[MS-OXODLGT]:

Delegate Access Configuration Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional

development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	Major	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	1.0.1	Editorial	Revised and edited technical content.
9/3/2008	1.0.2	Editorial	Revised and edited technical content.
12/3/2008	1.0.3	Editorial	Updated IP notice.
4/10/2009	2.0	Major	Updated technical content for new product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	3.1.0	Minor	Updated the technical content.
2/10/2010	4.0.0	Major	Updated and revised the technical content.
5/5/2010	4.1.0	Minor	Updated the technical content.
8/4/2010	4.2	Minor	Clarified the meaning of the technical content.
11/3/2010	4.2	No change	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	4.3	Minor	Clarified the meaning of the technical content.
8/5/2011	5.0	Major	Significantly changed the technical content.
10/7/2011	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	6.0	Major	Significantly changed the technical content.
4/27/2012	7.0	Major	Significantly changed the technical content.
7/16/2012	7.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	7.1	Minor	Clarified the meaning of the technical content.
2/11/2013	7.1	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	8.0	Major	Significantly changed the technical content.
11/18/2013	8.1	Minor	Clarified the meaning of the technical content.
2/10/2014	8.1	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	9.0	Major	Significantly changed the technical content.
7/31/2014	9.0	No Change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
10/30/2014	9.1	Minor	Clarified the meaning of the technical content.
3/16/2015	10.0	Major	Significantly changed the technical content.
5/26/2015	10.0	No Change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Delegate Access Configuration Protocol allows a user to delegate the responsibility for his or her **mailbox** to another user.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

action: A discrete operation that is executed on an incoming **Message object** when all conditions in the same **rule** are TRUE. A rule contains one or more actions.

Address Book object: An entity in an address book that contains a set of attributes (1), each attribute with a set of associated values.

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

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.

Calendar special folder: A Calendar folder that is in a user's **mailbox** and in which meetings are created by default.

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

delegate data folder: A **special folder** that contains the Delegate Information object.

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

delegate rule: A **server-side rule** that is used to send mail to delegates on behalf of a delegator.

delegator: A user or resource for which another user or resource has permission to act on its behalf.

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

handle: Any token that can be used to identify and access an object such as a device, file, or a window.

informational update: A Meeting Update object that includes a change that does not require attendees to respond again, such as additional agenda details.

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

Meeting Request object: A **Message object** that represents an invitation from the meeting organizer to an attendee.

Meeting Response object: A **Message object** that represents an attendee’s response to a meeting organizer's invitation. The response indicates whether the attendee accepted,

tentatively accepted, or declined the meeting request. The response can include a proposed new date or time for the meeting.

meeting-related object: A **Message object** that represents a relay of information between a meeting organizer and an attendee. It can be any of the following: **Meeting Request object**, Meeting Update object, Meeting Cancellation object, or **Meeting Response object**.

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.

multivalue property: A property that can contain multiple values of the same type.

permission: A rule that is associated with an object and that regulates which users can gain access to the object and in what manner. See also rights.

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.

remote procedure call (RPC): A context-dependent term commonly overloaded with three meanings. Note that much of the industry literature concerning RPC technologies uses this term interchangeably for any of the three meanings. Following are the three definitions: (*) The runtime environment providing remote procedure call facilities. The preferred usage for this meaning is "RPC runtime". (*) The pattern of request and response message exchange between two parties (typically, a client and a server). The preferred usage for this meaning is "RPC exchange". (*) A single message from an exchange as defined in the previous definition. The preferred usage for this term is "RPC message". For more information about RPC, see [C706].

remote user: A user who has a persistent identity within an enterprise and is connected from outside the enterprise network boundary.

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 (2) 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 response: See ROP response buffer.

rule: An item that defines a condition and an action. The condition is evaluated for each **Message object** as it is delivered, and the action is executed if the new Message object matches the condition.

send on behalf: A special permission that is granted to a **delegate**. It allows the delegate to send **Message objects** representing the **delegator**.

server-side rule: A **rule** for which all actions are executed by a server.

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

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

task request: A **Message object** that is used to issue a task assignment.

Unicode: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The **Unicode** standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

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-NSPI] Microsoft Corporation, "Name Service Provider Interface (NSPI) Protocol".

[MS-OXCDATA] Microsoft Corporation, "Data Structures".

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

[MS-OXCMAPIHTTP] Microsoft Corporation, "Messaging Application Programming Interface (MAPI) Extensions for HTTP".

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

[MS-OXCPERM] Microsoft Corporation, "Exchange Access and Operation Permissions Protocol".

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

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

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

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

[MS-OXDISCO] Microsoft Corporation, "Autodiscover HTTP Service Protocol".

[MS-OXOABK] Microsoft Corporation, "Address Book Object Protocol".

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

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

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

[MS-OXOSFLD] Microsoft Corporation, "Special Folders 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-OXPROTO] Microsoft Corporation, "Exchange Server Protocols System Overview".

1.3 Overview

The Delegate Access Configuration Protocol allows a **delegator** in an organization to delegate responsibility for several activities that are commonly performed on objects in the delegator's mailbox. The protocol also allows a delegator to configure delivery of **Meeting Request objects** directly to the **delegate**.

To enable a delegate to perform these activities, the delegator grants the delegate **permissions** to the resources required by the activity being performed. After permissions have been granted, the delegate is able to access the delegator's mailbox to complete the actions.

1.3.1 Granting Delegate Permissions

Three levels of permissions are commonly granted to a delegate: reviewer, author, and editor. These permissions are set on a specific set of **special folders**. The delegator sets the level of permissions based on the activities the delegate will be performing, as follows:

- Reviewer permissions give the delegate read-only access to items.
- Author permissions allow the delegate to read all items, create new items, and delete and modify the items that the delegate creates.
- Editor permissions provide full control of all items to the delegate.

Additionally, the delegate can be granted permission to send items on behalf of the delegator. This level of permission allows the delegate to respond to **Message objects**, manage **meeting-related objects**, and/or manage **Task objects**.

1.3.2 Accessing Delegator Information

To access the delegator's information, a delegate identifies and logs on to the delegator's mailbox. The delegate then identifies the special folder required to complete the action, opens the delegator's special folder, and manipulates items (for example, creates or modifies appointments) to complete the task.

1.3.3 Acting on Behalf of a Delegator

When the delegate sends messages on behalf of the delegator, the delegate's client sets properties on the Message object to indicate that the message is being sent on behalf of the delegator. The server then validates that the delegate has the appropriate permission to send on behalf of the delegator.

It is also possible for the delegate to receive meeting-related objects on behalf of the delegator. These objects can be acted on only if the delegate has the appropriate permission to the delegator's **Calendar special folder** and permission to send mail on behalf of the delegator. Both of these permissions are required to properly process and respond to meeting-related objects.

1.4 Relationship to Other Protocols

The Delegate Access Configuration Protocol depends on the following protocols:

- Message and Attachment Object Protocol, as described in [MS-OXCMSG].
- Folder Object Protocol, as described in [MS-OXCFOLD].
- Exchange Access and Operation Permissions Protocol, as described in [MS-OXCPERM].
- Email Rules Protocol, as described in [MS-OXORULE].
- Email Object Protocol, as described in [MS-OXOMSG].
- Address Book Object Protocol, as described in [MS-OXOABK].
- Appointment and Meeting Object Protocol, as described in [MS-OXOCAL].
- Task-Related Objects Protocol, as described in [MS-OXOTASK].

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

1.5 Prerequisites/Preconditions

In the case of a delegator, this protocol assumes that the client has previously resolved the name of the delegator, as described in [MS-NSPI], logged on to the server, and acquired a **handle** to the mailbox of the delegator.

In the case of the delegate, this protocol assumes that the messaging client has previously resolved the name of the delegate, as described in [MS-NSPI].

1.6 Applicability Statement

This protocol is implemented so that a user can manipulate the objects in another user's mailbox, send mail on another user's behalf, and/or manage meeting and **task requests** for another user.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol uses the protocols specified in [MS-OXCFOLD], [MS-OXCMSG], [MS-OXCPERM], [MS-OXOMSG], [MS-OXOABK], and [MS-OXORULE] as its underlying transport mechanism.

2.2 Message Syntax

This protocol uses the structures specified in [MS-OXCDATA] and the properties specified in [MS-OXPROPS] as the low-level syntax through which the following property/value pairs are encoded. For more details about the values stored in these properties, see section 3.

2.2.1 Delegate Data Folder

The **delegate data folder** is a special folder residing under the **Root folder** that contains the **Delegate Information object**.

2.2.1.1 Common Properties

In addition to Folder object properties specified in [MS-OXCFOLD] section 2.2.2, the delegate data folder contains the property listed in section 2.2.1.1.1.

2.2.1.1.1 PidTagDisplayName Property

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

The **PidTagDisplayName** property ([MS-OXPROPS] section 2.667) MUST be set to "Freebusy Data".

2.2.2 Delegate Information Object

The Delegate Information object is a special Message object used to store delegate access settings for a delegator. This Delegate Information object is stored in the delegate data folder for the delegator.

Unless otherwise specified, the Delegate Information object adheres to all property constraints specified in [MS-OXPROPS] and [MS-OXCMSG]. A Delegate Information object can also contain other properties, which are defined in [MS-OXPROPS], but these properties have no impact on the Delegate Access Configuration Protocol.

2.2.2.1 Common Properties

In addition to the Message object properties specified in [MS-OXCMSG], the Delegate Information object contains the properties listed in section 2.2.2.1.1 and section 2.2.2.1.2.

2.2.2.1.1 PidTagMessageClass Property

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

The **PidTagMessageClass** property ([MS-OXPROPS] section 2.776) MUST be set to "IPM.Microsoft.ScheduleData.FreeBusy".

2.2.2.1.2 PidTagNormalizedSubject Property

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

The **PidTagNormalizedSubject** property ([MS-OXPROPS] section 2.801) MUST be set to "LocalFreebusy".

2.2.2.2 Delegate Information Properties

2.2.2.2.1 PidTagScheduleInfoDelegatorWantsCopy Property

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

The **PidTagScheduleInfoDelegatorWantsCopy** property ([MS-OXPROPS] section 2.960) indicates whether the delegator wants to receive copies of the meeting-related objects that are sent to the delegate.

This property MUST be set on the Delegate Information object.

2.2.2.2.2 PidTagScheduleInfoDelegatorWantsInfo Property

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

The **PidTagScheduleInfoDelegatorWantsInfo** property ([MS-OXPROPS] section 2.961) indicates whether the delegator wants to receive **informational updates**, as specified in [MS-OXOCAL] section 3.1.4.7.4.1.<1> For more details about informational updates, see [MS-OXOCAL] section 3.1.5.6.

This property MUST be set on the Delegate Information object.

2.2.2.2.3 PidTagScheduleInfoDelegateNames Property

Data type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1)

The **PidTagScheduleInfoDelegateNames** property ([MS-OXPROPS] section 2.958) specifies the names of the delegates. Each entry contains the value of the **PidTagDisplayName** property (section 2.2.1.1.1) of each delegate's **Address Book object**. For details about the Address Book object, see [MS-OXOABK].

This property MAY<2> be accessed and manipulated as a **PtypMultipleString8** ([MS-OXCDATA] section 2.11.1) property, which can cause a loss of fidelity when converting from **Unicode**.

Requirements for this property are specified in section 3.1.4.3.3.

2.2.2.2.4 PidTagScheduleInfoDelegateNamesW Property

Data type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1)

The **PidTagScheduleInfoDelegateNamesW** property ([MS-OXPROPS] section 2.959) specifies the names of the delegates. Each entry contains the value of the **PidTagDisplayName** property (section 2.2.1.1.1) of each delegate's Address Book object. For more details about the Address Book object, see [MS-OXOABK].

This property is accessed and manipulated as a **PtypMultipleString** ([MS-OXCDATA] section 2.11.1) property, preserving the fidelity of Unicode information.

Requirements for this property are specified in section 3.1.4.3.3.

2.2.2.2.5 PidTagScheduleInfoDelegateEntryIds Property

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

The **PidTagScheduleInfoDelegateEntryIds** property ([MS-OXPROPS] section 2.957) specifies the **EntryIDs** of the delegates. Each entry contains the value of the **PidTagEntryId** property ([MS-OXCPERM] section 2.2.4) of each delegate's Address Book object. For more details about the Address Book object, see [MS-OXOABK].

This property MUST be set on the Delegate Information object.

2.2.2.2.6 PidTagDelegateFlags Property

Data type: **PtypMultipleInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagDelegateFlags** property ([MS-OXPROPS] section 2.658) indicates whether delegates can view Message objects that are marked as private, meaning the **PidTagSensitivity** property ([MS-OXCMSG] section 2.2.1.13) has a value of 0x00000002. Each entry of this property MUST be set to one of the following values.

Flag name	Value	Description
HidePrivate	0x00000000	The delegate SHOULD NOT be allowed to view the Message object that is marked as private.
ShowPrivate	0x00000001	The delegate SHOULD be allowed to view the Message object that is marked as private.

This property MUST be set on the Delegate Information object.

2.2.2.2.7 PidTagScheduleInfoDontMailDelegates Property

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

The **PidTagScheduleInfoDontMailDelegates** property ([MS-OXPROPS] section 2.964) is set to **TRUE** by the client, regardless of user input. The value of this property has no meaning in the context of this protocol.

2.2.3 Delegate Rule

To enable **calendar** workflow scenarios in which delegates receive copies of meeting-related objects that are sent to the delegator, a delegator's client creates a specific type of **server-side rule**, as specified in [MS-OXORULE] section 3.1.4.2.

2.2.3.1 Delegate Rule Properties

The **delegate rule** is specified by setting the properties listed in section 2.2.3.1.1 through section 2.2.3.1.6.

2.2.3.1.1 PidTagRuleState Property

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

The **PidTagRuleState** property ([MS-OXPROPS] section 2.952) MUST be set to 0x00000001.

2.2.3.1.2 PidTagRuleName Property

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

The **PidTagRuleName** property ([MS-OXPROPS] section 2.948) MUST be set to "" (a zero-length string).

2.2.3.1.3 PidTagRuleProvider Property

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

The **PidTagRuleProvider** property ([MS-OXPROPS] section 2.949) MUST be set to "Schedule+ EMS Interface".

2.2.3.1.4 PidTagRuleLevel Property

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

The **PidTagRuleLevel** property ([MS-OXPROPS] section 2.940) MUST be set to 0x00000000.

2.2.3.1.5 PidTagRuleCondition Property

Data type: **PtypRestriction** ([MS-OXCDATA] section 2.11.1)

The **PidTagRuleCondition** property ([MS-OXPROPS] section 2.935) contains a **restriction** of type RES_AND with the following restrictions:

- A restriction of type RES_CONTENT that limits a table view to rows that include the string "IPM.Schedule.Meeting" in the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) column. The level of precision, which is specified in the **FuzzyLevelLow** field of the **ContentRestriction** structure, is set to FL_PREFIX.
- A restriction of type RES_NOT with a restriction of type RES_EXIST that specifies the **PidTagDelegatedByRule** property ([MS-OXPROPS] section 2.657).
- A restriction of type RES_OR with the following restrictions:
    - A restriction of type RES_NOT with a restriction of type RES_EXIST that specifies the **PidTagSensitivity** property ([MS-OXCMSG] section 2.2.1.13)
    - A restriction of type RES_PROPERTY that specifies a comparison of the value of the **PidTagSensitivity** property to the value 0x00000002, indicating a private message. The relationship operator, which is specified in the **RelOp** field of the **PropertyRestriction** structure, is set to RELOP_NE.

For more details about restrictions, see [MS-OXCDATA] section 2.12.

2.2.3.1.6 PidTagRuleActions Property

Data type: **PtypRuleAction** ([MS-OXCDATA] section 2.11.1)

The **PidTagRuleActions** property ([MS-OXPROPS] section 2.933) specifies the delegate's **rule actions**, which are used to perform the following actions:

1. Send copies of meeting-related objects to delegates, by using the OP_DELEGATE action, as specified in [MS-OXORULE] section 2.2.5.1.2.4.
2. Delete the delegator's copy of meeting-related objects, by using the OP_DELETE action, as specified in [MS-OXORULE] section 2.2.5.1.2.7.

Sections 3.1.4.3.2.1 and 3.1.4.3.5 specify when these actions are specified in the delegate rule. For more details about rule actions, see [MS-OXORULE] section 2.2.5.

3 Protocol Details

3.1 Delegator's Client Details

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.

All abstract data model (ADM) elements maintained by the delegator's client are prefixed with "Delegator".

This protocol includes the following ADM type:

Mailbox, as specified in [MS-OXCMSG] section 3.1.1.2.

The following ADM types are defined in this section:

Delegator.Mailbox.DataFolder: A special folder that contains the Delegator.Mailbox.InformationObject ADM element. This folder is referred to as the delegate data folder in this document, and is specified in section 2.2.1.

Delegator.Mailbox.InformationObject: A Message object that contains properties specifying delegate access settings for resources in a delegator's mailbox. This object is referred to as the Delegate Information object in this document, and is specified in section 2.2.2.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Creating a Delegate Data Folder

The delegator's client creates the delegate data folder under the delegator's Root folder by sending the **RopCreateFolder** **remote operation (ROP)** ([MS-OXCROPS] section 2.2.4.2). The client then sets properties on the delegate data folder as specified in section 2.2.1 by sending the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6).

In addition, the EntryID for the delegate data folder is stamped in the **PidTagFreeBusyEntryIds** property ([MS-OXOSFLD] section 2.2.6).

3.1.4.2 Creating a Delegate Information Object

The delegator's client creates the Delegate Information object under the delegator's delegate data folder by sending the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2). The client then sets properties on the Delegate Information object as specified in section 2.2.2 by sending the

RopSetProperties ROP ([MS-OXCROPS] section 2.2.8.6). Finally, the client commits the changes to the Delegate Information object by sending the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3).

In addition, the EntryID for the Delegate Information object is stamped in the **PidTagFreeBusyEntryIds** property ([MS-OXOSFLD] section 2.2.6).

3.1.4.3 Creating a Delegation Relationship

The delegator's client establishes the delegation relationship by setting permissions and individual preferences for delegates, as well as by setting global delegate preferences. The delegator's client performs these tasks by using the steps specified in sections 3.1.4.3.1 through 3.1.4.3.5.

3.1.4.3.1 Setting Send on Behalf Permissions

The delegator's client SHOULD grant **send on behalf** permission to every delegate and stop creating a delegation relationship if send on behalf permissions cannot be granted. Granting send on behalf permissions is accomplished by adding the value of the **PidTagEntryId** property ([MS-OXCPERM] section 2.2.4) of the delegate's Address Book object to the **PidTagAddressBookPublicDelegates** property ([MS-OXOABK] section 2.2.5.5) of the delegator's Address Book object. This value is added by using the **NspiModLinkAtt** method, as specified in [MS-NSPI].

A client SHOULD<3> support delegation for a **remote user** if the remote user indicates that it supports sharing of the delegator's information. The remote user supports sharing of the delegator's information if the remote user's Address Book object has bit S set in the **PidTagDisplayTypeEx** property ([MS-OXOABK] section 2.2.3.12).

3.1.4.3.2 Setting Delegate Folder Permissions

The roles supported by this protocol are specified in the following table. Roles are a specific set of flags for the value of the **PidTagMemberRights** property ([MS-OXCPERM] section 2.2.7), which is used when setting folder permissions.

Role name	Numeric value	PidTagMemberRights flags	Description
None	0x00000000	None	The delegate is not able to view, create, modify, or delete any contents from the folder.
Reviewer	0x00000001	**ReadAny**	The delegate is able to view contents of the folder. However, the delegate is not able to create, modify, or delete any contents from the folder.
Author	0x0000001B	**ReadAny** **Create** **EditOwned** **DeleteOwned**	The delegate is able to view contents of the folder. In addition, the delegate is able to create, modify, and delete any items that this delegate created.
Editor	0x0000007B	**ReadAny** **Create** **EditOwned** **DeleteOwned** **EditAny** **DeleteAny**	The delegate is able to view, create, modify, and delete any items in the folder.

The delegator's client specifies a role for each of the following special folders, as specified in [MS-OXOSFLD]:

- Calendar
- Inbox
- Tasks
- Contacts
- Notes
- Journal

The delegator's special folders listed in sections 3.1.4.3.2.1 and 3.1.4.3.2.2 have additional constraints.

3.1.4.3.2.1 Additional Constraints for Calendar Folder

For a delegate to process meeting-related objects on behalf of the delegator, a client MUST:

1. Grant to the delegate the Author or Editor role on the delegator's Calendar special folder, as specified in [MS-OXCPERM] section 3.1.4.2.

   Note Granting the Author role allows the delegate to view contents of the folder. In addition, the delegate is able to create, modify, and delete any items created by this delegate.

2. Grant to the delegate the Editor role on the delegate data folder, as specified in [MS-OXCPERM] section 3.1.4.2.

   Note If the delegate data folder doesn't exist, the delegator's client creates it.

Additionally, if a delegate is to receive meeting-related objects on behalf of the delegator, a client MUST:

1. Grant to the delegate the Editor role on the delegator's Calendar special folder, as specified in [MS-OXCPERM] section 3.1.4.2.
2. Grant send on behalf permission to the delegate.
3. Add the OP_DELEGATE action, as specified in [MS-OXORULE] section 2.2.5.1.2.4, including all delegates that will receive meeting-related objects on behalf the delegator.

3.1.4.3.2.2 Additional Constraints for the Tasks Folder

If a delegate is to process task requests on behalf of the delegator, a client MUST:

1. Grant send on behalf permission to the delegate.
2. Grant, to the delegate, the Author or Editor role on the delegator's Task special folder, as specified in [MS-OXCPERM] section 3.1.4.2. Note that granting the Author role will allow the delegate to act only on new items and on items that the delegate creates.

3.1.4.3.3 Setting Individual Delegate Preferences

For each delegate being specified, the delegator's client MUST:

1. Specify the delegate's name, using the value of the **PidTagDisplayName** property (section 2.2.1.1.1) for the Address Book object of the delegate. This value MUST be specified as an entry

in either the **PidTagScheduleInfoDelegateNamesW** property (section 2.2.2.2.4) or the **PidTagScheduleInfoDelegateNames** property (section 2.2.2.2.3). The value MAY be specified as an entry in both the **PidTagScheduleInfoDelegateNamesW** and **PidTagScheduleInfoDelegateNames** properties. If a client uses only one of these properties, it SHOULD use the **PidTagScheduleInfoDelegateNamesW** property,<4> and it MUST use the same property for all delegates.

2. Specify the value of the **PidTagEntryId** property ([MS-OXCPERM] section 2.2.4) for the Address Book object of the delegate as an entry in the **PidTagScheduleInfoDelegateEntryIds** property (section 2.2.2.2.5).

3. Specify whether the delegate can view the delegator's Message objects that are marked as private as an entry in the **PidTagDelegateFlags** property (section 2.2.2.2.6). The value of the **PidTagSensitivity** property ([MS-OXCMSG] section 2.2.1.13) on a Message object that has been marked as private is 0x00000002. The ability to view Message objects that have been marked as private is applicable to all folders for which the delegate has a role of Reviewer, Author, or Editor, as specified in section 3.1.4.3.2.

The client then MUST send the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) with the values of the three properties generated by steps 1-3 in this secton, and commit the changes by sending the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3).

Note Because each **multivalue property** specified above has one entry for each delegate, they are correlated by their index into these multivalue properties, and are only valid if an entry is present for all three properties.

3.1.4.3.4 Setting Global Delegate Preferences

The following preferences are specific to calendar workflows and are applicable to all delegates. These preferences are used in conjunction with rules and allow a delegator to have greater control over which meeting-related objects are delivered to the delegator, the delegate, or both.

The following properties are set on the Delegate Information object by sending the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) and committed by sending the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3).

For more details about calendar workflows, see [MS-OXOCAL].

3.1.4.3.4.1 Setting the PidTagScheduleInfoDelegatorWantsCopy Property

The value of the **PidTagScheduleInfoDelegatorWantsCopy** property (section 2.2.2.2.1) is set to **TRUE** in the following cases:

- A delegator wants to receive meeting-related objects in their own mailbox.
- No delegates will receive meeting-related objects on behalf of the delegator.

Otherwise, the value MUST be set to **FALSE**.

3.1.4.3.4.2 Setting the PidTagScheduleInfoDelegatorWantsInfo Property

The value of the **PidTagScheduleInfoDelegatorWantsInfo** property (section 2.2.2.2.2) is set to **TRUE** when a delegator wants to receive informational updates, as specified in [MS-OXOCAL] section 3.1.4.7.4. Otherwise, it MUST be set to **FALSE**.

The value of this property is set to **FALSE** if the value of the **PidTagScheduleInfoDelegatorWantsCopy** property (section 2.2.2.2.1) is set to **FALSE**. For more details about how this property is used, see [MS-OXOCAL] section 3.1.5.6.

The following table illustrates valid combinations of the **PidTagScheduleInfoDelegatorWantsCopy** (WC) and **PidTagScheduleInfoDelegatorWantsInfo** (WI) properties.

WC	WI	Description
TRUE	**TRUE**	The delegator wants to receive copies and would like these copies to be informational updates when applicable.
TRUE	**FALSE**	The delegator wants to receive copies.
FALSE	**TRUE**	Invalid, as the delegator cannot receive informational updates unless the delegator receives copies.
FALSE	**FALSE**	The delegator doesn't want to receive copies or informational updates.

3.1.4.3.5 Setting the Delegate Rule

The delegator's client creates or updates the delegate rule, as specified in section 2.2.3, if, while creating the delegation relationship:

1. Any delegate is receiving Meeting Request objects on behalf of the delegator, because this adds the OP_DELEGATE action, as specified in [MS-OXORULE] section 2.2.5.1.2.4, to the delegate rule.

2. The delegator has elected not to receive copies of Meeting Request objects, because this adds the OP_DELETE action, as specified in [MS-OXORULE] section 2.2.5.1.2.7, to the delegate rule.

The client first retrieves the existing rules by sending the **RopGetRulesTable** ROP ([MS-OXCROPS] section 2.2.11.2 and then modifies the existing rules by sending the **RopModifyRules** ROP ([MS-OXCROPS] section 2.2.11.1).

The OP_DELETE action MUST NOT be present in the delegate rule when the **PidTagScheduleInfoDelegatorWantsCopy** property (section 2.2.2.2.1) has a value of **TRUE**. The value of this property is set to **TRUE** if the value of the **PidTagScheduleInfoDelegatorWantsInfo** property (section 2.2.2.2.2) is set to **TRUE**. The delegator adds the OP_DELETE action to the delegate rule when the **PidTagScheduleInfoDelegatorWantsCopy** property has a value of **FALSE**.

3.1.5 Message Processing Events and Sequencing Rules

The delegator's client depends on the message processing events and sequencing rules of the underlying Message and Attachment Object Protocol, as specified in [MS-OXCMSG] section 3.1.5.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Delegate's Client Details

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.

This protocol includes the following abstract data model (ADM) type:

Mailbox, as specified in [MS-OXCMSG] section 3.1.1.2.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Opening the Delegator's Special Folder

To open a special folder belonging to a delegator, the delegate's client takes the following steps:

1. Identify the delegator's server.

   The delegate's client identifies the delegator's server by using properties from the Address Book object that represents the delegator. The delegate's client retrieves the value of the **PidTagAddressBookHomeMessageDatabase** property ([MS-OXOABK] section 2.2.4.37) by using the **NspiGetProps** method, as specified in [MS-NSPI]. If the Address Book object has a value in the **PidTagAddressBookHomeMessageDatabase** property, this value is used to identify the delegator's server. Otherwise, if the delegator is a remote user, the client SHOULD<5> use the Autodiscover HTTP Service protocol to determine the correct server settings, as specified in [MS-OXDISCO]. A remote user can be identified by examining the **PidTagDisplayType** property ([MS-OXOABK] section 2.2.3.11) of the user's Address Book object. If the **PidTagDisplayType** property has the value **DT_REMOTE_MAILUSER**, then the delegator is a remote user.

2. Identify the delegator's mailbox.

   The delegate's client uses the value of the **PidTagAddressBookProxyAddresses** property ([MS-OXOABK] section 2.2.3.23) and value of the **PidTagEmailAddress** property ([MS-OXOABK] section 2.2.3.14) returned by the **NspiGetProps** method in step 1 to complete this step. If the Address Book object has a valid **PidTagAddressBookProxyAddresses** property and this property contains "MAILBOX" or "EX", then one of these strings is used, in the following order, to identify the delegator's mailbox:

   1. The "MAILBOX" entry, if present
   2. The "EX" entry, if present

If the Address Book object does not contain a valid **PidTagAddressBookProxyAddresses** property, or this property doesn't contain "MAILBOX" or "EX" entries, then the mailbox is identified by the value of the **PidTagEmailAddress** property.

3. Establish a connection to the delegator's server and log on to the delegator's mailbox.

   The delegate's client connects to the delegator's server either by using the **EcDoConnectEx remote procedure call (RPC)**, as specified in [MS-OXCRPC] section 3.1.4.1, or by using a **Connect** request type,<6> as specified in [MS-OXCMAPIHTTP] section 2.2.4.1, and then connects to the delegator's mailbox with the **HOME_LOGON** and **TAKE_OWNERSHIP** flags, as specified in [MS-OXCSTOR] section 2.2.1.1, by using the **RopLogon** ROP ([MS-OXCROPS] section 2.2.3.1).

4. Identify and open the delegator's special folder.

   After the client has logged on to the delegator's mailbox, the delegate's client identifies the required special folder within the delegator's mailbox. For more details about special folders, see [MS-OXOSFLD].

   The client opens the delegator's special folder, as specified in [MS-OXCFOLD] section 2.2.1.1, by using the **RopOpenFolder** ROP ([MS-OXCROPS] section 2.2.4.1).

3.2.4.2 Displaying the Delegator Contents

The delegate's client SHOULD NOT show Message objects the delegator has marked as private, meaning the **PidTagSensitivity** property ([MS-OXCMSG] section 2.2.1.13) has a value of 0x00000002, unless the **PidTagDelegateFlags** property (section 2.2.2.2.6) for the delegate has a value of **ShowPrivate**, indicating that the delegator wants Message objects that are marked private to be visible to the delegate.

3.2.4.3 Sending on Behalf of the Delegator

When sending Message objects on behalf of the delegator, the delegate's client populates the from properties, as specified in [MS-OXOMSG] section 2.2.1, with information from the delegator.

3.2.5 Message Processing Events and Sequencing Rules

The delegate's client can identify that a Message object has been received on behalf of the delegator when the received representing properties are present and different from the recipient properties. The received representing properties are specified in [MS-OXOMSG] section 2.2.1.

If the received representing properties are present, they take precedence over recipient properties to identify the delegator Address Book object.

When the delegate's client is processing a meeting-related object received on behalf of a delegator, the delegate's client creates or accesses the corresponding **Calendar object** in the delegator's special folder. For more details about processing a meeting-related object, see [MS-OXOCAL].

When the delegate's client is processing a task request on behalf of a delegator, the delegate's client creates the corresponding Task object in the delegator's special folder. For more details about processing a task request, see [MS-OXOTASK].

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

3.3 Server Details

3.3.1 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.

This protocol includes the following ADM type:

Mailbox, as specified in [MS-OXCMSG] section 3.2.1.2.

3.3.2 Timers

None.

3.3.3 Initialization

None.

3.3.4 Higher-Layer Triggered Events

3.3.4.1 Opening Delegator Root Folder

The server MUST provide read access to a delegator's Root folder and its properties, because the delegate needs to obtain the folder ID (FID), as specified in [MS-OXCDATA] section 2.2.1.1, for the delegator's special folders, as specified in [MS-OXOSFLD].

3.3.4.2 Submitting On Behalf Of Delegator

For non-meeting-related objects, a server MUST validate that the delegate, which is specified in the sender properties of the actual sender, has access to send on behalf of the delegator, which is specified in the from properties of the represented sender. For more details about sender properties and from properties, see [MS-OXOMSG] section 2.2.1.

3.3.4.3 Message Delivery to Delegator

A server MUST process the delegate rule when present, as specified in [MS-OXORULE].

When present, the delegator's rule actions accomplish the following:

- The OP_DELEGATE action, as specified in [MS-OXORULE] section 2.2.5.1.2.4, ensures that meeting-related objects are delivered to the delegate, and that these meeting-related objects are on behalf of the delegator, who is identified by received representing properties.
- The OP_DELETE action, as specified in [MS-OXORULE] section 2.2.5.1.2.7, ensures that the delegator does not receive unwanted Meeting Request objects.

3.3.4.4 Creating, Modifying, or Deleting Message Objects

When a delegate attempts to create, modify, or delete a Message object that resides in the delegator mailbox, a server MUST ensure that the delegate has sufficient permissions to the folder, as specified in [MS-OXCPERM] section 3.2.4.1.

Additionally, a server MUST track the creator and last modifier of any Message object by using the **PidTagCreatorName** ([MS-OXPROPS] section 2.647), **PidTagCreatorEntryId** ([MS-OXCMSG] section 2.2.1.31), **PidTagLastModifierName** ([MS-OXCPRPT] section 2.2.1.5), and **PidTagLastModifierEntryId** ([MS-OXCMSG] section 2.2.1.32) properties.

3.3.5 Message Processing Events and Sequencing Rules

The server depends on the message processing events and sequencing rules of the underlying Message and Attachment Object Protocol, as specified in [MS-OXCMSG] section 3.2.5.

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

4 Protocol Examples

4.1 Create Delegation Relationship with Multiple Delegates

The following example shows the ROP traces for the delegator named delegator1 creating a relationship with the delegate named delegate1 and the delegate named delegate2. The ROP traces in this example are truncated to more easily illustrate ROP information that is specific to this protocol.

This example shows the following steps when setting up the delegation relationship:

1. Identifying the delegator's special folders.
2. Setting permission to send on behalf of the delegator.
3. Updating the Delegate Information object.
4. Updating the delegate rule.
5. Setting permissions for the delegator's special folders.

4.1.1 Identify Delegator Special Folders

To identify the delegator's special folders, the client logs on to the delegator's mailbox and queries for the special folder properties from the Inbox special folder, which is provided in response to the **RopLogon** ROP ([MS-OXCROPS] section 2.2.3.1) request. For more information about ROPs, see [MS-OXCROPS].

```
RopLogon
    ROPid: 0xFE
    LogonFlags: 0x01 Private
    OpenFlags: 0x0100040C HOME_LOGON TAKE_OWNERSHIP NO_MAIL CLI_WITH_PER_MDB_FIX
    Private Logon    LegacyDN: /o=First Organization/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=delegator1
RopLogon
    ROPid: 0xFE
    FolderArray:
        ...
        FolderID  4: 0001-00174ea8cd9d IPM subtree
        FolderID  5: 0001-00174ea8cda0 Inbox
        FolderID  6: 0001-00174ea8cda1 Outbox
        ...
RopOpenFolder
    ROPid: 0x02
    FID: 0001-00174ea8cda0
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 1 (HSOT=0x00000085)
    ReturnValue: ecNone (success) (0x00000000)
RopGetPropertiesSpecific
    ROPid: 0x07
    HandleIndex: 2 (HSOT=0x00000085)
    ReturnValue: ecNone (success) (0x00000000)
    PropCount: 29
        ...
        0x36D00102 PidTagIpmAppointmentEntryId         PtypBinary      46 Bytes
            0000: 00 00 00 00 C3 E1 78 57-96 52 CE 46 A3 53 B3 E0 ......xW.R.F.S..
            0010: 7F 9B 97 BC 01 00 8B 8D-B1 82 AF 2E D0 48 93 47 .............H.G
            0020: 07 ED 54 48 84 0F 00 17-4E A8 9C 98 00 00       ..TH....N.....
        0x36D10102 PidTagIpmContactEntryId             PtypBinary      46 Bytes
            0000: 00 00 00 00 C3 E1 78 57-96 52 CE 46 A3 53 B3 E0 ......xW.R.F.S..
            0010: 7F 9B 97 BC 01 00 8B 8D-B1 82 AF 2E D0 48 93 47 .............H.G
```

```
        0020: 07 ED 54 48 84 0F 00 17-4E A8 9C 99 00 00       ..TH....N.....
    0x36D20102 PidTagIpmJournalEntryId             PtypBinary      46 Bytes
        0000: 00 00 00 00 C3 E1 78 57-96 52 CE 46 A3 53 B3 E0 ......xW.R.F.S..
        0010: 7F 9B 97 BC 01 00 8B 8D-B1 82 AF 2E D0 48 93 47 .............H.G
        0020: 07 ED 54 48 84 0F 00 17-4E A8 9C 9B 00 00       ..TH....N.....
    0x36D30102 PidTagIpmNoteEntryId                PtypBinary      46 Bytes
        0000: 00 00 00 00 C3 E1 78 57-96 52 CE 46 A3 53 B3 E0 ......xW.R.F.S..
        0010: 7F 9B 97 BC 01 00 8B 8D-B1 82 AF 2E D0 48 93 47 .............H.G
        0020: 07 ED 54 48 84 0F 00 17-4E A8 9C 9C 00 00       ..TH....N.....
    0x36D40102 PidTagIpmTaskEntryId                PtypBinary      46 Bytes
        0000: 00 00 00 00 C3 E1 78 57-96 52 CE 46 A3 53 B3 E0 ......xW.R.F.S..
        0010: 7F 9B 97 BC 01 00 8B 8D-B1 82 AF 2E D0 48 93 47 .............H.G
        0020: 07 ED 54 48 84 0F 00 17-4E A8 9C 9D 00 00       ..TH....N.....
    0x36E41102 PidTagFreeBusyEntryIds              PtypMultipleBinary
        PtypMultipleBinary[0] (0 bytes):
        PtypMultipleBinary[1] (70 bytes):
        0000: 00 00 00 00 C3 E1 78 57-96 52 CE 46 A3 53 B3 E0 ......xW.R.F.S..
        0010: 7F 9B 97 BC 07 00 8B 8D-B1 82 AF 2E D0 48 93 47 .............H.G
        0020: 07 ED 54 48 84 0F 00 17-4E A8 9C D4 00 00 8B 8D ..TH....N.......
        0030: B1 82 AF 2E D0 48 93 47-07 ED 54 48 84 0F 00 17 .....H.G..TH....
        0040: 4E A8 E7 68 00 00                              N..h..
        PtypMultipleBinary[2] (0 bytes):
        PtypMultipleBinary[3] (46 bytes):
        0000: 00 00 00 00 C3 E1 78 57-96 52 CE 46 A3 53 B3 E0 ......xW.R.F.S..
        0010: 7F 9B 97 BC 01 00 8B 8D-B1 82 AF 2E D0 48 93 47 .............H.G
        0020: 07 ED 54 48 84 0F 00 17-4E A8 9C D4 00 00       ..TH....N.....
    ...
```

4.1.2 Set Send on Behalf Permissions

Then, the delegator sets send on behalf permission by using the **NspiModLinkAtt** method, as described in [MS-NSPI].

```
NspiModLinkAtt
    ...
    0x8015000D PidTagAddressBookPublicDelegates
    ...
    PtypMultipleBinary
        PtypMultipleBinary[0] (128 bytes):
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 6F 3D 46 +/........../o=F
            0020: 69 72 73 74 20 4F 72 67-61 6E 69 7A 61 74 69 6F irst Organizatio
            0030: 6E 2F 6F 75 3D 45 78 63-68 61 6E 67 65 20 41 64 n/ou=Exchange Ad
            0040: 6D 69 6E 69 73 74 72 61-74 69 76 65 20 47 72 6F ministrative Gro
            0050: 75 70 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 up (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 63 6E 3D-52 65 63 69 70 69 65 6E DLT)/cn=Recipien
            0070: 74 73 2F 63 6E 3D 64 65-6C 65 67 61 74 65 32 00 ts/cn=delegate2.
        PtypMultipleBinary[1] (128 bytes):
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 6F 3D 46 +/........../o=F
            0020: 69 72 73 74 20 4F 72 67-61 6E 69 7A 61 74 69 6F irst Organizatio
            0030: 6E 2F 6F 75 3D 45 78 63-68 61 6E 67 65 20 41 64 n/ou=Exchange Ad
            0040: 6D 69 6E 69 73 74 72 61-74 69 76 65 20 47 72 6F ministrative Gro
            0050: 75 70 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 up (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 63 6E 3D-52 65 63 69 70 69 65 6E DLT)/cn=Recipien
            0070: 74 73 2F 63 6E 3D 64 65-6C 65 67 61 74 65 31 00 ts/cn=delegate1.
    ...
```

4.1.3 Update the Delegate Information Object

Updating the Delegate Information object requires two steps: opening the object and updating the properties.

4.1.3.1 Open the Delegator Information Object

The client opens the Delegate Information object by using the message ID (MID), as described in [MS-OXCDATA] section 2.2.1.2, and the FID, as described in [MS-OXCDATA] section 2.2.1.1, which are the second entry and the fourth entry, respectively, in the **PidTagFreeBusyEntryIds** property ([MS-OXOSFLD] section 2.2.6). The value of the **NormalizedSubject** field (in the **RopOpenMessage ROP response** ([MS-OXCROPS] section 2.2.6.1)) can be examined to determine whether the intended object has been opened. In this case, the value is "LocalFreebusy", verifying that the operation opened the correct object.

```
RopOpenMessage
    ROPid: 0x03
    FolderId: 0001-00174ea89cd4
    OpenModeFlags: 0x03 BestAccess rights
    MessageID: 0001-00174ea8e768
RopOpenMessage
    ROPid: 0x03
    HandleIndex: 1 (HSOT=0x00000062)
    ReturnValue: ecNone (success) (0x00000000)
    NormalizedSubject: LocalFreebusy
```

4.1.3.2 Update the Delegator Information Object Properties

The client updates the **PidTagScheduleInfoDelegatorWantsCopy** property (section 2.2.2.2.1) and the **PidTagScheduleInfoDelegatorWantsInfo** property (section 2.2.2.2.2) with the delegator's global settings. In this case, the delegator does want copies of meeting-related objects and would prefer to receive informational meeting-related objects if the client supports this workflow.

In addition, the delegator updates the **PidTagScheduleInfoDelegateNamesW** (section 2.2.2.2.4), **PidTagScheduleInfoDelegateEntryIds** (section 2.2.2.2.5), and **PidTagDelegateFlags** (section 2.2.2.2.6) properties for each delegate. delegate2 is stored in the first entry of these multivalue properties, and delegate1 is stored in the second entry of these multivalue properties. The delegator is only allowing delegate2 to see Message objects that are marked as private, meaning the **PidTagSensitivity** property ([MS-OXCMSG] section 2.2.1.13) on the Message object has a value of 0x00000002. The delegate2 preferences, as well as global delegator preferences, are shown in this example.

```
RopSetProperties
    ROPid: 0x0A
    HandleIndex: 0 (HSOT=0x00000062)
    PropCount: 12 (0x0C)
        ...
        0x6842000B PidTagScheduleInfoDelegatorWantsCopy    PtypBoolean  0x01 (TRUE)
        0x684A101F PidTagScheduleInfoDelegateNamesW        PtypMultipleString
            PtypMultipleString[0]:delegate2
            PtypMultipleString[1]:delegate1
        0x68451102 PidTagScheduleInfoDelegateEntryIds      PtypMultipleBinary
            PtypMultipleBinary[0] (128 bytes):
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 6F 3D 46 +/........../o=F
            0020: 69 72 73 74 20 4F 72 67-61 6E 69 7A 61 74 69 6F irst Organizatio
            0030: 6E 2F 6F 75 3D 45 78 63-68 61 6E 67 65 20 41 64 n/ou=Exchange Ad
            0040: 6D 69 6E 69 73 74 72 61-74 69 76 65 20 47 72 6F ministrative Gro
            0050: 75 70 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 up (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 63 6E 3D-52 65 63 69 70 69 65 6E DLT)/cn=Recipien
            0070: 74 73 2F 63 6E 3D 64 65-6C 65 67 61 74 65 32 00 ts/cn=delegate2.
            PtypMultipleBinary[1] (128 bytes):
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 6F 3D 46 +/........../o=F
            0020: 69 72 73 74 20 4F 72 67-61 6E 69 7A 61 74 69 6F irst Organizatio
            0030: 6E 2F 6F 75 3D 45 78 63-68 61 6E 67 65 20 41 64 n/ou=Exchange Ad
```

```
            0040: 6D 69 6E 69 73 74 72 61-74 69 76 65 20 47 72 6F ministrative Gro
            0050: 75 70 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 up (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 63 6E 3D-52 65 63 69 70 69 65 6E DLT)/cn=Recipien
            0070: 74 73 2F 63 6E 3D 64 65-6C 65 67 61 74 65 31 00 ts/cn=delegate1.
        0x686B1003 PidTagDelegateFlags                    PtypMultipleInteger32
            PtypMultipleInteger32[0]: 1
            PtypMultipleInteger32[1]: 0
        0x684B000B PidTagScheduleInfoDelegatorWantsInfo    PtypBoolean  0x01 (TRUE)
        ...
RopSaveChangesMessage
    ROPid: 0x0C
    LogonIndex: 0
    HandleIndex: 0 (HSOT=0x00000062)
    SaveOptions: 0x0A KeepOpenReadWrite DelayedCall
```

4.1.4 Update the Delegate Rule

The delegator's client updates the delegate rule on the receive folder rule by using the **RopModifyRules** ROP ([MS-OXCROPS] section 2.2.11.1), as described in [MS-OXORULE] section 2.2.1. Given the actions, only delegate1 is receiving meeting-related objects on behalf of the delegator.

```
RopGetReceiveFolder
    ROPid: 0x27
    HandleIndex: 0 (HSOT=0x0000006f)
    ReturnValue: ecNone (success) (0x00000000)
    FID: 0001-00174ea8cda0
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 0 (HSOT=0x0000006f)
    FID: 0001-00174ea8cda0
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 2 (HSOT=0x00000058)
    ReturnValue: ecNone (success) (0x00000000)
RopModifyRules
    ROPid: 0x41
    HandleIndex: 0 (HSOT=0x00000058)
    ModifyRulesFlags: 0x01 ROWLIST_REPLACE
    RulesCount: 1
Parsing row: 1
    RuleModificationFlag: 0x01 ROW ADD
    PropCount: 8 (0x08)
    0x66760003 PidTagRuleSequence      PtypInteger32   0x00000000 (0)
    0x66770003 PidTagRuleState         PtypInteger32   Flags: 0x00000001 ST_ENABLED
    0x667900FD PidTagRuleCondition     PtypRestriction Linked Restriction
        Linked Restriction:
        ConditionType: 0x00 RES AND:
            NoOfConditions: 3 restrictions
            ConditionType: 0x03 RES_CONTENT:
                FuzzyLevel: 0x00000002 FL_PREFIX
                0x001A001F PidTagMessageClass              PtypString
                    IPM.Schedule.Meeting
            ConditionType: 0x02 RES NOT
                Linked Restriction:
                ConditionType: 0x08 RES_EXIST:
                    PropertyTag: 0x3FE3000B PidTagDelegatedByRule
            ConditionType: 0x01 RES OR:
                NoOfConditions: 2 restrictions
                ConditionType: 0x02 RES NOT
                    Linked Restriction:
                    ConditionType: 0x08 RES_EXIST:
                        PropertyTag: 0x00360003 PidTagSensitivity
```

```
                ConditionType: 0x04 RES_PROPERTY:
                    RelationalOperator: 0x05 RELOP_NE
                    0x00360003 PidTagSensitivity       PtypInteger32
                        Flags: 0x00000002 SENSITIVITY_PRIVATE
    0x668000FE PidTagRuleActions                   PtypRuleAction
        NoOfActions: 0x0001 (1)
        Parsing action 1
        ActionType: 0x08 OP_DELEGATE
        Parsing action data:
        RecipientCount: 0x0001 (1)
        Recipient 1:
        PropCount: 12 (0x0C)
        0x0FFF0102 PidTagEntryId                   PtypBinary      128 Bytes
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 6F 3D 46 +/.........../o=F
            0020: 69 72 73 74 20 4F 72 67-61 6E 69 7A 61 74 69 6F irst Organizatio
            0030: 6E 2F 6F 75 3D 45 78 63-68 61 6E 67 65 20 41 64 n/ou=Exchange Ad
            0040: 6D 69 6E 69 73 74 72 61-74 69 76 65 20 47 72 6F ministrative Gro
            0050: 75 70 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 up (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 63 6E 3D-52 65 63 69 70 69 65 6E DLT)/cn=Recipien
            0070: 74 73 2F 63 6E 3D 64 65-6C 65 67 61 74 65 31 00 ts/cn=delegate1.
        0x3001001F PidTagDisplayName               PtypString      delegate1
        ...
    0x6681001F PidTagRuleProvider      PtypString      Schedule+ EMS Interface
    0x66830003 PidTagRuleLevel         PtypInteger32   0x00000000 (0)
    0x6682001F PidTagRuleName          PtypString      (null)
    0x66780003 PidTagRuleUserFlags     PtypInteger32   0x00000000 (0)
```

4.1.5 Set Permissions for Delegator Special Folders

Lastly, the client applies folder permissions to all delegator special folders that are described in this protocol.

In this example, the client is granting the same role to both delegates. The client grants the following permissions:

- Editor role to the Calendar (the only ROP trace shown for both delegates) and Task special folders.
- None role to the Inbox, Contacts, Notes, and Journal special folders.
- Editor role to the Freebusy Data folder.

```
RopOpenFolder
    ROPid: 0x02
    FID: 0001-00174ea89c98
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 9 (HSOT=0x00000055)
    ReturnValue: ecNone (success) (0x00000000)
RopModifyPermissions
    ROPid: 0x40
    HandleIndex: 0 (HSOT=0x00000055)
    ACLTableFlags: 0x01 ROWLIST_REPLACE
    RecipientRowCount: 3
    Parsing row: 3
    ACLFlag: 0x01 ROW ADD
        PropCount: 2 (0x02)
        0x0FFF0102 PidTagEntryId                   PtypBinary      128 Bytes
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 6F 3D 46 +/.........../o=F
            0020: 69 72 73 74 20 4F 72 67-61 6E 69 7A 61 74 69 6F irst Organizatio
            0030: 6E 2F 6F 75 3D 45 78 63-68 61 6E 67 65 20 41 64 n/ou=Exchange Ad
```

```
            0040: 6D 69 6E 69 73 74 72 61-74 69 76 65 20 47 72 6F ministrative Gro
            0050: 75 70 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 up (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 63 6E 3D-52 65 63 69 70 69 65 6E DLT)/cn=Recipien
            0070: 74 73 2F 63 6E 3D 64 65-6C 65 67 61 74 65 32 00 ts/cn=delegate2.
        0x66730003 PidTagMemberRights              PtypInteger32   0x0000007B (123)
    ACLFlag: 0x01 ROW_ADD
    PropCount: 2 (0x02)
        0x0FFF0102 PidTagEntryId                   PtypBinary      128 Bytes
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 6F 3D 46 +/........../o=F
            0020: 69 72 73 74 20 4F 72 67-61 6E 69 7A 61 74 69 6F irst Organizatio
            0030: 6E 2F 6F 75 3D 45 78 63-68 61 6E 67 65 20 41 64 n/ou=Exchange Ad
            0040: 6D 69 6E 69 73 74 72 61-74 69 76 65 20 47 72 6F ministrative Gro
            0050: 75 70 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 up (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 63 6E 3D-52 65 63 69 70 69 65 6E DLT)/cn=Recipien
            0070: 74 73 2F 63 6E 3D 64 65-6C 65 67 61 74 65 31 00 ts/cn=delegate1.
        0x66730003 PidTagMemberRights              PtypInteger32   0x0000007B (123)
    ACLFlag: 0x01 ROW_ADD
    PropCount: 2 (0x02)
        0x0FFF0102 PidTagEntryId                   PtypBinary      0 Bytes
        0x66730003 PidTagMemberRights              PtypInteger32   0x00000000 (0)
RopOpenFolder
    ROPid: 0x02
    FID: 0001-00174ea89c9d
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 1 (HSOT=0x0000004d)
    ReturnValue: ecNone (success) (0x00000000)
RopModifyPermissions
    ROPid: 0x40
    HandleIndex: 0 (HSOT=0x0000004d)
    ACLTableFlags: 0x01 ROWLIST_REPLACE
    RecipientRowCount: 3
    Parsing row: 3
    ACLFlag: 0x01 ROW_ADD
    PropCount: 2 (0x02)
        ...
        0x66730003 PidTagMemberRights              PtypInteger32   0x0000007B (123)
    ...
RopOpenFolder
    ROPid: 0x02
    FID: 0001-00174ea8cda0
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 1 (HSOT=0x00000066)
    ReturnValue: ecNone (success) (0x00000000)
RopModifyPermissions
    ROPid: 0x40
    LogonIndex: 0
    HandleIndex: 0 (HSOT=0x00000066)
    ACLTableFlags: 0x01 ROWLIST_REPLACE
    RecipientRowCount: 3
    Parsing row: 3
    ACLFlag: 0x01 ROW_ADD
    PropCount: 2 (0x02)
        ...
        0x66730003 PidTagMemberRights              PtypInteger32   0x00000000 (0)
    ...
RopOpenFolder
    ROPid: 0x02
    FID: 0001-00174ea89c99
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 1 (HSOT=0x00000086)
    ReturnValue: ecNone (success) (0x00000000)
```

```
RopModifyPermissions
    ROPid: 0x40
    HandleIndex: 0 (HSOT=0x00000086)
    ACLTableFlags: 0x01 ROWLIST_REPLACE
    RecipientRowCount: 3
    Parsing row: 3
    ACLFlag: 0x01 ROW_ADD
    PropCount: 2 (0x02)
        ...
        0x66730003 PidTagMemberRights              PtypInteger32   0x00000000 (0)
    ...
RopOpenFolder
    ROPid: 0x02
    FID: 0001-00174ea89c9c
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 1 (HSOT=0x0000008f)
    ReturnValue: ecNone (success) (0x00000000)
RopModifyPermissions
    ROPid: 0x40
    HandleIndex: 0 (HSOT=0x0000008f)
    ACLTableFlags: 0x01 ROWLIST_REPLACE
    RecipientRowCount: 3
    Parsing row: 3
    ACLFlag: 0x01 ROW ADD
    PropCount: 2 (0x02)
        ...
        0x66730003 PidTagMemberRights              PtypInteger32   0x00000000 (0)
    ...
RopOpenFolder
    ROPid: 0x02
    FID: 0001-00174ea89c9b
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 1 (HSOT=0x000000d3)
    ReturnValue: ecNone (success) (0x00000000)
RopModifyPermissions
    ROPid: 0x40
    HandleIndex: 0 (HSOT=0x000000d3)
    ACLTableFlags: 0x01 ROWLIST_REPLACE
    RecipientRowCount: 3
    Parsing row: 3
    ACLFlag: 0x01 ROW ADD
    PropCount: 2 (0x02)
        ...
        0x66730003 PidTagMemberRights              PtypInteger32   0x00000000 (0)
    ...
RopOpenFolder
    ROPid: 0x02
    FID: 0001-00174ea89cd4
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 1 (HSOT=0x0000006a)
    ReturnValue: ecNone (success) (0x00000000)
RopModifyPermissions
    ROPid: 0x40
    HandleIndex: 0 (HSOT=0x0000006a)
    ACLTableFlags: 0x01 ROWLIST_REPLACE
    RecipientRowCount: 3
    Parsing row: 3
    ACLFlag: 0x01 ROW_ADD
    PropCount: 2 (0x02)
        ...
        0x66730003 PidTagMemberRights              PtypInteger32   0x0000007B (123)
```

...

4.2 Accept Meeting Request Object on Behalf of Delegator

The following example shows the ROP traces for delegate1 receiving and processing a Meeting Request object on behalf of delegator1. The ROP traces in this example are truncated to more easily illustrate ROP information that is specific to this protocol.

This example shows that the delegator's Calendar special folder is opened to process the Meeting Request object, but the example does not go into the details of the creation of the Calendar object in the delegator's mailbox, which is described in [MS-OXOCAL].

This example also shows the creation and submission of a **Meeting Response object** on behalf of the delegator, which illustrates setting the from properties, as described in [MS-OXOMSG] section 2.2.1.

This example highlights the following steps when accepting a Meeting Request object on behalf of the delegator:

1. Identify that the meeting-related object is received on behalf of the delegator.
2. Identify the delegator's server and mailbox.
3. Access the delegator's Calendar special folder.
4. Send a Meeting Response object on behalf of the delegator.

4.2.1 Identify Meeting Request Object Received on Behalf of Delegator

In the following ROP traces, the client is logged on to the delegate1 mailbox and opens a Meeting Request object from the Inbox special folder. Received representing properties are present and different from recipient properties; therefore, as described in section 3.2.4.3, the Meeting Request object is being received on behalf of the delegator.

```
RopLogon
    ROPid: 0xFE
    LogonFlags: 0x01 Private
    OpenFlags: 0x0100040C HOME_LOGON TAKE_OWNERSHIP NO_MAIL CLI_WITH_PER_MDB_FIX
    Private Logon    LegacyDN: /o=First Organization/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=delegate1
RopLogon
    ROPid: 0xFE
    HandleIndex: 0 (HSOT=0x00000049)
    ReturnValue: ecNone (success) (0x00000000)
    FolderArray:
        ...
        FolderID  4: 0001-00174ea8cfdc IPM subtree
        FolderID  5: 0001-00174ea8cfdf Inbox
        FolderID  6: 0001-00174ea8cfe0 Outbox
        ...
RopOpenMessage
    ROPid: 0x03
    FolderId: 0001-00174ea8cfdf
    OpenModeFlags: 0x03 BestAccess rights
    MessageID: 0001-00174ea8d45b
RopOpenMessage
    ROPid: 0x03
    HandleIndex: 1 (HSOT=0x0000007b)
    ReturnValue: ecNone (success) (0x00000000)
    NormalizedSubject: delegatetest
    ...
```

```
RopGetPropertiesSpecific
    ROPid: 0x07
    HandleIndex: 0 (HSOT=0x0000007b)
    ReturnValue: ecNone (success) (0x00000000)
    HasError: 1
    PropertyArray:
    PropCount: 349
        ...
        0x0040001F PidTagReceivedByName                PtypString      delegate1
        0x0075001F PidTagReceivedByAddressType            PtypString      EX
        0x003F0102 PidTagReceivedByEntryId             PtypBinary      128 Bytes
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 4F 3D 46 +/.........../O=F
            0020: 49 52 53 54 20 4F 52 47-41 4E 49 5A 41 54 49 4F IRST ORGANIZATIO
            0030: 4E 2F 4F 55 3D 45 58 43-48 41 4E 47 45 20 41 44 N/OU=EXCHANGE AD
            0040: 4D 49 4E 49 53 54 52 41-54 49 56 45 20 47 52 4F MINISTRATIVE GRO
            0050: 55 50 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 UP (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 43 4E 3D-52 45 43 49 50 49 45 4E DLT)/CN=RECIPIEN
            0070: 54 53 2F 43 4E 3D 44 45-4C 45 47 41 54 45 31 00 TS/CN=DELEGATE1.
        0x0076001F PidTagReceivedByEmailAddress        PtypString      /O=FIRST
ORGANIZATION/OU=EXCHANGE ADMINISTRATIVE GROUP (FYDIBOHF23SPDLT)/CN=RECIPIENTS/CN=DELEGATE1
        0x00510102 PidTagReceivedBySearchKey           PtypBinary      103 Bytes
            0000: 45 58 3A 2F 4F 3D 46 49-52 53 54 20 4F 52 47 41 EX:/O=FIRST ORGA
            0010: 4E 49 5A 41 54 49 4F 4E-2F 4F 55 3D 45 58 43 48 NIZATION/OU=EXCH
            0020: 41 4E 47 45 20 41 44 4D-49 4E 49 53 54 52 41 54 ANGE ADMINISTRAT
            0030: 49 56 45 20 47 52 4F 55-50 20 28 46 59 44 49 42 IVE GROUP (FYDIB
            0040: 4F 48 46 32 33 53 50 44-4C 54 29 2F 43 4E 3D 52 OHF23SPDLT)/CN=R
            0050: 45 43 49 50 49 45 4E 54-53 2F 43 4E 3D 44 45 4C ECIPIENTS/CN=DEL
            0060: 45 47 41 54 45 31 00                            EGATE1.
        0x0044001F PidTagReceivedRepresentingName      PtypString      delegator1
        0x0077001F PidTagReceivedRepresentingAddressType  PtypString      EX
        0x00430102 PidTagReceivedRepresentingEntryId   PtypBinary      129 Bytes
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 4F 3D 46 +/.........../O=F
            0020: 49 52 53 54 20 4F 52 47-41 4E 49 5A 41 54 49 4F IRST ORGANIZATIO
            0030: 4E 2F 4F 55 3D 45 58 43-48 41 4E 47 45 20 41 44 N/OU=EXCHANGE AD
            0040: 4D 49 4E 49 53 54 52 41-54 49 56 45 20 47 52 4F MINISTRATIVE GRO
            0050: 55 50 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 UP (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 43 4E 3D-52 45 43 49 50 49 45 4E DLT)/CN=RECIPIEN
            0070: 54 53 2F 43 4E 3D 44 45-4C 45 47 41 54 4F 52 31 TS/CN=DELEGATOR1
            0080: 00                                              .
        0x0078001F PidTagReceivedRepresentingEmailAddress  PtypString      /O=FIRST
ORGANIZATION/OU=EXCHANGE ADMINISTRATIVE GROUP (FYDIBOHF23SPDLT)/CN=RECIPIENTS/CN=DELEGATOR1
        0x00520102 PidTagReceivedRepresentingSearchKey PtypBinary      104 Bytes
            0000: 45 58 3A 2F 4F 3D 46 49-52 53 54 20 4F 52 47 41 EX:/O=FIRST ORGA
            0010: 4E 49 5A 41 54 49 4F 4E-2F 4F 55 3D 45 58 43 48 NIZATION/OU=EXCH
            0020: 41 4E 47 45 20 41 44 4D-49 4E 49 53 54 52 41 54 ANGE ADMINISTRAT
            0030: 49 56 45 20 47 52 4F 55-50 20 28 46 59 44 49 42 IVE GROUP (FYDIB
            0040: 4F 48 46 32 33 53 50 44-4C 54 29 2F 43 4E 3D 52 OHF23SPDLT)/CN=R
            0050: 45 43 49 50 49 45 4E 54-53 2F 43 4E 3D 44 45 4C ECIPIENTS/CN=DEL
            0060: 45 47 41 54 4F 52 31 00                         EGATOR1.
        0x001A001F PidTagMessageClass                  PtypString
IPM.Schedule.Meeting.Request
        ...
```

4.2.2 Identify Delegator Server and Mailbox

Because this Meeting Request object is received on behalf of the delegator, the next step is to identify the server and mailbox for the delegator and log on to the delegator's mailbox. The server is identified by the value of the **PidTagAddressBookHomeMessageDatabase** property ([MS-OXOABK] section 2.2.4.37) returned by the **NspiGetProps** method, as described in [MS-NSPI], and because the **PidTagAddressBookProxyAddresses** property ([MS-OXOABK] section 2.2.3.23) does not have a "MAILBOX" or "EX" entry, the mailbox is identified by the value of the **PidTagEmailAddress** property ([MS-OXOABK] section 2.2.3.14).

```
NspiGetProps
    ...
    0x8006001f PidTagAddressBookHomeMessageDatabase          PtypString
        /o=First Organization/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=3659R9-A13/cn=Microsoft Private MDB
    0x3003001f PidTagEmailAddress           PtypString
        /o=First Organization/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=delegator1
    0x800f101f PidTagAddressBookProxyAddresses    PtypMultipleString
        PtypMultipleString[0]: SMTP:delegator1@jlvpno-dom.extest.microsoft.com
    ...
```

4.2.3 Access Delegator Calendar Special Folder

Because this is a Meeting Request object, the client needs to access the delegator's Calendar special folder. The delegator's Calendar special folder is identified by using the value of the **PidTagIpmAppointmentEntryId** property ([MS-OXPROPS] section 2.739), which is found in the delegator's Root folder.

The following example shows the ROP traces to accomplish this.

```
RopLogon
    ROPid: 0xFE
    LogonFlags: 0x01 Private
    OpenFlags: 0x0100040C HOME_LOGON TAKE_OWNERSHIP NO_MAIL CLI_WITH_PER_MDB_FIX
    Private Logon    LegacyDN: /o=First Organization/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=delegator1
RopLogon
    ROPid: 0xFE
    HandleIndex: 0 (HSOT=0x0000008f)
    ReturnValue: ecNone (success) (0x00000000)
    FolderArray:
        FolderID  1: 0001-00174ea8cd9c Root Folder
        ...
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 0 (HSOT=0x0000008f)
    FID: 0001-00174ea8cd9c
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 1 (HSOT=0x00000068)
    ReturnValue: ecNone (success) (0x00000000)
RopGetPropertiesSpecific
    ROPid: 0x07
    HandleIndex: 0 (HSOT=0x00000068)
    ReturnValue: ecNone (success) (0x00000000)
    PropCount: 11
        0x36D00102 PidTagIpmAppointmentEntryId         PtypBinary      46 Bytes
            0000: 00 00 00 00 C3 E1 78 57-96 52 CE 46 A3 53 B3 E0 ......xW.R.F.S..
            0010: 7F 9B 97 BC 01 00 8B 8D-B1 82 AF 2E D0 48 93 47 .............H.G
            0020: 07 ED 54 48 84 0F 00 17-4E A8 9C 98 00 00       ..TH....N.....
        ...
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 0 (HSOT=0x0000008f)
    FID: 0001-00174ea89c98
    OpenModeFlags: 0x00 ReadOnly
RopOpenFolder
    ROPid: 0x02
    HandleIndex: 1 (HSOT=0x00000080)
    ReturnValue: ecNone (success) (0x00000000)
RopGetPropertiesSpecific
    ROPid: 0x07
```

```
    HandleIndex: 1 (HSOT=0x00000080)
    ReturnValue: ecNone (success) (0x00000000)
    PropertyArray:
    PropCount: 17
        ...
        0x3001001F PidTagDisplayName                    PtypString      Calendar
        ...
```

4.2.4 Send a Meeting Response Object on Behalf of the Delegator

The final step in this example is to create and submit a Meeting Response object on behalf of the delegator. The following example shows that the client is populating the from properties with delegator information.

```
RopCreateMessage
    ROPid: 0x06
    HandleIndex: 1 (HSOT=0x000000a4)
    ReturnValue: ecNone (success) (0x00000000)
RopSetProperties
    ROPid: 0x0A
    LogonIndex: 0
    HandleIndex: 0 (HSOT=0x000000a4)
    PropertySize: 0x0A46 (2630)
    PropCount: 123 (0x7B)
        ...
        0x0042001F PidTagSentRepresentingName
               PidTagSentRepresentingName          PtypString      delegator1
        0x0064001F PidTagSentRepresentingAddressType      PtypString      (null)
        0x00410102 PidTagSentRepresentingEntryId       PtypBinary      129 Bytes
            0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 ......@..B......
            0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 4F 3D 46 +/........../O=F
            0020: 49 52 53 54 20 4F 52 47-41 4E 49 5A 41 54 49 4F IRST ORGANIZATIO
            0030: 4E 2F 4F 55 3D 45 58 43-48 41 4E 47 45 20 41 44 N/OU=EXCHANGE AD
            0040: 4D 49 4E 49 53 54 52 41-54 49 56 45 20 47 52 4F MINISTRATIVE GRO
            0050: 55 50 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 UP (FYDIBOHF23SP
            0060: 44 4C 54 29 2F 43 4E 3D-52 45 43 49 50 49 45 4E DLT)/CN=RECIPIEN
            0070: 54 53 2F 43 4E 3D 44 45-4C 45 47 41 54 4F 52 31 TS/CN=DELEGATOR1
            0080: 00                                            .
        0x001A001F PidTagMessageClass                  PtypString
IPM.Schedule.Meeting.Resp.Pos
        ...
RopSetProperties
    ROPid: 0x0A
    HandleIndex: 2 (HSOT=0x000000a4)
    PropertySize: 0x003A (58)
    PropCount: 2 (0x02)
        0x003D001F PidTagSubjectPrefix                 PtypString      Accepted:
        0x0E1D001F PidTagNormalizedSubject             PtypString      delegatetest
RopSubmitMessage
    ROPid: 0x32
    HandleIndex: 2 (HSOT=0x000000a4)
    SubmitMessageFlags: 0x00
```

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to the Delegate Access Configuration Protocol. General security considerations pertaining to the underlying RPC-based transport apply. For more information about these security considerations, see [MS-OXCROPS].

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 Preview
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016 Preview

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

<1> Section 2.2.2.2.2: Office Outlook 2003 does not support the **PidTagScheduleInfoDelegatorWantsInfo** property.

<2> Section 2.2.2.2.3: Office Outlook 2003 uses a **PtypMultipleString8** ([MS-OXCDATA] section 2.11.1) internal representation. Therefore, Office Outlook 2003 is unable to preserve the fidelity for all Unicode strings.

<3> Section 3.1.4.3.1: Office Outlook 2003, Office Outlook 2007, and Microsoft Outlook 2010 do not support delegation for a remote user when running against Exchange 2003.

<4> Section 3.1.4.3.3: Office Outlook 2003 uses the **PidTagScheduleInfoDelegateNames** property (section 2.2.2.2.3).

<5> Section 3.2.4.1: Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not support delegation for a remote user when running against Exchange 2003.

<6> Section 3.2.4.1: Exchange 2003, Exchange 2007, Exchange 2010, the initial release version of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release version of Outlook 2013 do not support the **Connect** request type. The **Connect** request type was introduced in Microsoft Exchange Server 2013 Service Pack 1 (SP1) and Microsoft Outlook 2013 Service Pack 1 (SP1).

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
6 Appendix A: Product Behavior	Added Exchange 2016 and Outlook 2016 to the list of applicable products.	Y	Content update.

8 Index