EF 4F 98 14 81
00e0: 7E 2C 82 BD C2 00 00 01-50 4D F8 00 00 00 2E 00
00f0: 00 00 00 00 6A 3C B8 FA-3B A9 F0 46 B4 F4 E4 B6
...
```

RopGetPropertiesSpecific ([\[MS-OXCROPS\]](#) section 2.2.8.3)

RopId: 0x07

InputHandleIndex: 1 (Server object handle = 0x000000E2)

ReturnValue: Success (0x00000000)

PropertyArray:

PropCount: 29

PidTagIpmAppointmentEntryId 46 bytes (section [2.2.3](#))

```
0000: 00 00 00 00 6A 3C B8 FA-3B A9 F0 46 B4 F4 E4 B6
0010: C7 74 45 09 01 00 02 27-39 56 14 8B EF 4F 98 14
```

```
0020: 81 7E 2C 82 BD C2 00 00-01 50 4D F6 00 00
...
```

Note The entry ID contains 00 00-01 50 4D F6 that is part of the FID ([\[MS-OXCDATA\]](#) section 2.2.1.1): 0001-000001504DF6 used in the **RopOpenFolder** ROP request that is shown in section [4.1.1](#).

```
0000: 02 01 00 00 00 00 00 00
```

RopOpenFolder

RopId: 0x02

OutputHandleIndex: 1 (Server object handle = 0x000000E2)

Note Successfully opened the Calendar folder, and the Calendar folder has the Server object handle 0x000000E2.

HasRules: 0x00 (FALSE)

IsGhosted: 0x00 (FALSE)

...

4.2 Creating a Special Folder

Using the Calendar folder as an example, creating a special folder involves the following procedure:

1. Open the parent folder, in this case the Top of Personal Folders folder, as specified in the folder hierarchy in section [3.1.1.1](#), by using the Top of Personal Folders FID ([\[MS-OXCADATA\]](#) section 2.2.1.1) and the Folder Object Protocol, as described in [\[MS-OXCFOLD\]](#). This operation returns a Folder object.
2. Create a new folder in the opened parent folder by using the Folder Object Protocol with the name "calendar" in English locales, reusing the existing folder if one already exists by that name. The entry ID of the newly created folder will be used in the following steps.
3. Open the Inbox folder by using the Inbox folder FID and the Folder Object Protocol. This operation returns a Folder object.
4. Set the value of the [PidTagIpmAppointmentEntryId](#) property (section [2.2.3](#)) on the Inbox Folder object returned in step 3 to the entry ID value of the folder created in step 2, by using the Property and Stream Object Protocol, as specified in [\[MS-OXCPRPT\]](#).
5. Set the value of the [PidTagContainerClass](#) property (section [2.2.7](#)) on the new Calendar folder to the value "IPF.Appointment".

4.2.1 Client Request for Creating a Special Folder

A complete set of ROP requests to create the Calendar folder would look like the following.

```
0000: 02 00 00 01 01 00 00 00-01 42 0E 41 00
```

RopOpenFolder ([\[MS-OXCROPS\]](#) section 2.2.4.1)

RopId: 0x02

LogonId: 0x00

InputHandleIndex: 0 (Server object handle = 0x00000059)

FolderId: 0001-000001420E41

Note The FID ([\[MS-OXCDATA\]](#) section 2.2.1.1) 0001-000001420E41 is for the Top of Personal Folders folder. This identifier is received from the **RopLogon** ROP ([\[MS-OXCROPS\]](#) section 2.2.3.1).

OpenModeFlags: 0x00 (ReadOnly)

...

```
0000: 1C 00 00 01 01 01 00 00-43 00 61 00 6C 00 65 00
0010: 6E 00 64 00 61 00 72 00-00 00 43 00 61 00 6C 00
0020: 65 00 6E 00 64 00 61 00-72 00 20 00 43 00 6F 00
0030: 6D 00 6D 00 65 00 6E 00-74 00 00 00
```

RopCreateFolder ([\[MS-OXCROPS\]](#) section 2.2.4.2)

RopId: 0x1C

LogonId: 0

InputHandleIndex: 0 (Server object handle = 0x00000497)

FolderType: 0x01

UseUnicodeStrings: (0x01) (TRUE)

OpenExisting: 0x00 (FALSE)

Reserved: 0x00 (FALSE)

DisplayName: **Calendar**

Comment: Calendar Comment

```
0000: 02 00 00 01 01 00 00 00-01 42 0E 44 00
```

RopOpenFolder

RopId: 0x02

LogonId: 0

InputHandleIndex: 0 (Server object handle = 0x00000059)

FolderId: 0001-000001420E44

Note Open the Inbox folder with the FID given in the **FolderId** field.

OpenModeFlags: 0x00 (Open existing folder)

```
0000: 0A 00 00 36 00 01 00 02-01 D0 36 2E 00 00 00 00
```

```
0010: 00 6A 3C B8 FA 3B A9 F0-46 B4 F4 E4 B6 C7 74 45
0020: 09 01 00 02 27 39 56 14-8B EF 4F 98 14 81 7E 2C
0030: 82 BD C2 00 00 01 50 4D-F6 00 00
```

RopSetProperties ([\[MS-OXCROPS\]](#) section 2.2.8.6)

RopId: 0x0A

LogonId: 0

InputHandleIndex: 0 (Server object handle = 0x000004E4)

Note 0x000004E4 is a handle to the Inbox folder.

PropertyValueSize: 0x0036 (54)

PropertyValueCount: 1 (0x01)

0x36D00102 **PidTagIpmAppointmentEntryId** 46 bytes (section [2.2.3](#))

```
0000: 00 00 00 00 6A 3C B8 FA-3B A9 F0 46 B4 F4 E4 B6
0010: C7 74 45 09 01 00 02 27-39 56 14 8B EF 4F 98 14
0020: 81 7E 2C 82 BD C2 00 00-01 50 4D F6 00 00
```

```
0000: 0A 00 00 26 00 01 00 1F-00 13 36 49 00 50 00 46
0010: 00 2E 00 41 00 70 00 70-00 6F 00 69 00 6E 00 74
0020: 00 6D 00 65 00 6E 00 74-00 00 00
```

RopSetProperties

RopId: 0x0A

LogonId: 0

InputHandleIndex: 0 (Server object handle = 0x0000042E)

PropertyValueSize: 0x0026 (38)

PropertyValueCount: 1 (0x01)

PidTagContainerClass (section [2.2.7](#)): IPF.Appointment

4.2.2 Server Response for Creating a Special Folder

```
0000: 02 01 00 00 00 00 00 00
```

RopOpenFolder ([\[MS-OXCROPS\]](#) section 2.2.4.1)

RopId: 0x02

OutputHandleIndex: 1 (Server object handle = 0x00000497)

Note 0x00000497 is the handle of the Top of Personal Folders folder, and it is used in the **RopCreateFolder** ROP request ([\[MS-OXCROPS\]](#) section 2.2.4.2) to create the Calendar folder as shown in section [4.2.1](#).

ReturnValue: Success (0x00000000)

HasRules: 0x00 (FALSE)

IsGhosted: 0x00 (FALSE)

```
0000: 1C 01 00 00 00 00 01 00-00 00 01 50 4D F6 00
```

RopCreateFolder ([\[MS-OXCROPS\]](#) section 2.2.4.2)

RopId: 0x1C

OutputHandleIndex: 1 (Server object handle = 0x0000042E)

Note 0x0000042E is the handle to the Calendar folder that was created.

ReturnValue: ecNone (success) (0x00000000)

FolderId: 0001-000001504df6

IsExistingFolder: 0x00 (FALSE)

```
0000: 02 01 00 00 00 00 00 00
```

RopOpenFolder

RopId: 0x02

OutputHandleIndex: 1 (Server object handle = 0x000004E4)

Note 0x000004E4 is the handle of the Inbox folder. It is used in the **RopSetProperties** ROP request ([\[MS-OXCROPS\]](#) section 2.2.8.6) to set the **PidTagIpmAppointmentEntryId** property (section [2.2.3](#)) of the Inbox folder as shown in section [4.2.1](#).

ReturnValue: Success (0x00000000)

HasRules: 0x00 (FALSE)

IsGhosted: 0x00 (FALSE)

```
0000: 0A 00 00 00 00 00 00 00
```

RopSetProperties

RopId: 0x0A

InputHandleIndex: 0 (Server object handle = 0x0000042E)

ReturnValue: ecNone (success) (0x00000000)

PropertyProblemCount: 0

PropertyProblems:

0000: 0A 00 00 00 00 00 00 00

RopSetProperties

RopId: 0x0A

InputHandleIndex: 0 (Server object handle = 0x0000042E)

ReturnValue: ecNone (success) (0x00000000)

PropertyProblemCount: 0

PropertyProblems:

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to the Special Folders Protocol. General security considerations pertaining to the underlying transport apply, as described in [\[MS-OXCFLD\]](#).

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® Office Outlook® 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Outlook® 2010

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 2.2.5.1](#): Office Outlook 2003 clients and Exchange 2003 servers do not recognize the RSS Feeds, Tracked Mail Processing, and To-Do folders as special folders.

<2> [Section 3.1.4.1.1](#): Office Outlook 2003 sets the search criteria to include only the Calendar, Tasks, Inbox, and Contacts folders, and it sets a restriction (2) only for the **PidLidReminderSet** ([\[MS-OXPROPS\]](#) section 2.216) and **PidLidRecurring** ([\[MS-OXPROPS\]](#) section 2.209) properties. In addition, it does not include RECURSIVE_SEARCH.

<3> [Section 3.1.4.1.2](#): Office Outlook 2003 and Exchange 2003 do not recognize search criteria for the To-Do folder.

<4> [Section 3.1.4.1.3](#): Office Outlook 2003 and Exchange 2003 do not recognize search criteria for the Tracked Mail Processing folder.

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

A

Abstract data model – client
[folder hierarchy](#) 15
[overview](#) 15
[search criteria for search special folders](#) 16

Abstract data model – server
[folder hierarchy](#) 15
[overview](#) 15
[search criteria for search special folders](#) 16

[Applicability](#) 8

B

[Binary Identification Properties message](#) 11

C

[Capability negotiation](#) 8
[Change tracking](#) 35

Client
[initialization](#) 18
[local events](#) 24
[message processing](#) 24
[overview](#) 15
[sequencing rules](#) 24
[timer events](#) 24
[timers](#) 18

Client - abstract data model
[folder hierarchy](#) 15
[overview](#) 15
[search criteria for search special folders](#) 16

Client - higher layer triggered events
[folder creation](#) 18
[overview](#) 18

Creating a special folder example
[client request for creating a special folder](#) 28
[overview](#) 28
[server response for creating a special folder](#) 30

D

Data model – abstract
[client](#) 15
[server](#) 15

E

Examples - creating a special folder
[client request for creating a special folder](#) 28
[overview](#) 28
[server response for creating a special folder](#) 30

Examples - opening a special folder
[client request for opening a special folder](#) 25
[overview](#) 25
[server response for opening a special folder](#) 27

[Examples - overview](#) 25

F

[Fields - vendor-extensible](#) 8

G

[Glossary](#) 5

H

Higher layer triggered events - client
[folder creation](#) 18
[overview](#) 18

Higher layer triggered events - server
[folder creation](#) 18
[overview](#) 18

I

[Identifiers for Special Folders message](#) 11
[Implementer - security considerations](#) 33
[Inbox Identification message](#) 14
[Index of security parameters](#) 33
[Informative references](#) 7

Initialization
[client](#) 18
[server](#) 18

[Introduction](#) 5

L

[List of Special Folders message](#) 9

Local events
[client](#) 24
[server](#) 24

M

Message processing
[client](#) 24
[server](#) 24

Messages
[Binary Identification Properties](#) 11
[Identifiers for Special Folders](#) 11
[Inbox Identification](#) 14
[List of Special Folders](#) 9
[PidTagAdditionalRenEntryIds Property](#) 12
[PidTagAdditionalRenEntryIdsEx Property](#) 12
[PidTagContainerClass Property](#) 14
[transport](#) 9

N

[Normative references](#) 6

O

Opening a special folder example
[client request for opening a special folder](#) 25
[overview](#) 25
[server response for opening a special folder](#) 27

[Overview \(synopsis\)](#) 7

P

[Parameters - security index](#) 33

[PersistData structure](#) 12

[PersistElement structure](#) 13

[PidTagAdditionalRenEntryIds Property message](#) 12

[PidTagAdditionalRenEntryIdsEx Property message](#)
12

[PersistData structure](#) 12

[PersistElement structure](#) 13

[PidTagContainerClass Property message](#) 14

[Preconditions](#) 7

[Prerequisites](#) 7

[Product behavior](#) 34

R

References

[informative](#) 7

[normative](#) 6

[Relationship to other protocols](#) 7

S

Security

[implementer considerations](#) 33

[parameter index](#) 33

Sequencing rules

[client](#) 24

[server](#) 24

Server

[initialization](#) 18

[local events](#) 24

[message processing](#) 24

[overview](#) 15

[sequencing rules](#) 24

[timer events](#) 24

[timers](#) 18

Server - abstract data model

[folder hierarchy](#) 15

[overview](#) 15

[search criteria for search special folders](#) 16

Server - higher layer triggered events

[folder creation](#) 18

[overview](#) 18

[Standards assignments](#) 8

T

Timer events

[client](#) 24

[server](#) 24

Timers

[client](#) 18

[server](#) 18

[Tracking changes](#) 35

[Transport](#) 9

Triggered events - client

[folder creation](#) 18

[overview](#) 18

Triggered events - server

[folder creation](#) 18

[overview](#) 18

V

[Vendor-extensible fields](#) 8

[Versioning](#) 8