[MS-OXORMDR]:
Reminder Settings Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Copyright © 2013 Microsoft Corporation.

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.
04/27/2012	6.0	Major	Significantly changed the technical content.
07/16/2012	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	6.1	Minor	Clarified the meaning of the technical content.
02/11/2013	6.1	No change	No changes to the meaning, language, or formatting of the technical content.

Copyright © 2013 Microsoft Corporation.

Table of Contents

1 Introduction 5
1.1 Glossary 5
1.2 References 6
1.2.1 Normative References 6
1.2.2 Informative References 7
1.3 Overview 7
1.4 Relationship to Other Protocols 8
1.5 Prerequisites/Preconditions 8
1.6 Applicability Statement 8
1.7 Versioning and Capability Negotiation 8
1.8 Vendor-Extensible Fields 8
1.9 Standards Assignments 8

2 Messages 9
2.1 Transport 9
2.2 Message Syntax 9
2.2.1 Properties Used to Specify and/or Decode Reminders 9
2.2.1.1 PidLidReminderSet Property 9
2.2.1.2 PidLidReminderSignalTime Property 9
2.2.1.3 PidLidReminderDelta Property 9
2.2.1.4 PidLidReminderTime Property 10
2.2.1.5 PidLidReminderOverride Property 10
2.2.1.6 PidLidReminderPlaySound Property 10
2.2.1.7 PidLidReminderFileParameter Property 10
2.2.1.8 PidTagReplyTime Property 10
2.2.1.9 PidLidReminderType Property 10
2.2.2 Shared Properties 11
2.2.2.1 Properties Shared with the Message and Attachment Object Protocol 11
2.2.2.2 Properties Shared with the Informational Flagging Protocol 11
2.2.2.3 Properties Shared with the Task-Related Objects Protocol 11
2.2.2.3.1 PidLidTaskDueDate Property 11
2.2.2.4 Properties Shared with the Appointment and Meeting Object Protocol 11
2.2.3 Properties Used to Specify User Preferences That Are Roamed on the Server 12
2.2.3.1 piReminderUpgradeTime Property 12

3 Protocol Details 13
3.1 Client Details 13
3.1.1 Abstract Data Model 13
3.1.2 Timers 13
3.1.3 Initialization 13
3.1.3.1 Scope of the Search for Active Reminders 13
3.1.3.2 Loading and Processing Reminders 14
3.1.4 Higher-Layer Triggered Events 14
3.1.4.1 Setting a Reminder 14
3.1.4.1.1 Reminders on Task Objects 14
3.1.4.1.2 Reminders on Message Objects Other Than Calendar Objects and Task Objects 14
3.1.4.1.3 Reminders on Calendar Objects 14
3.1.4.1.4 Reminders on Recurring Calendar Objects 15
3.1.4.1.5 Setting Reminders on Draft Message Objects 15

Copyright © 2013 Microsoft Corporation.

3.1.4.1.5.1 Setting a Reminder for all Recipients 16
3.1.4.1.5.2 Setting a Reminder for the Sender 16
3.1.4.2 Post-Transmit Processing 16
3.1.4.3 Post-Receive Processing 16
3.1.4.4 Removing a Reminder 16
3.1.4.4.1 Single Instance Objects 16
3.1.4.4.2 Recurring Calendar Objects 16
3.1.4.4.3 Recurring Task Objects 17
3.1.4.5 When a Reminder Becomes Overdue 17
3.1.4.5.1 Reminders That Are Ignored When They Become Overdue 17
3.1.4.5.2 Reminders That Are Automatically Dismissed When Overdue 17
3.1.4.5.3 Actions for Overdue Reminders That Are Not Ignored or Automatically Dismissed 17
3.1.4.5.3.1 Starting a Conference 18
3.1.4.6 Dismissing a Reminder 19
3.1.4.6.1 Dismissing Single Instance Objects 19
3.1.4.6.2 Dismissing Recurring Calendar Objects 19
3.1.4.6.3 Dismissing for Recurring Task Objects 19
3.1.4.7 Snoozing a Reminder 19
3.1.4.7.1 Snoozing for Single Instance Objects 19
3.1.4.7.2 Snoozing for Recurring Calendar Objects and Recurring Task Objects 20
3.1.4.8 Generating Instances for Recurring Task Objects 20
3.1.5 Message Processing Events and Sequencing Rules 20
3.1.6 Timer Events 20
3.1.7 Other Local Events 20
3.2 Server Details 20
3.2.1 Abstract Data Model 21
3.2.2 Timers 21
3.2.3 Initialization 21
3.2.4 Higher-Layer Triggered Events 21
3.2.5 Message Processing Events and Sequencing Rules 21
3.2.6 Timer Events 21
3.2.7 Other Local Events 21

4 Protocol Examples 22
4.1 Setting a Reminder on a Single Instance Appointment 23
4.2 Setting a Reminder on a Message Object 24
4.3 Dismissing a Reminder on a Task 25
4.4 Dismissing a Reminder on a Recurring Calendar Object 26
4.5 Snoozing a Reminder on a Contact Object 28
4.6 Removing a Reminder from an Instance of a Recurring Calendar Object 29

5 Security 35
5.1 Security Considerations for Implementers 35
5.2 Index of Security Parameters 35

6 Appendix A: Product Behavior 36

7 Change Tracking 38

8 Index 39

Copyright © 2013 Microsoft Corporation.

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 can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. 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)
flags
handle
object

The following terms are defined in [MS-OXGLOS]:

active reminder
Calendar object
calendar options dictionary
contact
Contact object
Draft Message object
Exception Attachment object
Exception Embedded Message object
Folder 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

The following terms are specific to this document:

Copyright © 2013 Microsoft Corporation.

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 Email, 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 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 Specifications documentation do not include a publishing year because links are to the latest version of the technical 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. 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".

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

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

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

Copyright © 2013 Microsoft Corporation.

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

[MS-OXCTABL] Microsoft Corporation, "Table Object Protocol".

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

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

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

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

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol".

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

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

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

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

1.2.2 Informative References

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

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

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

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

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

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.

The client creates, modifies, or removes reminders by setting **reminder properties** on Message objects, as specified in section 3.1.4. 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. 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.

The client processes **overdue reminders** as specified in section 3.1.4.5. Before signaling an overdue reminder, the client is responsible for determining which reminders to ignore and which to automatically **dismiss**.

The client can dismiss or **snooze** overdue reminders as specified in section 3.1.4.6 and section 3.1.4.7.

Copyright © 2013 Microsoft Corporation.

1.4 Relationship to Other Protocols

The Reminder Settings Protocol extends the Message and Attachment Object Protocol, as specified in [MS-OXCMSG], and therefore has the same dependencies.

This protocol uses data structures in Message objects, as described in [MS-OXCDATA].

This protocol creates, sends, and stores e-mail messages by representing them as Message objects, as described in [MS-OXOMSG]. It flags Message objects as described in [MS-OXOFLAG].

This protocol works with **Folder objects**, as described in [MS-OXCFOLD]; creates and reads **special folders**, as described in [MS-OXOSFLD]; and uses **search folders (2)**, as described in [MS-OXOSRCH].

This protocol reads tabular data, as described in [MS-OXCTABL].

This protocol shares configuration data between clients and servers, as described in [MS-OXOCFG].

This protocol uses **Calendar objects** and **meeting-related objects**, as described [MS-OXOCAL].

This protocol uses **Task objects**, as described in [MS-OXOTASK].

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

1.5 Prerequisites/Preconditions

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-OXCMSG].

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-OXCMSG].

1.9 Standards Assignments

None.

Copyright © 2013 Microsoft Corporation.

2 Messages

2.1 Transport

This protocol uses the transport specified in [MS-OXCMSG].

2.2 Message Syntax

Clients can create and modify reminders on any type of Message object by using the Message and Attachment Object Protocol, as specified in [MS-OXCMSG].

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-OXOTASK] section 2.2.2.2.6.

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

Copyright © 2013 Microsoft Corporation.

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] section 3.1.4.1.

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 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 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<1> that a client SHOULD 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<2> be set, and its value MUST be ignored.

Copyright © 2013 Microsoft Corporation.

2.2.2 Shared Properties

Shared properties are set when the user configures the reminder.

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-OXOFLAG] section 2.2.1.7)
- **PidTagSwappedToDoStore** ([MS-OXOFLAG] section 2.2.1.8)

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

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<3> 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. The "due in" time is the time remaining until a Message object is due.

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)

Copyright © 2013 Microsoft Corporation.

- **PidLidConferencingCheck** ([MS-OXOCAL] section 2.2.1.50.2)

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].

Copyright © 2013 Microsoft Corporation.

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

Messages with **active reminders** can be discovered by using a persistent search folder (2) as specified in [MS-OXCFOLD]. 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.

3.1.2 Timers

None.

3.1.3 Initialization

3.1.3.1 Scope of the Search for Active Reminders

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<4> 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.

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<5> 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.

Copyright © 2013 Microsoft Corporation.

3.1.3.2 Loading and Processing Reminders

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 3.1.4.5.1.
- Automatically dismiss certain Message objects, as specified in section 3.1.4.5.2.
- Include any remaining Message objects that have active reminders (pending or overdue reminders).

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Setting a Reminder

To enable the signaling of a reminder, in addition to the behaviors specified in section 3.1.4.1.1 through section 3.1.4.1.5.2, the client sets the **PidLidReminderSet** property (section 2.2.1.1) to TRUE on a Message object. 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).

3.1.4.1.1 Reminders on Task Objects

To enable the signaling of a reminder on a Task object, the client sets the **PidLidReminderTime** property (section 2.2.1.4) to the specified time at which the reminder becomes overdue.

3.1.4.1.2 Reminders on Message Objects Other Than Calendar Objects and Task Objects

To enable the signaling of reminders on Message objects that are not Calendar objects or Task objects, the client sets the **PidLidReminderTime** property (section 2.2.1.4) and SHOULD<6> set the **PidTagReplyTime** ([MS-OXOFLAG] section 2.2.3.1) property to the specified signal time.

In summary, the client sets the properties in the following table (or ensures that they are already set<7> for Message objects that are neither Calendar objects nor Task objects).

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

3.1.4.1.3 Reminders on Calendar Objects

To enable the signaling of a reminder on Calendar objects, the client sets the **PidLidReminderTime** property (section 2.2.1.4), and its value SHOULD<8> correspond to the appointment start time.

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.

Copyright © 2013 Microsoft Corporation.

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

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

3.1.4.1.4 Reminders on Recurring Calendar Objects

To enable the signaling of a reminder on 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** property ([MS-OXCMSG] 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 the **StartTime** field in the recurrence BLOB, the **PidLidReminderDelta** property (section 2.2.1.3) or **PidLidReminderSet** property (section 2.2.1.1). For example recurrence BLOBs, see [MS-OXOCAL] section 4.1.1.

Setting the **PidLidReminderSet** property 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, the client sets the **PidLidReminderSet** property to FALSE on the Recurring Calendar object.

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

3.1.4.1.5 Setting Reminders on Draft Message Objects

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) 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).

If either a sender or a 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] section 3.1.4.3.

Copyright © 2013 Microsoft Corporation.

3.1.4.1.5.1 Setting a Reminder for all Recipients

To set a reminder for all recipients (1) 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.

3.1.4.1.5.2 Setting a Reminder for the Sender

To enable the signaling of a reminder for the sender, the client MUST set the **rtmReminder** field to the required signal time and the **fReminderSet** field to TRUE in the **PidTagSwappedToDoData** property ([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.

3.1.4.2 Post-Transmit Processing

The post-transmit processing required by the Reminder Settings Protocol is identical to the post-transmit processing specified in [MS-OXOFLAG] section 3.1.4.3.

3.1.4.3 Post-Receive Processing

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

3.1.4.4 Removing a Reminder

3.1.4.4.1 Single Instance Objects

To remove a reminder for a single instance, 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<9> 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 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.

3.1.4.4.2 Recurring Calendar Objects

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

To remove the reminder for a single instance on a Recurring Calendar object, 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 set the **PidLidAutoStartCheck** property ([MS-

Copyright © 2013 Microsoft Corporation.

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), see [MS-OXOCAL]. The Exception Embedded Message object is stored in the **PidTagAttachDataObject** property ([MS-OXCMSG] section 2.2.2.8).

3.1.4.4.3 Recurring Task Objects

To remove the reminder for all instances on a Recurring Task object, the client sets the **PidLidReminderSet** property (section 2.2.1.1) to FALSE and SHOULD 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**.

3.1.4.5 When a Reminder Becomes Overdue

3.1.4.5.1 Reminders That Are Ignored When They Become Overdue

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 value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) contains the prefix "IPM.Schedule.".
- Unsent mail - any Message object that contains the **PidTagMessageFlags** property ([MS-OXCMSG] section 2.2.1.6) with the flag set corresponding to **MSGFLAG_UNSENT**.
- Message objects that have the **PidTagSwappedToDoStore** property ([MS-OXOFLAG] section 2.2.1.8) set.<10>

3.1.4.5.2 Reminders That Are Automatically Dismissed When Overdue

Clients following the full reminder domain scope strategy, as specified in section 3.1.3.1, SHOULD automatically dismiss any Message objects that are found outside the minimal reminder domain and for which the value of the **PidLidReminderSignalTime** property (section 2.2.1.2) is earlier than the value of the **piReminderUpgradeTime** property (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.

3.1.4.5.3 Actions for Overdue Reminders That Are Not Ignored or Automatically Dismissed

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<11> 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.

Copyright © 2013 Microsoft Corporation.

Case	Due time
For Message objects other than Calendar objects<12>	If the **PidLidTaskDueDate** property ([MS-OXOTASK] section 2.2.2.2.5) exists, the client SHOULD<13> use "TaskDueDate + default End-of-Day time"; otherwise use the value of the **PidLidReminderTime** property (section 2.2.1.4) as the due time. For details about End-of-Day time, see the **End** element in [MS-OXOCFG] section 2.2.5.2.1.
For single instance Calendar objects	The client uses the value of the **PidLidReminderTime** 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 (**StartTime** – **ReminderDelta**) is less than or equal to the time encoded in the value of the **PidLidReminderSignalTime** property (section 2.2.1.2) on the Recurring Calendar object. **Note** **StartTime** and **ReminderDelta** 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 play the default reminder sound.

If the **PidLidReminderOverride** property is set to TRUE, the client SHOULD 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 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 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.

3.1.4.5.3.1 Starting a Conference

The client can launch a conference if the **PidLidConferencingCheck** property ([MS-OXOCAL] section 2.2.1.50.2) and the **PidLidAutoStartCheck**<14> 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 following properties:

- **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)
- **PidLidAppointmentStateFlags** ([MS-OXOCAL] section 2.2.1.10)

Copyright © 2013 Microsoft Corporation.

3.1.4.6 Dismissing a Reminder

3.1.4.6.1 Dismissing Single Instance Objects

To dismiss a reminder 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 3.1.4.6.2 for dismissing a Recurring Task object.

3.1.4.6.2 Dismissing Recurring Calendar Objects

To dismiss a reminder for a Recurring Calendar object if there is a future instance (including exceptions) with a pending reminder (in other words, the reminder is not disabled individually on all future instances), the client MUST set the value of the **PidLidReminderSignalTime** property (section 2.2.1.2) to the start time of that instance minus the value of the **PidLidReminderDelta** property (section 2.2.1.3).

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 Low:0xA3DD4000 High:0x0CB34557 (4501/01/01 00:00:00.000).

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.

Dismissing a reminder for a Recurring Calendar object never causes an instance to become an exception.

3.1.4.6.3 Dismissing for Recurring Task Objects

To dismiss a reminder for a Recurring Task object, the client MUST set the **PidLidReminderSet** property (section 2.2.1.1) to FALSE and SHOULD<15> set the **PidLidTaskResetReminder** property ([MS-OXOTASK] section 2.2.2.2.6) to TRUE.

For Task objects, if the value of the **PidLidReminderTime** property (section 2.2.1.4) is in the future, the client SHOULD 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.

Dismissing a reminder for a Recurring Task object never causes an instance to become an exception.

3.1.4.7 Snoozing a Reminder

3.1.4.7.1 Snoozing for Single Instance Objects

To snooze a reminder for a single instance object, the client MUST set the **PidLidReminderSignalTime** property (section 2.2.1.2) to the time specified by the user (or user agent).

Copyright © 2013 Microsoft Corporation.

3.1.4.7.2 Snoozing for Recurring Calendar Objects and Recurring Task Objects

To snooze a reminder for a Recurring Calendar object or Recurring Task object, 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, the start time of the next instance minus the value of the **PidLidReminderDelta** property (section 2.2.1.3)), whichever is earlier.

Snoozing a reminder for a Recurring Calendar object or Recurring Task object never causes an instance to become an exception.

3.1.4.8 Generating Instances for Recurring Task 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 **PidLidTaskResetReminder** property is TRUE or the value of the **PidLidReminderSet** property (section 2.2.1.1) is TRUE, a reminder SHOULD 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 (DST) 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 be TRUE. The now-completed instance MUST have the **PidLidReminderSet** property set to FALSE, and the **PidLidTaskResetReminder** property SHOULD NOT be TRUE.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

The server fulfills the server role as specified in the following documents:

- [MS-OXPROPS]
- [MS-OXOMSG]
- [MS-OXCFOLD]
- [MS-OXCTABL]

Copyright © 2013 Microsoft Corporation.

- [MS-OXOSFLD]
- [MS-OXOSRCH]
- [MS-OXOCFG]
- [MS-OXOCAL]
- [MS-OXOTASK]
- [MS-OXOFLAG]

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

None.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

Copyright © 2013 Microsoft Corporation.

4 Protocol Examples

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

Property	Property Set GUID	Name or ID
PidLidReminderSet (section 2.2.1.1)	{00062008-0000-0000-C000-000000000046}	0x8503
PidLidReminderTime (section 2.2.1.4)	{00062008-0000-0000-C000-000000000046}	0x8502
PidLidReminderDelta (section 2.2.1.3)	{00062008-0000-0000-C000-000000000046}	0x8501
PidLidReminderSignalTime (section 2.2.1.2)	{00062008-0000-0000-C000-000000000046}	0x8560
PidLidTaskDueDate ([MS-OXOTASK] section 2.2.2.2.5)	{00062003-0000-0000-C000-000000000046}	0x8105
PidLidTaskResetReminder ([MS-OXOTASK] section 2.2.2.2.6)	{00062003-0000-0000-C000-000000000046}	0x8107
PidLidAppointmentRecur ([MS-OXOCAL] section 2.2.1.44)	{00062002-0000-0000-C000-000000000046}	0x8216
PidLidAutoStartCheck ([MS-OXPROPS] section 2.41)	{00062002-0000-0000-C000-000000000046}	0x8244
PidLidFExceptionalAttendees ([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 ID
PidLidReminderSet	0x8004
PidLidReminderTime	0x8005
PidLidReminderDelta	0x81FF
PidLidReminderSignalTime	0x8006
PidLidTaskDueDate	0x8144
PidLidTaskResetReminder	0x815B
PidLidAppointmentRecur	0x81AE
PidLidAutoStartCheck	0x82E0
PidLidFExceptionalAttendees	0x82D7

Copyright © 2013 Microsoft Corporation.

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. Some of the properties of the Message object are as follows.

Property	Property ID	Property type	Data	Value
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	**PtypString** ([MS-OXCDATA] 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"
PidLidLocation ([MS-OXOCAL] section 2.2.1.4)	0x8009	**PtypString**	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"
PidTagStartDate ([MS-OXOCAL] section 2.2.1.30)	0x0060	**PtypTime** ([MS-OXCDATA] section 2.11.1)	00 10 00 A3 3F 70 C8 01	2008/02/16 02:00:00.000
PidTagEndDate ([MS-OXOCAL] section 2.2.1.31)	0x0061	**PtypTime**	00 78 C4 04 48 70 C8 01	2008/02/16 03:00:00.000
PidLidReminderSet (section 2.2.1.1)	0x8004	**PtypBoolean** ([MS-OXCDATA] section 2.11.1)	00	FALSE
PidLidReminderTime (section 2.2.1.4)	0x8005	**PtypTime**	Deleted	N/A
PidLidReminderSignalTime (section 2.2.1.2)	0x8006	**PtypTime**	Deleted	N/A
PidLidReminderDelta (section 2.2.1.3)	0x81FF	**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	0F 00 00 00	15 minutes

Around 11:44 A.M., John sets a reminder on the appointment for 30 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.

Copyright © 2013 Microsoft Corporation.

Property	Property ID	Property type	Data	Value
PidLidReminderSet	0x8004	**PtypBoolean**	01	TRUE
PidLidReminderDelta	0x81FF	**PtypInteger32**	1E 00 00 00	30 minutes
PidLidReminderSignalTime	0x8006	**PtypTime**	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.

Some of the properties of the Message object are as follows.

Property	Property ID	Property type	Data	Value
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	**PtypString** ([MS-OXCDATA] 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"
PidLidTaskStartDate ([MS-OXOTASK] section 2.2.2.2.4)	0x8143	**PtypTime** ([MS-OXCDATA] section 2.11.1)	00 C0 A3 8A 9C 6E C8 01	2008/02/14 00:00:00.000
PidLidTaskDueDate ([MS-OXOTASK] section 2.2.2.2.5)	0x8144	**PtypTime**	00 C0 A3 8A 9C 6E C8 01	2008/02/14 00:00:00.000
PidLidReminderSet (section 2.2.1.1)	0x8004	**PtypBoolean** ([MS-OXCDATA] section 2.11.1)	00	FALSE
PidLidReminderTime (section 2.2.1.4)	0x8005	**PtypTime**	Deleted	N/A
PidLidReminderSignalTime (section 2.2.1.2)	0x8006	**PtypTime**	Deleted	N/A

To ensure that the message is not forgotten, John sets a reminder for 6:00 P.M. 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.

Copyright © 2013 Microsoft Corporation.

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
PidLidReminderTime	0x8005	**PtypTime**	00 50 96 78 76 6f c8 01	2008/02/15 02:00:00.000
PidLidReminderSignalTime	0x8006	**PtypTime**	00 50 96 78 76 6f c8 01	2008/02/15 02:00:00.000
PidLidReminderSet (section 2.2.1.1)	0x8004	**PtypBoolean**	01	TRUE
PidTagReplyTime ([MS-OXOFLAG] section 2.2.3.1)	0x0030	**PtypTime**	00 50 96 78 76 6f c8 01	2008/02/15 02:00:00.000
PidTagReplyRequested ([MS-OXOMSG] section 2.2.1.42)	0x0C17	**PtypBoolean**	01	TRUE
PidTagResponseRequested ([MS-OXOMSG] section 2.2.1.43)	0x0063	**PtypBoolean**	01	TRUE
PidTagFlagStatus ([MS-OXOFLAG] section 2.2.1.1)	0x1090	**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	02 00 00 00	0x00000002

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

Property	Property ID	Property type
PidTagFlagCompleteTime ([MS-OXOFLAG] section 2.2.1.3)	0x1091	**PtypTime**

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.

Some of the properties of the reminder are as follows.

Property	Property ID	Property type	Data	Value
PidTagNormalizedSubject ([MS-OXCMSG] section	0x0E1D	**PtypString** ([MS-OXCDATA]	50 00 72 00 65 00 70 00 61 00 72 00 65 00 20 00 66 00	"Prepare for Contoso

Copyright © 2013 Microsoft Corporation.

Property	Property ID	Property type	Data	Value
2.2.1.10)		section 2.11.1)	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"
PidLidTaskStartDate ([MS-OXOTASK] section 2.2.2.2.4)	0x8143	**PtypTime** ([MS-OXCDATA] section 2.11.1)	00 80 0D B5 65 6F C8 01	2008/02/15 00:00:00.000
PidLidTaskDueDate ([MS-OXOTASK] section 2.2.2.2.5)	0x8144	**PtypTime**	00 80 0D B5 65 6F C8 01	2008/02/15 00:00:00.000
PidLidReminderSet (section 2.2.1.1)	0x8004	**PtypBoolean** ([MS-OXCDATA] section 2.11.1)	01	TRUE
PidLidReminderTime (section 2.2.1.4)	0x8005	**PtypTime**	00 6C 83 27 09 70 C8 01	2008/02/15 19:30:00.000
PidLidReminderSignalTime (section 2.2.1.2)	0x8006	**PtypTime**	00 6C 83 27 09 70 C8 01	2008/02/15 19:30:00.000
PidLidTaskResetReminder ([MS-OXOTASK] section 2.2.2.2.6)	0x815B	**PtypBoolean**	Deleted	N/A

Because he is done, John dismisses the reminder at 11:31 A.M. 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
PidLidReminderSet	0x8004	**PtypBoolean**	00	FALSE
PidLidTaskResetReminder	0x815B	**PtypBoolean**	01	TRUE

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.

Some of the properties of the Message object are as follows.

Copyright © 2013 Microsoft Corporation.

Property Name	Property ID	Property Type	Data	Value
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	**PtypString** ([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"
PidLidLocation ([MS-OXOCAL] section 2.2.1.4)	0x8009	**PtypString**	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"
PidTagStartDate ([MS-OXOCAL] section 2.2.1.30)	0x0060	**PtypTime** ([MS-OXCDATA] section 2.11.1)	00 A0 65 58 0D 70 C8 01	2008/02/15 20:00:00.000
PidTagEndDate ([MS-OXOCAL] section 2.2.1.31)	0x0061	**PtypTime**	00 08 2A BA 15 70 C8 01	2008/02/15 21:00:00.000
PidLidReminderSet (section 2.2.1.1)	0x8004	**PtypBoolean** ([MS-OXCDATA] section 2.11.1)	01	TRUE
PidLidReminderTime (section 2.2.1.4)	0x8005	**PtypTime**	00 A0 65 58 0D 70 C8 01	2008/02/15 20:00:00.000
PidLidReminderSignalTime (section 2.2.1.2)	0x8006	**PtypTime**	00 28 24 8D 0A 70 C8 01	2008/02/15 19:40:00.000
PidLidReminderDelta (section 2.2.1.3)	0x81FF	**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	14 00 00 00	20 minutes
PidLidAppointmentRecur ([MS-OXOCAL] section 2.2.1.44)	0x81AE	**PtypBinary** ([MS-OXCDATA] 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 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 00
```

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

The reminder for the first instance is displayed, and John dismisses the reminder before leaving for the Coho Winery. In response to the user-entered data, the client sends a **RopOpenMessage** ROP

Copyright © 2013 Microsoft Corporation.

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
PidLidReminderSignalTime	0x8006	**PtypTime**	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.

Some of the properties of the Message object are as follows.

Property	Property ID	Property type	Data	Value
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	**PtypString** ([MS-OXCDATA] 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"
PidLidTaskStartDate ([MS-OXOTASK] section 2.2.2.2.4)	0x8143	**PtypTime** ([MS-OXCDATA] section 2.11.1)	00 80 0D B5 65 6F C8 01	2008/02/15 00:00:00.000
PidLidTaskDueDate ([MS-OXOTASK] section 2.2.2.2.5)	0x8144	**PtypTime**	00 80 0D B5 65 6F C8 01	2008/02/15 00:00:00.000
PidLidReminderSet (section 2.2.1.1)	0x8004	**PtypBoolean** ([MS-OXCDATA] section 2.11.1)	01	TRUE
PidLidReminderTime (section 2.2.1.4)	0x8005	**PtypTime**	00 52 12 0F 07 70 C8 01	2008/02/15 19:15:00.000
PidLidReminderSignalTime (section 2.2.1.2)	0x8006	**PtypTime**	00 52 12 0F 07 70 C8 01	2008/02/15 19:15:00.000

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. 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.

Copyright © 2013 Microsoft Corporation.

Property	Property ID	Property type	Data	Value
PidLidReminderSignalTime	0x8006	**PtypTime**	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).

Some of the properties of the Recurring Calendar object are as follows.

Property	Property ID	Property type	Data	Value
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	**PtypString** ([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"
PidTagStartDate ([MS-OXOCAL] section 2.2.1.30)	0x0060	**PtypTime** ([MS-OXCDATA] section 2.11.1)	00 A0 65 58 0D 70 C8 01	2008/02/15 20:00:00.000
PidTagEndDate ([MS-OXOCAL] section 2.2.1.31)	0x0061	**PtypTime**	00 08 2A BA 15 70 C8 01	2008/02/15 21:00:00.000
PidLidReminderSet (section 2.2.1.1)	0x8004	**PtypBoolean** ([MS-OXCDATA] section 2.11.1)	01	TRUE
PidLidReminderTime (section 2.2.1.4)	0x8005	**PtypTime**	00 A0 65 58 0D 70 C8 01	2008/02/15 20:00:00.000
PidLidReminderSignalTime (section 2.2.1.2)	0x8006	**PtypTime**	00 00 44 54 82 75 C8 01	2008/02/22 06:40:00.000
PidLidAppointmentRecur ([MS-OXOCAL] section 2.2.1.44)	0x81AE	**PtypBinary** ([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:

Copyright © 2013 Microsoft Corporation.

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

John opens that instance, sets the signal time to none, and saves the Message object, disabling the reminder for just that instance. 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] section 4.2.2.7.

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	Property ID	Property type	Data	Value
PidLidAutoStartCheck ([MS-OXPROPS] section 2.41)	0x82E0	**PtypBoolean**	00	FALSE
PidLidReminderSet	0x8004	**PtypBoolean**	00	FALSE

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):

Copyright © 2013 Microsoft Corporation.

Property	Property ID	Property type	Data	Value
PidTagAttachMethod ([MS-OXCMSG] section 2.2.2.9)	0x3705	**PtypInteger32**	05 00 00 00	5
PidTagRenderingPosition ([MS-OXCMSG] section 2.2.2.16)	0x370B	**PtypInteger32**	FF FF FF FF	-1
PidTagExceptionStartTime ([MS-OXOCAL] section 2.2.10.1.4)	0x7FFB	**PtypTime**	00 38 62 11 42 75 C8 01	2008/02/22 11:00:00.000
PidTagAttachEncoding ([MS-OXCMSG] section 2.2.2.20)	0x3702	**PtypBinary**	00 00	Size: 0 bytes
PidTagExceptionEndTime ([MS-OXOCAL] section 2.2.10.1.5)	0x7FFC	**PtypTime**	00 a0 26 73 4A 75 C8 01	2008/02/22 12:00:00.000
PidTagAttachmentFlags ([MS-OXCMSG] section 2.2.2.23)	0x7FFD	**PtypInteger32**	02 00 00 00	Exception to a recurrence
PidTagDisplayName ([MS-OXCFOLD] section 2.2.2.2.4)	0x3001	**PtypString**	55 00 6E 00 74 00 69 00 74 00 6C 00 65 00 64 00 00 00	"Untitled"
PidTagAttachmentLinkId ([MS-OXCMSG] section 2.2.2.22)	0x7FFA	**PtypInteger32**	00 00 00 00	0
PidTagAttachFlags ([MS-OXCMSG] section 2.2.2.18)	0x3714	**PtypInteger32**	00 00 00 00	0
PidTagAttachmentHidden ([MS-OXCMSG] section 2.2.2.24)	0x7FFE	**PtypBoolean**	01	TRUE
PidTagAttachmentContactPhoto ([MS-OXOCNTC] section 2.2.1.8)	0x7FFF	**PtypBoolean**	00	FALSE

Next, in response to user-entered data for the Recurring Calendar 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	Property ID	Property type	Data	Value
PidLidAppointmentRecur	0x81AE	**PtypBinary**	Cb: 76 00 Lpb: The value is listed later in this section.	Cb: 118 bytes Lpb: The value is listed later in this section.
PidLidFExceptionalAttendees ([MS-OXOCAL] section 2.2.2.3)	0x82D7	**PtypBoolean**	00	FALSE

Copyright © 2013 Microsoft Corporation.

Property	Property ID	Property type	Data	Value
PidLidReminderSignalTime	0x8006	**PtypTime**	00 A8 EC DE 0A 7B C8 01	2008/02/29 19:40:00.000

Then the client releases the Exception Embedded Message object by sending a **RopRelease** ROP request ([MS-OXCROPS] section 2.2.15.3) and releases the Exception Attachment object by sending a **RopRelease** ROP request.

Finally, the client releases the attachment table by sending a **RopRelease** ROP request and releases the Recurring Calendar object by sending a **RopRelease** ROP request.

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 **PidLidAppointmentRecur** property. 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 information about how to determine the byte position in other cases, see [MS-OXOCAL] section 2.2.1.44.1.

Field name	Type	Size, in bytes	Data	Description
ReaderVersion	**WORD** ([MS-DTYP])	2	04 30	This field is set to 0x3004.
WriterVersion	**WORD**	2	04 30	This field is set to 0x3004.
UIGroup	**WORD**	2	0B 20	The pattern of the recurrence is weekly (0x200b).
PatternType	**WORD**	2	01 00	The pattern type is weekly (0x0001).
CalendarType	**WORD**	2	00 00	The calendar type is Gregorian (0x0000).
FirstDateTime	**ULONG** ([MS-DTYP])	4	C0 21 00 00	For information about how this property was calculated based on the start date, see [MS-OXOCAL] section 2.2.1.44.1.

Copyright © 2013 Microsoft Corporation.

Field name	Type	Size, in bytes	Data	Description
Period	**ULONG**	4	01 00 00 00	The recurrence occurs every week (0x00000001).
SlidingFlag	**ULONG**	4	00 00 00 00	The recurring instances do not rely on completion of the previous instances.
PatternTypeSpecific	**PtypBinary**	Varies (4 bytes, in this case, for Weekly pattern)	20 00 00 00	The recurring appointment occurs on Friday. For information about how the value is determined, see [MS-OXOCAL] section 2.2.1.44.1.2.
EndType	**ULONG**	4	23 20 00 00	No End. (0x00002023)
OccurrenceCount	**ULONG**	4	0A 00 00 00	Ignored, because recurrence does not have an end.
FirstDOW	**ULONG**	4	00 00 00 00	The first day of the week on the calendar is Sunday (the default value).
DeletedInstanceCount	**ULONG**	4	01 00 00 00	One deleted instance.
Deleted Instance Start	**ULONG**	4	A0 71 C3 0C	0x0CC371A0 == February 22, 2008, at 00:00 in UTC.
ModifiedInstanceCount	**ULONG**	4	01 00 00 00	One modified instance.
Modified Instance Start	**ULONG**	4	A0 71 C3 0C	0x0CC371A0 == February 22, 2008, at 00:00 in UTC.
Start Date of entire series	**ULONG**	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	**ULONG**	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	**ULONG**	4	06 30 00 00	This value is set to 0x00003006.
WriterVersion2	**ULONG**	4	09 30 00 00	This value is set to 0x00003009.
StartTimeOffset	**ULONG**	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

Copyright © 2013 Microsoft Corporation.

Field name	Type	Size, in bytes	Data	Description
				P.M.
EndTimeOffset	**ULONG**	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	**WORD**	2	01 00	There is one exception in this recurrence BLOB.
ExceptionInfo block for exception 1:				
Start Date Time	**ULONG**	4	34 74 C3 0C	0x0CC37434 == February 22, 2008, at 11:00 A.M.
End Date Time	**ULONG**	4	70 74 C3 0C	0x0CC37470 == February 22, 2008, at 12:00 P.M.
Original Start Time	**ULONG**	4	70 74 C3 0C	0x0CC37470 == February 22, 2008, at 12:00 P.M.
OverrideFlags	**WORD**	2	08 00	0x0008 corresponds to the **fReminderSet** flag, meaning just that field is specified for this exception.
fReminderSet	**ULONG**	4	00 00 00 00	0x00000000 == FALSE, meaning ReminderSet is overridden with the value FALSE for this exception.
ReservedBlock1Size	**ULONG**	4	00 00 00 00	There is no data in the reserved block.
ExtendedException block for exception 1:				
ChangeHighlight	**Byte Array**	Varies	04 00 00 00 00 00 00 00	The size of the **ChangeHighlight** field is 4 bytes. The value of the **PidLidChangeHighlight** property ([MS-OXOCAL] section 2.2.6.2) is zero for this exception.
ReservedBlockEE1Size	**ULONG**	4	00 00 00 00	There is no data in the reserved block.
ReservedBlock2Size	**ULONG**	4	00 00 00 00	There is no data in the reserved block.

Copyright © 2013 Microsoft Corporation.

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-OXCMSG] and [MS-OXCPRPT].

5.2 Index of Security Parameters

None.

Copyright © 2013 Microsoft Corporation.

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

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 2.2.1.7: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 support whatever the Windows API WINMM!PlaySound is able to play.

<2> Section 2.2.1.9: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 preserve the value if already set.

<3> 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, Outlook 2010, and Outlook 2013 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].

<4> Section 3.1.3.1: Office Outlook 2003 follows the minimal reminder domain scope strategy.

<5> Section 3.1.3.1: Office Outlook 2003 does not set the **piReminderUpgradeTime** property (section 2.2.3.1). Under normal operation, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 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, Outlook 2010, and Outlook 2013 set the **piReminderUpgradeTime** property to 0 to clean the reminders search folder (1), after which all reminders would be eligible to signal again.

<6> Section 3.1.4.1.2: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 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.

Copyright © 2013 Microsoft Corporation.

<7> Section 3.1.4.1.2: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 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, Outlook 2010, and Outlook 2013 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, Outlook 2010, and Outlook 2013 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.

<8> Section 3.1.4.1.3: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 set the **PidLidReminderTime** property (section 2.2.1.4) to the start time of the first instance. They set the property only when creating the Recurring Calendar object or modifying the series such that the series start date changes. Therefore the **PidLidReminderTime** property cannot be assumed to have a specific value for Recurring Calendar objects.

<9> Section 3.1.4.4.1: 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.

<10> Section 3.1.4.5.1: 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 such as the **Sent Items folder**.

<11> Section 3.1.4.5.3: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 use the **PidTagNormalizedSubject** ([MS-OXCMSG] 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, Outlook 2010, and Outlook 2013 use the **PidLidTaskDueDate** property ([MS-OXOTASK] section 2.2.2.2.5).

<12> Section 3.1.4.5.3: Office Outlook 2003 displays DueIn "Now" when first signaling a reminder for a Task object or flagged Message object; Office Outlook 2007, Outlook 2010, and Outlook 2013 display a more intuitive time-until-DueDate value for these Message object types. Office Outlook 2007, Outlook 2010, and Outlook 2013 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 uses the value of the **PidLidReminderTime** property (section 2.2.1.4) as the due time.

<13> Section 3.1.4.5.3: Office Outlook 2003 uses the signal time as the due time.

<14> Section 3.1.4.5.3.1: The **PidLidAutoStartCheck** property ([MS-OXPROPS] section 2.41) is deprecated and SHOULD NOT be used. Office Outlook 2003 reads and writes this property. This property is not read by Office Outlook 2007, Outlook 2010, or Outlook 2013.

<15> Section 3.1.4.6.3: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 set the **PidLidTaskResetReminder** property ([MS-OXOTASK] section 2.2.2.2.6).

Copyright © 2013 Microsoft Corporation.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

Copyright © 2013 Microsoft Corporation.

8 Index

A

Abstract data model
  client 13
  server 21
Applicability 8

C

Capability negotiation 8
Change tracking 38
Client
  abstract data model 13
  message processing 20
  other local events 20
  sequencing rules 20
  timer events 20
  timers 13
Client - higher layer triggered events
  generating instances for recurring Task objects 20
  post-receive processing 16
  post-transmit processing 16
  setting a reminder 14
Client - intialization
  loading and processing reminders 14
  scope of the search for active reminders 13

D

Data model - abstract
  client 13
  server 21
Dismissing a reminder on a recurring Calendar object 26
Dismissing a reminder on a task 25

E

Examples
  dismissing a reminder on a recurring Calendar object 26
  dismissing a reminder on a task 25
  overview 22
  removing a reminder from an instance of a recurring Calendar object 29
  setting a reminder on a Message object 24
  setting a reminder on a single instance appointment 23
  snoozing a reminder on a Contact object 28

F

Fields - vendor-extensible 8

G

Glossary 5

H

Higher layer triggered events - client
  generating instances for recurring Task objects 20
  post-receive processing 16
  post-transmit processing 16
  setting a reminder 14
Higher-layer triggered events
  server 21

I

Implementer - security considerations 35
Index of security parameters 35
Informative references 7
Initialization
  server 21
Initialization - client
  loading and processing reminders 14
  scope of the search for active reminders 13
Introduction 5

M

Message processing
  client 20
  server 21
Messages
  Properties Used to Specify and/or Decode Reminders 9
  Properties Used to Specify User Preferences That Are Roamed on the Server 12
  Shared Properties 11
  syntax 9
  transport 9

N

Normative references 6

O

Other local events
  client 20
  server 21
Overview (synopsis) 7

P

Parameters - security index 35
PidLidReminderDelta property used to specify and/or decode reminders 9
PidLidReminderFileParameter property used to specify and/or decode reminders 10
PidLidReminderOverride property used to specify and/or decode reminders 10

Copyright © 2013 Microsoft Corporation.

PidLidReminderPlaySound property used to specify and/or decode reminders 10
PidLidReminderSet property used to specify and/or decode reminders 9
PidLidReminderSignalTime property used to specify and/or decode reminders 9
PidLidReminderTime property used to specify and/or decode reminders 10
PidLidReminderType property used to specify and/or decode reminders 10
PidTagReplyTime property used to specify and/or decode reminders 10
piReminderUpgradeTime property used to specify user preferences that are roamed on the server 12
Preconditions 8
Prerequisites 8
Product behavior 36
Properties shared with the Appointment and Meeting Object Protocol 11
Properties shared with the Informational Flagging Protocol 11
Properties shared with the Message and Attachment Object Protocol 11
Properties shared with the Task-Related Objects Protocol 11
Properties used to specify and/or decode reminders
PidLidReminderDelta property 9
PidLidReminderFileParameter property 10
PidLidReminderOverride property 10
PidLidReminderPlaySound property 10
PidLidReminderSet property 9
PidLidReminderSignalTime property 9
PidLidReminderTime property 10
PidLidReminderType property 10
PidTagReplyTime property 10
Properties Used to Specify and/or Decode Reminders message 9
Properties used to specify user preferences that are roamed on the server
- iReminderUpgradeTime property 12
Properties Used to Specify User Preferences That Are Roamed on the Server message 12
Protocol examples 22

R

References 6
informative 7
normative 6
Relationship to other protocols 8
Removing a reminder from an instance of a recurring Calendar object 29

S

Security
implementer considerations 35
parameter index 35
Sequencing rules
client 20
server 21
Server
abstract data model 21
higher-layer triggered events 21
initialization 21
message processing 21
other local events 21
overview 20
sequencing rules 21
timer events 21
timers 21
Setting a reminder on a Message object 24
Setting a reminder on a single instance appointment example 23
Shared properties
properties shared with the Appointment and Meeting Object Protocol 11
properties shared with the Informational Flagging Protocol 11
properties shared with the Message and Attachment Object Protocol 11
properties shared with the Task-Related Objects Protocol 11
Shared Properties message 11
Snoozing a reminder on a Contact object 28
Standards assignments 8
Syntax 9

T

Timer events
client 20
server 21
Timers
client 13
server 21
Tracking changes 38
Transport 9
Triggered events - client
generating instances for recurring Task objects 20
post-receive processing 16
post-transmit processing 16
setting a reminder 14
Triggered events - higher-layer
server 21

V

Vendor-extensible fields 8
Versioning 8

Copyright © 2013 Microsoft Corporation.