

[MS-OXODLGT]: Delegate Access Configuration Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1	Major	Initial Availability.
04/25/2008	0.2	Minor	Revised and updated property names and other technical content.
06/27/2008	1.0	Major	Initial Release.
08/06/2008	1.0.1	Editorial	Revised and edited technical content.
09/03/2008	1.0.2	Editorial	Revised and edited technical content.
12/03/2008	1.0.3	Editorial	Updated IP notice.
04/10/2009	2.0	Major	Updated technical content for new product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	3.1.0	Minor	Updated the technical content.
02/10/2010	4.0.0	Major	Updated and revised the technical content.
05/05/2010	4.1.0	Minor	Updated the technical content.
08/04/2010	4.1.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/03/2010	4.1.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction	6
1.1 Glossary	6
1.2 References.....	7
1.2.1 Normative References.....	7
1.2.2 Informative References	8
1.3 Overview	8
1.3.1 Granting Delegate Permissions.....	8
1.3.2 Accessing Delegator Information	8
1.3.3 Acting on Behalf of a Delegator	9
1.4 Relationship to Other Protocols.....	9
1.5 Prerequisites/Preconditions	9
1.6 Applicability Statement.....	9
1.7 Versioning and Capability Negotiation.....	9
1.8 Vendor-Extensible Fields.....	9
1.9 Standards Assignments	10
2 Messages	11
2.1 Transport.....	11
2.2 Message Syntax	11
2.2.1 Delegate Data Folder	11
2.2.1.1 Common Properties	11
2.2.1.1.1 PidTagDisplayName	11
2.2.2 Delegate Information Object.....	11
2.2.2.1 Common Properties	12
2.2.2.1.1 PidTagMessageClass	12
2.2.2.1.2 PidTagNormalizedSubject	12
2.2.2.2 Delegate Information Properties	12
2.2.2.2.1 PidTagScheduleInfoDelegatorWantsCopy	12
2.2.2.2.2 PidTagScheduleInfoDelegatorWantsInfo	12
2.2.2.2.3 PidTagScheduleInfoDelegateNames	12
2.2.2.2.4 PidTagScheduleInfoDelegateNamesW	12
2.2.2.2.5 PidTagScheduleInfoDelegateEntryIds	13
2.2.2.2.6 PidTagDelegateFlags.....	13
2.2.3 Delegate Rule	13
2.2.3.1 Delegate Rule Properties	13
2.2.3.1.1 PidTagRuleState.....	13
2.2.3.1.2 PidTagRuleName	13
2.2.3.1.3 PidTagRuleProvider.....	13
2.2.3.1.4 PidTagRuleLevel.....	13
2.2.3.1.5 PidTagRuleCondition	13
2.2.3.1.6 PidTagRuleActions	14
3 Protocol Details	15
3.1 Client Details.....	15
3.1.1 Abstract Data Model	15
3.1.1.1 Delegator Client	15
3.1.1.2 Delegate Client.....	15
3.1.2 Timers	15
3.1.3 Initialization	15
3.1.4 Higher-Layer Triggered Events.....	15

3.1.4.1	Creating Delegate Data Folder.....	15
3.1.4.2	Creating Delegate Information Object	15
3.1.4.3	Creating Delegate Relationship.....	16
3.1.4.3.1	Set Send-On-Behalf-Of Delegator Permissions	16
3.1.4.3.2	Set Delegate Folder Permissions	16
3.1.4.3.2.1	Additional Constraints for Calendar Folder	17
3.1.4.3.2.2	Additional Constraints for the Tasks Folder	17
3.1.4.3.3	Set Individual Delegate Preferences.....	18
3.1.4.3.4	Set Global Delegate Preferences.....	18
3.1.4.3.4.1	Set the PidTagScheduleInfoDelegatorWantsCopy Property	18
3.1.4.3.4.2	Set the PidTagScheduleInfoDelegatorWantsInfo Property	19
3.1.4.3.5	Set Delegate Rule	19
3.1.4.4	Opening Delegator's Special Folder	19
3.1.4.5	Display Delegator Contents.....	20
3.1.4.6	Send On Behalf Of Delegator.....	20
3.1.4.7	Receive/Process On Behalf Of Delegator	20
3.1.5	Message Processing Events and Sequencing Rules.....	21
3.1.6	Timer Events	21
3.1.7	Other Local Events	21
3.2	Server Details	21
3.2.1	Abstract Data Model	21
3.2.2	Timers	21
3.2.3	Initialization	21
3.2.4	Higher-Layer Triggered Events.....	21
3.2.4.1	Opening Delegator Root Folder.....	21
3.2.4.2	External Higher-Layer Triggered Events.....	21
3.2.4.2.1	Submitting On Behalf Of Delegator.....	21
3.2.4.2.2	Message Delivery to Delegator.....	22
3.2.4.2.3	Creating, Modifying, or Deleting Message Objects.....	22
3.2.5	Message Processing Events and Sequencing Rules.....	22
3.2.6	Timer Events	22
3.2.7	Other Local Events	22
4	Protocol Examples.....	23
4.1	Create Delegate Relationship with Multiple Delegates	23
4.1.1	Identify Delegator Special Folders	23
4.1.2	Set Send-On-Behalf-Of Delegator Permissions	24
4.1.3	Update the Delegate Information Object	25
4.1.3.1	Open the Delegator Information Object	25
4.1.3.2	Update the Delegator Information Object Properties	25
4.1.4	Update the Delegate Rule.....	26
4.1.5	Set Permissions for Delegator Special Folders	27
4.2	Accept Meeting Request Object On Behalf Of Delegator	30
4.2.1	Identify Meeting Request Object Received on Behalf of Delegator.....	31
4.2.2	Identify Delegator Server and Mailbox	32
4.2.3	Access Delegator Calendar Special Folder.....	33
4.2.4	Send a Meeting Response Object on Behalf of the Delegator.....	34
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.....	37
8 Index	38

1 Introduction

This document specifies the Delegate Access Configuration Protocol, which allows a user to delegate the responsibility for his or her **mailbox** to another user.

The Delegate Access Configuration Protocol defines the following:

- The format to enable a user to send mail on behalf of the delegating user.
- The format to enable a user to send or receive meeting requests on behalf of the delegating user.
- The format for granting permissions to a user to read from or write to all or part of the delegating user's mailbox.
- The mechanism for accessing the delegating user's mailbox.

1.1 Glossary

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

action
Address Book object
calendar
Calendar object
Calendar special folder
delegate
Delegate Information object
delegator
EntryID
folder
folder ID (FID)
from properties
handle
informational update
mailbox
meeting-related object
Meeting Request object
Meeting Response object
message ID (MID)
Message object
property (1)
recipient properties
remote operation (ROP)
remote procedure call (RPC)
restriction
Root folder
rule
sender properties
server-side rule
special folder
Task object
task request
Unicode

The following terms are specific to this document:

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

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

Private Message object: A **Message object** with [PidTagSensitivity](#) set to Private ([\[MS-OXCMSG\]](#) section 2.2.1.13), indicating that it contains sensitive information.

received representing properties: A group of **properties** that identifies the end user represented by the receiving **mailbox** owner.

remote user: An **Address Book object** known to be from a foreign or remote messaging system. For more information about remote users, see [\[MS-OXOABK\]](#) section 2.2.3.11 and section [2.2.3.12](#).

send-on-behalf-of: A special permission granted to a **delegate** allowing them to send **Message objects** representing the **delegator**.

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

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-OXCDATA] Microsoft Corporation, "[Data Structures](#)", April 2008.

[MS-OXCFOLD] Microsoft Corporation, "[Folder Object Protocol Specification](#)", April 2008.

[MS-OXCMSG] Microsoft Corporation, "[Message and Attachment Object Protocol Specification](#)", April 2008.

[MS-OXCPerm] Microsoft Corporation, "[Exchange Access and Operation Permissions Protocol Specification](#)", April 2008.

[MS-OXCROPS] Microsoft Corporation, "[Remote Operations \(ROP\) List and Encoding Protocol Specification](#)", April 2008.

[MS-OXCSTOR] Microsoft Corporation, "[Store Object Protocol Specification](#)", April 2008.

[MS-OXDISCO] Microsoft Corporation, "[Autodiscover HTTP Service Protocol Specification](#)", April 2008.

[MS-OXOABK] Microsoft Corporation, "[Address Book Object Protocol Specification](#)", April 2008.

[MS-OXOCAL] Microsoft Corporation, "[Appointment and Meeting Object Protocol Specification](#)", April 2008.

[MS-OXOMSG] Microsoft Corporation, "[E-Mail Object Protocol Specification](#)", April 2008.

[MS-OXOPFFB] Microsoft Corporation, "[Public Folder-Based Free/Busy Protocol Specification](#)", April 2008.

[MS-OXORULE] Microsoft Corporation, "[E-Mail Rules Protocol Specification](#)", April 2008.

[MS-OXOSFLD] Microsoft Corporation, "[Special Folders Protocol Specification](#)", April 2008.

[MS-OXOTASK] Microsoft Corporation, "[Task-Related Objects Protocol Specification](#)", April 2008.

[MS-OXPROPS] Microsoft Corporation, "[Exchange Server Protocols Master Property List](#)", April 2008.

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

1.2.2 Informative References

[MS-OXGLOS] Microsoft Corporation, "[Exchange Server Protocols Master Glossary](#)", April 2008.

1.3 Overview

The Delegate Access Configuration Protocol can be used to allow a **delegator** in an organization to delegate responsibility for several activities that are commonly performed on objects in the delegator's mailbox, or a delegator can configure delivery of **Meeting Request objects** directly to the **delegate**.

To allow 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 and complete the desired **actions**.

1.3.1 Granting Delegate Permissions

There are three sets of permissions that are commonly granted to a delegate: reviewer, author, and editor. These permissions are set on a specific set of **special folders**. The delegator decides on 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 to the delegate.

Additionally, the delegate can be granted permissions to send on behalf of the delegator. This can be useful if the delegate will be responding to **Message objects**, managing **meeting-related objects**, and/or managing **Task objects**.

1.3.2 Accessing Delegator Information

To access the delegator's information, a delegate will identify and log on to the delegator's mailbox. The delegate will then identify the desired special folder, open the delegator's special folder, and manipulate items (such as creating or modifying appointments) to complete the task.

1.3.3 Acting on Behalf of a Delegator

If the delegate desires to send on behalf of the delegator, the delegate sets **properties** on the Message object to indicate that it is being sent on behalf of the delegator. The server will then validate that the delegate has the appropriate permissions 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 only be acted on if the delegate has the appropriate permissions to the delegator's **Calendar special folder** and permission to send mail on behalf of the delegator. This is due to the fact that 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:

- Message and Attachment Object Protocol, as specified in [\[MS-OXCMMSG\]](#).
- Folder Object Protocol, as specified in [\[MS-OXCFOLD\]](#).
- Exchange Access and Operation Permissions, as specified in [\[MS-OXCPerm\]](#).
- E-mail Rules Protocol, as specified in [\[MS-OXORULE\]](#).
- E-mail Object Protocol, as specified in [\[MS-OXOMSG\]](#).
- Address Book Object Protocol, as specified in [\[MS-OXOABK\]](#).
- Appointment and Meeting Object Protocol, as specified in [\[MS-OXOCAL\]](#).
- Task-Related Objects Protocol, as specified in [\[MS-OXOTASK\]](#).

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 specified in [\[MS-OXOABK\]](#), 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 delegator, as specified in [\[MS-OXOABK\]](#).

1.6 Applicability Statement

This protocol is implemented when a user wants to 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](#).

Delegate Information object properties can be of several different data types. The following data types are used by Delegate Information object properties, as specified in [\[MS-OXCData\]](#) section 2.11.1.

- PtypBinary
- PtypBoolean
- PtypFloating64
- PtypInteger32
- PtypString
- PtypTime

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 2.3, the delegate data **folder** contains the properties listed in the following section.

2.2.1.1.1 PidTagDisplayName

This is a **PtypString** property and its value 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 MAY also contain other properties, [<1>](#) 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 the following sections.

2.2.2.1.1 PidTagMessageClass

This is a **PtypString** property and its value MUST be set to "IPM.Microsoft.ScheduleData.FreeBusy".

2.2.2.1.2 PidTagNormalizedSubject

This is a **PtypString** property and its value MUST be set to "LocalFreebusy".

2.2.2.2 Delegate Information Properties

2.2.2.2.1 PidTagScheduleInfoDelegatorWantsCopy

This **PtypBoolean** property 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

This **PtypBoolean** property indicates whether the delegator wants to receive **informational updates**, as specified in [\[MS-OXOCAL\]](#) section 3.1.4.6.2.1 and [\[MS-OXOCAL\]](#) section 3.1.4.6.4.1. 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

This **PtypMultipleString** property specifies the names of the delegates. Each entry contains the value of the [PidTagDisplayName](#) property 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** 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

This **PtypMultipleString** property specifies the names of the delegates. Each entry contains the value of the [PidTagDisplayName](#) property 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** 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

This **PtypMultipleBinary** property specifies the **EntryIDs** of the delegates. Each entry contains the value of the [PidTagEntryId](#) property 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

This **PtypMultipleInteger32** property indicates whether delegates can view **Private Message objects**. Each entry of this property MUST be set to one of the following values.

Flag	Value	Description
HidePrivate	0x00000000	The delegate SHOULD NOT be allowed to view Private Message objects.
ShowPrivate	0x00000001	The delegate SHOULD be allowed to view Private Message objects.

This property MUST be set on the Delegate Information object.

2.2.3 Delegate Rule

To enable **calendar** workflow scenarios where 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 3.1.4.3.

2.2.3.1 Delegate Rule Properties

The **delegate rule** is specified by setting the properties listed in the following sections.

2.2.3.1.1 PidTagRuleState

This is a **PtypInteger32** property and its value MUST be set to "0x00000001".

2.2.3.1.2 PidTagRuleName

This is a **PtypString** property and its value MUST be set to "" (a zero-length string).

2.2.3.1.3 PidTagRuleProvider

This is a **PtypString** property and its value MUST be set to "Schedule+ EMS Interface".

2.2.3.1.4 PidTagRuleLevel

This is a **PtypInteger32** property and its value MUST be set to "0x00000000".

2.2.3.1.5 PidTagRuleCondition

This is a **PtypRestriction** property and its value is set to a **restriction** of type RES_AND with the following subclauses:

- A restriction of type RES_CONTENT that limits a table view to rows that include the string "IPM.Schedule.Meeting" in the [PidTagMessageClass](#) property 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 the following subclause:
 - A restriction of type RES_EXIST that specifies the [PidTagDelegatedByRule](#) property, as specified in [\[MS-OXPROPS\]](#) section 2.740.
- A restriction of type RES_OR with the following subclauses:
 - A restriction of type RES_NOT with the following subclause:
 - A restriction of type RES_EXIST that specifies the [PidTagSensitivity](#) property, as specified in [\[MS-OXCMSG\]](#) 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 "Private" ("0x00000002"). 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

This **PtypRuleAction** property specifies the delegate's rule actions, which are used to:

1. Send copies of meeting-related objects to delegates.

Use the OP_DELEGATE action, as specified in [\[MS-OXORULE\]](#) section 2.2.5.1.3.4.

2. Delete the delegator's copy of meeting-related objects.

Use the OP_DELETE action, as specified in [\[MS-OXORULE\]](#) section 2.2.5.1.3.7.

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 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.

3.1.1.1 Delegator Client

A delegator client is used by a delegator to establish delegation relationships with one or more delegates, and is used to store delegator preferences in the Delegate Information object.

3.1.1.2 Delegate Client

A delegate client is used to perform actions on behalf of the delegator. To perform these actions, the delegate client will:

- Access the delegator's mailbox to create, modify, or delete objects.
- Honor preferences stored by the delegator client in the Delegate Information object.
- Send Message objects on behalf of the delegator.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Creating Delegate Data Folder

The client for a delegator creates the delegate data folder under the delegator's Root folder by sending [RopCreateFolder](#) ([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 [RopSetProperties](#) ([MS-OXCROPS] section 2.2.8.6).

In addition, the EntryID for the delegate data folder is stamped in the [PidTagFreeBusyEntryIds](#) property, as specified in [MS-OXOPFFB] section 2.2.2.1.

3.1.4.2 Creating Delegate Information Object

The client for a delegator creates the Delegate Information object under the delegator's delegate data folder by sending [RopCreateMessage](#) ([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

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

In addition, the EntryID for the Delegate Information object is stamped in the [PidTagFreeBusyEntryIds](#) property, as specified in [MS-OXOPFFB] section 2.2.2.1.

3.1.4.3 Creating Delegate Relationship

The client for a delegator establishes the delegate relationship by setting permissions and individual preferences for delegates, as well as setting global delegate preferences. A client for the delegator accomplishes this by performing the following steps, as specified in sections [3.1.4.3.1](#) through [3.1.4.3.5](#).

3.1.4.3.1 Set Send-On-Behalf-Of Delegator Permissions

The delegator's client SHOULD grant **send-on-behalf-of** permissions to every delegate. [<3>](#) This is accomplished by adding the value of the [PidTagEntryId](#) property of the delegate's Address Book object to the [PidTagAddressBookPublicDelegates](#) property of the delegator's address book container, as specified in [MS-OXOABK].

A client MAY [<4>](#) support delegation for a **remote user** if it 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, as specified in [MS-OXOABK] section 2.2.3.12.

3.1.4.3.2 Set Delegate Folder Permissions

The following table specifies roles supported by this protocol. Roles are a specific set of flags for the value of the [PidTagMemberRights](#) property, which is used when setting folder permissions, as specified in [MS-OXCPERM] section 2.2.1.6.

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 will not be able to create, modify, or delete any contents from the folder.
Author	0x0000001B	ReadAny Create EditOwned DeleteOwn	The delegate is able to view contents of the folder. In addition, the delegate is able to create, modify, and delete any items created by this delegate.
Editor	0x0000007B	ReadAny Create EditOwned DeleteOwn EditAny DeleteAny	The delegate is able to view, create, modify, and delete any items in the folder.

The delegator client specifies a role for each of the following special folders, as specified in [\[MS-OXOSFLD\]](#):

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

The delegator 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

If a delegate will 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 will allow 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 client for the delegator creates it.

Additionally, if a delegate will 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-of permissions to the delegate.
3. Add the OP_DELEGATE action, including all delegates that will receive meeting-related objects on behalf the delegator. The OP_DELEGATE action is specified in [\[MS-OXORULE\]](#) section 2.2.5.1.3.4.

3.1.4.3.2.2 Additional Constraints for the Tasks Folder

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

1. 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 Granting the Author role will allow the delegate to act only on new items or ones created by the same delegate.

2. Grant send-on-behalf-of permissions to the delegate.

3.1.4.3.3 Set Individual Delegate Preferences

For each delegate being specified, the client for the delegator MUST:

1. Specify the delegate's name, using the value of the [PidTagDisplayName](#) property for the Address Book object of the delegate. This value MUST be specified as an entry in either the [PidTagScheduleInfoDelegateNamesW](#) property or the [PidTagScheduleInfoDelegateNames](#) property. 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 [PidTagScheduleInfoDelegateNamesW,<5>](#) and it MUST use the same property for all delegates.
2. Specify the value of the [PidTagEntryId](#) property for the Address Book object of the delegate as an entry in the [PidTagScheduleInfoDelegateEntryIds](#) property.
3. Specify whether the delegate can view the delegator's private Message objects as an entry in the [PidTagDelegateFlags](#) property. This preference is applicable to all folders for which the delegate has a role of Reviewer, Author, or Editor.

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

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

3.1.4.3.4 Set 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 [RopSetProperties](#) ([MS-OXCROPS] section 2.2.8.6), and committed by sending [RopSaveChangesMessage](#) ([MS-OXCROPS] section 2.2.6.3).

For more details about calendar workflows, see [\[MS-OXOCAL\]](#).

3.1.4.3.4.1 Set the PidTagScheduleInfoDelegatorWantsCopy Property

The value of this property is set to TRUE (0x01) in the following cases:

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

Otherwise, the value MUST be set to FALSE (0x00).

The OP_DELETE action MUST NOT be present when this property has a value of TRUE. The value of this property is set to TRUE (0x01) if the value of the [PidTagScheduleInfoDelegatorWantsInfo](#) property is set to TRUE (0x01).

The delegator adds the OP_DELETE action, as specified in [\[MS-OXORULE\]](#) section 2.2.5.1.3.7, when this property has a value of FALSE (0x00).

For more information about how this property is used, see [\[MS-OXOCAL\]](#) section 3.1.4.

3.1.4.3.4.2 Set the PidTagScheduleInfoDelegatorWantsInfo Property

The value of this property is set to "0x01" when a delegator wants to receive informational updates, as specified in [\[MS-OXOCAL\]](#) section 3.1.4.6.2.1. Otherwise, it MUST be set to "0x00".

The value of this property is set to "0x00" if the value of property [PidTagScheduleInfoDelegatorWantsCopy](#) is set to "0x00". For more information about how this property is used, see [\[MS-OXOCAL\]](#) section 3.1.4.

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

WC	WI	Description
1	1	The delegator wants to receive copies and would like these copies to be informational updates when applicable.
1	0	The delegator wants to receive copies.
0	1	Invalid, as the delegator cannot receive informational updates unless the delegator receives copies.
0	0	The delegator doesn't want to receive copies or informational updates.

3.1.4.3.5 Set Delegate Rule

The client for the delegator creates or updates the delegate rule, which is specified in section [2.2.3](#), if, while creating the delegate relationship:

1. Any delegate will be receiving Meeting Request objects on behalf of the delegator, as this adds the OP_DELEGATE action to the delegate rule.
2. The delegator has elected not to receive copies of Meeting Request objects, as this adds the OP_DELETE action to the delegate rule.

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

3.1.4.4 Opening Delegator's Special Folder

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

1. Identify and establish a connection to the delegator's server.
2. Identify and log on to the delegator's mailbox.
3. Identify and open the delegator's special folder.

The delegate's client uses properties from the Address Book object that represents the delegator to identify the delegator's server and mailbox.

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 MAY use the delegate's own server and let the Autodiscover HTTP Service protocol handle redirection to the correct server [\[MS-OXDISCO\]](#). A remote user can be identified by examining the

[PidTagDisplayType](#) property of the user's Address Book object. If the [PidTagDisplayType](#) property has the value DT_REMOTE_MAILUSER, then the delegate is a remote user. For more information about the [PidTagDisplayType](#) property, see [\[MS-OXOABK\]](#) section 2.2.3.11.

If the Address Book object has a valid [PidTagAddressBookProxyAddresses](#) property ([\[MS-OXOABK\]](#) section 2.2.3.23) 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 doesn't 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 ([\[MS-OXOABK\]](#) section 2.2.3.14).

After establishing a connection to the server, the client uses [RopLogon](#) to connect to the delegator's private mailbox with HOME_LOGON and TAKE_OWNERSHIP flags, as specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.

After the client has logged on to the delegator's mailbox, the delegate's client identifies the desired special folder within the delegator's mailbox. See [\[MS-OXOSFLD\]](#) for more information about special folders.

The client uses [RopOpenFolder<7>](#) to open the delegator's special folder as specified in [\[MS-OXCFOLD\]](#) section 2.2.1.

3.1.4.5 Display Delegator Contents

The client for a delegate SHOULD NOT show private Message objects of the delegator unless the [PidTagDelegateFlags](#) property for the delegate has a value of "ShowPrivate", indicating that the delegator wants to make Private Message objects visible.

3.1.4.6 Send On Behalf Of Delegator

When sending Message objects on behalf of the delegator, the client of the delegate populates **from properties**, as specified in [\[MS-OXOMSG\]](#) section 3.1.1.2.1, with information from the delegator.

3.1.4.7 Receive/Process On Behalf Of Delegator

The client for a delegate can identify that a Message object has been received on behalf of the delegator when the **received representing properties** are present and different than the **recipient properties**. The received representing properties are specified in [\[MS-OXOMSG\]](#) section 3.1.1.2.2.1.

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

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

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

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 is primarily client driven; however, the server plays an important role in ensuring that delegate scenarios are possible, which is accomplished mostly by implementing underlying protocols. As such, the abstract data model for this protocol is not significantly different than that of its underlying protocols.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Opening Delegator Root Folder

The server MUST provide read access to a delegator's Root folder and its properties, as the delegate will need to obtain the **folder ID (FID)** for the delegator's special folders, as specified in [\[MS-OXOSFLD\]](#).

3.2.4.2 External Higher-Layer Triggered Events

This section specifies higher-layer triggered events implemented in external protocols that are required for scenarios in this protocol.

3.2.4.2.1 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 3.1.1.2.

3.2.4.2.2 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 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 ensures that the delegator does not receive unwanted Meeting Request objects.

3.2.4.2.3 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, as specified in [\[MS-OXCPerm\]](#).

Additionally, a server MUST track the creator and last modifier of any Message object as specified in [\[MS-OXCMSG\]](#).

3.2.5 Message Processing Events and Sequencing Rules

None.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Create Delegate Relationship with Multiple Delegates

The following example shows the **remote operation (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 highlights the following steps when setting up the delegate relationship:

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

4.1.1 Identify Delegator Special Folders

To identify the delegator special folders, the client will log on to the delegator mailbox and query for the special folder properties from the Inbox special folder, which is provided in response to the [RopLogon](#) request. <8> 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-Of Delegator Permissions

Then, the delegator will set send on behalf of permissions by using the **NspiModLinkAtt** method, as specified in [\[MS-OXOABK\]](#) section 2.2.5.

```

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

There are two steps necessary to update the Delegate Information object: opening the object and updating the properties.

4.1.3.1 Open the Delegator Information Object

The client will open the Delegate Information object by using the **message ID (MID)** and the FID, which are the second entry and the fourth entry, respectively in the [PidTagFreeBusyEntryIds](#) property. The value of the **NormalizedSubject** field (in the [RopOpenMessage<9>](#) response) 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 will update the [PidTagScheduleInfoDelegatorWantsCopy](#) property and the [PidTagScheduleInfoDelegatorWantsInfo](#) property with the delegator global settings. In this case, the delegator does not 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 will update the [PidTagScheduleInfoDelegateNamesW](#), [PidTagScheduleInfoDelegateEntryIds](#), and [PidTagDelegateFlags](#) properties for each delegate. "delegate2" is stored in the first entry of these multiple-valued properties, and "delegate1" is stored in the second entry of these multiple-valued properties. The delegator is only allowing "delegate2" to see private Message objects. The "delegate2" preferences, as well as global delegator preferences, are highlighted.

```
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 first 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 first 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 will update the delegate rule on the receive folder rule by using [RopModifyRules](#), as specified in [\[MS-OXORULE\]](#) section 2.2.1. Given the actions, only "delegate1" will be 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=delegatel.
  0x3001001F PidTagDisplayName       PtypString     delegatel
  ...
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 will apply folder permissions to all delegator special folders that are specified in this protocol.

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

- 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)
  0xFFFF0102 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)
  0xFFFF0102 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)
  0xFFFF0102 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 the delegate, "delegate1", receiving and processing a Meeting Request object on behalf of the delegator, "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 doesn't go into the details of the creation of the Calendar object in the delegator's mailbox, as this is specified in [\[MS-OXOCAL\]](#).

This example also shows the creation and submission of a **Meeting Response object** on behalf of the delegator, as this illustrates setting the from properties.

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 delegate "delegate1" mailbox, and opens a Meeting Request object from the Inbox special folder. Received representing properties are present and different than recipient properties; therefore, as specified in section 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: 0001-00174ea8cfd0 Inbox
    FolderID 6: 0001-00174ea8cfe0 Outbox
    ...
RopOpenMessage
  ROPid: 0x03
  FolderId: 0001-00174ea8cfd0
  OpenModeFlags: 0x03 BestAccess rights
  MessageID: 0001-00174ea8d45b
RopOpenMessage
  ROPid: 0x03
  HandleIndex: 1 (HSOT=0x0000007b)
  ReturnValue: ecNone (success) (0x00000000)
  NormalizedSubject: delegatetest
  ...
RopGetPropertySpecific
  ROPid: 0x07
  HandleIndex: 0 (HSOT=0x0000007b)
  ReturnValue: ecNone (success) (0x00000000)
  HasError: 1
  PropertyArray:
  PropCount: 349
  ...
  0x0040001F PidTagReceivedByName PtypString delegat1
  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 MINISTERIAL 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 MINISTERIAL 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
PidTagRemoteHeaderLoc 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 will be identified by the value of the [PidTagAddressBookHomeMessageDatabase](#) property, and because the [PidTagAddressBookProxyAddresses](#) property doesn't have a "MAILBOX" or "EX" entry, the mailbox will be identified by the value of the [PidTagEmailAddress](#) property.

```

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@jlvyno-dom.extest.microsoft.com
...

```

4.2.3 Access Delegator Calendar Special Folder

Because this is a Meeting Request object, the client will need to access the delegator's Calendar special folder. The delegator's Calendar special folder is identified by using the value of the [PidTagIpmAppointmentEntryId](#) ([MS-OXPROPS] section 2.849) property, 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 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
    PidTagSentRepresentingAddressType          PtypString    (null)
  0x00410102 PidTagSentRepresentingEntryId
    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 MINISTERATIVE 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
    PidTagMessageClass                          PtypString    IPM.Schedule.Meeting.Resp.Pos
  ...
RopSetProperties
  ROPid: 0x0A
  HandleIndex: 2 (HSOT=0x000000a4)
  PropertySize: 0x003A (58)
  PropCount: 2 (0x02)
  0x003D001F PidTagSubjectPrefix
    PidTagSubjectPrefix                          PtypString    Accepted:
  0x0E1D001F PidTagNormalizedSubject
    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 **remote procedure call (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:

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

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified. 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 prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that product does not follow the prescription.

[<1> Section 2.2.2:](#) Office Outlook 2003, Office Outlook 2007, and Outlook 2010 set the following property regardless of user input; its value has no meaning in the context of this protocol: [PidTagScheduleInfoDontMailDelegates](#).

[<2> Section 2.2.2.3:](#) Office Outlook 2003 uses a **PtypMultipleString8** internal representation. Therefore, this version is unable to preserve the fidelity for all Unicode strings.

[<3> Section 3.1.4.3.1:](#) Office Outlook 2007 and Outlook 2010 will stop creating a delegate relationship if send-on-behalf-of permissions cannot be granted.

[<4> Section 3.1.4.3.1:](#) This is only supported by Office Outlook 2007 and Outlook 2010 when running against Exchange 2007 or Exchange 2010.

[<5> Section 3.1.4.3.3:](#) Office Outlook 2003 uses the [PidTagScheduleInfoDelegateNames](#) property.

[<6> Section 3.1.4.4:](#) This is only supported by Office Outlook 2007 and Outlook 2010 when running against Exchange 2007 or Exchange 2010.

[<7> Section 3.1.4.4:](#) Exchange 2010 can output unexpected results when using [RopOpenFolder](#) when client connection services are deployed on an Exchange server that does not also have a mailbox store installed.

[<8> Section 4.1.1:](#) Exchange 2010 can output unexpected results when using [RopLogon](#) when client connection services are deployed on an Exchange server that does not also have a mailbox store installed.

[<9> Section 4.1.3.1:](#) Exchange 2010 can output unexpected results when using [RopOpenMessage](#) when client connection services are deployed on an Exchange server that does not also have a mailbox store installed.

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](#) 15
[server](#) 21
[Applicability](#) 9

C

[Change tracking](#) 37
Client
[abstract data model](#) 15

D

Data model – abstract
[client](#) 15
[server](#) 21

E

[Examples - overview](#) 23

G

[Glossary](#) 6

I

[Implementer - security considerations](#) 35
[Introduction](#) 6

M

Messages
[overview](#) 11
[transport](#) 11

N

[Normative references](#) 7

O

[Overview \(synopsis\)](#) 8

P

[Preconditions](#) 9
[Prerequisites](#) 9
[Product behavior](#) 36

R

References
[normative](#) 7
[Relationship to other protocols](#) 9

S

Security
[implementer considerations](#) 35
[overview](#) 35

Server
[abstract data model](#) 21

T

[Tracking changes](#) 37
[Transport](#) 11