

# [MS-OXORMDR]: Reminder Settings Protocol Specification

---

## Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

**Preliminary Documentation.** This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final

documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

## Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability.
04/25/2008	0.2		Revised and updated property names and other technical content.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Revised and edited technical content.
09/03/2008	1.02		Updated references.
12/03/2008	1.03		Updated IP notice.
04/10/2009	2.0		Updated technical content for new product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	4.1.0	Minor	Updated the technical content.
05/05/2010	4.2.0	Minor	Updated the technical content.
08/04/2010	4.3	Minor	Clarified the meaning of the technical content.
11/03/2010	4.4	Minor	Clarified the meaning of the technical content.
03/18/2011	4.4	No change	No changes to the meaning, language, and formatting of the technical content.
08/05/2011	4.5	Minor	Clarified the meaning of the technical content.
10/07/2011	4.5	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	5.0	Major	Significantly changed the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>6</b>
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	Creating, Modifying, and Removing Reminders.....	8
1.3.2	Processing Overdue Reminders .....	9
1.3.3	Dismissing and Snoozing Reminders .....	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 .....	9
<b>2</b>	<b>Messages.....</b>	<b>10</b>
2.1	Transport.....	10
2.2	Message Syntax .....	10
2.2.1	Properties Used to Specify and/or Decode Reminders .....	10
2.2.1.1	PidLidReminderSet Property .....	10
2.2.1.2	PidLidReminderSignalTime Property .....	10
2.2.1.3	PidLidReminderDelta Property .....	11
2.2.1.4	PidLidReminderTime Property.....	11
2.2.1.5	PidLidReminderOverride Property .....	11
2.2.1.6	PidLidReminderPlaySound Property .....	11
2.2.1.7	PidLidReminderFileParameter Property .....	11
2.2.1.8	PidTagReplyTime Property .....	11
2.2.1.9	PidLidReminderType Property.....	12
2.2.2	Shared Properties.....	12
2.2.2.1	Properties Shared with the Message and Attachment Object Protocol .....	12
2.2.2.2	Properties Shared with the Informational Flagging Protocol.....	12
2.2.2.3	Properties Shared with the Task-Related Objects Protocol.....	12
2.2.2.3.1	PidLidTaskDueDate Property.....	12
2.2.2.4	Properties Shared with the Appointment and Meeting Object Protocol .....	13
2.2.3	Properties Used to Specify User Preferences that are Roamed on the Server.....	13
2.2.3.1	piReminderUpgradeTime Property.....	13
<b>3</b>	<b>Protocol Details.....</b>	<b>14</b>
3.1	Common Details .....	14
3.2	Client Details.....	14
3.2.1	Abstract Data Model .....	14
3.2.2	Timers .....	14
3.2.3	Initialization .....	14
3.2.3.1	Populating the Reminder Queue .....	14
3.2.3.1.1	Scope of the Search for Active Reminders .....	14
3.2.3.1.1.1	Requirements for Clients Using a Full Reminder Domain .....	15
3.2.3.1.2	Loading/Processing.....	15
3.2.4	Higher-Layer Triggered Events.....	15
3.2.4.1	Setting a Reminder .....	15

3.2.4.1.1	Reminders on Task Objects .....	15
3.2.4.1.2	Reminders on Message Objects other than Calendar Objects and Task Objects .....	16
3.2.4.1.3	Reminders on Calendar Objects .....	16
3.2.4.1.4	Reminders on Recurring Calendar Objects .....	16
3.2.4.1.5	Setting Reminders on Draft Message Objects .....	17
3.2.4.1.5.1	Setting a Reminder for all Recipients .....	17
3.2.4.1.5.2	Setting a Reminder for the Sender .....	17
3.2.4.2	Post-Transmit Processing .....	18
3.2.4.3	Post-Receive Processing .....	18
3.2.4.4	Removing a Reminder .....	18
3.2.4.4.1	Single Instance Objects .....	18
3.2.4.4.2	Recurring Calendar Objects .....	18
3.2.4.4.3	Recurring Task Objects .....	19
3.2.4.5	When a Reminder Becomes Overdue .....	19
3.2.4.5.1	Reminders That Are Ignored When They Become Overdue .....	19
3.2.4.5.2	Reminders That Are Automatically Dismissed When Overdue .....	19
3.2.4.5.3	Actions for Overdue Reminders That Are Not Ignored or Automatically Dismissed .....	20
3.2.4.5.3.1	Starting a Conference .....	20
3.2.4.6	Dismissing a Reminder .....	21
3.2.4.6.1	Dismissing for Single Instance Objects .....	21
3.2.4.6.2	Dismissing for Recurring Calendar Objects .....	21
3.2.4.6.3	Dismissing for Recurring Task Objects .....	21
3.2.4.7	Snoozing a Reminder .....	22
3.2.4.7.1	Snoozing for Single Instance Objects .....	22
3.2.4.7.2	Snoozing for Recurring Objects .....	22
3.2.4.8	Generating Instances for Recurring Task Objects .....	22
3.2.5	Message Processing Events and Sequencing Rules .....	22
3.2.6	Timer Events .....	22
3.2.7	Other Local Events .....	23
3.3	Server Details .....	23
3.3.1	Abstract Data Model .....	23
3.3.2	Timers .....	23
3.3.3	Initialization .....	23
3.3.4	Higher-Layer Triggered Events .....	23
3.3.5	Message Processing Events and Sequencing Rules .....	23
3.3.6	Timer Events .....	23
3.3.7	Other Local Events .....	23
<b>4</b>	<b>Protocol Examples .....</b>	<b>24</b>
4.1	Setting a Reminder on a Single Instance Appointment .....	25
4.2	Setting a Reminder on a Message Object .....	26
4.3	Dismissing a Reminder on a Task .....	27
4.4	Dismissing a Reminder on a Recurring Calendar Object .....	28
4.5	Snoozing a Reminder on a Contact Object .....	30
4.6	Removing a Reminder from an Instance of a Recurring Calendar Object .....	31
<b>5</b>	<b>Security .....</b>	<b>37</b>
5.1	Security Considerations for Implementers .....	37
5.2	Index of Security Parameters .....	37
<b>6</b>	<b>Appendix A: Product Behavior .....</b>	<b>38</b>

<b>7</b>	<b>Change Tracking.....</b>	<b>42</b>
<b>8</b>	<b>Index .....</b>	<b>44</b>

Preliminary

# 1 Introduction

The Reminder Settings Protocol enables a user to discover and act upon appointments, tasks, messages, or **contacts (3)** that have a deadline or for which follow-up is necessary.

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

## 1.1 Glossary

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

**Coordinated Universal Time (UTC)**  
**handle**  
**object**

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

**active reminder**  
**Calendar folder**  
**Calendar object**  
**calendar options dictionary**  
**contact**  
**Contact object**  
**Draft Message object**  
**Exception Attachment object**  
**Exception Embedded Message object**  
**instance**  
**mailbox**  
**meeting-related object**  
**Message object**  
**named property**  
**property ID**  
**recipient**  
**recurrence BLOB**  
**recurrence pattern**  
**Recurring Calendar object**  
**recurring series**  
**recurring task**  
**reminder**  
**reminder properties**  
**ROP request**  
**search folder**  
**Sent Items folder**  
**signal time**  
**special folder**  
**store**  
**Task object**  
**Tasks folder**

The following terms are specific to this document:

**dismiss:** A process that disables an overdue reminder. After a reminder is dismissed, it is not considered overdue anymore and is not signaled or displayed to a user or any agents who are acting on behalf of that user.

**due time:** The time after which a user is considered late, such as the start time of an appointment or the time at which a work item is expected to be complete.

**full reminder domain:** The maximum scope that a client is allowed to use when searching for objects that have reminders enabled. The full reminder domain includes all folders except the following: Deleted Items, Junk E-mail, Drafts, Outbox, Conflicts, Local Failures, Server Failures, and Sync Issues.

**minimal reminder domain:** The smallest scope that a client is allowed to use when searching for objects that have an active reminder. The minimal reminder domain includes the following folders: Inbox, primary Contacts, primary Calendar, and primary Tasks. It does not include sub-folders.

**overdue reminder:** An active reminder whose signal time has passed.

**Recurring object:** An object that is a Recurring Calendar object or a Recurring Task object.

**Recurring Task object:** A Task object that represents a recurring task.

**reminder domain:** A set of folders that are searched for objects that have an active reminder.

**reminder queue:** A sorted list of objects that are in a reminder domain and have been stamped with properties implying that they could have an active reminder.

**snooze:** A process that delays an overdue reminder by a specified time interval. At the end of the time interval, the reminder becomes overdue again.

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

## 1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 1.2.1 Normative References

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

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

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

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

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

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

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

[MS-OXCTABL] Microsoft Corporation, "[Table Object Protocol Specification](#)".

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

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

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

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

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

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

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

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

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

### 1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

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

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

## 1.3 Overview

This protocol allows the user or an agent who is acting on behalf of the user to associate a **reminder** with a given **Message object** with the intention that a client will somehow signal the user or agent when the **signal time** is reached.

This protocol specifies the following:

- Methods by which a client can add, remove, or modify reminders on a Message object.
- Methods by which a client can **dismiss** or **snooze** a reminder.
- Details that allow the client to process **overdue reminders**.

For example, a user might have a recurring appointment but might want to remove the reminder for a particular instance of the appointment. For examples of how a client sets the properties on a Message object such as an appointment, see section [4](#).

### 1.3.1 Creating, Modifying, and Removing Reminders

The client creates, modifies, or removes reminders by setting **reminder properties** on Message objects, as specified in section [3.2.4](#). The client can specify such things as when to signal a reminder, the **due time** for the Message object about which the user is to be reminded, whether to play a sound when the reminder is signaled, which sound to play, and whether the reminder is



enabled. On a **Draft Message object**, reminders for the **recipient (1)** or recipients (1) can be specified independently from those for the sender.

### 1.3.2 Processing Overdue Reminders

The client processes overdue reminders as specified in section 3.2.4.5. Before signaling an overdue reminder, the client is responsible for determining which reminders to ignore and which to automatically dismiss.

### 1.3.3 Dismissing and Snoozing Reminders

The client can dismiss or snooze overdue reminders, as specified in section 3.2.4.6 and section 3.2.4.7.

## 1.4 Relationship to Other Protocols

The Reminder Settings Protocol extends the Message and Attachment Object Protocol, as specified in [MS-OXCMMSG], and relies on an understanding of how to work with folders, messages, and tables. For information, see [MS-OXCDATA], [MS-OXPROPS], [MS-OXOMSG], [MS-OXCFOLD], [MS-OXCTABL], [MS-OXOSFLD], [MS-OXOSRCH], [MS-OXOCFG], [MS-OXOCAL], [MS-OXOTASK], and [MS-OXOFLAG].

## 1.5 Prerequisites/Preconditions

For requesting and configuring reminders, the Reminder Settings Protocol assumes that the client has previously logged on to the server and has acquired a **handle** to the Message object for which it intends to create or update reminder properties, as specified in [MS-OXCFOLD] and [MS-OXCMMSG].

For discovering Message objects that have reminders, the Reminder Settings Protocol assumes that the client has previously logged on to the server and has acquired a handle to the **mailbox store** in which it intends to search, as specified in [MS-OXCSTOR].

## 1.6 Applicability Statement

The Reminder Settings Protocol is used to specify and raise time-specific reminders to a user or an agent who is acting on behalf of a user.

## 1.7 Versioning and Capability Negotiation

None.

## 1.8 Vendor-Extensible Fields

This protocol provides no vendor extensibility beyond what is already specified in [MS-OXCMMSG].

## 1.9 Standards Assignments

None.

## 2 Messages

### 2.1 Transport

This protocol uses the transport specified in [\[MS-OXCMMSG\]](#).

### 2.2 Message Syntax

Reminders can be created and modified on any type of Message object by clients and servers. Except where noted, this section defines constraints to which both clients and servers adhere when operating on the reminder properties of Message objects.

Clients operate on the reminder properties of Message objects by using the Message and Attachment Object Protocol, as specified in [\[MS-OXCMMSG\]](#). How a server operates on the reminder properties of Message objects is implementation-dependent. The results of any such operation are exposed to clients in a manner that is consistent with the Reminder Settings Protocol.

#### 2.2.1 Properties Used to Specify and/or Decode Reminders

Properties that are used to specify the reminder for a given Message object are specified in section [2.2.1.1](#) through section [2.2.1.9](#). These properties are set by the client to ensure that the user is reminded at the time and in the manner that the user wants.

##### 2.2.1.1 PidLidReminderSet Property

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

The **PidLidReminderSet** property ([\[MS-OXPROPS\]](#) section 2.223) specifies whether a reminder is set on the Message object.

If a **Recurring Calendar object** has the **PidLidReminderSet** property set to **TRUE**, the client can override this value for exceptions. For details, see the definition of the **PidLidAppointmentRecur** property in [\[MS-OXOCAL\]](#) section 2.2.1.44.

If the **PidLidReminderSet** property is set to **FALSE** on a Recurring Calendar object, reminders are disabled for the entire series, including exceptions.

For **Recurring Task objects**, the **PidLidReminderSet** property cannot be overridden by exceptions. For details, see [\[MS-OXOCAL\]](#) and [\[MS-OXOTASK\]](#).

##### 2.2.1.2 PidLidReminderSignalTime Property

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

The **PidLidReminderSignalTime** property ([\[MS-OXPROPS\]](#) section 2.224) specifies the time when a reminder transitions from pending to overdue. This property **MUST** be set if the **PidLidReminderSet** property (section [2.2.1.1](#)) is set to **TRUE**. Clients **MUST** set the value of this property in **Coordinated Universal Time (UTC)**.

This property is not overridden by exceptions; there is a single signal time per Recurring Calendar object.

The value of this property is not interpreted strictly as **UTC**. For details, see [\[MS-OXOCAL\]](#) section 3.1.5.5.

### 2.2.1.3 PidLidReminderDelta Property

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

The **PidLidReminderDelta** property ([\[MS-OXPROPS\]](#) section 2.219) specifies the interval, in minutes, between the time at which the reminder first becomes overdue and the start time of the **Calendar object**. This property MUST be set on Calendar objects. For all Message objects other than Calendar objects, this property SHOULD be set to 0x00000000 and is ignored.

When a reminder is dismissed for one **instance** of a Recurring Calendar object, the value of this property is used in the calculation of the signal time for the next instance. For details about creating Calendar objects, see [\[MS-OXOCAL\]](#).

### 2.2.1.4 PidLidReminderTime Property

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

The **PidLidReminderTime** property ([\[MS-OXPROPS\]](#) section 2.225) specifies the initial signal time for Message objects other than Calendar objects. For Calendar objects, the value of this property represents the time after which the user would be late; that is, the start time of the appointment. Clients MUST set the value in UTC.

### 2.2.1.5 PidLidReminderOverride Property

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

The **PidLidReminderOverride** property ([\[MS-OXPROPS\]](#) section 2.221), if set to **TRUE**, specifies that the client SHOULD [<1>](#) respect the values of the **PidLidReminderPlaySound** property (section [2.2.1.6](#)) and the **PidLidReminderFileParameter** property (section [2.2.1.7](#)). Otherwise, a client can use default values in place of the values of the **PidLidReminderPlaySound** and **PidLidReminderFileParameter** properties.

### 2.2.1.6 PidLidReminderPlaySound Property

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

The **PidLidReminderPlaySound** property ([\[MS-OXPROPS\]](#) section 2.222), if set to **TRUE**, specifies that the client SHOULD [<2>](#) play a sound when the reminder becomes overdue.

### 2.2.1.7 PidLidReminderFileParameter Property

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

The **PidLidReminderFileParameter** property ([\[MS-OXPROPS\]](#) section 2.220) SHOULD specify the full path (but MAY specify only the file name) of the sound [<3>](#) that a client SHOULD [<4>](#) play when the reminder for the Message object becomes overdue. If this property is not present, the client can use a default value.

### 2.2.1.8 PidTagReplyTime Property

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

The **PidTagReplyTime** property ([\[MS-OXOFLAG\]](#) section 2.2.3.1), upon receipt of a Message object, specifies the due time that the sender wants, presumably for an associated work item. The value of this property is otherwise ignored. Clients MUST set the value in UTC.

### 2.2.1.9 PidLidReminderType Property

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

The **PidLidReminderType** property ([MS-OXPROPS] section 2.228) SHOULD NOT [<5>](#) be set, and its value MUST be ignored.

## 2.2.2 Shared Properties

Shared properties are set when the user configures the reminder, processed, and presented to the user (or an agent who is acting on behalf of the user) as specified in section [2.2.2.1](#) through section [2.2.2.3.1](#).

### 2.2.2.1 Properties Shared with the Message and Attachment Object Protocol

This protocol shares the following properties with the Message and Attachment Object Protocol:

- **PidTagAttachDataObject** ([MS-OXCMSG] section 2.2.2.8)
- **PidTagMessageClass** ([MS-OXCMSG] section 2.2.1.3)
- **PidTagMessageFlags** ([MS-OXCMSG] section 2.2.1.6)

The semantics and accepted values are identical to those specified in [MS-OXCMSG].

### 2.2.2.2 Properties Shared with the Informational Flagging Protocol

This protocol shares the following properties with the Informational Flagging Protocol:

- **PidTagSwappedToDoData** ([MS-EXOFLAG] section 2.2.1.7)
- **PidTagSwappedToDoStore** ([MS-EXOFLAG] section 2.2.1.8)

The semantics and accepted values are identical to those specified in [MS-EXOFLAG].

### 2.2.2.3 Properties Shared with the Task-Related Objects Protocol

This protocol shares the following properties with the Task-Related Objects Protocol:

- **PidLidTaskDueDate** ([MS-OXOTASK] section 2.2.2.2.5)
- **PidLidTaskRecurrence** ([MS-OXOTASK] section 2.2.2.2.15)
- **PidLidTaskResetReminder** ([MS-OXOTASK] section 2.2.2.2.6)

Unless noted in section [2.2.2.3.1](#), the semantics and accepted values are identical to those specified in [MS-OXOTASK].

#### 2.2.2.3.1 PidLidTaskDueDate Property

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

The **PidLidTaskDueDate** property ([MS-OXOTASK] section 2.2.2.2.5) SHOULD [<6>](#) be set when the client creates a reminder on an **object (1)** that is neither a **Task object** nor a Calendar object.

Setting the **PidLidTaskDueDate** property allows for a more intuitive "due in" time when the reminder signals.

#### 2.2.2.4 Properties Shared with the Appointment and Meeting Object Protocol

This protocol shares the following properties with the Appointment and Meeting Object Protocol:

- **PidLidAppointmentRecur** ([\[MS-OXOCAL\]](#) section 2.2.1.44)
- **PidLidAutoStartCheck** ([\[MS-OXPROPS\]](#) section 2.41) [<7>](#)
- **PidLidConferencingCheck** ([\[MS-OXOCAL\]](#) section 2.2.1.50.2 [<8>](#))

Unless otherwise noted, the semantics and accepted values are identical to those specified in [\[MS-OXOCAL\]](#).

#### 2.2.3 Properties Used to Specify User Preferences that are Roamed on the Server

The property specified in section [2.2.3.1](#) is used to specify global state with respect to reminders. This property uses the protocol defined in [\[MS-OXOCFG\]](#) as a transport.

##### 2.2.3.1 piReminderUpgradeTime Property

The **piReminderUpgradeTime** property is a 32-bit integer value (specified in number of minutes since midnight, January 1, 1601) stored in the **calendar options dictionary** that specifies the first time a **full reminder domain** client was used on the mailbox. The time is interpreted in the user's current time zone, not UTC.

For details about the calendar options dictionary, see [\[MS-OXOCFG\]](#).

## 3 Protocol Details

### 3.1 Common Details

Except where noted, section [3.2.1](#) through section [3.2.4.8](#) define behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

### 3.2 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.

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

Messages with **active reminders** can be discovered by using a persistent **search folder (2)** as specified in [\[MS-OXCFOOLD\]](#). Those messages can then be loaded into a list, referred to as the **reminder queue**, that holds the relevant information in memory for pending and/or overdue reminders. This list is sorted by the value of the **PidLidReminderSignalTime** property (section [2.2.1.2](#)).

Message objects can be in one of the following states:

- No active reminder
- Pending reminder
- Overdue reminders

The reminder queue contains Message objects that have a pending or overdue reminder.

#### 3.2.2 Timers

None.

#### 3.2.3 Initialization

##### 3.2.3.1 Populating the Reminder Queue

###### 3.2.3.1.1 Scope of the Search for Active Reminders

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

One of two approaches is typically used to specify the scope of the search for active reminders in the reminder queue: **minimal reminder domain** or full reminder domain.

Clients SHOULD [<9>](#) use a full reminder domain scope strategy. Using multiple clients that use different scope strategies to access the same mailbox is not recommended.

Using multiple clients that use different **reminder domains** can lead to inconsistent reminder signaling behavior. For example, different overdue reminders might signal in different clients, which can lead to missed reminders in the client with the smaller reminder domain.

#### 3.2.3.1.1.1 Requirements for Clients Using a Full Reminder Domain

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

Clients that use a full reminder domain scope strategy MUST ensure that the **piReminderUpgradeTime** property (section [2.2.3.1](#)) is set. If the **piReminderUpgradeTime** property does not exist, the client MUST set it. The value SHOULD correspond to the current local time to reflect the time of upgrade to the expanded scope strategy of the newer client. The **piReminderUpgradeTime** property SHOULD [<10>](#) be set only once. If the value of the **piReminderUpgradeTime** property is already set, it SHOULD NOT be changed because it represents the first time an "upgraded" client was used.

#### 3.2.3.1.2 Loading/Processing

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

When populating or updating the reminder queue with Message objects found in the reminder domain, the client SHOULD:

- Ignore certain Message objects, as specified in section [3.2.4.5.1](#).
- Automatically dismiss certain Message objects, as specified in section [3.2.4.5.2](#).
- Include any remaining Message objects that have active reminders (pending or overdue reminders).

### 3.2.4 Higher-Layer Triggered Events

#### 3.2.4.1 Setting a Reminder

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

In addition to the behaviors specified in section [3.2.4.1.1](#) through section [3.2.4.1.5.2](#), the client sets the **PidLidReminderSet** property (section [2.2.1.1](#)) to **TRUE** on a Message object to enable the signaling of a reminder. The client sets the **PidLidReminderSignalTime** property (section [2.2.1.2](#)) to the specified time the reminder becomes overdue and sets the **PidLidReminderTime** property (section [2.2.1.4](#)).

##### 3.2.4.1.1 Reminders on Task Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The client sets the **PidLidReminderTime** property (section [2.2.1.4](#)) to the specified time at which the reminder becomes overdue.

### 3.2.4.1.2 Reminders on Message Objects other than Calendar Objects and Task Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The client sets the **PidLidReminderTime** property (section 2.2.1.4) and SHOULD<11> set the **PidTagReplyTime** ([MS-OXOFLAG] section 2.2.3.1) property to the specified signal time.

In summary, the client sets the following properties (or ensures that they are already set<12> for Message objects that are neither Calendar objects nor Task objects):

Property	Value
<b>PidLidReminderSet</b> (section 2.2.1.1)	<b>TRUE</b>
<b>PidLidReminderTime</b>	The specified signal time
<b>PidLidReminderSignalTime</b> (section 2.2.1.2)	The specified signal time
<b>PidTagReplyTime</b>	The specified signal time

### 3.2.4.1.3 Reminders on Calendar Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The client sets the **PidLidReminderTime** property (section 2.2.1.4), and its value SHOULD<13> correspond to the appointment start time (see [MS-OXOCAL] for details).

For Calendar objects, the client MUST set the **PidLidReminderDelta** property (section 2.2.1.3) to reflect the specified interval, in minutes, between the signal time of the reminder and the start time of the Calendar object.

In summary, the client sets the following properties (or otherwise ensures that the properties are already set).

Property	Value
<b>PidLidReminderSet</b> (section 2.2.1.1)	<b>TRUE.</b>
<b>PidLidReminderTime</b>	MUST be set, and the value SHOULD<14> be the appointment start time.
<b>PidLidReminderDelta</b>	The interval, in minutes, between signal time and the appointment start time.
<b>PidLidReminderSignalTime</b> (section 2.2.1.2)	The signal time in UTC.

### 3.2.4.1.4 Reminders on Recurring Calendar Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

For Recurring Calendar objects, the client also sets or modifies the **recurrence BLOB** of the **PidLidAppointmentRecur** property ([MS-OXOCAL] section 2.2.1.44) on the Recurring Calendar



object, and the **Exception Embedded Message object** **PidTagAttachDataObject** ([MS-OXCMMSG] section 2.2.2.8) on the **Exception Attachment object**, as specified in the following paragraphs.

To modify the reminder for specific instances in a **recurring series**, the client MUST create an exception if one does not already exist for that instance and then modify the recurrence BLOB and Exception Embedded Message object to store whichever properties are different from the Recurring Calendar object, such as **StartTime**, **ReminderDelta**, or **ReminderSet**. For details about the recurrence BLOB **PidLidAppointmentRecur** and the Exception Attachment object, see [MS-OXOCAL]. For details about accessing the Exception Embedded Message object, which is stored in the **PidTagAttachDataObject** property, see [MS-OXCMMSG].

Setting the **PidLidReminderSet** property (section 2.2.1.1) to **FALSE** on the Recurring Calendar object MUST take precedence over any reminders that are specified for exceptions. To disable reminders for the entire series, including all exceptions, set the **PidLidReminderSet** property to **FALSE** on the Recurring Calendar object.

If one or more instances, but not all instances, need a reminder, set the **PidLidReminderSet** property to **TRUE** on the Recurring Calendar object. Then set the **ReminderSet** field to **FALSE** in the recurrence BLOB for each instance that does not have a reminder, and set the **PidLidReminderSet** property on the Exception Embedded Message object to **FALSE**, creating exceptions as appropriate.

#### 3.2.4.1.5 Setting Reminders on Draft Message Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

By using the same mechanism that allows the client to set flagging properties for the sender of a Message object independently from the flagging properties specified in [MS-OXOFLAG] that are sent to recipients (1), a client can set the reminder properties for the sender of a Message object independently from the reminder properties that are sent to recipients (1).

If either a sender or recipient (1) reminder is specified by a client that supports sender reminders, the **PidTagSwappedToDoStore** property ([MS-OXOFLAG] section 2.2.1.8) MUST be set to enable post-transmit processing. The remaining work to process recipient (1) and sender reminders is covered during post-transmit processing as specified in [MS-OXOFLAG].

##### 3.2.4.1.5.1 Setting a Reminder for all Recipients

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

To set a recipient (1) reminder on a Draft Message object, the client sets the **PidLidReminderSet** property (section 2.2.1.1) to **TRUE** and sets the **PidLidReminderTime** property (section 2.2.1.4), the **PidLidReminderSignalTime** property (section 2.2.1.2), and the **PidTagReplyTime** property ([MS-OXOFLAG] section 2.2.3.1) to the required signal time.

##### 3.2.4.1.5.2 Setting a Reminder for the Sender

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The client MUST set the **rtmReminder** field to the required signal time and the **fReminderSet** field to **TRUE** in the **PtypBinary** property **PidTagSwappedToDoData** ([MS-OXOFLAG] section 2.2.1.7).

The client sets the corresponding validity bits (0x00000040 and 0x00000080) to 1 in the **dwFlags** field in the **PidTagSwappedToDoData** property.

### 3.2.4.2 Post-Transmit Processing

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The post-transmit processing required by the Reminder Settings Protocol is identical to the post-transmit processing specified in [\[MS-OXOFLAG\]](#).

### 3.2.4.3 Post-Receive Processing

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

For received Message objects that are not time-flagged Message objects [\[MS-OXOFLAG\]](#), if the **PidTagReplyTime** property ([\[MS-OXOFLAG\]](#) section 2.2.3.1) exists, the value of the **PidTagReplyTime** property SHOULD<15> be copied into the **PidLidReminderTime** (section 2.2.1.4) and **PidLidReminderSignalTime** (section 2.2.1.2) properties, and the **PidLidReminderSet** property (section 2.2.1.1) SHOULD be set to **TRUE**.

### 3.2.4.4 Removing a Reminder

#### 3.2.4.4.1 Single Instance Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The client sets the **PidLidReminderSet** property (section 2.2.1.1) to **FALSE**, which disables the reminder.

If the Message object is neither a Task object nor a Calendar object, the client MAY<16> delete the **PidLidReminderTime** property (section 2.2.1.4) and the **PidLidReminderSignalTime** property (section 2.2.1.2) and SHOULD delete the **PidTagReplyTime** property ([\[MS-OXOFLAG\]](#) section 2.2.3.1).

If the Message object is a Calendar object, the client SHOULD<17> set the **PidLidAutoStartCheck** property ([\[MS-OXPROPS\]](#) section 2.41) to **FALSE**.

If the Message object is a Task object, the client SHOULD NOT change the value of the **PidLidAutoStartCheck** property.

#### 3.2.4.4.2 Recurring Calendar Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

To remove the reminder for all instances, including any exceptions, the client MUST set the **PidLidReminderSet** property (section 2.2.1.1) to **FALSE** on the Recurring Calendar object.

To remove the reminder for a single instance, the client ensures that an exception exists for the instance, sets the value of the **ReminderSet** field for the corresponding exception in the recurrence BLOB to **FALSE**, sets the **PidLidReminderSet** property on the Exception Embedded Message object to **FALSE**, and SHOULD<18> set the **PidLidAutoStartCheck** property ([\[MS-OXPROPS\]](#) section 2.41) on the Exception Embedded Message object to **FALSE**. For details about how to modify the

appointment recurrence BLOB of the **PidLidAppointmentRecur** property ([MS-OXOCAL] section 2.2.1.44) and the Exception Attachment object, see [MS-OXOCAL]. For details about accessing the Exception Embedded Message object, see ([MS-OXCMSG] section 2.2.2.8).

### 3.2.4.4.3 Recurring Task Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

To remove the reminder for all instances, the client sets the **PidLidReminderSet** property (section 2.2.1.1) to **FALSE** on the Recurring Task object, and SHOULD<19> ensure that the **PidLidTaskResetReminder** property ([MS-OXOTASK] section 2.2.2.2.6) either is not present or has the value **FALSE**.

Removing the reminder for a single instance of a Recurring Task object is not possible because Recurring Task objects do not support exceptions. Therefore, the reminder can be enabled or disabled only for the entire **recurrence pattern**.

### 3.2.4.5 When a Reminder Becomes Overdue

#### 3.2.4.5.1 Reminders That Are Ignored When They Become Overdue

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

Minimal reminder domain clients SHOULD ignore reminders for the following types of Message objects, and full reminder domain clients MUST ignore reminders for the following types of Message objects:

- **Meeting-related objects** - any Message object where the **PtypString** property ([MS-OXCDATA] section 2.11.1) **PidTagMessageClass** ([MS-OXCMSG] section 2.2.1.3) contains the prefix "IPM.Schedule".<20>
- Unsent mail - any Message object that contains the **PtypInteger32** ([MS-OXCDATA] section 2.11.1) property **PidTagMessageFlags** ([MS-OXCMSG] section 2.2.1.6) with the flag set corresponding to **MSGFLAG\_UNSENT**.
- Message objects that have the **PtypBinary** ([MS-OXCDATA] section 2.11.1) property **PidTagSwappedToDoStore** ([MS-OXOFLAG] section 2.2.1.8) set.<21>

#### 3.2.4.5.2 Reminders That Are Automatically Dismissed When Overdue

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

Clients following the full reminder domain scope strategy, as specified in section 3.2.3.1.1, SHOULD<22> automatically dismiss any Message objects found outside the minimal reminder domain that have a signal time (**PidLidReminderSignalTime** (section 2.2.1.2)) that is earlier than the upgrade time (**piReminderUpgradeTime** (section 2.2.3.1)), so that the Message objects never have to be considered again. A reminder is automatically dismissed by setting the **PidLidReminderSet** property (section 2.2.1.1) to **FALSE**.

### 3.2.4.5.3 Actions for Overdue Reminders That Are Not Ignored or Automatically Dismissed

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

When a reminder becomes overdue, the client commonly prompts the user to take action, such as dismissing or snoozing. To identify the reminder to the user (or user agent), the client can use any properties<23> in addition to the following properties that the client SHOULD use to determine the due time for displaying how long until the Message object is due or how much time has passed since the Message object became due.

When displaying how long until the Message object is due, the time is calculated from the due time, as summarized in the following table.

Case	Due time
For Message objects other than Calendar objects (Task objects and flagged Message objects)<24>	If the <b>PidLidTaskDueDate</b> property ([MS-OXOTASK] section 2.2.2.2.5) exists, the client SHOULD<25> use "TaskDueDate + default End-of-Day time"; otherwise use the value of the <b>PidLidReminderTime</b> property (section 2.2.1.4) as the due time. For details about End-of-Day time, see the <b>End</b> element in [MS-OXOCFG] section 2.2.5.2.1.
For single instance Calendar objects	Use the value of the <b>PidLidReminderTime</b> property.
For Recurring Calendar objects	The client SHOULD use the start time of the latest instance with enabled reminder for which the time defined by evaluating the expression ( <b>StartTime</b> – <b>ReminderDelta</b> ) is less than or equal to the time encoded in the value of the <b>PidLidReminderSignalTime</b> property (section 2.2.1.2) on the Recurring Calendar object. <b>Note</b> <b>StartTime</b> and <b>ReminderDelta</b> could be overridden by exceptions.

If the **PidLidReminderOverride** property (section 2.2.1.5) is set to **FALSE** or is not set, the client SHOULD<26> play the default reminder sound.

If the **PidLidReminderOverride** property is set to **TRUE**, the client SHOULD<27> use the values of the **PidLidReminderPlaySound** (section 2.2.1.6) and **PidLidReminderFileParameter** (section 2.2.1.7) properties from the reminder properties of the given Message object.

If the **PidLidReminderPlaySound** property is **TRUE**, the client SHOULD<28> use the value of the **PidLidReminderFileParameter** property to locate the specified sound file and then play it.

If the **PidLidReminderPlaySound** property is **FALSE**, the client SHOULD NOT<29> play a reminder sound.

If the client uses the value of the **PidLidReminderFileParameter** property to play a sound and it does not represent a full path, the client can search for a matching file name.

#### 3.2.4.5.3.1 Starting a Conference

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The client can launch a conference if the **PidLidConferencingCheck** property ([MS-OXOCAL] section 2.2.1.50.2) and the **PidLidAutoStartCheck** property ([MS-OXPROPS] section 2.41) are both set to **TRUE**. The client MAY launch a conference if the **PidLidConferencingType** property

([\[MS-OXOCAL\]](#) section 2.2.1.50.3) is set to 0x00000000 and use the values of the **PidLidConferencingCheck**, **PidLidConferencingType**, **PidLidOrganizerAlias** ([\[MS-OXOCAL\]](#) section 2.2.1.50.6), **PidLidDirectory** ([\[MS-OXPROPS\]](#) section 2.94), **PidTagConversationTopic** ([\[MS-OXOMSG\]](#) section 2.2.1.5), **PidLidCollaborateDoc** ([\[MS-OXOCAL\]](#) section 2.2.1.50.7), **PidLidNetShowUrl** ([\[MS-OXOCAL\]](#) section 2.2.1.50.8), and **PidLidAppointmentStateFlags** ([\[MS-OXOCAL\]](#) section 2.2.1.10) properties.

### 3.2.4.6 Dismissing a Reminder

#### 3.2.4.6.1 Dismissing for Single Instance Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

For all single instance Message objects, the client MUST set the **PidLidReminderSet** property (section [2.2.1.1](#)) to **FALSE**.

In addition, for single instance Task objects, the client can follow the guidance in section [3.2.4.6.2](#) for dismissing a Recurring Task object.

#### 3.2.4.6.2 Dismissing for Recurring Calendar Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

If there is a future instance (including exceptions) with a pending reminder (in other words, not disabled individually on all future instances), the client MUST set the **PidLidReminderSignalTime** property (section [2.2.1.2](#)) on the Recurring Calendar object based on the **PidLidReminderDelta** property (section [2.2.1.3](#)) and the start of that instance (that is, **NextInstanceStartTime** - **PidLidReminderDelta**). If no more instances (including exceptions) have a pending reminder, it is recommended that the client avoid setting the **PidLidReminderSet** property (section [2.2.1.1](#)) to **FALSE**, and the client MUST set the **PidLidReminderSignalTime** property to the **PtypTime** ([\[MS-OXCDATA\]](#) section 2.11.1) value "Midnight (UTC) January 1, 4501" (value Low:0xA3DD4000 High:0x0CB34557).

It is recommended that the client avoid setting the **PidLidReminderSet** property to **FALSE** when dismissing reminders for Recurring Calendar objects, even when no more instances require a reminder to signal. This is to preserve the user's intent to signal reminders, in case the recurrence is extended at a later date, to include instances in the future.

**Note** Instances never turn into exceptions as a result of dismissing/snoozing.

#### 3.2.4.6.3 Dismissing for Recurring Task Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The client MUST set the **PidLidReminderSet** property (section [2.2.1.1](#)) to **FALSE** and SHOULD [<30>](#) set the **PidLidTaskResetReminder** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.6) to **TRUE** to dismiss the reminder.

For Task objects, if the value of the **PidLidReminderTime** property (section [2.2.1.4](#)) is in the future, the client SHOULD [<31>](#) set the **PidLidReminderSignalTime** property (section [2.2.1.2](#)) to the same value as the **PidLidReminderTime** property so that the reminder will be signaled at the appropriate time for the next instance.

### 3.2.4.7 Snoozing a Reminder

#### 3.2.4.7.1 Snoozing for Single Instance Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The client MUST set the **PidLidReminderSignalTime** property (section [2.2.1.2](#)) to the time specified by the user (or user agent).

#### 3.2.4.7.2 Snoozing for Recurring Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

The client MUST set the **PidLidReminderSignalTime** property (section [2.2.1.2](#)) to either the specified time or the signal time for the next instance with an active reminder (for example, **NextInstanceStart - ReminderDelta**), whichever is earlier.

**Note** Instances never turn into exceptions as a result of dismissing or snoozing.

### 3.2.4.8 Generating Instances for Recurring Task Objects

Except where noted, this section defines behavioral constraints to which both clients and servers MUST adhere when operating on the reminder properties of Message objects.

In addition to the details related to the semantics and the use of the **PidLidTaskResetReminder** property specified in [\[MS-OXOTASK\]](#) section 2.2.2.2.6, the following applies to generating instances for Recurring Task objects:

When a Recurring Task object is updated after one instance of the task is marked complete, if the value of the **PtypBoolean** ([\[MS-OXCDATA\]](#) section 2.11.1) property **PidLidTaskResetReminder** is **TRUE** or the value of the **PidLidReminderSet** property (section [2.2.1.1](#)) is **TRUE**, a reminder SHOULD [<32>](#) be set on the Recurring Task object by setting both the **PidLidReminderTime** property (section [2.2.1.4](#)) and the **PidLidReminderSignalTime** property (section [2.2.1.2](#)) to the next signal time. These values SHOULD correspond to the same time as the previous reminder, but on a different day, such that the difference between the new signal time and the new task due date is the same as the difference between the old signal time and old task due date values. The client adjusts for Daylight Saving Time when necessary (if a change in DST occurs between the current and next instance), so the reminder signals at the same time of day.

After an instance of a **recurring task** that has an active reminder is complete, the future instance MUST have the **PidLidReminderSet** property set to **TRUE**, and the **PidLidTaskResetReminder** property SHOULD NOT [<33>](#) be **TRUE**. The now-completed instance MUST have the **PidLidReminderSet** property set to **FALSE**, and the **PidLidTaskResetReminder** property SHOULD NOT [<34>](#) be **TRUE**.

### 3.2.5 Message Processing Events and Sequencing Rules

None.

### 3.2.6 Timer Events

None.



### 3.2.7 Other Local Events

None.

## 3.3 Server Details

The server fulfills the server role as specified in the following protocols.

- Office Exchange Protocols Master Property List Specification [\[MS-OXPROPS\]](#)
- E-Mail Object Protocol Specification [\[MS-OXOMSG\]](#)
- Folder Object Protocol Specification [\[MS-OXCFOLD\]](#)
- Table Object Protocol Specification [\[MS-OXCTABL\]](#)
- Special Folders Protocol Specification [\[MS-OXOSFLD\]](#)
- Search Folder List Configuration Protocol Specification [\[MS-OXOSRCH\]](#)
- Configuration Information Protocol Specification [\[MS-OXOCFG\]](#)
- Appointment and Meeting Object Protocol Specification [\[MS-OXOCAL\]](#)
- Task-Related Objects Protocol Specification [\[MS-OXOTASK\]](#)
- Informational Flagging Protocol Specification [\[MS-OXOFLAG\]](#)

### 3.3.1 Abstract Data Model

None.

### 3.3.2 Timers

None.

### 3.3.3 Initialization

None.

### 3.3.4 Higher-Layer Triggered Events

None.

### 3.3.5 Message Processing Events and Sequencing Rules

None.

### 3.3.6 Timer Events

None.

### 3.3.7 Other Local Events

None.

## 4 Protocol Examples

Before adding, removing, or modifying a reminder on any Message objects, the client sends a **RopGetPropertyIdsFromNames** ROP request ([MS-OXCROPS] section 2.2.8.1) to ask the server to map from **named properties** to **property IDs**, as follows.

Property	Property Set GUID	Name or ID
<b>PidLidReminderSet</b> (section 2.2.1.1)	{00062008-0000-0000-C000-000000000046}	0x8503
<b>PidLidReminderTime</b> (section 2.2.1.4)	{00062008-0000-0000-C000-000000000046}	0x8502
<b>PidLidReminderDelta</b> (section 2.2.1.3)	{00062008-0000-0000-C000-000000000046}	0x8501
<b>PidLidReminderSignalTime</b> (section 2.2.1.2)	{00062008-0000-0000-C000-000000000046}	0x8560
<b>PidLidTaskDueDate</b> ([MS-OXOTASK] section 2.2.2.2.5)	{00062003-0000-0000-C000-000000000046}	0x8105
<b>PidLidTaskResetReminder</b> ([MS-OXOTASK] section 2.2.2.2.6)	{00062003-0000-0000-C000-000000000046}	0x8107
<b>PidLidAppointmentRecur</b> ([MS-OXOCAL] section 2.2.1.44)	{00062002-0000-0000-C000-000000000046}	0x8216
<b>PidLidAutoStartCheck</b> ([MS-OXPROPS] section 2.41)	{00062002-0000-0000-C000-000000000046}	0x8244
<b>PidLidFExceptionalAttendees</b> ([MS-OXOCAL] section 2.2.2.3)	{00062002-0000-0000-C000-000000000046}	0x822B

The server responds with the following identifiers, which will be used in the examples that follow (the actual identifiers are at the discretion of the server).

Property	Property IDs
<b>PidLidReminderSet</b>	0x8004
<b>PidLidReminderTime</b>	0x8005
<b>PidLidReminderDelta</b>	0x81FF
<b>PidLidReminderSignalTime</b>	0x8006
<b>PidLidTaskDueDate</b>	0x8144
<b>PidLidTaskResetReminder</b>	0x815B
<b>PidLidAppointmentRecur</b>	0x81AE
<b>PidLidAutoStartCheck</b>	0x82E0
<b>PidLidFExceptionalAttendees</b>	0x82D7



## 4.1 Setting a Reminder on a Single Instance Appointment

John has an existing appointment for dinner with Robin Counts at Coho Vineyard on February 15, 2008, from 6:00 P.M. to 7:00 P.M., Pacific Standard Time. Around 11:44 A.M., John sets a reminder on the appointment for 30 minutes.

Before John modified the Message object, some of the properties were as follows:

Property Name	Property IDs	Property Type	Data	Value
<b>PidTagNormalizedSubject</b> ( <a href="#">[MS-OXCMSG]</a> section 2.2.1.10)	0x0E1D	<b>PtypString</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	44 00 69 00 6E 00 6E 00 65 00 72 00 20 00 77 00 69 00 74 00 68 00 20 00 52 00 6F 00 62 00 69 00 6E 00 20 00 43 00 6F 00 75 00 6E 00 74 00 73 00 00 00	"Dinner with Robin Counts"
<b>PidLidLocation</b> ( <a href="#">[MS-OXOCAL]</a> section 2.2.1.4)	0x8009	<b>PtypString</b>	43 00 6F 00 68 00 6F 00 20 00 56 00 69 00 6E 00 65 00 79 00 61 00 72 00 64 00 00 00	"Coho Vineyard"
<b>PidTagStartDate</b> ( <a href="#">[MS-OXOCAL]</a> section 2.2.1.30)	0x0060	<b>PtypTime</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	00 10 00 A3 3F 70 C8 01	2008/02/16 02:00:00.000
<b>PidTagEndDate</b> ( <a href="#">[MS-OXOCAL]</a> section 2.2.1.31)	0x0061	<b>PtypTime</b>	00 78 C4 04 48 70 C8 01	2008/02/16 03:00:00.000
<b>PidLidReminderSet</b> (section <a href="#">2.2.1.1</a> )	0x8004	<b>PtypBoolean</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	00	<b>FALSE</b>
<b>PidLidReminderTime</b> (section <a href="#">2.2.1.4</a> )	0x8005	<b>PtypTime</b>	00 10 00 A3 3F 70 C8 01	2008/02/16 02:00:00.000
<b>PidLidReminderSignalTime</b> (section <a href="#">2.2.1.2</a> )	0x8006	<b>PtypTime</b>	00 F6 8E 8A 3D 70 C8 01	2008/02/16 01:45:00.000
<b>PidLidReminderDelta</b> (section <a href="#">2.2.1.3</a> )	0x81FF	<b>PtypInteger32</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	0F 00 00 00	15 minutes

In response to the user-entered data, the client sends a **RopOpenMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.1) and waits for the server to respond. The server response contains a handle to the Message object.

The client then sends a **RopSetProperties** ROP request ([\[MS-OXCROPS\]](#) section 2.2.8.6) with the following properties.

Property	Property IDs	Property Type	Data	Value
<b>PidLidReminderSet</b>	0x8004	<b>PtypBoolean</b>	01	<b>TRUE</b>
<b>PidLidReminderDelta</b>	0x81FF	<b>PtypInteger32</b>	1E 00 00 00	30 minutes
<b>PidLidReminderSignalTime</b>	0x8006	<b>PtypTime</b>	00 DC 1D 72 3B 70 C8 01	2008/02/16 01:30:00.000

Finally, to persist the Message object on the server, the client sends a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3) and releases the Message object by sending a **RopRelease** ROP request ([\[MS-OXCROPS\]](#) section 2.2.15.3).

## 4.2 Setting a Reminder on a Message Object

John, who is currently working in the Pacific Standard time zone, has an existing message titled "Important Project Information" that was previously flagged for follow-up by February 14, 2008. To ensure that it is not forgotten, John sets a reminder for 6:00 P.M.

Before John modified the Message object, some of its properties were as follows.

Property Name	Property IDs	Property Type	Data	Value
<b>PidTagNormalizedSubject</b> ( <a href="#">[MS-OXCMSG]</a> section 2.2.1.10)	0x0E1D	<b>PtypString</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	49 00 6D 00 70 00 6F 00 72 00 74 00 61 00 6E 00 74 00 20 00 50 00 72 00 6F 00 6A 00 65 00 63 00 74 00 20 00 49 00 6E 00 66 00 6F 00 72 00 6D 00 61 00 74 00 69 00 6F 00 6E 00 00 00	"Important Project Information"
<b>PidLidTaskStartDate</b> ( <a href="#">[MS-OXOTASK]</a> section 2.2.2.2.4)	0x8143	<b>PtypTime</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	00 C0 A3 8A 9C 6E C8 01	2008/02/14 00:00:00.000
<b>PidLidTaskDueDate</b> ( <a href="#">[MS-OXOTASK]</a> section 2.2.2.2.5)	0x8144	<b>PtypTime</b>	00 C0 A3 8A 9C 6E C8 01	2008/02/14 00:00:00.000
<b>PidLidReminderSet</b> (section 2.2.1.1)	0x8004	<b>PtypBoolean</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	00	<b>FALSE</b>
<b>PidLidReminderTime</b> (section 2.2.1.4)	0x8005	<b>PtypTime</b>	Deleted	N/A
<b>PidLidReminderSignalTime</b> (section 2.2.1.2)	0x8006	<b>PtypTime</b>	Deleted	N/A

In response to the user-entered data, the client sends a **RopOpenMessage ROP request** ([\[MS-OXCROPS\]](#) section 2.2.6.1) and waits for the server to respond. The server response contains a handle to the Message object.

The client then sends a **RopSetProperties** ROP request ([MS-OXCROPS] section 2.2.8.6) with the following properties.

Property	Property IDs	Property Type	Data	Value
<b>PidLidReminderTime</b>	0x8005	<b>PtypTime</b>	00 50 96 78 76 6f c8 01	2008/02/15 02:00:00.000
<b>PidLidReminderSignalTime</b>	0x8006	<b>PtypTime</b>	00 50 96 78 76 6f c8 01	2008/02/15 02:00:00.000
<b>PidLidReminderSet</b> (section 2.2.1.1)	0x8004	<b>PtypBoolean</b>	01	<b>TRUE</b>
<b>PidTagReplyTime</b> ([MS-OXOFLAG] section 2.2.3.1)	0x0030	<b>PtypTime</b>	00 50 96 78 76 6f c8 01	2008/02/15 02:00:00.000
<b>PidTagReplyRequested</b> ([MS-OXOMSG] section 2.2.1.37)	0x0C17	<b>PtypBoolean</b>	01	<b>TRUE</b>
<b>PidTagResponseRequested</b> ([MS-OXOMSG] section 2.2.1.37)	0x0063	<b>PtypBoolean</b>	01	<b>TRUE</b>
<b>PidTagFlagStatus</b> ([MS-OXOFLAG] section 2.2.1.1)	0x1090	<b>PtypInteger32</b> ([MS-OXCADATA] section 2.11.1)	02 00 00 00	follow-up marked

Next, the client sends a **RopDeletePropertiesNoReplicate** ROP request ([MS-OXCROPS] section 2.2.8.9) for the following property.

Property	Property ID	Property Type
<b>PidTagFlagCompleteTime</b> ([MS-OXOFLAG] section 2.2.1.3)	0x1091	<b>PtypTime</b>

Finally, the client persists the Message object on the server by sending a **RopSaveChangesMessage** ROP request ([MS-OXCROPS] section 2.2.6.3) and releases the Message object by sending a **RopRelease** ROP request ([MS-OXCROPS] section 2.2.15.3).

### 4.3 Dismissing a Reminder on a Task

John had previously created a Task object titled "Prepare for Contoso presentation" with a reminder that was signaled on February 15, 2008, at 11:30 A.M. Pacific Standard Time. Because he is done, John dismisses the reminder at 11:31 A.M.

Before John dismissed the reminder, some of its properties were as follows:

Property Name	Property ID	Property Type	Data	Value
<b>PidTagNormalizedSubject</b> ([MS-OXCMSG] section 2.2.1.1)	0x0E1D	<b>PtypString</b> ([MS-OXCSTR] section 2.11.1)	50 00 72 00 65 00 70 00 61 00 72 00	"Prepare for Contoso"

Property Name	Property ID	Property Type	Data	Value
2.2.1.10)		<a href="#">OXCDATA1</a> section 2.11.1)	65 00 20 00 66 00 6F 00 72 00 20 00 43 00 6F 00 6E 00 74 00 6F 00 73 00 6F 00 20 00 70 00 72 00 65 00 73 00 65 00 6E 00 74 00 61 00 74 00 69 00 6F 00 6E 00 00 00	presentation"
<b>PidLidTaskStartDate</b> ( <a href="#">[MS-OXOTASK]</a> section 2.2.2.2.4)	0x8143	<b>PtypTime</b> ( <a href="#">[MS-OXCDATA1]</a> section 2.11.1)	00 80 0D B5 65 6F C8 01	2008/02/15 00:00:00.000
<b>PidLidTaskDueDate</b> ( <a href="#">[MS-OXOTASK]</a> section 2.2.2.2.5)	0x8144	<b>PtypTime</b>	00 80 0D B5 65 6F C8 01	2008/02/15 00:00:00.000
<b>PidLidReminderSet</b> (section <a href="#">2.2.1.1</a> )	0x8004	<b>PtypBoolean</b> ( <a href="#">[MS-OXCDATA1]</a> section 2.11.1)	01	<b>TRUE</b>
<b>PidLidReminderTime</b> (section <a href="#">2.2.1.4</a> )	0x8005	<b>PtypTime</b>	00 6C 83 27 09 70 C8 01	2008/02/15 19:30:00.000
<b>PidLidReminderSignalTime</b> (section <a href="#">2.2.1.2</a> )	0x8006	<b>PtypTime</b>	00 6C 83 27 09 70 C8 01	2008/02/15 19:30:00.000
<b>PidLidTaskResetReminder</b> ( <a href="#">[MS-OXOTASK]</a> section 2.2.2.2.6)	0x815B	<b>PtypBoolean</b>	Deleted	N/A

In response to the user-entered data, the client sends a **RopOpenMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.1) and waits for the server to respond. The server response contains a handle to the Message object.

The client then sends a **RopSetProperties** ROP request ([\[MS-OXCROPS\]](#) section 2.2.8.6) with the following properties.

Property	Property ID	Property Type	Data	Value
<b>PidLidReminderSet</b>	0x8004	<b>PtypBoolean</b>	00	<b>FALSE</b>
<b>PidLidTaskResetReminder</b>	0x815B	<b>PtypBoolean</b>	01	<b>TRUE</b>

Finally, the client persists the Message object on the server by sending a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3) and releases the Message object by sending a **RopRelease** ROP request ([\[MS-OXCROPS\]](#) section 2.2.15.3).

#### 4.4 Dismissing a Reminder on a Recurring Calendar Object

John has a recurring appointment for lunch with Ben Smith every Friday at noon Pacific Standard Time, with a 20-minute reminder. The reminder for the first instance is displayed, and John dismisses the reminder before leaving for the Coho Winery.

Before John dismissed the Message object, some of its properties were as follows:

Property Name	Property ID	Property Type	Data	Value
<b>PidTagNormalizedSubject</b> ([MS-OXCMMSG] section 2.2.1.10)	0x0E1D	<b>PtypString</b> ([MS-OXCADATA] section 2.11.1)	4C 00 75 00 6E 00 63 00 68 00 20 00 77 00 69 00 74 00 68 00 20 00 42 00 65 00 6E 00 20 00 53 00 6D 00 69 00 74 00 68 00 00 00	"Lunch with Ben Smith"
<b>PidLidLocation</b> ([MS-OXOCAL] section 2.2.1.4)	0x8009	<b>PtypString</b>	43 00 6F 00 68 00 6F 00 20 00 57 00 69 00 6E 00 65 00 72 00 79 00 00 00	"Coho Winery"
<b>PidTagStartDate</b> ([MS-OXOCAL] section 2.2.1.30)	0x0060	<b>PtypTime</b> ([MS-OXCADATA] section 2.11.1)	00 A0 65 58 0D 70 C8 01	2008/02/15 20:00:00.000
<b>PidTagEndDate</b> ([MS-OXOCAL] section 2.2.1.31)	0x0061	<b>PtypTime</b>	00 08 2A BA 15 70 C8 01	2008/02/15 21:00:00.000
<b>PidLidReminderSet</b> (section 2.2.1.1)	0x8004	<b>PtypBoolean</b> ([MS-OXCADATA] section 2.11.1)	01	<b>TRUE</b>
<b>PidLidReminderTime</b> (section 2.2.1.4)	0x8005	<b>PtypTime</b>	00 A0 65 58 0D 70 C8 01	2008/02/15 20:00:00.000
<b>PidLidReminderSignalTime</b> (section 2.2.1.2)	0x8006	<b>PtypTime</b>	00 28 24 8D 0A 70 C8 01	2008/02/15 19:40:00.000
<b>PidLidReminderDelta</b> (section 2.2.1.3)	0x81FF	<b>PtypInteger32</b> ([MS-OXCADATA] section 2.11.1)	14 00 00 00	20 minutes
<b>PidLidAppointmentRecur</b> ([MS-OXOCAL] section 2.2.1.44)	0x81AE	<b>PtypBinary</b> ([MS-OXCADATA] section 2.11.1)	Cb: 50 00 Lpb: (Information follows the table)	Cb: 80 bytes Lpb: (See [MS-OXOCAL])

The value of the **PidLidAppointmentRecur** property is as follows:

Cb: 50 00

Lpb:

```

0x0000: 04 30 04 30 0B 20 01 00-00 00 C0 21 00 00 01 00
0x0010: 00 00 00 00 00 00 20 00-00 00 23 20 00 00 0A 00
0x0020: 00 00 00 00 00 00 00 00-00 00 00 00 00 40 4A
0x0030: C3 0C DF 80 E9 5A 06 30-00 00 09 30 00 00 D0 02
0x0040: 00 00 0C 03 00 00 00 00-00 00 00 00 00 00 00

```

For information about interpreting the data, see [MS-OXOCAL].

In response to the user-entered data, the client sends a **RopOpenMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.1) and waits for the server to respond. The server response contains a handle to the Message object.

The client then sends a **RopSetProperties** ROP request ([\[MS-OXCROPS\]](#) section 2.2.8.6) with the following property.

Property	Property ID	Property Type	DATA	Value
<b>PidLidReminderSignalTime</b>	0x8006	<b>PtypTime</b>	00 68 08 B6 8A 75 C8 01	2008/02/22 19:40:00.000

Finally, the client persists the Message object on the server by sending a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3) and releases the Message object by sending a **RopRelease** ROP request ([\[MS-OXCROPS\]](#) section 2.2.15.3).

## 4.5 Snoozing a Reminder on a Contact Object

John has an existing **Contact object** for Adam Barr with a reminder set so that he is reminded to call on February 15, 2008, at 11:15 A.M. Pacific Standard Time. The reminder is displayed, but John is still in the middle of a project, so at 11:18 A.M., John snoozes the reminder for 1 hour.

Before John snoozed the Message object, some of its properties were as follows.

Property Name	Property ID	Property Type	Data	Value
<b>PidTagNormalizedSubject</b> ( <a href="#">[MS-OXCMSG]</a> section 2.2.1.10)	0x0E1D	<b>PtypString</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	41 00 64 00 61 00 6D 00 20 00 42 00 61 00 72 00 72 00 00 00	"Adam Barr"
<b>PidLidTaskStartDate</b> ( <a href="#">[MS-OXOTASK]</a> section 2.2.2.2.4)	0x8143	<b>PtypTime</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	00 80 0D B5 65 6F C8 01	2008/02/15 00:00:00.000
<b>PidLidTaskDueDate</b> ( <a href="#">[MS-OXOTASK]</a> section 2.2.2.2.5)	0x8144	<b>PtypTime</b>	00 80 0D B5 65 6F C8 01	2008/02/15 00:00:00.000
<b>PidLidReminderSet</b> (section <a href="#">2.2.1.1</a> )	0x8004	<b>PtypBoolean</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	01	<b>TRUE</b>
<b>PidLidReminderTime</b> (section <a href="#">2.2.1.4</a> )	0x8005	<b>PtypTime</b>	00 52 12 0F 07 70 C8 01	2008/02/15 19:15:00.000
<b>PidLidReminderSignalTime</b> (section <a href="#">2.2.1.2</a> )	0x8006	<b>PtypTime</b>	00 52 12 0F 07 70 C8 01	2008/02/15 19:15:00.000

In response to the user-entered data, the client sends a **RopOpenMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.1) and waits for the server to respond. The server response contains a handle to the Message object.

The client then sends a **RopSetProperties** ROP request ([\[MS-OXCROPS\]](#) section 2.2.8.6) with the following property.

Property	Property ID	Property Type	Data	Value
<b>PidLidReminderSignalTime</b>	0x8006	<b>PtypTime</b>	00 8C 20 DC 0F 70 C8 01	2008/02/15 20:18:00.000

Finally, the client persists the Message object on the server by sending a **RopSaveChangesMessage** ROP request ([MS-OXCROPS] section 2.2.6.3) and releases the Message object by sending a **RopRelease** ROP request ([MS-OXCROPS] section 2.2.15.3).

#### 4.6 Removing a Reminder from an Instance of a Recurring Calendar Object

John has an existing recurring appointment for lunch on Fridays at noon, Pacific Standard Time, but the instance for next week, which occurs on February 22, 2008, was previously changed to 11:00 A.M. (that is, the next instance already has an exception). John opens that instance, sets the signal time to none, and saves the Message object, disabling the reminder for just that instance.

Before John modified the Recurring Calendar object, some of its properties were as follows.

Property Name	Property ID	Property Type	Data	Value
<b>PidTagNormalizedSubject</b> ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	<b>PtypString</b> ([MS-OXCDATA] section 2.11.1)	4C 00 75 00 6E 00 63 00 68 00 20 00 77 00 69 00 74 00 68 00 20 00 42 00 65 00 6E 00 20 00 53 00 6D 00 69 00 74 00 68 00 00 00	"Lunch with Ben Smith"
<b>PidTagStartDate</b> ([MS-OXOCAL] section 2.2.1.30)	0x0060	<b>PtypTime</b> ([MS-OXCDATA] section 2.11.1)	00 A0 65 58 0D 70 C8 01	2008/02/15 20:00:00.000
<b>PidTagEndDate</b> ([MS-OXOCAL] section 2.2.1.31)	0x0061	<b>PtypTime</b>	00 08 2A BA 15 70 C8 01	2008/02/15 21:00:00.000
<b>PidLidReminderSet</b> (section 2.2.1.1)	0x8004	<b>PtypBoolean</b> ([MS-OXCDATA] section 2.11.1)	01	<b>TRUE</b>
<b>PidLidReminderTime</b> (section 2.2.1.4)	0x8005	<b>PtypTime</b>	00 A0 65 58 0D 70 C8 01	2008/02/15 20:00:00.000
<b>PidLidReminderSignalTime</b> (section 2.2.1.2)	0x8006	<b>PtypTime</b>	00 00 44 54 82 75 C8 01	2008/02/22 06:40:00.000
<b>PidLidAppointmentRecur</b> ([MS-OXOCAL] section 2.2.1.44)	0x81AE	<b>PtypBinary</b> ([MS-OXCDATA] section 2.11.1)	Cb: 72 00 Lpb: (Information follows the table)	Cb: 114 bytes Lpb: (See [MS-OXOCAL])

The value of the **PidLidAppointmentRecur** property is as follows:

Cb: 72 00

Lpb:

```
0x0000: 04 30 04 30 0B 20 01 00-00 00 C0 21 00 00 01 00
0x0010: 00 00 00 00 00 00 20 00-00 00 23 20 00 00 0A 00
0x0020: 00 00 00 00 00 00 01 00-00 00 A0 71 C3 0C 01 00
0x0030: 00 00 A0 71 C3 0C 40 4A-C3 0C DF 80 E9 5A 06 30
0x0040: 00 00 09 30 00 00 D0 02-00 00 0C 03 00 00 01 00
0x0050: 34 74 C3 0C 70 74 C3 0C-70 74 C3 0C 00 00 00 00
0x0060: 00 00 04 00 00 00 00 00-00 00 00 00 00 00 00 00
0x0070: 00 00
```

For details about interpreting the data, see [MS-OXOCAL].

Some properties from the Exception Attachment object are listed in the following table.

Property Name	Property ID	Property Type	Data	Value
<b>PidTagAttachSize</b> ([MS-OXCMSG] section 2.2.2.5)	0x0E20	<b>PtypInteger32</b>	DC 0E 00 00	3804 bytes
<b>PidTagLastModificationTime</b> ([MS-OXCMSG] section 2.2.2.2)	0x3008	<b>PtypTime</b>	62 BC D2 86 20 70 C8 01	2008/02/15 22:17:18.328

In response to the user-entered data, the client sends a **RopOpenMessage** ROP request ([MS-OXCROPS] section 2.2.6.1) and waits for the server to respond. The server response contains a handle to the Recurring Calendar object.

Then the client sends a **RopOpenAttachment** ROP request ([MS-OXCROPS] section 2.2.6.12) and waits for the server to respond. The server response contains a handle to the Exception Attachment object. For information about using the attachment table to find the attachment that corresponds to a given exception, see [MS-OXOCAL].

Then the client sends a **RopOpenEmbeddedMessage** ROP request ([MS-OXCROPS] section 2.2.6.16) and waits for the server to respond. The server response contains a handle to the Exception Embedded Message object.

Next, in response to the user-entered data for the Exception Embedded Message object, the client sends a **RopSetProperties** ROP request ([MS-OXCROPS] section 2.2.8.6) with the following properties (followed by a **RopSaveChangesMessage** ROP request ([MS-OXCROPS] section 2.2.6.3) to persist the object (1) on the server).

Property Name	Property ID	Property Type	Data	Value
<b>PidLidAutoStartCheck</b> ([MS-OXPROPS] section 2.41)	0x82E0	<b>PtypBoolean</b>	00	<b>FALSE</b>
<b>PidLidReminderSet</b>	0x8004	<b>PtypBoolean</b>	00	<b>FALSE</b>

Next, in response to the user-entered data for the Exception Attachment object, the client sends a **RopSetProperties** ROP request ([MS-OXCROPS] section 2.2.8.6) with the following properties (followed by a **RopSaveChangesAttachment** ROP request ([MS-OXCROPS] section 2.2.6.15) to persist the object (1) on the server):



Property Name	Property ID	Property Type	Data	Value
<b>PidTagAttachMethod</b> ([MS-OXCMSG] section 2.2.2.9)	0x3705	<b>PtypInteger32</b>	05 00 00 00	5
<b>PidTagRenderingPosition</b> ([MS-OXCMSG] section 2.2.2.16)	0x370B	<b>PtypInteger32</b>	FF FF FF FF	-1
<b>PidTagExceptionStartTime</b> ([MS-OXOCAL] section 2.2.10.1.4)	0x7FFB	<b>PtypTime</b>	00 38 62 11 42 75 c8 01	2008/02/22 11:00:00.000
<b>PidTagAttachEncoding</b> ([MS-OXCMSG] section 2.2.2.20)	0x3702	<b>PtypBinary</b>	00 00	Size: 0 bytes
<b>PidTagExceptionEndTime</b> ([MS-OXOCAL] section 2.2.10.1.5)	0x7FFC	<b>PtypTime</b>	00 a0 26 73 4a 75 c8 01	2008/02/22 12:00:00.000
<b>PidTagAttachmentFlags</b> ([MS-OXCMSG] section 2.2.2.23)	0x7FFD	<b>PtypInteger32</b>	02 00 00 00	Exception to a recurrence
<b>PidTagDisplayName</b> ([MS-OXCFOLD] section 2.2.2.2.4)	0x3001	<b>PtypString</b>	55 00 6e 00 74 00 69 00 74 00 6c 00 65 00 64 00 00 00	"Untitled"
<b>PidTagAttachmentLinkId</b> ([MS-OXCMSG] section 2.2.2.22)	0x7FFA	<b>PtypInteger32</b>	00 00 00 00	0
<b>PidTagAttachFlags</b> ([MS-OXCMSG] section 2.2.2.18)	0x3714	<b>PtypInteger32</b>	00 00 00 00	0
<b>PidTagAttachmentHidden</b> ([MS-OXCMSG] section 2.2.2.24)	0x7FFE	<b>PtypBoolean</b>	01	<b>TRUE</b>
<b>PidTagAttachmentContactPhoto</b> ([MS-OXOCNTC] section 2.2.1.8)	0x7FFF	<b>PtypBoolean</b>	00	<b>FALSE</b>

Next, in response to user-entered data for the Recurring Calendar object, the client sends a **RopSetProperties** ([MS-OXCROPS] section 2.2.8.6) request with the following properties (followed by a **RopSaveChangesMessage** ([MS-OXCROPS] section 2.2.6.3) request to persist the object (1) on the server).

Property Name	Property ID	Property Type	Data	Value
<b>PidLidAppointmentRecur</b>	0x81AE	<b>PtypBinary</b>	Cb: 76 00 Lpb: See below	Cb: 118 bytes Lpb: See below
<b>PidLidFExceptionalAttendees</b> ([MS-OXOCAL] section 2.2.2.3)	0x82D7	<b>PtypBoolean</b>	00	<b>FALSE</b>
<b>PidLidReminderSignalTime</b>	0x8006	<b>PtypTime</b>	00 A8 EC DE 0A 7B	2008/02/29 19:40:00.000

Property Name	Property ID	Property Type	Data	Value
			C8 01	

Then the client sends a **RopRelease** ([MS-OXCROPS] section 2.2.15.3) request to release the Exception Embedded Message object, and a **RopRelease** request to release the Exception Attachment object.

Finally, the client sends a **RopRelease** ([MS-OXCROPS] section 2.2.15.3) request to release the attachment table, and a **RopRelease** request to release the Recurring Calendar object.

The value of the **PidLidAppointmentRecur** property is as follows.

Cb: 76 00

Lpb:

```

0x0000: 04 30 04 30 0B 20 01 00-00 00 C0 21 00 00 01 00
0x0010: 00 00 00 00 00 00 20 00-00 00 23 20 00 00 0A 00
0x0020: 00 00 00 00 00 00 01 00-00 00 A0 71 C3 0C 01 00
0x0030: 00 00 A0 71 C3 0C 40 4A-C3 0C DF 80 E9 5A 06 30
0x0040: 00 00 09 30 00 00 D0 02-00 00 0C 03 00 00 01 00
0x0050: 34 74 C3 0C 70 74 C3 0C-70 74 C3 0C 08 00 00 00
0x0060: 00 00 00 00 00 00 04 00-00 00 00 00 00 00 00 00
0x0070: 00 00 00 00 00 00

```

This corresponds to the following table, in which the **OverrideFlags** and **fReminder** fields for the first exception are the only two fields that changed within the **PtypBinary** property **PidLidAppointmentRecur**. In this specific case, the 2-byte value, starting at the 93rd byte (flags), was changed from 0x0000 to 0x0008, and a 4-byte value representing FALSE (0x00000000) was inserted between the 94th and 95th bytes, extending the size of the property from 114 bytes to 118 bytes. For details about how to determine the byte position in other cases, see [MS-OXOCAL].

Name	Type	Size	Data	Description
ReaderVersion	<b>WORD</b>	2	04 30	This field is set to 0x3004.
WriterVersion	<b>WORD</b>	2	04 30	This field is set to 0x3004.
UIGroup	<b>WORD</b>	2	0b 20	The pattern of the recurrence is weekly (0x200b).
PatternType	<b>WORD</b>	2	01 00	The pattern type is weekly (0x0001).
CalendarType	<b>WORD</b>	2	00 00	The calendar type is Gregorian (0x0000).
FirstDateTime	<b>ULONG</b>	4	c0 21 00 00	For details about how this property was calculated based on the start date, see [MS-OXOCAL].
Period	<b>ULONG</b>	4	01 00 00 00	The recurrence occurs every week (0x00000001).
SlidingFlag	<b>ULONG</b>	4	00 00 00 00	The recurring instances do not rely on completion of the previous instances.

Name	Type	Size	Data	Description
PatternTypeSpecific	<b>Byte Array</b>	Varies (4 bytes, in this case, for Weekly pattern)	20 00 00 00	The recurring appointment occurs on Friday. For details about how the value is determined, see [MS-OXOCAL].
EndType	<b>ULONG</b>	4	23 20 00 00	No End. (0x00002023)
OccurrenceCount	<b>ULONG</b>	4	0A 00 00 00	Ignored, because recurrence does not have an end.
FirstDOW	<b>ULONG</b>	4	00 00 00 00	The first day of the week on the calendar is Sunday (the default value).
DeletedInstanceCount	<b>ULONG</b>	4	01 00 00 00	One deleted instance.
Deleted Instance Start	<b>ULONG</b>	4	A0 71 C3 0C	0x0CC371A0 == February 22, 2008, at 00:00 in UTC.
ModifiedInstanceCount	<b>ULONG</b>	4	01 00 00 00	One modified instance.
Modified Instance Start	<b>ULONG</b>	4	A0 71 C3 0C	0x0CC371A0 == February 22, 2008, at 00:00 in UTC.
Start Date of entire series	<b>ULONG</b>	4	40 4A C3 0C	The start date of the recurrence given in minutes since midnight January 1, 1601, corresponds to February 15, 2008, 12:00:00 A.M. 0x0CC34A40
EndDate of entire series	<b>ULONG</b>	4	DF 80 E9 5A	The end date of the recurrence given in minutes since midnight January 1, 1601; however, because there is no end date, this value corresponds to the "last valid" date: December 31, 4500, 11:59 P.M. (0x5AE980DF)
ReaderVersion2	<b>ULONG</b>	4	06 30 00 00	This value is set to 0x00003006.
WriterVersion2	<b>ULONG</b>	4	09 30 00 00	This value is set to 0x00003009. <a href="#">&lt;35&gt;</a>
StartTimeOffset	<b>ULONG</b>	4	D0 02 00 00	The hexadecimal start time of the recurrence is 0x000002D0, which corresponds to 720 in decimal. 720 minutes is 12 hours, which is 12:00 P.M.
EndTimeOffset	<b>ULONG</b>	4	0C 03 00 00	The hexadecimal end time of the recurrence is 0x00000030C, which corresponds to 780 minutes, which is 1:00 P.M.
ExceptionCount	<b>WORD</b>	2	01 00	There is one exception in this recurrence BLOB.

Name	Type	Size	Data	Description
ExceptionInfo block for exception 1:				
Start Date Time	<b>ULONG</b>	4	34 74 C3 0C	0x0CC37434 == February 22, 2008, at 11:00 A.M.
End Date Time	<b>ULONG</b>	4	70 74 C3 0C	0x0CC37470 == February 22, 2008, at 12:00 P.M.
Original Start Time	<b>ULONG</b>	4	70 74 C3 0C	0x0CC37470 == February 22, 2008, at 12:00 P.M.
OverrideFlags	<b>WORD</b>	2	08 00	0x0008 corresponds to the <b>fReminderSet</b> flag, meaning just that field is specified for this exception.
fReminderSet	<b>ULONG</b>	4	00 00 00 00	0x00000000 == FALSE, meaning ReminderSet is overridden with the value FALSE for this exception.
ReservedBlock1Size	<b>ULONG</b>	4	00 00 00 00	There is no data in the reserved block.
ExtendedException block for exception 1:				
ChangeHighlight	<b>Byte Array</b>	Varies	04 00 00 00 00 00 00 00	The size of the ChangeHighlight is 4. The value of the <b>PidLidChangeHighlight</b> property ( <a href="#">[MS-OXOCAL]</a> section 2.2.6.2) is zero for this exception.
ReservedBlockEE1Size	<b>ULONG</b>	4	00 00 00 00	There is no data in the reserved block.
ReservedBlock2Size	<b>ULONG</b>	4	00 00 00 00	There is no data in the reserved block.

## 5 Security

### 5.1 Security Considerations for Implementers

There are no special security considerations specific to the Reminder Settings Protocol. General security considerations pertaining to the underlying transport apply, as specified in [\[MS-OXCMMSG\]](#) and [\[MS-OXCPRPT\]](#).

### 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 15 Technical Preview
- Microsoft® Office Outlook® 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Outlook® 2010
- Microsoft® Outlook® 15 Technical Preview

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

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

[<1> Section 2.2.1.5:](#) In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidReminderOverride** property.

[<2> Section 2.2.1.6:](#) In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidReminderPlaySound** property (section [2.2.1.6](#)).

[<3> Section 2.2.1.7:](#) Office Outlook 2003, Office Outlook 2007, and Outlook 2010 support whatever the Windows API WINMM!PlaySound is able to play.

[<4> Section 2.2.1.7:](#) In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidReminderFileParameter** property (section [2.2.1.7](#)).

[<5> Section 2.2.1.9:](#) Office Outlook 2003, Office Outlook 2007, and Outlook 2010 preserve the value if already set.

[<6> Section 2.2.2.3.1:](#) Office Outlook 2003 does not set the **PidLidTaskDueDate** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.5) for Message objects other than Task objects, so when a reminder signals for a Message object that can be flagged, the "due in" time is calculated based on the value of the **PidLidReminderTime** property (section [2.2.1.4](#)). Office Outlook 2007 SP1 does set the **PidLidTaskDueDate** property for Message objects that can be flagged, and so has a more intuitive "due in" time when the reminder signals. Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not set the **PidLidTaskDueDate** property for tasks when the user only specifies a reminder on a Task object without also specifying a due date. In other words, this property can be set on a Task object in the context of setting a due date. The task due date is independent of the signal time. For more information about the conditions under which the **PidLidTaskDueDate** property is set on a Task object, see [\[MS-OXOTASK\]](#).

<7> [Section 2.2.2.4](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidAutoStartCheck** property ([\[MS-OXPROPS\]](#) section 2.41).

<8> [Section 2.2.2.4](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidConferencingCheck** property ([\[MS-OXOCAL\]](#) section 2.2.1.50.2).

<9> [Section 3.2.3.1.1](#): Office Outlook 2003 follows the minimal reminder domain scope strategy. Exchange 2003 signals reminders only from the **Calendar folder** and **Tasks folder**.

<10> [Section 3.2.3.1.1.1](#): Setting the **piReminderUpgradeTime** property is not supported in Office Outlook 2003. Under normal operation, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not change the **piReminderUpgradeTime** property after the value is initialized. However, when the "cleanreminders" feature is invoked via command-line parameter, Office Outlook 2007 and Outlook 2010 set the **piReminderUpgradeTime** property (section [2.2.3.1](#)) to 0 to clean the reminders search folder (1), after which all reminders would be eligible to signal again.

<11> [Section 3.2.4.1.2](#): Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 user interfaces set the **PidTagReplyTime** property ([\[MS-OXOFLAG\]](#) section 2.2.3.1) when setting a reminder on an already-flagged Message object. Setting the **PidTagReplyTime** property is not required for reminders to signal locally, but this property is used to transmit a reminder on a sent message. The Exchange 2003 user interface does not expose a way to set reminders on Message objects that are neither Calendar objects nor Task objects, so this constraint does not apply.

<12> [Section 3.2.4.1.2](#): Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not set all the specified properties when the reminder is created, because some of the properties were set in a previous operation (such as Message object creation or flagging for follow-up), or are set independent of the reminder itself (such as for Task objects), and the values did not change during the reminder creation process. Because Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not expose UI to just set a reminder on an e-mail message without also flagging, some properties, such as the **PidLidFlagRequest** property ([\[MS-OXOFLAG\]](#) section 2.2.1.9), happen to get set by Office Outlook 2003, Office Outlook 2007, and Outlook 2010 when a reminder is added to an e-mail message. These properties are not required for the reminder to function but are listed for completeness.

<13> [Section 3.2.4.1.3](#): The user interfaces of Exchange 2007 and Exchange 2010 set the **PidLidReminderTime** property (section [2.2.1.4](#)) to the start time for the next instance with an active reminder that has an end date in the future. This is true regardless of whether the Recurring Calendar object is created where part of the instances occur in the past, or when dismissing a reminder. When a reminder is dismissed (or a recurrence is created where part of the recurrence is in the past), the **PidLidReminderTime** property is updated to match the start time of the next instance with an active reminder. Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the Exchange 2003 user interface set the **PidLidReminderTime** property to the start time of the first instance and do not modify the **PidLidReminderTime** property during snooze or dismiss operations after the property is set on a Recurring Calendar object, even if the value has been modified. To clarify, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the Exchange 2003 user interface only set the **PidLidReminderTime** property when creating the Recurring Calendar object or modifying the series such that the series start date changes and otherwise do not modify the property. In other words, the **PidLidReminderTime** property cannot be assumed to have a specific value for **Recurring objects**.

<14> [Section 3.2.4.1.3](#): The Exchange 2007 and Exchange 2010 user interfaces set the **PidLidReminderTime** property (section [2.2.1.4](#)) to the start time for the next instance with an active reminder that has an end date in the future. This is true regardless of whether the Recurring Calendar object is created where part of the instances occur in the past, or when dismissing a reminder. When a reminder is dismissed (or a recurrence is created where part of the recurrence is

in the past), the **PidLidReminderTime** property is updated to match the start time of the next instance with an active reminder. Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the Exchange 2003 user interface set the **PidLidReminderTime** property to the start time of the first instance, and do not modify the **PidLidReminderTime** property during snooze or dismiss operations after the property is set on a Recurring Calendar object, even if the value has been modified. To clarify, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the Exchange 2003 user interface only set the **PidLidReminderTime** property when creating the Recurring Calendar object or modifying the series such that the series start date changes, and otherwise do not modify the property. In other words, the **PidLidReminderTime** property cannot be assumed to have a specific value for Recurring Calendar objects.

<15> [Section 3.2.4.4.3](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interface does not perform this processing.

<16> [Section 3.2.4.4.1](#): In addition to the properties specified, Office Outlook 2003 deletes the **PidTagReplyTime** property ([\[MS-OXOFLAG\]](#) section 2.2.3.1) for a Message object that is neither a Calendar object nor a Task object. Office Outlook 2007 deletes the **PidLidReminderTime** (section 2.2.1.4), **PidLidReminderSignalTime** (section 2.2.1.2), and **PidTagReplyTime** properties when removing the reminder from a Message object that is neither a Calendar object nor a Task object. For the remaining cases, Office Outlook 2003 and Office Outlook 2007 do not delete the **PidLidReminderTime** or **PidLidReminderSignalTime** properties when removing the reminder for Calendar objects or Task objects; this is for user convenience to persist the signal time values even when the reminder is disabled.

<17> [Section 3.2.4.4.1](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidAutoStartCheck** property ([\[MS-OXPROPS\]](#) section 2.41).

<18> [Section 3.2.4.4.2](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidAutoStartCheck** property ([\[MS-OXPROPS\]](#) section 2.41).

<19> [Section 3.2.4.4.3](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidTaskResetReminder** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.6).

<20> [Section 3.2.4.5.1](#): The Exchange 2007 and Exchange 2010 user interface use stricter criteria when ignoring meeting-related Message objects by ignoring messages with the **PidTagMessageClass** property ([\[MS-OXCMMSG\]](#) section 2.2.1.3) prefixed with "IPM.Schedule.meeting". Office Outlook 2007 and Outlook 2010 use slightly more lenient criteria: "IPM.Schedule" (note the missing trailing dot).

<21> [Section 3.2.4.5.1](#): Office Outlook 2003 does not ignore Message objects that have the **PidTagSwappedToDoStore** property ([\[MS-OXOFLAG\]](#) section 2.2.1.8) set. Office Outlook 2003 also does not ignore meeting-related Message objects, but this is not a problem because it uses the minimal reminder domain, which does not include **special folders** like the **Sent Items folder**.

<22> [Section 3.2.4.5.2](#): Exchange 2003, Exchange 2007, and Exchange 2010 never read the **piReminderUpgradeTime** property (section 2.2.3.1). Exchange 2003 does not automatically dismiss reminders that appear outside the minimal reminder domain. Exchange 2007 and Exchange 2010 do automatically dismiss said reminders but only on initial transition to using the full reminder domain on a given mailbox store.

<23> [Section 3.2.4.5.3](#): Office Outlook 2003, Office Outlook 2007, and Outlook 2010 use the **PidTagNormalizedSubject** ([\[MS-OXCMMSG\]](#) section 2.2.1.10), **PidLidFlagRequest** ([\[MS-OXOFLAG\]](#) section 2.2.1.9), and **PidLidLocation** ([\[MS-OXOCAL\]](#) section 2.2.1.4) properties. Office Outlook 2007 and Outlook 2010 use the **PidLidTaskDueDate** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.5).



<24> [Section 3.2.4.5.3](#): Office Outlook 2003 displays DueIn "Now" when first signaling a reminder for a Task object or flagged Message object; Office Outlook 2007 and Outlook 2010 display a more intuitive time-until-DueDate value for these Message object types. Office Outlook 2007 and Outlook 2010 use the **PidLidTaskDueDate** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.5) and the "end of workday" value to determine the due time. Office Outlook 2003 and the Exchange 2003 user interface use the value of the **PidLidReminderTime** property (section [2.2.1.4](#)) as the due time.

<25> [Section 3.2.4.5.3](#): Office Outlook 2003 uses the signal time as the due time.

<26> [Section 3.2.4.5.3](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not use the **PidLidReminderOverride** (section [2.2.1.5](#)), **PidLidReminderPlaySound** (section [2.2.1.6](#)), or **PidLidReminderFileParameter** (section [2.2.1.7](#)) properties.

<27> [Section 3.2.4.5.3](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not use the **PidLidReminderOverride** (section [2.2.1.5](#)), **PidLidReminderPlaySound** (section [2.2.1.6](#)), or **PidLidReminderFileParameter** (section [2.2.1.7](#)) properties.

<28> [Section 3.2.4.5.3](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not use the **PidLidReminderOverride** (section [2.2.1.5](#)), **PidLidReminderPlaySound** (section [2.2.1.6](#)), or **PidLidReminderFileParameter** (section [2.2.1.7](#)) properties.

<29> [Section 3.2.4.5.3](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not use the **PidLidReminderOverride** (section [2.2.1.5](#)), **PidLidReminderPlaySound** (section [2.2.1.6](#)), or **PidLidReminderFileParameter** (section [2.2.1.7](#)) properties.

<30> [Section 3.2.4.6.3](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not set the **PidLidTaskResetReminder** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.6) when dismissing a reminder on a Task object. This means that Exchange 2003, Exchange 2007, and Exchange 2010 do not support dismissing reminders on single instances of Recurring Task objects. Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do set the **PidLidTaskResetReminder** property.

<31> [Section 3.2.4.6.3](#): The Exchange 2003 user interface sets the **PidLidReminderSignalTime** property (section [2.2.1.2](#)) when dismissing Task objects. Exchange 2007 does not set the **PidLidReminderSignalTime** property when dismissing Task objects.

<32> [Section 3.2.4.8](#): The Exchange 2003 user interface sets these properties if the value of the **PidLidReminderSet** property (section [2.2.1.1](#)) is **TRUE**.

<33> [Section 3.2.4.8](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidTaskResetReminder** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.6).

<34> [Section 3.2.4.8](#): In Exchange 2003, Exchange 2007, and Exchange 2010, the user interfaces do not honor or set the **PidLidTaskResetReminder** property ([\[MS-OXOTASK\]](#) section 2.2.2.2.6).

<35> [Section 4.6](#): Office Outlook 2003 uses a different format for this property stream and sets the WriterVersion2 property to 0x00003008.

## 7 Change Tracking

This section identifies changes that were made to the [MS-OXORMDR] protocol document between the October 2011 and January 2012 releases. 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.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 language and 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 or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- 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 [protocol@microsoft.com](mailto:protocol@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">3.2 Client Details</a>	Changed section title from "Client and Server Details" to "Client Details".	N	Content updated for template compliance.
<a href="#">3.3 Server Details</a>	Added section 3.2 for and subsections 3.2.1 through 3.2.7 server details.	N	Content updated for template compliance.
<a href="#">6 Appendix A: Product Behavior</a>	Added Exchange 15 Technical Preview and Outlook 15 Technical Preview to the list of applicable product versions.	Y	Content updated.

## 8 Index

### A

Abstract data model  
[client](#) 14  
[server](#) 23  
[Applicability](#) 9

### C

[Capability negotiation](#) 9  
[Change tracking](#) 42  
Client  
    [abstract data model](#) 14  
    [message processing](#) 22  
    [other local events](#) 23  
    [overview](#) 14  
    [sequencing rules](#) 22  
    [timer events](#) 22  
    [timers](#) 14  
Client - higher layer triggered events  
    [generating instances for recurring Task objects](#) 22  
    [post-receive processing](#) 18  
    [post-transmit processing](#) 18  
    [setting a reminder](#) 15  
Creating\  
    modifying\  
        [and removing reminders overview](#) 8

### D

Data model - abstract  
    [client](#) 14  
    [server](#) 23  
[Dismissing a reminder on a recurring Calendar object](#) 28  
[Dismissing a reminder on a task](#) 27  
[Dismissing and snoozing reminders overview](#) 9

### E

Examples  
    [dismissing a reminder on a recurring Calendar object](#) 28  
    [dismissing a reminder on a task](#) 27  
    [overview](#) 24  
    [removing a reminder from an instance of a recurring Calendar object](#) 31  
    [setting a reminder on a Message object](#) 26  
    [setting a reminder on a single instance appointment](#) 25  
    [snoozing a reminder on a Contact object](#) 30

### F

[Fields - vendor-extensible](#) 9

### G

[Glossary](#) 6

### H

Higher layer triggered events - client  
    [generating instances for recurring Task objects](#) 22  
    [post-receive processing](#) 18  
    [post-transmit processing](#) 18  
    [setting a reminder](#) 15  
Higher-layer triggered events  
    [server](#) 23

### I

[Implementer - security considerations](#) 37  
[Index of security parameters](#) 37  
[Informative references](#) 8  
Initialization  
    [server](#) 23  
[Introduction](#) 6

### M

Message processing  
    [client](#) 22  
    [server](#) 23  
[Message syntax](#) 10  
Messages  
    [Properties Used to Specify and/or Decode Reminders](#) 10  
    [Properties Used to Specify User Preferences that are Roamed on the Server](#) 13  
    [Shared Properties](#) 12  
    [syntax](#) 10  
    [transport](#) 10

### N

[Normative references](#) 7

### O

Other local events  
    [client](#) 23  
    [server](#) 23  
Overview  
    creating\  
        modifying\  
            [and removing reminders](#) 8  
            [dismissing and snoozing reminders](#) 9  
            [processing overdue reminders](#) 9  
    [Overview \(synopsis\)](#) 8

### P

[Parameters - security index](#) 37

[PidLidReminderDelta property used to specify and/or decode reminders](#) 11  
[PidLidReminderFileParameter property used to specify and/or decode reminders](#) 11  
[PidLidReminderOverride property used to specify and/or decode reminders](#) 11  
[PidLidReminderPlaySound property used to specify and/or decode reminders](#) 11  
[PidLidReminderSet property used to specify and/or decode reminders](#) 10  
[PidLidReminderSignalTime property used to specify and/or decode reminders](#) 10  
[PidLidReminderTime property used to specify and/or decode reminders](#) 11  
[PidLidReminderType property used to specify and/or decode reminders](#) 12  
[PidTagReplyTime property used to specify and/or decode reminders](#) 11  
[piReminderUpgradeTime property used to specify user preferences that are roamed on the server](#) 13  
[Preconditions](#) 9  
[Prerequisites](#) 9  
[Processing overdue reminders overview](#) 9  
[Product behavior](#) 38  
[Properties shared with the Appointment and Meeting Object Protocol](#) 13  
[Properties shared with the Informational Flagging Protocol](#) 12  
[Properties shared with the Message and Attachment Object Protocol](#) 12  
[Properties shared with the Task-Related Objects Protocol](#) 12  
 Properties used to specify and/or decode reminders  
[PidLidReminderDelta property](#) 11  
[PidLidReminderFileParameter property](#) 11  
[PidLidReminderOverride property](#) 11  
[PidLidReminderPlaySound property](#) 10  
[PidLidReminderSignalTime property](#) 10  
[PidLidReminderTime property](#) 11  
[PidLidReminderType property](#) 12  
[PidTagReplyTime property](#) 11  
[Properties Used to Specify and/or Decode Reminders message](#) 10  
 Properties used to specify user preferences that are roamed on the server  
[piReminderUpgradeTime property](#) 13  
[Properties Used to Specify User Preferences that are Roamed on the Server message](#) 13  
 Protocol examples  
[overview](#) 24

## R

References  
[informative](#) 8  
[normative](#) 7  
[Relationship to other protocols](#) 9  
[Removing a reminder from an instance of a recurring Calendar object](#) 31

## S

Security  
[implementer considerations](#) 37  
[parameter index](#) 37  
 Sequencing rules  
[client](#) 22  
[server](#) 23  
 Server  
[abstract data model](#) 23  
[higher-layer triggered events](#) 23  
[initialization](#) 23  
[message processing](#) 23  
[other local events](#) 23  
 overview ([section 3.1](#) 14, [section 3.3](#) 23)  
[sequencing rules](#) 23  
[timer events](#) 23  
[timers](#) 23  
[Setting a reminder on a Message object](#) 26  
[Setting a reminder on a single instance appointment example](#) 25  
 Shared properties  
[properties shared with the Appointment and Meeting Object Protocol](#) 13  
[properties shared with the Informational Flagging Protocol](#) 12  
[properties shared with the Message and Attachment Object Protocol](#) 12  
[properties shared with the Task-Related Objects Protocol](#) 12  
[Shared Properties message](#) 12  
[Snoozing a reminder on a Contact object](#) 30  
[Standards assignments](#) 9

## T

Timer events  
[client](#) 22  
[server](#) 23  
 Timers  
[client](#) 14  
[server](#) 23  
[Tracking changes](#) 42  
[Transport](#) 10  
 Triggered events - client  
[generating instances for recurring Task objects](#) 22  
[post-receive processing](#) 18  
[post-transmit processing](#) 18  
[setting a reminder](#) 15  
 Triggered events - higher-layer  
[server](#) 23

## V

[Vendor-extensible fields](#) 9  
[Versioning](#) 9