

[MS-OXOSFLD]: Special Folders Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation 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 might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability.
4/25/2008	0.2	Minor	Revised and updated property names and other technical content.
6/27/2008	1.0	Major	Initial Release.
8/6/2008	1.01	Minor	Updated references to reflect date of initial release.
9/3/2008	1.02	Minor	Revised and edited technical content.
12/3/2008	1.03	Minor	Revised and edited technical content.
3/4/2009	1.04	Minor	Revised and edited technical content.
4/10/2009	2.0	Major	Updated technical content and applicable product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	3.1	Minor	Updated the technical content.
2/10/2010	3.2	Minor	Updated the technical content.
5/5/2010	3.3	Minor	Updated the technical content.
8/4/2010	3.4	Minor	Clarified the meaning of the technical content.
11/3/2010	3.4	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	3.4	None	No changes to the meaning, language, and formatting of the technical content.
8/5/2011	4.0	Major	Significantly changed the technical content.
10/7/2011	4.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	5.0	Major	Significantly changed the technical content.
4/27/2012	6.0	Major	Significantly changed the technical content.
7/16/2012	7.0	Major	Significantly changed the technical content.
10/8/2012	8.0	Major	Significantly changed the technical content.
2/11/2013	8.0	None	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	9.0	Major	Significantly changed the technical content.
11/18/2013	9.1	Minor	Clarified the meaning of the technical content.
2/10/2014	9.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	9.2	Minor	Clarified the meaning of the technical content.
7/31/2014	9.2	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
10/30/2014	9.2	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	10.0	Major	Significantly changed the technical content.
5/26/2015	10.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2015	10.0	None	No changes to the meaning, language, or formatting of the technical content.
6/13/2016	10.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	10.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	9
1.2.1	Normative References	9
1.2.2	Informative References	9
1.3	Overview	10
1.4	Relationship to Other Protocols	10
1.5	Prerequisites/Preconditions	10
1.6	Applicability Statement	10
1.7	Versioning and Capability Negotiation	10
1.8	Vendor-Extensible Fields	10
1.9	Standards Assignments.....	10
2	Messages.....	11
2.1	Transport	11
2.2	Message Syntax.....	11
2.2.1	List of Special Folders.....	11
2.2.2	Identifiers for Special Folders	13
2.2.3	Binary Identification Properties.....	13
2.2.4	PidTagAdditionalRenEntryIds Property	14
2.2.5	PidTagAdditionalRenEntryIdsEx Property	14
2.2.5.1	PersistData Structure	15
2.2.5.2	PersistElement Structure.....	16
2.2.6	PidTagFreeBusyEntryIds Property	16
2.2.7	Inbox Identification.....	17
2.2.8	PidTagContainerClass Property	17
3	Protocol Details	18
3.1	Client Details	18
3.1.1	Abstract Data Model.....	18
3.1.1.1	Folder Hierarchy	18
3.1.1.2	Search Criteria for Search Special Folders	19
3.1.2	Timers	21
3.1.3	Initialization.....	21
3.1.4	Higher-Layer Triggered Events	21
3.1.4.1	Folder Creation	21
3.1.4.1.1	Creating the Reminders Folder	21
3.1.4.1.2	Creating the To-Do Folder	22
3.1.4.1.3	Creating the Tracked Mail Processing Folder.....	25
3.1.4.1.4	Creating Other Special Folders	25
3.1.5	Message Processing Events and Sequencing Rules	26
3.1.6	Timer Events.....	26
3.1.7	Other Local Events.....	26
3.2	Server Details.....	26
3.2.1	Abstract Data Model.....	26
3.2.2	Timers	27
3.2.3	Initialization	27
3.2.4	Higher-Layer Triggered Events	27
3.2.5	Message Processing Events and Sequencing Rules	27
3.2.6	Timer Events.....	27
3.2.7	Other Local Events.....	27
4	Protocol Examples	28
4.1	Opening a Special Folder.....	28
4.1.1	Client Request for Opening a Special Folder	28

4.1.2	Server Response for Opening a Special Folder	30
4.2	Creating a Special Folder	31
4.2.1	Client Request for Creating a Special Folder	31
4.2.2	Server Response for Creating a Special Folder	33
5	Security	35
5.1	Security Considerations for Implementers	35
5.2	Index of Security Parameters	35
6	Appendix A: Product Behavior	36
7	Change Tracking	38
8	Index	39

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 [\[MS-OXCFLD\]](#).

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

address book: A collection of Address Book objects, each of which are contained in any number of address lists.

calendar: A date range that shows availability, meetings, and appointments for one or more users or resources. See also **Calendar object**.

Calendar folder: A **Folder object** that contains **Calendar objects**.

Calendar object: A **Message object** that represents an event, which can be a one-time event or a recurring event. The Calendar object includes properties that specify event details such as description, organizer, date and time, and status.

client-side rule: A rule that has at least one action that is executed by a client because it cannot be executed by a server.

Common Views folder: A **special folder** that contains the data for default views that are standard for a message store and can be used by any user of a client that accesses the message store.

contact: A person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and attributes (2), such as an Internet message address or login name.

Contact object: A **Message object** that contains properties pertaining to a **contact**.

Contacts folder: A **Folder object** that contains **Contact objects**.

Container class: The value of the PidTagContainerClass property on a folder, which indicates the default **Message object** type for the folder.

conversation action: A limited set of actions that a user applies to all **Message objects** that have the same PidTagConversationId value. The action is applied to all Message objects that are currently in the store or are delivered in the future.

Deferred Action Message (DAM): A hidden message indicating to a client that it needs to execute one or more rules on another user-visible message in the store.

Deferred Error Message (DEM): A hidden message indicating to a client that it needs to present the user with an error indicating that a server-side rule failed to execute.

delegate: A user or resource that has permissions to act on behalf of another user or resource.

Delegate Information object: A **Message object** that contains properties specifying delegate access settings for resources in a delegator's mailbox.

Deleted Items folder: A **special folder** that is the default location for objects that have been deleted.

Drafts folder: A **special folder** that is the default location for **Message objects** that have been saved but not sent.

Email object: A **Message object** that represents an email message in a message store and adheres to the property descriptions that are described in in [\[MS-OXOMSG\]](#).

entry ID: See **EntryID**.

EntryID: A sequence of bytes that is used to identify and access an object.

folder associated information (FAI): A collection of **Message objects** that are stored in a Folder object and are typically hidden from view by email applications. An FAI Message object is used to store a variety of settings and auxiliary data, including forms, views, calendar options, favorites, and category lists.

Folder object: A messaging construct that is typically used to organize data into a hierarchy of objects containing Message objects and **folder associated information (FAI)** Message objects.

free/busy message: A message that is stored in a public folder and contains free/busy data.

Inbox folder: A **special folder** that is the default location for **Message objects** received by a user or resource.

Journal object: A **Message object** that represents an entry in a journal or log and adheres to the property descriptions that are described in in [\[MS-OXOJRN\]](#).

Junk Email folder: A **special folder** that is the default location for **Message objects** that are determined to be junk email by a Junk Email rule.

Junk Email rule: An extended rule that describes a spam filter.

little-endian: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

locale: A collection of rules and data that are specific to a language and a geographical area. A locale can include information about sorting rules, date and time formatting, numeric and monetary conventions, and character classification.

mailbox: A **message store** that contains email, calendar items, and other **Message objects** for a single recipient.

message class: A property that loosely defines the type of a message, contact, or other Personal Information Manager (PIM) object in a mailbox.

Message object: A set of properties that represents an email message, appointment, contact, or other type of personal-information-management object. In addition to its own properties, a Message object contains recipient properties that represent the addressees to which it is addressed, and an attachments table that represents any files and other Message objects that are attached to it.

message store: A unit of containment for a single hierarchy of Folder objects, such as a mailbox or public folders.

Note object: A **Message object** that represents a simple text note in a messaging store and that adheres to the property descriptions that are described in [\[MS-OXONOTE\]](#). A Note object functions as an electronic equivalent of a paper sticky note.

Outbox folder: A **special folder** that contains **Message objects** that are submitted to be sent.

Personal Distribution List object: A **Message object** that contains properties pertaining specifically to user-created distribution lists.

public folder: A **Folder object** that is stored in a location that is publicly available.

Really Simple Syndication (RSS): An XML-based syndication format for content, as described in [\[RSS2.0\]](#).

Receive folder: A **Folder object** that is configured to be the destination for email messages that are delivered.

recipient: An entity that can receive email messages.

reminder: A generally user-visible notification that a specified time has been reached. A reminder is most commonly related to the beginning of a meeting or the due time of a task but it can be applied to any object type.

remote operation (ROP): An operation that is invoked against a server. Each ROP represents an action, such as delete, send, or query. A ROP is contained in a ROP buffer for transmission over the wire.

restriction: A filter used to map some domain into a subset of itself, by passing only those items from the domain that match the filter. Restrictions can be used to filter existing Table objects or to define new ones, such as **search folder** or rule criteria.

Root folder: The **special folder** that is the top-level folder in a message store hierarchy. It contains all other **Folder objects** in that message store.

ROP request: See ROP request buffer.

ROP response: See ROP response buffer.

search criteria: A criteria used to determine which messages are included in a folder with specific characteristics. It is composed of a restriction, which is the filter to be applied, and a search scope, which are the folders that contain the content to search.

search folder: A **Folder object** that provides a means of querying for items that match certain criteria. The search folder includes the search folder definition message and the search folder container.

Sent Items folder: A **special folder** that is the default location for storing copies of **Message objects** after they are submitted or sent.

Server object: An object on a server that is used as input or created as output for **remote operations (ROPs)**.

Server object handle: A 32-bit value that identifies a **Server object**.

special folder: One of a default set of **Folder objects** that can be used by an implementation to store and retrieve user data objects.

Store object: An object that is used to store **mailboxes** and **public folder** content.

Task object: A **Message object** that represents an assignment to be completed.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

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.

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

[MS-OXCADATA] Microsoft Corporation, "[Data Structures](#)".

[MS-OXCFOLD] Microsoft Corporation, "[Folder Object Protocol](#)".

[MS-OXCMSG] Microsoft Corporation, "[Message and Attachment Object Protocol](#)".

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

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

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

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

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

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

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

[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-OXCFXICS] Microsoft Corporation, "[Bulk Data Transfer Protocol](#)".

[MS-OXCPRPT] Microsoft Corporation, "[Property and Stream Object Protocol](#)".

[MS-OXCRPC] Microsoft Corporation, "[Wire Format Protocol](#)".

[MS-OXCSPAM] Microsoft Corporation, "[Spam Confidence Level Protocol](#)".

[MS-OXOCFG] Microsoft Corporation, "[Configuration Information Protocol](#)".

[MS-OXOCNTC] Microsoft Corporation, "[Contact Object Protocol](#)".

[MS-OXODOC] Microsoft Corporation, "[Document Object Protocol](#)".

[MS-OXOJRN] Microsoft Corporation, "[Journal Object Protocol](#)".

[MS-OXOMSG] Microsoft Corporation, "[Email Object Protocol](#)".

[MS-OXONOTE] Microsoft Corporation, "[Note Object Protocol](#)".

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

[MS-OXORSS] Microsoft Corporation, "[RSS Object Protocol](#)".

[MS-OXORULE] Microsoft Corporation, "[Email Rules Protocol](#)".

[MS-OXOSRCH] Microsoft Corporation, "[Search Folder List Configuration Protocol](#)".

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**. 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 [\[MS-OXCFOLD\]](#).

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [\[MS-OXPROTO\]](#).

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 items, as described in section [2.2.1](#), 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 [\[MS-OXCFLD\]](#).

2.2 Message Syntax

2.2.1 List of Special Folders

The set of folders that are special folders, <1> 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 message store hierarchy's top-level folder, which contains all other Folder objects in that message store.	None	[MS-OXCSTOR]
Finder	Contains the default search folders .	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 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 Message (DEM) that results from the execution of client-side rules .	None	[MS-OXORULE]
Calendar	Contains Calendar objects , such as	"IPF.Appointment"	[MS-

Special folder name	Description	Container class	More information
	appointments.		OXOCAL
Contacts	Contains Contact objects .	"IPF.Contact"	[MS- OXOCNTC]
Suggested Contacts	Contains Contact objects that are created when a recipient is not in an address book .	"IPF.Contact"	[MS- OXOCNTC]
Quick Contacts	Contains Contact objects for the user's favorite contacts and instant messaging contacts.	"IPF.Contact.MOC.QuickContacts"	[MS- OXOCNTC]
IM Contacts List	Contains Personal Distribution List objects of favorite contacts and instant messaging contacts.	"IPF.Contact.MOC.ImContactList"	[MS- OXOCNTC]
Contacts Search	Search folder that displays a list of contacts that fit search criteria .	"IPF.Contact"	[MS- OXOCNTC]
Document Libraries	Contains documents to be uploaded to a shared location.	"IPF.ShortcutFolder"	[MS- OXODOC]
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 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]
Sync Issues	Contains folders that contain messages that indicate particular issues encountered during synchronization between client and server. This is the parent folder of the Conflicts, Local Failures, and Server Failures folders.	"IPF.Note"	[MS- OXCFXICS] *
Conflicts	Contains Message objects that indicate synchronization conflicts between client and server.	"IPF.Note"	[MS- OXCFXICS]*
Local Failures	Contains messages that indicate client-side synchronization failures.	"IPF.Note"	[MS- OXCFXICS]*
Server Failures	Contains messages that indicate server-side synchronization failures.	"IPF.Note"	[MS- OXCFXICS]*
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- OXCSHAM]
RSS Feeds	Contains Really Simple Syndication (RSS) feed messages.	"IPF.Note.OutlookHomepage"	[MS- OXORSS]

Special folder name	Description	Container class	More information
Tracked Mail Processing	Search folder that contains flagged objects.	"IPF.Note"	[MS-OXOFLAG]
To-Do	Search folder 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]
Spooler Queue	Contains E-mail objects that have been sent or received.	"IPF.Note"	[MS-OXOMSG]

*The Sync Issues folder and its subfolders (Conflicts, Local Failures, and Server Failures) are optional and are not synchronized between client and server. These folders are used in one possible implementation for handling synchronization errors that occur during bulk data transfer. This implementation involves moving the problem items into an unsynchronized folder, such as Conflicts, Local Failures, or Server Failures. By moving the items to a folder that is not synchronized, the error does not continue to occur during each subsequent synchronization.

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) ([\[MS-OXCADATA\]](#) 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 (section [2.2.6](#)) 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 folder that is set to be the **Receive folder**. For details about setting the Receive folder, see [\[MS-OXCSTOR\]](#).

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** ([\[MS-OXCDATA\]](#) section 2.11.1).

Property	Description
PidTagIpmAppointmentEntryId ([MS-OXPROPS] section 2.741)	Contains the entry ID of the Calendar folder
PidTagIpmContactEntryId ([MS-OXPROPS] section 2.742)	Contains the entry ID of the Contacts folder
PidTagIpmJournalEntryId ([MS-OXPROPS] section 2.744)	Contains the entry ID of the Journal folder
PidTagIpmNoteEntryId ([MS-OXPROPS] section 2.745)	Contains the entry ID of the Notes folder
PidTagIpmTaskEntryId ([MS-OXPROPS] section 2.746)	Contains the entry ID of the Tasks folder
PidTagRemindersOnlineEntryId ([MS-OXPROPS] section 2.903)	Contains the entry ID of the Reminders folder
PidTagIpmDraftsEntryId ([MS-OXPROPS] section 2.743)	Contains the entry ID of the Drafts folder

2.2.4 PidTagAdditionalRenEntryIds Property

Type: **PtypMultipleBinary** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidTagAdditionalRenEntryIds** property ([\[MS-OXPROPS\]](#) section 2.500) 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** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidTagAdditionalRenEntryIdsEx** property ([\[MS-OXPROPS\]](#) section 2.501) of the Store object specifies the identifiers of several special folders. If this property is present, its value is an array of **PersistData** structures (section [2.2.5.1](#)), each of which pertains to a particular special folder. The last **PersistData** structure in the array is indicated either by the PERSIST_SENTINEL value in the **PersistID** field of the **PersistData** structure or by the end of the data stream.

2.2.5.1 PersistData Structure

The **PersistData** structure contains the entry ID of a special folder and other data related to a special folder. The **PersistData** structure has the following format.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
PersistID											DataElementsSize																				
DataElements (variable)																															
...																															

PersistID (2 bytes): An integer that identifies the type of special folder to which the **PersistData** structure pertains.

The value of the **PersistID** field SHOULD [<2>](#) 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.

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.
RSF_PID_CONTACT_SEARCH	0x8009	Indicates that the structure contains data for the Contacts Search folder.
RSF_PID_BUDDYLIST_PDLS	0x800A	Indicates that the structure contains data for the IM Contacts List folder.
RSF_PID_BUDDYLIST_CONTACTS	0x800B	Indicates that the structure contains data for the Quick Contacts folder.
PERSIST_SENTINEL	0x0000	Indicates that the PersistData structure is the last one contained in the PidTagAdditionalRenEntryIdsEx property (section 2.2.5). The PERSIST_SENTINEL is optional; if it is not included, processing continues to the end of the data stream.

DataElementsSize (2 bytes): The size, in bytes, of the **DataElements** field.

DataElements (variable): An array of **PersistElement** structures (section [2.2.5.2](#)), each of which contains the data about the special folder. The last **PersistElement** structure in the array is

indicated either by the ELEMENT_SENTINEL value in the **ElementID** field of the **PersistElement** structure or by the end of the data stream.

2.2.5.2 PersistElement Structure

The **PersistElement** structure contains either the entry ID of the special folder or other data related to the special folder. The **PersistElement** structure has the following format.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
ElementID											ElementDataSize																				
ElementData (variable)																															
...																															

ElementID (2 bytes): An integer that identifies the type of data contained in the **ElementData** field.

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.

Value name	Value	Size of the ElementData field	Meaning
RSF_ELID_HEADER	0x0002	4 bytes	Indicates that the ElementData field contains a 4-byte header value equal to 0x00000000.
RSF_ELID_ENTRYID	0x0001	variable	Indicates that the ElementData field contains the entry ID of the special folder that is of the type indicated by the value of the PersistID field of the PersistData structure.
ELEMENT_SENTINEL	0x0000	0 bytes	Indicates that the PersistElement structure is the last one contained in the DataElements field of the PersistData structure.

ElementDataSize (2 bytes): The size, in bytes, of the **ElementData** field. The value of this field is 0 (zero) if the **ElementID** field is set to ELEMENT_SENTINEL.

ElementData (variable): The data corresponding to the special folder that is identified by the value of the **PersistID** field of the **PersistData** structure (section 2.2.5.1). The type of data in this field is indicated by the value of the **ElementID** field. If the value of the **ElementID** field is RSF_ELID_HEADER, the **ElementData** field MUST be set to 0x00000000.

This field is not present if the **ElementID** field is set to ELEMENT_SENTINEL.

2.2.6 PidTagFreeBusyEntryIds Property

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

The **PidTagFreeBusyEntryIds** property ([MS-OXPROPS] section 2.696) is set on the Root folder and on the Inbox folder of the local message store. The value on the Root folder of the local message store MUST be equal to the value on the Inbox folder. The property contains four binary values, as follows:

- The first value is null.

- The second value is the **EntryID** of the **Delegate Information object**. For details about the Delegate Information object, see [\[MS-OXODLGT\]](#).
- The third value can be the EntryID of the **free/busy message** of the logged-on user. If the free/busy **public folder** is unavailable, the third value is null. This value is set when a client initially creates the free/busy message for a user.
- The fourth value is the EntryID of the public folder that has its **PidTagDisplayName** property ([\[MS-OXCFOLD\]](#) section 2.2.2.2.5) set to "Freebusy Data". This folder is a child folder of the Root folder of the public folders message store.

2.2.7 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 ([\[MS-OXCDATA\]](#) section 2.2.1.1) for the default Receive folder for the Store object. For details about the Store Object Protocol, see [\[MS-OXCSTOR\]](#).

2.2.8 PidTagContainerClass Property

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

The **PidTagContainerClass** property ([\[MS-OXPROPS\]](#) section 2.633), 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 Client Details

Special folders can be opened or created by clients. This section defines constraints to which clients adhere when interacting with special folders. In all other respects, clients 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, [<3>](#) which is the top-level folder, in a message store:

1. Deferred Action
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. Suggested Contacts
 8. Quick Contacts
 9. IM Contact List
 10. Contacts Search

11. Journal
 12. Notes
 13. Tasks
 14. Drafts
 15. Sync Issues
 1. Conflicts
 2. Local Failures
 3. Server Failures
 16. Junk E-mail
 17. RSS Feeds
 18. Conversation Action Settings
9. Spooler Queue
 10. FreeBusy Data
 11. Document Libraries

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, as specified in [\[MS-OXCFLD\]](#), and the functionality supported by these folders depends on specific search criteria. The search criteria are set on each search folder 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**:

- 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:

- 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:

- 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 a client tries to read or write Message objects within a special folder, the client **MUST** obtain a handle to the special folder by using the following steps:

1. The client **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 message store, the client **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, a client **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 client'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 [2.2.8](#)) 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, a client **SHOULD** [<4>](#) set the search criteria for the Reminders folder. The search criteria are composed of restriction structures, as specified in [\[MS-OXCADATA\]](#) 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 with the following two subclauses.

First, a **RES_AND** restriction with any of the following subclauses. Each of these subclauses is included in the **RES_AND** restriction only if the special folder exists within the message 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 with a relational operator (specified by the **RelOp** field) value of **RELOP_NE**, comparing the value of the **PidTagParentEntryId** property ([MS-OXCFOOLD] section 2.2.2.2.1.7) with the FID ([MS-OXCADATA] section 2.2.1.1)/message ID (MID) ([MS-OXCADATA] section 2.2.1.2) pair of the Deleted Items 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 Junk E-mail 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 Drafts 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 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 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 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 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 with the following three subclauses:

1. A **RES_NOT** restriction that contains a **RES_AND** restriction with the following two subclauses:
 - A **RES_EXIST** restriction that specifies the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3).
 - A **RES_CONTENT** restriction 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 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 with the following two subclauses:
 1. A **RES_PROPERTY** restriction 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 with the following two subclauses:
 - A **RES_EXIST** restriction that specifies the **PidLidRecurring** property ([MS-OXOCAL] section 2.2.1.12).
 - A **RES_PROPERTY** restriction 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, a client SHOULD [<5>](#) set the search criteria for the To-Do folder. The search criteria are composed of restriction 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 with the following two subclauses.

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

1. A **RES_NOT** restriction 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 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 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 with the following two subclauses:

1. A **RES_AND** restriction with any of the following subclauses. Each of these subclauses is included in the **RES_AND** restriction only if the special folder exists within the message 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 with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property ([\[MS-OXCFOLD\]](#) section 2.2.2.1.7) 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 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 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 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 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 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 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 with the following four subclauses:
 1. A **RES_AND** restriction with the following two subclauses:

1. A **RES_OR** restriction with the following two subclauses:
 - A **RES_CONTENT** restriction 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 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 with a **RES_AND** subclause. The **RES_AND** subclause has the following two subclauses:
 - A **RES_PROPERTY** restriction with a relational operator value of RELOP_EQ, comparing the value of the **PidLidTaskState** ([\[MS-OXPROPS\]](#) section 2.334) property to the **LONG** ([\[MS-DTYP\]](#)) value 2.
 - A **RES_PROPERTY** restriction 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 with the following two subclauses:
 - A **RES_EXIST** restriction that specifies the **PidTagFollowupIcon** property ([\[MS-OXOFLAG\]](#) section 2.2.1.2).
 - A **RES_PROPERTY** restriction 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 with the following two subclauses:
 - A **RES_EXIST** restriction that specifies the **PidTagToDoItemFlags** property ([\[MS-OXOFLAG\]](#) section 2.2.1.6).
 - A **RES_BITMASK** restriction 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 with the following two subclauses:
 1. A **RES_AND** restriction with the following three subclauses:
 1. A **RES_OR** restriction with the following two subclauses:
 - A **RES_NOT** restriction with a **RES_EXIST** subclause that specifies the **PidTagFollowupIcon** property.
 - A **RES_PROPERTY** restriction 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 that specifies the **PidTagFlagStatus** property ([\[MS-OXOFLAG\]](#) section 2.2.1.1).
 3. A **RES_PROPERTY** restriction, 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 with the following two subclauses:
 - A **RES_EXIST** restriction that specifies the **PidLidTaskStatus** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.2).

- A **RES_PROPERTY** restriction 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, a client SHOULD [<6>](#) set the search criteria for the Tracked Mail Processing folder. The search criteria are composed of restriction 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 with the following two subclauses.

First, a **RES_AND** restriction with any of the following subclauses. Each of these subclauses is included in the **RES_AND** restriction only if the special folder exists within the message 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 with a relational operator value of RELOP_NE, comparing the value of the **PidTagParentEntryId** property ([\[MS-OXCFOLD\]](#) section 2.2.2.2.1.7) with the FID ([\[MS-OXCADATA\]](#) section 2.2.1.1)/MID ([\[MS-OXCADATA\]](#) section 2.2.1.2) pair of the Deleted Items 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 Junk E-mail 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 Drafts 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 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 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 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 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 with the following two subclauses:

- A **RES_EXIST** restriction that specifies the **PidTagSwappedToDoStore** property ([\[MS-OXOFLAG\]](#) section 2.2.1.8).
- A **RES_BITMASK** restriction with the **BitMapRelOp** field set to of BMR_EQZ to compare the value of the **PidTagMessageFlags** property ([\[MS-OXPROPS\]](#) section 2.782) 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

A client 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, a client 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

A client MUST treat any failure to open the Root folder as a failure of the entire Special Folders Protocol. For all other special folders, a client 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.

3.2 Server Details

The server processes a client's requests regarding special folders. In all other respects, the server operates as specified in [\[MS-OXCFOLD\]](#).

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

The server uses the same abstract data model as the client, as specified in section [3.1.1](#).

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 [\[MS-OXCFLD\]](#) section 3.2.5.

3.2.6 Timer Events

None.

3.2.7 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 and obfuscated, as described in [\[MS-OXCRPC\]](#) section 3. 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.

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 [\[MS-OXCFCOLD\]](#). 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: 1

InputHandleIndex: 0 (Server object handle = 0x00000160)

FolderId: 0001-000001420e44

Note This is the FID ([\[MS-OXCADATA\]](#) 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: 0

InputHandleIndex: 2 (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 ([\[MS-OXCADATA\]](#) 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)

IsGhosed: 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
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 ([\[MS-OXCADATA\]](#) 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 ([\[MS-OXCDATA\]](#) 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 [2.2.8](#)) 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-OXCADATA] 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 [2.2.8](#)): 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 [\[MS-OXCFOLD\]](#).

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
- Microsoft Exchange Server 2016
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016

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](#): Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not support the Quick Contacts folder, the IM Contacts List folder, the Contacts Search folder, and the Document Libraries folder. Exchange 2003, Exchange 2007, Office Outlook 2003, and Office Outlook 2007 support the Conversation Action Settings and Suggested Contacts folders, but do not recognize them as special folders. Exchange 2003 and Office Outlook 2003 support the RSS Feeds, Tracked Mail Processing, and To-Do folders, but do not recognize them as special folders.

<2> [Section 2.2.5.1](#): Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010, do not support the RSF_PID_CONTACT_SEARCH, RSF_PID_BUDDYLIST_PDLS, and RSF_PID_BUDDYLIST_CONTACTS values for the **PersistID** field.

<3> [Section 3.1.1.1](#): Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not support the Quick Contacts folder, the IM Contacts List folder, the Contacts Search folder, or the Document Libraries folder. Exchange 2003, Exchange 2007, Office Outlook 2003, and Office Outlook 2007 support the Conversation Action Settings and Suggested Contacts folders, but do not recognize them as special folders. Exchange 2003 and Office Outlook 2003 support the RSS Feeds, Tracked Mail Processing, and To-Do folders, but do not recognize them as special folders.

<4> [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 only for the **PidLidReminderSet** ([\[MS-OXPROPS\]](#) section 2.222) and **PidLidRecurring** ([\[MS-OXPROPS\]](#) section 2.216) properties. In addition, it does not include RECURSIVE_SEARCH.

[<5> Section 3.1.4.1.2](#): Office Outlook 2003 and Exchange 2003 do not recognize search criteria for the To-Do folder.

[<6> 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](#) 18
[server](#) 26

Abstract data model – client
[folder hierarchy](#) 18
[overview](#) 18
[search criteria for search special folders](#) 19

Abstract data model – server
[folder hierarchy](#) 18
[overview](#) 18
[search criteria for search special folders](#) 19

[Applicability](#) 10

B

[Binary Identification Properties message](#) 13

C

[Capability negotiation](#) 10
[Change tracking](#) 38

Client
[abstract data model](#) 18
[higher-layer triggered events](#) 21
[initialization](#) 21
[local events](#) 26
[message processing](#) 26
[other local events](#) 26
[overview](#) 18
[sequencing rules](#) 26
[timer events](#) 26
[timers](#) 21

Client - abstract data model
[folder hierarchy](#) 18
[overview](#) 18
[search criteria for search special folders](#) 19

Client - higher layer triggered events
[folder creation](#) 21
[overview](#) 21

Creating a special folder example
[client request for creating a special folder](#) 31
[overview](#) 31
[server response for creating a special folder](#) 33

D

Data model - abstract
[client](#) 18
[server](#) 26

Data model – abstract
[client](#) 18
[server](#) 18

E

Examples - creating a special folder
[client request for creating a special folder](#) 31
[overview](#) 31
[server response for creating a special folder](#) 33

Examples - opening a special folder

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

[Examples - overview](#) 28

F

[Fields - vendor-extensible](#) 10

G

[Glossary](#) 6

H

Higher layer triggered events - client
[folder creation](#) 21
[overview](#) 21

Higher layer triggered events - server
[folder creation](#) 21
[overview](#) 21

Higher-layer triggered events
[client](#) 21
[server](#) 27

I

[Identifiers for Special Folders message](#) 13
[Implementer - security considerations](#) 35
[Inbox Identification message](#) 17
[Index of security parameters](#) 35
[Informative references](#) 9

Initialization
[client](#) 21
server ([section 3.1.3](#) 21, [section 3.2.3](#) 27)

[Introduction](#) 6

L

[List of Special Folders message](#) 11

Local events
[client](#) 26
[server](#) 26

M

Message processing
[client](#) 26
server ([section 3.1.5](#) 26, [section 3.2.5](#) 27)

Messages
[Binary Identification Properties](#) 13
[Identifiers for Special Folders](#) 13
[Inbox Identification](#) 17
[List of Special Folders](#) 11
[PidTagAdditionalRenEntryIds Property](#) 14
[PidTagAdditionalRenEntryIdsEx Property](#) 14
[PidTagContainerClass Property](#) 17
[PidTagFreeBusyEntryIds Property](#) 16
[transport](#) 11

N

[Normative references](#) 9

O

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

Other local events
[client](#) 26
[server](#) 27

[Overview \(synopsis\)](#) 10

P

[Parameters - security index](#) 35

[PersistData structure](#) 15

[PersistElement structure](#) 16

[PidTagAdditionalRenEntryIds Property message](#) 14

[PidTagAdditionalRenEntryIdsEx Property message](#) 14

[PersistData structure](#) 15

[PersistElement structure](#) 16

[PidTagContainerClass Property message](#) 17

[PidTagFreeBusyEntryIds Property message](#) 16

[Preconditions](#) 10

[Prerequisites](#) 10

[Product behavior](#) 36

R

[References](#) 9

[informative](#) 9

[normative](#) 9

[Relationship to other protocols](#) 10

S

Security
[implementer considerations](#) 35
[parameter index](#) 35

Sequencing rules
[client](#) 26
server ([section 3.1.5](#) 26, [section 3.2.5](#) 27)

Server
[abstract data model](#) 26
[higher-layer triggered events](#) 27
initialization ([section 3.1.3](#) 21, [section 3.2.3](#) 27)
[local events](#) 26
message processing ([section 3.1.5](#) 26, [section 3.2.5](#) 27)
[other local events](#) 27
overview ([section 3.1](#) 18, [section 3.2](#) 26)
sequencing rules ([section 3.1.5](#) 26, [section 3.2.5](#) 27)
timer events ([section 3.1.6](#) 26, [section 3.2.6](#) 27)
timers ([section 3.1.2](#) 21, [section 3.2.2](#) 27)

Server - abstract data model
[folder hierarchy](#) 18
[overview](#) 18
[search criteria for search special folders](#) 19

Server - higher layer triggered events
[folder creation](#) 21
[overview](#) 21

[Standards assignments](#) 10

T

Timer events
[client](#) 26
server ([section 3.1.6](#) 26, [section 3.2.6](#) 27)

Timers
[client](#) 21
server ([section 3.1.2](#) 21, [section 3.2.2](#) 27)

[Tracking changes](#) 38

[Transport](#) 11

Triggered events - client
[folder creation](#) 21
[overview](#) 21

Triggered events - higher-layer
[client](#) 21
[server](#) 27

Triggered events - server
[folder creation](#) 21
[overview](#) 21

V

[Vendor-extensible fields](#) 10

[Versioning](#) 10