Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.

- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

**Support.** For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/4/2008</td>
<td>0.1</td>
<td>New</td>
<td>Initial Availability.</td>
</tr>
<tr>
<td>4/25/2008</td>
<td>0.2</td>
<td>Minor</td>
<td>Revised and updated property names and other technical content.</td>
</tr>
<tr>
<td>6/27/2008</td>
<td>1.0</td>
<td>Major</td>
<td>Initial Release.</td>
</tr>
<tr>
<td>8/6/2008</td>
<td>1.01</td>
<td>Minor</td>
<td>Revised and edited technical content.</td>
</tr>
<tr>
<td>9/3/2008</td>
<td>1.02</td>
<td>Minor</td>
<td>Updated references.</td>
</tr>
<tr>
<td>12/3/2008</td>
<td>1.03</td>
<td>Minor</td>
<td>Updated IP notice.</td>
</tr>
<tr>
<td>4/10/2009</td>
<td>2.0</td>
<td>Major</td>
<td>Updated technical content for new product releases.</td>
</tr>
<tr>
<td>7/15/2009</td>
<td>3.0</td>
<td>Major</td>
<td>Revised and edited for technical content.</td>
</tr>
<tr>
<td>11/4/2009</td>
<td>4.0.0</td>
<td>Major</td>
<td>Updated and revised the technical content.</td>
</tr>
<tr>
<td>2/10/2010</td>
<td>4.1.0</td>
<td>Minor</td>
<td>Updated the technical content.</td>
</tr>
<tr>
<td>5/5/2010</td>
<td>4.2.0</td>
<td>Minor</td>
<td>Updated the technical content.</td>
</tr>
<tr>
<td>8/4/2010</td>
<td>4.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>11/3/2010</td>
<td>4.4</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>4.4</td>
<td>None</td>
<td>No changes to the meaning, language, and formatting of the technical content.</td>
</tr>
<tr>
<td>8/5/2011</td>
<td>4.5</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/7/2011</td>
<td>4.5</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/27/2012</td>
<td>6.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>6.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>6.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>6.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/26/2013</td>
<td>7.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>7.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>7.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>7.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>7.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>Date</td>
<td>Revision History</td>
<td>Revision Class</td>
<td>Comments</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>8.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>3/16/2015</td>
<td>9.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>5/26/2015</td>
<td>10.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/14/2015</td>
<td>10.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/13/2016</td>
<td>10.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>10.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>11.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>12.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/22/2021</td>
<td>13.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/17/2021</td>
<td>14.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
</tbody>
</table>
# Table of Contents

1 Introduction ........................................................................................................... 6
  1.1 Glossary ............................................................................................................. 6
  1.2 References ........................................................................................................ 8
  1.2.1 Normative References .................................................................................. 8
  1.2.2 Informative References ................................................................................ 9
  1.3 Overview .......................................................................................................... 9
  1.4 Relationship to Other Protocols ....................................................................... 10
  1.5 Prerequisites/Preconditions ............................................................................. 10
  1.6 Applicability Statement .................................................................................... 10
  1.7 Versioning and Capability Negotiation ............................................................ 10
  1.8 Vendor-Extensible Fields ................................................................................. 10
  1.9 Standards Assignments ..................................................................................... 10

2 Messages .............................................................................................................. 11
  2.1 Transport .......................................................................................................... 11
  2.2 Message Syntax ................................................................................................ 11
    2.2.1 Properties Used to Specify and/or Decode Reminders ......................... 11
      2.2.1.1 PidLidReminderSet Property ......................................................... 11
      2.2.1.2 PidLidReminderSignalTime Property ......................................... 11
      2.2.1.3 PidLidReminderDelta Property .................................................. 11
      2.2.1.4 PidLidReminderTime Property .................................................... 12
      2.2.1.5 PidLidReminderOverride Property ............................................. 12
      2.2.1.6 PidLidReminderPlaySound Property ........................................... 12
      2.2.1.7 PidLidReminderFileParameter Property ..................................... 12
      2.2.1.8 PidTagReplyTime Property ......................................................... 12
      2.2.1.9 PidLidReminderType Property ..................................................... 12
    2.2.2 Shared Properties ....................................................................................... 13
      2.2.2.1 Properties Shared with the Message and Attachment Object Protocol .. 13
      2.2.2.2 Properties Shared with the Informational Flagging Protocol ........ 13
      2.2.2.3 Properties Shared with the Task-Related Objects Protocol ............ 13
      2.2.2.3.1 PidLidTaskDueDate Property .................................................. 13
      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 14
      2.2.3.1 PidReminderUpgradeTime Property ........................................... 14

3 Protocol Details .................................................................................................... 15
  3.1 Client Details .................................................................................................... 15
    3.1.1 Abstract Data Model .................................................................................. 15
    3.1.2 Timers ......................................................................................................... 15
    3.1.3 Initialization ............................................................................................... 15
      3.1.3.1 Scope of the Search for Active Reminders ..................................... 15
      3.1.3.2 Loading and Processing Reminders .............................................. 16
    3.1.4 Higher-Layer Triggered Events .................................................................. 16
      3.1.4.1 Setting a Reminder ........................................................................... 16
        3.1.4.1.1 Reminders on Task Objects .................................................... 16
        3.1.4.1.2 Reminders on Message Objects Other Than Calendar Objects and Task Objects ................................................................. 16
        3.1.4.1.3 Reminders on Calendar Objects ............................................. 16
        3.1.4.1.4 Reminders on Recurring Calendar Objects .......................... 17
        3.1.4.1.5 Setting Reminders on Draft Message Objects ....................... 17
          3.1.4.1.5.1 Setting a Reminder for all Recipients ............................... 17
          3.1.4.1.5.2 Setting a Reminder for the Sender .................................. 18
        3.1.4.2 Post-Transmit Processing ............................................................... 18
        3.1.4.3 Post-Receive Processing ............................................................... 18
        3.1.4.4 Removing a Reminder .................................................................. 18

..............................................................
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.4.4.1</td>
<td>Single Instance Objects</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>3.1.4.4.2</td>
<td>Recurring Calendar Objects</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>3.1.4.4.3</td>
<td>Recurring Task Objects</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>3.1.4.5</td>
<td>When a Reminder Becomes Overdue</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>3.1.4.5.1</td>
<td>Reminders That Are Ignored When They Become Overdue</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>3.1.4.5.2</td>
<td>Reminders That Are Automatically Dismissed When Overdue</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>3.1.4.5.3</td>
<td>Actions for Overdue Reminders That Are Not Ignored or Automatically Dismissed</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>3.1.4.5.3.1</td>
<td>Starting a Conference</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>3.1.4.6</td>
<td>Dismissing a Reminder</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>3.1.4.6.1</td>
<td>Dismissing Single Instance Objects</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>3.1.4.6.2</td>
<td>Dismissing Recurring Calendar Objects</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>3.1.4.6.3</td>
<td>Dismissing for Recurring Task Objects</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>3.1.4.7</td>
<td>Snoozing a Reminder</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>3.1.4.7.1</td>
<td>Snoozing for Single Instance Objects</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>3.1.4.7.2</td>
<td>Snoozing for Recurring Calendar Objects and Recurring Task Objects</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>3.1.4.8</td>
<td>Generating Instances for Recurring Task Objects</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>3.1.5</td>
<td>Message Processing Events and Sequencing Rules</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>3.1.6</td>
<td>Timer Events</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>3.1.7</td>
<td>Other Local Events</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>Server Details</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Abstract Data Model</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Timers</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Initialization</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Higher-Layer Triggered Events</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Message Processing Events and Sequencing Rules</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>3.2.6</td>
<td>Timer Events</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>3.2.7</td>
<td>Other Local Events</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>Protocol Examples</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>4.1</td>
<td>Setting a Reminder on a Single Instance Appointment</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>4.2</td>
<td>Setting a Reminder on a Message Object</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>4.3</td>
<td>Dismissing a Reminder on a Task</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>4.4</td>
<td>Dismissing a Reminder on a Recurring Calendar Object</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>4.5</td>
<td>Snoozing a Reminder on a Contact Object</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>4.6</td>
<td>Removing a Reminder from an Instance of a Recurring Calendar Object</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>Security</td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>5.1</td>
<td>Security Considerations for Implementers</td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>5.2</td>
<td>Index of Security Parameters</td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>Appendix A: Product Behavior</td>
<td></td>
<td>37</td>
</tr>
<tr>
<td>7</td>
<td>Change Tracking</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>Index</td>
<td></td>
<td>41</td>
</tr>
</tbody>
</table>
1 Introduction

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

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

1.1 Glossary

This document uses the following terms:

**active reminder**: A reminder that is enabled on an object and is either pending or overdue, depending on whether the signal time has passed.

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

**calendar options dictionary**: A dictionary that contains calendar configuration data. It is stored in a folder associated information (FAI) message that is in a Calendar special folder.

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

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

**Coordinated Universal Time (UTC)**: A high-precision atomic time standard that approximately tracks Universal Time (UT). It is the basis for legal, civil time all over the Earth. Time zones around the world are expressed as positive and negative offsets from UTC. In this role, it is also referred to as Zulu time (Z) and Greenwich Mean Time (GMT). In these specifications, all references to UTC refer to the time at UTC-0 (or GMT).

**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.

**Draft Message object**: A Message object that has not been sent.

**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.

**Exception Attachment object**: An Attachment object on a Recurring Calendar object that contains the data for an exception, including an Exception Embedded Message object.

**Exception Embedded Message object**: An Embedded Message object that contains the changes for an Exception object.

**flags**: A set of values used to configure or report options or settings.

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

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

**instance**: A unique publication of data for a category. It enables a publisher to publish data for the same category multiple times. An example is a publisher who uses two different endpoints to publish data. These endpoints can publish the same category. However, each endpoint requires a different instance number to be considered a distinct publication by the server. An instance number is provided by the publishing client.

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

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

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

**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 subfolders.

**named property**: A property that is identified by both a GUID and either a string name or a 32-bit identifier.

**object**: A set of attributes, each with its associated values. Two attributes of an object have special significance: an identifying attribute and a parent-identifying attribute. An identifying attribute is a designated single-valued attribute that appears on every object; the value of this attribute identifies the object. For the set of objects in a replica, the values of the identifying attribute are distinct. A parent-identifying attribute is a designated single-valued attribute that appears on every object; the value of this attribute identifies the object's parent. That is, this attribute contains the value of the parent's identifying attribute, or a reserved value identifying no object. For the set of objects in a replica, the values of this parent-identifying attribute define a tree with objects as vertices and child-parent references as directed edges with the child as an edge's tail and the parent as an edge's head. Note that an object is a value, not a variable; a replica is a variable. The process of adding, modifying, or deleting an object in a replica replaces the entire value of the replica with a new value. As the word replica suggests, it is often the case that two replicas contain "the same objects". In this usage, objects in two replicas are considered the same if they have the same value of the identifying attribute and if there is a process in place (replication) to converge the values of the remaining attributes. When the members of a set of replicas are considered to be the same, it is common to say "an object" as shorthand referring to the set of corresponding objects in the replicas.

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

**property ID**: A 16-bit numeric identifier of a specific attribute. A property ID does not include any property type information.

**recipient**: An entity that can receive email messages.

**recurrence BLOB**: The binary large object (BLOB) encoding of a recurrence pattern, a recurrence range, and recurrence exceptions.

**recurrence pattern**: Information for a repeating event, such as the start and end time, the number of occurrences, and how occurrences are spaced, such as daily, weekly, or monthly.
**Recurring Calendar object**: A Calendar object that describes an event that repeats according to a recurrence pattern.

**recurring series**: An event that repeats at specific intervals of time according to a recurrence pattern.

**recurring task**: A series of Task objects that are described by a recurrence pattern.

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

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

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

**reminder properties**: A set of properties that specify the attributes of a reminder. These attributes include the time at which and the method by which a reminder is signaled or displayed.

**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.

**ROP request**: See ROP request buffer.

**search folder**: (1) A collection of related items to be crawled by a search service.

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

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

**signal time**: The time at which a reminder has been specified to notify the user or an agent acting on behalf of the user. For example, the signal time for a meeting that starts at 11:00 A.M. can be 10:45 A.M., thus allowing the user 15 minutes to prepare for or travel to the meeting upon receiving the notification.

**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.

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

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

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

### 1.2 References

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

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[MS-OXCDATA] Microsoft Corporation, "Data Structures".
[MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol".
[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-OXOSTOR] Microsoft Corporation, "Special Folders Protocol".
[MS-OXOTASK] Microsoft Corporation, "Task-Related Objects Protocol".

1.2.2 Informative References

[MS-DTYP] Microsoft Corporation, "Windows Data Types".
[MS-OXOCTNC] Microsoft Corporation, "Contact Object Protocol".

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 or recipients 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.

### 1.4 Relationship to Other Protocols

The Reminder Settings Protocol extends the Message and Attachment Object Protocol, as specified in [MS-OXCMMSG], 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 in [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-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** in which it intends to search, as specified in [MS-OXCMMSG].

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

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 2.222) 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 2.2.1.4.

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.

2.2.1.2 PidLidReminderSignalTime Property
Type: PtypTime ([MS-OXCDATA] section 2.11.1)
The PidLidReminderSignalTime property ([MS-OXPROPS] section 2.223) 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 2.218) 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] 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 2.224) 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 2.220), 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 2.221), 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 2.219) 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-OXFLAG] 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 2.227) SHOULD NOT<2> be set, and its value MUST be ignored.
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 be set when the client creates a reminder on an object 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-OPROPS] section 2.41)
- PidLidConferencingCheck ([MS-OXOCAL] section 2.2.1.56.2)
 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].
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.
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 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 for Message objects that are neither Calendar objects nor Task objects).

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidReminderSet</td>
<td>TRUE</td>
</tr>
<tr>
<td>PidLidReminderTime</td>
<td>The specified signal time</td>
</tr>
<tr>
<td>PidLidReminderSignalTime</td>
<td>The specified signal time</td>
</tr>
<tr>
<td>PidTagReplyTime</td>
<td>The specified signal time</td>
</tr>
</tbody>
</table>

3.1.4.1.3 Reminders on Calendar Objects

To enable the signaling of a reminder on Calendar objects, the client sets properties (or otherwise ensures that the properties are already set) as shown in the following table. The PidLidReminderTime property (section 2.2.1.4) and the PidLidReminderDelta property (section 2.2.1.3) MUST be set.
### 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 PidLidAppointmentRecurr property ([MS-OXOCAL] section 2.2.1.4) 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 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.

If either a sender or a recipient 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 and sender reminders is covered during post-transmit processing as specified in [MS-OXOFLAG] section 3.1.4.3.

### 3.1.4.1.5.1 Setting a Reminder for all Recipients

To set a reminder for all recipients 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, 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. For details about flagging and flagged states of a Message object, see [MS-OXOFLAG].

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 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-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.

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 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.

<table>
<thead>
<tr>
<th>Case</th>
<th>Due time</th>
</tr>
</thead>
<tbody>
<tr>
<td>For Message objects other than Calendar objects &lt;12&gt;</td>
<td>If the PidLidTaskDueDate property ([MS-OXOTASK] section 2.2.2.2.5) exists, the client SHOULD &lt;13&gt; use &quot;TaskDueDate + default End-of-Day time&quot;; otherwise use the value of the PidLidReminderSignalTime property (section 2.2.1.2) as the due time. For details about End-of-Day time, see the End element in [MS-OXOCFG] section 2.2.5.2.1.</td>
</tr>
<tr>
<td>For single instance Calendar objects</td>
<td>The client uses the value of the PidLidReminderTime property.</td>
</tr>
<tr>
<td>For Recurring Calendar objects</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 2.2.1.56.2) 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 2.2.1.56.3) is set to 0x00000000 and use the values of the following properties:

- `PidLidConferencingCheck`
- `PidLidConferencingType`
- `PidLidOrganizerAlias` ([MS-OXOCAL] section 2.2.1.56.6)
- `PidLidDirectory` ([MS-OXPROPS] section 2.94)
- `PidTagConversationTopic` ([MS-OXOMSG] section 2.2.1.5)
- `PidLidCollaborateDoc` ([MS-OXOCAL] section 2.2.1.56.7)
- `PidLidNetShowUrl` ([MS-OXOCAL] section 2.2.1.56.8)
- `PidLidAppointmentStateFlags` ([MS-OXOCAL] section 2.2.1.10)

### 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** 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).

#### 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 2.2.2.6, 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]
- [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.
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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Set GUID</th>
<th>Name or ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidReminderSet (section 2.2.1.1)</td>
<td>{00062008-0000-0000-C000-000000000046}</td>
<td>0x8004</td>
</tr>
<tr>
<td>PidLidReminderTime (section 2.2.1.4)</td>
<td>{00062008-0000-0000-C000-000000000046}</td>
<td>0x8005</td>
</tr>
<tr>
<td>PidLidReminderDelta (section 2.2.1.3)</td>
<td>{00062008-0000-0000-C000-000000000046}</td>
<td>0x8004</td>
</tr>
<tr>
<td>PidLidReminderSignalTime (section 2.2.1.2)</td>
<td>{00062008-0000-0000-C000-000000000046}</td>
<td>0x8006</td>
</tr>
<tr>
<td>PidLidTaskDueDate ([MS-OXOTASK] section 2.2.2.2.5)</td>
<td>{00062003-0000-0000-C000-000000000046}</td>
<td>0x8105</td>
</tr>
<tr>
<td>PidLidTaskResetReminder ([MS-OXOTASK] section 2.2.2.6)</td>
<td>{00062003-0000-0000-C000-000000000046}</td>
<td>0x8107</td>
</tr>
<tr>
<td>PidLidAppointmentRecur ([MS-OXOCAL] section 2.2.1.44)</td>
<td>{00062002-0000-0000-C000-000000000046}</td>
<td>0x8216</td>
</tr>
<tr>
<td>PidLidAutoStartCheck ([MS-OXPROPS] section 2.41)</td>
<td>{00062002-0000-0000-C000-000000000046}</td>
<td>0x8244</td>
</tr>
<tr>
<td>PidLidFExceptionalAttendees ([MS-OXOCAL] section 2.2.2.3)</td>
<td>{00062002-0000-0000-C000-000000000046}</td>
<td>0x822B</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidReminderSet</td>
<td>0x8004</td>
</tr>
<tr>
<td>PidLidReminderTime</td>
<td>0x8005</td>
</tr>
<tr>
<td>PidLidReminderDelta</td>
<td>0x81FF</td>
</tr>
<tr>
<td>PidLidReminderSignalTime</td>
<td>0x8006</td>
</tr>
<tr>
<td>PidLidTaskDueDate</td>
<td>0x8144</td>
</tr>
<tr>
<td>PidLidTaskResetReminder</td>
<td>0x815B</td>
</tr>
<tr>
<td>PidLidAppointmentRecur</td>
<td>0x81AE</td>
</tr>
<tr>
<td>PidLidAutoStartCheck</td>
<td>0x82E0</td>
</tr>
<tr>
<td>PidLidFExceptionalAttendees</td>
<td>0x82D7</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidTagNormalizedSubject</td>
<td>0x0E1D</td>
<td>PtypString</td>
<td>44 00 69 00 6E 00 6E 00 6E 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</td>
<td>&quot;Dinner with Robin Counts&quot;</td>
</tr>
<tr>
<td>PidLidLocation</td>
<td>0x8009</td>
<td>PtypString</td>
<td>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</td>
<td>&quot;Coho Vineyard&quot;</td>
</tr>
<tr>
<td>PidTagStartDate</td>
<td>0x0060</td>
<td>PtypTime</td>
<td>00 10 00 A3 3F 70 C8 01</td>
<td>2008/02/16 02:00:00.000</td>
</tr>
<tr>
<td>PidTagEndDate</td>
<td>0x0061</td>
<td>PtypTime</td>
<td>00 78 C4 04 48 70 C8 01</td>
<td>2008/02/16 03:00:00.000</td>
</tr>
<tr>
<td>PidLidReminderSet</td>
<td>0x8004</td>
<td>PtypBoolean</td>
<td>00</td>
<td>FALSE</td>
</tr>
<tr>
<td>PidLidReminderTime</td>
<td>0x8005</td>
<td>PtypTime</td>
<td>Deleted</td>
<td>N/A</td>
</tr>
<tr>
<td>PidLidReminderSignalTime</td>
<td>0x8006</td>
<td>PtypTime</td>
<td>Deleted</td>
<td>N/A</td>
</tr>
<tr>
<td>PidLidReminderDelta</td>
<td>0x81FF</td>
<td>PtypInteger32</td>
<td>0F 00 00 00</td>
<td>15 minutes</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidReminderSet</td>
<td>0x8004</td>
<td>PtypBoolean</td>
<td>01</td>
<td>TRUE</td>
</tr>
<tr>
<td>PidLidReminderDelta</td>
<td>0x81FF</td>
<td>PtypInteger32</td>
<td>1E 00 00 00</td>
<td>30 minutes</td>
</tr>
<tr>
<td>PidLidReminderSignalTime</td>
<td>0x8006</td>
<td>PtypTime</td>
<td>00 DC 1D 72 3B 70 C8 01</td>
<td>2008/02/16 01:30:00.000</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidReminderSet ([MS-OXOTASK] section 2.2.2.2.5)</td>
<td>0x8004</td>
<td>PtypBoolean ([MS-OXCDATA] section 2.11.1)</td>
<td>00</td>
<td>FALSE</td>
</tr>
<tr>
<td>PidLidReminderTime ([MS-OXOTASK] section 2.2.2.2.4)</td>
<td>0x8143</td>
<td>PtypTime ([MS-OXCDATA] section 2.11.1)</td>
<td>00 C0 A3 8A 9C 6E C8 01</td>
<td>2008/02/14 00:00:00.000</td>
</tr>
<tr>
<td>PidLidReminderSignalTime ([MS-OXOTASK] section 2.2.2.2.4)</td>
<td>0x8144</td>
<td>PtypTime ([MS-OXCDATA] section 2.11.1)</td>
<td>00 C0 A3 8A 9C 6E C8 01</td>
<td>2008/02/14 00:00:00.000</td>
</tr>
<tr>
<td>PidLidTaskDueDate ([MS-OXOTASK] section 2.2.2.2.4)</td>
<td>0x8005</td>
<td>PtypTime ([MS-OXCDATA] section 2.11.1)</td>
<td>Deleted</td>
<td>N/A</td>
</tr>
<tr>
<td>PidLidTaskStartDate ([MS-OXOTASK] section 2.2.2.2.4)</td>
<td>0x8006</td>
<td>PtypTime ([MS-OXCDATA] section 2.11.1)</td>
<td>Deleted</td>
<td>N/A</td>
</tr>
</tbody>
</table>

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.

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

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidReminderTime</td>
<td>0x8005</td>
<td>PtypTime</td>
<td>00 50 96 78 76 6f c8 01</td>
<td>2008/02/15 02:00:00.000</td>
</tr>
<tr>
<td>PidLidReminderSignalTime</td>
<td>0x8006</td>
<td>PtypTime</td>
<td>00 50 96 78 76 6f c8 01</td>
<td>2008/02/15 02:00:00.000</td>
</tr>
<tr>
<td>PidLidReminderSet (section 2.2.2.2.5)</td>
<td>0x8004</td>
<td>PtypBoolean</td>
<td>01</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
Next, the client sends a **RopDeletePropertiesNoReplicate** ROP request ([MS-OXCROPS] section 2.2.8.9) for the following property.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PidTagFlagCompleteTime</strong></td>
<td>0x1091</td>
<td>PtypTime</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PidTagNormalizedSubject</strong></td>
<td>0x0E1D</td>
<td>PtypString</td>
<td>50 00 72 00 65 00 70 00 61 00 72 00 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</td>
<td>&quot;Prepare for Contoso presentation&quot;</td>
</tr>
<tr>
<td><strong>PidLidTaskStartDate</strong></td>
<td>0x8143</td>
<td>PtypTime</td>
<td>00 80 0D B5 65 6F C8 01</td>
<td>2008/02/15 00:00:00.000</td>
</tr>
<tr>
<td><strong>PidLidTaskDueDate</strong></td>
<td>0x8144</td>
<td>PtypTime</td>
<td>00 80 0D B5 65 6F C8 01</td>
<td>2008/02/15 00:00:00.000</td>
</tr>
<tr>
<td><strong>PidLidReminderSet</strong></td>
<td>0x8004</td>
<td>PtypBoolean</td>
<td>01</td>
<td>TRUE</td>
</tr>
<tr>
<td><strong>PidLidReminderTime</strong></td>
<td>0x8005</td>
<td>PtypTime</td>
<td>00 6C 83 27 09 70 C8 01</td>
<td>2008/02/15 19:30:00.000</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidReminderSet</td>
<td>0x8004</td>
<td>PtypBoolean</td>
<td>00</td>
<td>FALSE</td>
</tr>
<tr>
<td>PidLidTaskResetReminder</td>
<td>0x815B</td>
<td>PtypBoolean</td>
<td>01</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property ID</th>
<th>Property Type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)</td>
<td>0x0E1D</td>
<td>PtypString ([MS-OXCDATA] section 2.11.1)</td>
<td>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</td>
<td>&quot;Lunch with Ben Smith&quot;</td>
</tr>
<tr>
<td>PidLidLocation ([MS-OXOCAL] section 2.2.1.4)</td>
<td>0x8009</td>
<td>PtypString</td>
<td>43 00 6F 00 68 00 6F 00 20 00 57 00 69 00 6E 00 65 00 72 00 79 00 00 00</td>
<td>&quot;Coho Winery&quot;</td>
</tr>
<tr>
<td>PidTagStartDate ([MS-OXOCAL] section 2.2.1.30)</td>
<td>0x0060</td>
<td>PtypTime ([MS-OXCDATA] section 2.11.1)</td>
<td>00 A0 65 58 0D 70 C8 01</td>
<td>2008/02/15 20:00:00.000</td>
</tr>
<tr>
<td>PidTagEndDate ([MS-OXOCAL] section 2.2.1.31)</td>
<td>0x0061</td>
<td>PtypTime</td>
<td>00 08 2A BA 15 70 C8 01</td>
<td>2008/02/15 21:00:00.000</td>
</tr>
<tr>
<td>PidLidReminderSet (section 2.2.1.1)</td>
<td>0x8004</td>
<td>PtypBoolean ([MS-OXCDATA] section 2.11.1)</td>
<td>01</td>
<td>TRUE</td>
</tr>
<tr>
<td>PidLidReminderTime (section 2.2.1.4)</td>
<td>0x8005</td>
<td>PtypTime</td>
<td>00 A0 65 58 0D 70 C8 01</td>
<td>2008/02/15 20:00:00.000</td>
</tr>
</tbody>
</table>
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 00 00 23 20 00 00 0A 00
0x0020: 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 02
0x0040: 00 00 0C 03 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 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 Name | Property ID | Property Type | Data | Value
--- | --- | --- | --- | ---
PidLidReminderSignalTime | 0x8006 | PtypTime | 00 28 24 8D 0A 70 C8 01 | 2008/02/15 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 Name | Property ID | Property Type | Data | Value
--- | --- | --- | --- | ---
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10) | 0x0E1D | PtypString ([MS-OXCDATA] section | 41 00 64 00 61 00 6D 00 20 00 42 00 61 00 72 | "Adam Barr"
```
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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidReminderSignalTime</td>
<td>0x8006</td>
<td>PtypTime</td>
<td>00 8C 20 DC 0F 70 C8 01</td>
<td>2008/02/15 20:18:00.000</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)</td>
<td>0x0E1D</td>
<td>PtypString</td>
<td>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</td>
<td>&quot;Lunch with Ben Smith&quot;</td>
</tr>
<tr>
<td>PidTagStartDate ([MS-OXOCAL] section 2.2.1.30)</td>
<td>0x0060</td>
<td>PtypTime</td>
<td>00 A0 65 58 0D 70 C8 01</td>
<td>2008/02/15 20:00:00.000</td>
</tr>
</tbody>
</table>
The value of the **PidLidAppointmentRecur** property is as follows:

Cb: 72 00

Lpb:

```
0x0000: 04 30 04 30 0B 20 01 00-00 00 00 00 C0 21 00 00 01 00
0x0010: 00 00 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 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 DF 70 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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidTagAttachSize</td>
<td>0x0E20</td>
<td>PtypInteger32</td>
<td>DC 0E 00 00</td>
<td>3804 bytes</td>
</tr>
<tr>
<td>PidTagLastModificationTime</td>
<td>0x3008</td>
<td>PtypTime</td>
<td>62 BC D2 86</td>
<td>2008/02/15 22:17:18.328</td>
</tr>
</tbody>
</table>

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 on the server):

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidAutoStartCheck ([MS-OXPROPS] section 2.41)</td>
<td>0x82E0</td>
<td>PtypBoolean</td>
<td>00</td>
<td>FALSE</td>
</tr>
<tr>
<td>PidLidReminderSet</td>
<td>0x8004</td>
<td>PtypBoolean</td>
<td>00</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

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 on the server):

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Property type</th>
<th>Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidTagAttachMethod ([MS-OXCMSG] section 2.2.2.9)</td>
<td>0x3705</td>
<td>PtypInteger32</td>
<td>05 00 00 00</td>
<td>5</td>
</tr>
<tr>
<td>PidTagRenderingPosition ([MS-OXCMSG] section 2.2.2.16)</td>
<td>0x370B</td>
<td>PtypInteger32</td>
<td>FF FF FF FF</td>
<td>-1</td>
</tr>
<tr>
<td>PidTagExceptionStartTime ([MS-OXOCAL] section 2.2.10.1.4)</td>
<td>0x7FFB</td>
<td>PtypTime</td>
<td>00 38 62 11 42 75 C8 01</td>
<td>2008/02/22 11:00:00.000</td>
</tr>
<tr>
<td>PidTagAttachEncoding ([MS-OXCMSG] section 2.2.2.20)</td>
<td>0x3702</td>
<td>PtypBinary</td>
<td>00 00</td>
<td>Size: 0 bytes</td>
</tr>
<tr>
<td>PidTagExceptionEndTime ([MS-OXOCAL] section 2.2.10.1.5)</td>
<td>0x7FFC</td>
<td>PtypTime</td>
<td>00 a0 26 73 4A 75 C8 01</td>
<td>2008/02/22 12:00:00.000</td>
</tr>
<tr>
<td>PidTagAttachmentFlags ([MS-OXCMSG] section 2.2.2.23)</td>
<td>0x7FFD</td>
<td>PtypInteger32</td>
<td>02 00 00 00</td>
<td>Exception to a recurrence</td>
</tr>
<tr>
<td>PidTagDisplayName ([MS-OXCFOLD] section 2.2.2.2.2.5)</td>
<td>0x3001</td>
<td>PtypString</td>
<td>55 00 6E 00 74 00 69 00 74 00 6C 00 65 00 64 00 00 00</td>
<td>&quot;Untitled&quot;</td>
</tr>
<tr>
<td>PidTagAttachmentLinkId ([MS-OXCMSG] section 2.2.2.22)</td>
<td>0x7FFA</td>
<td>PtypInteger32</td>
<td>00 00 00 00</td>
<td>0</td>
</tr>
<tr>
<td>PidTagAttachFlags ([MS-OXCMSG] section 2.2.2.18)</td>
<td>0x3714</td>
<td>PtypInteger32</td>
<td>00 00 00 00</td>
<td>0</td>
</tr>
<tr>
<td>PidTagAttachmentHidden ([MS-OXCMSG] section 2.2.2.24)</td>
<td>0x7FFE</td>
<td>PtypBoolean</td>
<td>01</td>
<td>TRUE</td>
</tr>
<tr>
<td>PidTagAttachmentContactPhoto ([MS-OXOCNCTC] section 2.2.1.8)</td>
<td>0x7FFF</td>
<td>PtypBoolean</td>
<td>00</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

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 on the server).
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 20 00-00 00 23 20 00 00 0A 00
0x0020: 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
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.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Size, in bytes</th>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>ULONG</td>
<td>4</td>
<td>01 00 00 00</td>
<td>The recurrence occurs every week (0x00000001).</td>
</tr>
<tr>
<td>SlidingFlag</td>
<td>ULONG</td>
<td>4</td>
<td>00 00 00 00</td>
<td>The recurring instances do not rely on completion of the previous instances.</td>
</tr>
<tr>
<td>PatternTypeSpecific</td>
<td>PtypBinary</td>
<td>Varies (4 bytes, in this case, for Weekly pattern)</td>
<td>20 00 00 00</td>
<td>The recurring appointment occurs on Friday. For information about how the value is determined, see [MS-OXOCAL] section 2.2.1.44.1.4.</td>
</tr>
<tr>
<td>EndType</td>
<td>ULONG</td>
<td>4</td>
<td>23 20 00 00</td>
<td>No End. (0x00002023)</td>
</tr>
<tr>
<td>OccurrenceCount</td>
<td>ULONG</td>
<td>4</td>
<td>0A 00 00 00</td>
<td>Ignored, because recurrence does not have an end.</td>
</tr>
<tr>
<td>FirstDOW</td>
<td>ULONG</td>
<td>4</td>
<td>00 00 00 00</td>
<td>The first day of the week on the calendar is Sunday (the default value).</td>
</tr>
<tr>
<td>DeletedInstanceCount</td>
<td>ULONG</td>
<td>4</td>
<td>01 00 00 00</td>
<td>One deleted instance.</td>
</tr>
<tr>
<td>Deleted Instance Start</td>
<td>ULONG</td>
<td>4</td>
<td>A0 71 C3 0C</td>
<td>0x0CC371A0 == February 22, 2008, at 00:00 in UTC.</td>
</tr>
<tr>
<td>ModifiedInstanceCount</td>
<td>ULONG</td>
<td>4</td>
<td>01 00 00 00</td>
<td>One modified instance.</td>
</tr>
<tr>
<td>Modified Instance Start</td>
<td>ULONG</td>
<td>4</td>
<td>A0 71 C3 0C</td>
<td>0x0CC371A0 == February 22, 2008, at 00:00 in UTC.</td>
</tr>
<tr>
<td>Start Date of entire series</td>
<td>ULONG</td>
<td>4</td>
<td>40 4A C3 0C</td>
<td>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</td>
</tr>
<tr>
<td>EndDate of entire series</td>
<td>ULONG</td>
<td>4</td>
<td>DF 80 E9 5A</td>
<td>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)</td>
</tr>
<tr>
<td>ReaderVersion2</td>
<td>ULONG</td>
<td>4</td>
<td>06 30 00 00</td>
<td>This value is set to 0x00003006.</td>
</tr>
<tr>
<td>WriterVersion2</td>
<td>ULONG</td>
<td>4</td>
<td>09 30 00 00</td>
<td>This value is set to 0x00003009.</td>
</tr>
<tr>
<td>StartTimeOffset</td>
<td>ULONG</td>
<td>4</td>
<td>D0 02 00 00</td>
<td>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.</td>
</tr>
<tr>
<td>EndTimeOffset</td>
<td>ULONG</td>
<td>4</td>
<td>0C 03 00 00</td>
<td>The hexadecimal end time of the recurrence is 0x00000030C, which corresponds to 780 minutes, which is 1:00 P.M.</td>
</tr>
</tbody>
</table>
### ExceptionInfo block for exception 1:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Size, in bytes</th>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExceptionCount</td>
<td>WORD</td>
<td>2</td>
<td>01 00</td>
<td>There is one exception in this <strong>recurrence BLOB</strong>.</td>
</tr>
</tbody>
</table>

**Start Date Time**

- **Type**: ULONG
- **Size**: 4
- **Data**: 34 74 C3 0C
- **Description**: 0x0CC37434 == February 22, 2008, at 11:00 A.M.

**End Date Time**

- **Type**: ULONG
- **Size**: 4
- **Data**: 70 74 C3 0C
- **Description**: 0x0CC37470 == February 22, 2008, at 12:00 P.M.

**Original Start Time**

- **Type**: ULONG
- **Size**: 4
- **Data**: 70 74 C3 0C
- **Description**: 0x0CC37470 == February 22, 2008, at 12:00 P.M.

**OverrideFlags**

- **Type**: WORD
- **Size**: 2
- **Data**: 08 00
- **Description**: 0x0008 corresponds to the `fReminderSet` flag, meaning just that field is specified for this exception.

**fReminderSet**

- **Type**: ULONG
- **Size**: 4
- **Data**: 00 00 00 00
- **Description**: 0x00000000 == FALSE, meaning ReminderSet is overridden with the value FALSE for this exception.

**ReservedBlock1Size**

- **Type**: ULONG
- **Size**: 4
- **Data**: 00 00 00 00
- **Description**: There is no data in the reserved block.

### ExtendedException block for exception 1:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Size, in bytes</th>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChangeHighlight</td>
<td>Byte Array</td>
<td>Varies</td>
<td>04 00</td>
<td>The size of the <code>ChangeHighlight</code> field is 4 bytes. The value of the <code>PidLidChangeHighlight</code> property ([MS-OXOCAL] section 2.2.6.2) is zero for this exception.</td>
</tr>
<tr>
<td>ReservedBlockEE1Size</td>
<td>ULONG</td>
<td>4</td>
<td>00 00</td>
<td>There is no data in the reserved block.</td>
</tr>
<tr>
<td>ReservedBlock2Size</td>
<td>ULONG</td>
<td>4</td>
<td>00 00</td>
<td>There is no data in the reserved block.</td>
</tr>
</tbody>
</table>
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.
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 updates to those products.

- Microsoft Exchange Server 2003
- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2019
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016
- Microsoft Outlook 2019
- Microsoft Outlook 2021

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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, Outlook 2013, Outlook 2016, and Outlook 2019 support whatever the Microsoft Windows API WINMM!PlaySound is able to play.

<2> Section 2.2.1.9: Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 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 2.2.2.5) 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). Microsoft Office Outlook 2007 Service Pack 1 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, Outlook 2013, Outlook 2016, and Outlook 2019 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].

[MS-OXORMDR] - v20210817
Reminder Settings Protocol
Copyright © 2021 Microsoft Corporation
Release: August 17, 2021
<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 2007, Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 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, Outlook 2013, Outlook 2016, and Outlook 2019 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, Outlook 2013, Outlook 2016, and Outlook 2019 set the **PidTagReplyTime** property ([MS-OXOFLAG] section 2.2.3.1) when setting a reminder on a Message object that is already flagged. 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.

<7> Section 3.1.4.1.2: Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 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, Outlook 2013, Outlook 2016, and Outlook 2019 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, Outlook 2013, Outlook 2016, and Outlook 2019 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, Outlook 2013, Outlook 2016, and Outlook 2019 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, Outlook 2013, Outlook 2016, and Outlook 2019 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, Outlook 2013, Outlook 2016, and Outlook 2019 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, Outlook 2013, Outlook 2016, and Outlook 2019 display a more intuitive time-until-DueDate value for these Message object types. Office Outlook 2007, Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 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, Outlook 2013, Outlook 2016, and Outlook 2019.

**<15> Section 3.1.4.6.3**: Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 set the **PidLidTaskResetReminder** property ([MS-OXOTASK] section 2.2.2.2.6).
7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

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.
- A document revision that captures changes to protocol functionality.

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 **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Revision class</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Appendix A: Product Behavior</td>
<td>Updated list of supported products.</td>
<td>major</td>
</tr>
</tbody>
</table>
8 Index

A

Abstract data model
client 15
server 22
Applicability 10

C

Capability negotiation 10
Change tracking 40
Client
abstract data model 15
message processing 22
other local events 22
sequencing rules 22
timer events 22
timers 15
Client - higher layer triggered events
generating instances for recurring Task objects 21
post-receive processing 18
post-transmit processing 18
setting a reminder 16
Higher-layer triggered events
server 23

I

Implementer - security considerations 36
Index of security parameters 36
Informative references 9
Initialization
server 22
Initialization - client
loading and processing reminders 16
scope of the search for active reminders 15
Introduction 6

M

Message processing
client 22
server 23
Messages
Properties Used to Specify and/or Decode
Reminders 11
Properties Used to Specify User Preferences That
Are Roamed on the Server 14
Shared Properties 13
syntax 11
transport 11

N

Normative references 8

O

Other local events
client 22
server 23
Overview (synopsis) 9

P

Parameters - security index 36
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 12
PidLidReminderPlaySound property used to specify
and/or decode reminders 12
PidLidReminderSet property used to specify and/or
decode reminders 11
PidLidReminderSignalTime property used to specify
and/or decode reminders 11
PidLidReminderTime property used to specify and/or
decode reminders 12
PidLidReminderType property used to specify and/or decode reminders 12

PidTagReplyTime property used to specify and/or decode reminders 12

piReminderUpgradeTime property used to specify user preferences that are roamed on the server 14

Preconditions 10
Prerequisites 10

Product behavior 37

Properties shared with the Appointment and Meeting Object Protocol 13
Properties shared with the Informational Flagging Protocol 13
Properties shared with the Message and Attachment Object Protocol 13
Properties shared with the Task-Related Objects Protocol 13

Properties used to specify and/or decode reminders

PidLidReminderDelta property 11
PidLidReminderFileParameter property 12
PidLidReminderOverride property 12
PidLidReminderPlaySound property 12
PidLidReminderSet property 11
PidLidReminderSignalTime property 11
PidLidReminderTime property 12
PidLidReminderType property 12
PidTagReplyTime property 12

Properties Used to Specify and/or Decode Reminders

Properties used to specify user preferences that are roamed on the server
- iReminderUpgradeTime property 14

Properties Used to Specify User Preferences That Are Roamed on the Server message 14

Protocol examples 24

R

References 8
informative 9
normative 8

Relationship to other protocols 10
Removing a reminder from an instance of a recurring Calendar object 30

S

Security
implementer considerations 36
parameter index 36

Sequencing rules
client 22
server 23

Server
abstract data model 22
higher-layer triggered events 23
initialization 22
message processing 23
other local events 23
overview 22
sequencing rules 23
timer events 23
timers 22
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 13
properties shared with the Message and Attachment Object Protocol 13
properties shared with the Task-Related Objects Protocol 13

Shared Properties message 13
Snoozing a reminder on a Contact object 29

Standards assignments 10
Syntax 11

T

Timer events
client 22
server 23

Timers
client 15
server 22

Tracking changes 40

Transport 11

Triggered events - client
generating instances for recurring Task objects 21
post-receive processing 18
post-transmit processing 18
setting a reminder 16

Triggered events - higher-layer
server 23

V

Vendor-extensible fields 10
Versioning 10