[MS-OXOCAL]:
Appointment and Meeting Object Protocol

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1	Major	Initial Availability.
04/25/2008	0.2	Minor	Revised and updated property names and other technical content.
06/27/2008	1.0	Major	Initial Release.
08/06/2008	1.0.1	Editorial	Revised and edited technical content.
09/03/2008	1.0.2	Editorial	Updated references.
12/03/2008	1.0.3	Editorial	Revised and edited technical content.
12/02/2009	1.0.4	Editorial	Revised and edited technical content.
04/10/2009	2.0	Major	Updated technical content and applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	3.1.0	Minor	Updated the technical content.
02/10/2010	3.2.0	Minor	Updated the technical content.
05/05/2010	3.3.0	Minor	Updated the technical content.
08/04/2010	4.0	Major	Significantly changed the technical content.
11/03/2010	4.1	Minor	Clarified the meaning of the technical content.
03/18/2011	4.1	No change	No changes to the meaning, language, and formatting of the technical content.
08/05/2011	5.0	Major	Significantly changed the technical content.
10/07/2011	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	6.0	Major	Significantly changed the technical content.
04/27/2012	7.0	Major	Significantly changed the technical content.
07/16/2012	7.1	Minor	Clarified the meaning of the technical content.
10/08/2012	8.0	Major	Significantly changed the technical content.
02/11/2013	8.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/26/2013	9.0	Major	Significantly changed the technical content.
11/18/2013	10.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
02/10/2014	10.0	No change	No changes to the meaning, language, or formatting of the technical content.
04/30/2014	11.0	Major	Significantly changed the technical content.
07/31/2014	11.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	11.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction

The Appointment and Meeting Object Protocol allows a user to manage schedules electronically. This protocol extends the Message and Attachment Object Protocol, which is described in [MS-OXCMSG].

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

1.1 Glossary

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

Address Book object
Appointment object
Attachment object
attachments table
base64 encoding
best body
big-endian
binary large object (BLOB)
blind carbon copy (Bcc) recipient
calendar
Calendar folder
Calendar object
calendar options dictionary
Calendar special folder
condition
contents table
Coordinated Universal Time (UTC)
counter proposal
delegate
Delegate Information object
delegator
Deleted Items folder
Embedded Message object
EntryID
Exception Attachment object
Exception Embedded Message object
Exception object
flags
free/busy status
handle
Inbox folder
informational update
little-endian
mailbox
meeting
Meeting Cancellation object
Meeting Forward Notification object
Meeting object
meeting request
Meeting Request object

Meeting Response object
meeting update
Meeting Update object
Meeting Workspace
meeting-related object
Message object
message store
optional attendee
organizer
orphan instance
Out of Office (OOF)
Outbox folder
property ID
property name
public folder
recipient
recurrence BLOB
recurrence pattern
Recurring Calendar object
recurring series
recurring task
reminder
reminder properties
remote operation (ROP)
remote procedure call (RPC)
required attendee
Resource object
Rich Text Format (RTF)
rule
search key
sendable attendee
Sent Items folder
sequence number
signal time
significant change
single-instance object
skip block
stream
Task object
tentative
tombstone
Unicode
Uniform Resource Locator (URL)
unsendable attendee

The following terms are specific to this document:

floating appointment: An appointment that starts and ends at the same local time regardless of any time zone considerations.

recurrence range: The range of time for which a recurrence pattern continues.

time zone update: Any change to a time zone that occurs when a time zone changes the dates in which it observes Daylight Saving Time (DST) or changes its offset from **Coordinated Universal Time (UTC)**.

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

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

1.2.1 Normative References

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

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

[MS-MEETS] Microsoft Corporation, "Meetings Web Services Protocol".

[MS-OXBBODY] Microsoft Corporation, "Best Body Retrieval Algorithm".

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

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

[MS-OXCICAL] Microsoft Corporation, "iCalendar to Appointment Object Conversion Algorithm".

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

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

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

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

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

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

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

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

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

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

[MS-OXODLGT] Microsoft Corporation, "Delegate Access Configuration Protocol".

[MS-OXODOC] Microsoft Corporation, "Document Object Protocol".

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

[MS-OXOPFFB] Microsoft Corporation, "Public Folder-Based Free/Busy Protocol".

[MS-OXORMDR] Microsoft Corporation, "Reminder Settings Protocol".

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

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

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

[MS-OXRTFCP] Microsoft Corporation, "Rich Text Format (RTF) Compression Algorithm".

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

1.2.2 Informative References

[MS-OXCFXICS] Microsoft Corporation, "Bulk Data Transfer Protocol".

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

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

1.3 Overview

The Appointment and Meeting Object Protocol enables a user to manage schedules electronically by creating events on **calendars** and optionally requesting others to attend the events. The events can be made to recur at specific intervals. Upon receiving an invitation to a calendar event, users can accept, decline, or propose a different date and/or time for the event.

The Appointment and Meeting Object Protocol also enables one user, a **delegate**, to manage the calendar of another user, the **delegator**, by allowing interaction between the delegate and the delegator's calendar.

The Appointment and Meeting Object Protocol defines various types of **Message objects** that the client uses for working with a user's electronic schedule. These Message objects can be broadly categorized as either a **Calendar object** or a **meeting-related object**.

The Appointment and Meeting Object Protocol extends the Message and Attachment Object Protocol, as described in [MS-OXCMSG], by defining new properties for a Message object and by adding constraints to existing properties of Message object.

1.4 Relationship to Other Protocols

The Appointment and Meeting Object Protocol extends the Message and Attachment Object Protocol, as described in [MS-OXCMSG], for use with Calendar objects and relies on the Property and Stream Object Protocol, as described in [MS-OXCPRPT], and the Email Object Protocol, as described in [MS-OXOMSG], for message transport and delivery.

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

1.5 Prerequisites/Preconditions

The Appointment and Meeting Object Protocol requires that a client acquires a **handle** to the object on which it will operate. It also requires that the client acquires a handle to the **Calendar folder** to access Calendar objects when required. For more information on working with folders, messages, **recipients** (1), and tables, see [MS-OXCPRPT], [MS-OXCMSG], and [MS-OXCFOLD].

1.6 Applicability Statement

The Appointment and Meeting Object Protocol is appropriate for clients and servers that manage user appointments and **meetings** and their associated resources.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

This protocol does not provide any vendor extensibility beyond what is already specified in [MS-OXCMSG].

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Appointment and Meeting Object Protocol extends and relies on [MS-OXCMSG] and uses the Property Stream Object Protocol, as specified in [MS-OXCPRPT], and the Email Object Protocol, as specified in [MS-OXOMSG], for transport.

2.2 Message Syntax

Calendar objects and meeting-related objects can be created and modified by clients and servers.

Clients operate on Calendar objects and meeting-related objects by using the Message and Attachment Object Protocol, as specified in [MS-OXCMSG]. How servers operate on these objects is implementation-dependent, but the results of any such operations MUST be exposed to clients as specified by the Appointment and Meeting Object Protocol.

Unless otherwise specified, Calendar objects and meeting-related objects MUST adhere to all property constraints specified in [MS-OXCMSG]. An object can contain other properties but these properties do not have any impact on the Appointment and Meeting Object Protocol.<1>

When a property is referred to as "read-only for the client", it means that a client SHOULD NOT attempt to change the value of this property and that a server returns an error and ignores any request to change the value of this property.

2.2.1 Common Properties

Properties that are common to all object types in the Appointment and Meeting Object Protocol are specified in sections 2.2.1.1 through 2.2.1.51.9. Unless otherwise specified, these properties exist on all Calendar objects and meeting-related objects. Unless otherwise specified, all common properties are ordered **little-endian**.

2.2.1.1 PidLidAppointmentSequence Property

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

The **PidLidAppointmentSequence** property ([MS-OXPROPS] section 2.25) specifies the **sequence number** (2) of a **Meeting object**. A Meeting object begins with the sequence number (2) set to 0 (zero) and is incremented each time the **organizer** sends out a **Meeting Update object**. The sequence number (2) is copied onto the **Meeting Response object** so that the client or server knows which version of the meeting is being responded to. This property is unsigned. For more details about when and how a client increments the sequence number (2), see section 3.1.5.4.

2.2.1.2 PidLidBusyStatus Property

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

The **PidLidBusyStatus** property ([MS-OXPROPS] section 2.47) specifies the availability of a user for the event described by the object. This property MUST be one of the values specified in the following table.

Status	Value	Meaning
olFree	0x00000000	The user is available.

Status	Value	Meaning
olTentative	0x00000001	The user has a **tentative** event scheduled.
olBusy	0x00000002	The user is busy.
olOutOfOffice	0x00000003	The user is **Out of Office (OOF)**.
olWorkingElsewhere<2>	0x00000004	The user is working from a location other than the office.

2.2.1.3 PidLidAppointmentAuxiliaryFlags Property

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

The **PidLidAppointmentAuxiliaryFlags** property ([MS-OXPROPS] section 2.8) specifies a bit field that describes the auxiliary state of the object. This property is not required. The individual **flags** that can be set are specified in the following diagram.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	B	C	D	E	F	G	unused																								

A - auxApptFlagCopied (1 bit): This flag indicates that the Calendar object was copied from another Calendar folder.

B - auxApptFlagForceMtgResponse (1 bit): This flag on a **Meeting Request object** indicates that the client or server can require that a Meeting Response object be sent to the organizer when a response is chosen.

C - auxApptFlagForwarded (1 bit): This flag on a Meeting Request object indicates that it was forwarded by the organizer or another recipient (2), rather than sent directly from the organizer.

D - Reserved (1 bit): This flag is reserved for future use and MUST NOT be set.

E - Reserved (1 bit): This flag is reserved for future use and MUST NOT be set.

F - auxApptFlagRepairUpdateMessage (1 bit): This flag is set when the **meeting request** is a Repair Update Message sent from a server-side calendar repair system.

G - Reserved (1 bit): This flag is reserved for future use and MUST NOT be set.

unused (25 bits): This flag is not used, MUST be zero, and MUST be ignored.

2.2.1.4 PidLidLocation Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidLocation** property ([MS-OXPROPS] section 2.159) specifies the location of the event. This property is not required.

2.2.1.5 PidLidAppointmentStartWhole Property

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

The **PidLidAppointmentStartWhole** property ([MS-OXPROPS] section 2.29) specifies the start date and time of the event in **Coordinated Universal Time (UTC)**. The value of this property MUST be less than or equal to the value of the **PidLidAppointmentEndWhole** property (section 2.2.1.6). For a **recurring series**, the value of this property is the start date and time of the first instance according to the **recurrence pattern**. Note that for some appointments, the value of this time property is not interpreted strictly as a UTC time. For more details about how this property is interpreted, see section 3.1.5.5.<3>

2.2.1.6 PidLidAppointmentEndWhole Property

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

The **PidLidAppointmentEndWhole** property ([MS-OXPROPS] section 2.14) specifies the end date and time for the event in **UTC**. The value of this property MUST be greater than or equal to the value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5). For a recurring series, the value of this property is the end date and time of the first instance according to the recurrence pattern. Note that for some appointments, the value of this time property is not interpreted strictly as a UTC time. For more details about how this property is interpreted, see section 3.1.5.5.<4>

2.2.1.7 PidLidAppointmentDuration Property

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

The **PidLidAppointmentDuration** property ([MS-OXPROPS] section 2.11) specifies the length of the event, in minutes. This property is not required. If set, the value MUST be the number of minutes between the value of the **PidLidAppointmentStartWhole** property section 2.2.1.5) and the value of the **PidLidAppointmentEndWhole** property (section 2.2.1.6).<5>

2.2.1.8 PidNameKeywords Property

Type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1.2)

The **PidNameKeywords** property ([MS-OXCMSG] section 2.2.1.17) specifies the color to be used when displaying a Calendar object. If the **PidNameKeywords** property contains the name of a category that is described in the category list, as specified in [MS-OXOCFG] section 2.2.5.2.2, the client displays the Calendar object in the color that is specified for the category. Otherwise, the client displays the Calendar object in the default color.<6>

2.2.1.9 PidLidAppointmentSubType Property

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

The **PidLidAppointmentSubType** property ([MS-OXPROPS] section 2.31) specifies whether the event is an all-day event, as specified by the user. A value of TRUE indicates that the event is an all-day event, in which case the values of the **PidLidAppointmentStartWhole** property (section 2.2.1.5) and the **PidLidAppointmentEndWhole** property (section 2.2.1.6) MUST both be midnight so that the duration is a multiple of 24 hours and is at least 24 hours. A value of FALSE or the absence of this property indicates that the event is not an all-day event. Note that the client or server cannot infer the value as TRUE when a user happens to create an event that is 24 hours long, even if the event starts and ends at midnight.

2.2.1.10 PidLidAppointmentStateFlags Property

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

The **PidLidAppointmentStateFlags** property ([MS-OXPROPS] section 2.31) is a bit field that specifies the appointment state of the object. This property is not required. The individual flags that can be set are as follows.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
M	R	C	unused																												

M - asfMeeting (1 bit): This flag indicates that the object is a Meeting object or a meeting-related object.

R - asfReceived (1 bit): This flag indicates that the represented object was received from someone else.

C - asfCanceled (1 bit): This flag indicates that the Meeting object that is represented by the object has been canceled.

unused (29 bits): These bits are not used. MUST be zero, and MUST be ignored.

2.2.1.11 PidLidResponseStatus Property

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

The **PidLidResponseStatus** property ([MS-OXPROPS] section 2.231) specifies the response status of an attendee. This property MUST be set to one of the values listed in the following table.

Response status	Value	Meaning
respNone	0x00000000	No response is required for this object. This is the case for **Appointment objects** and Meeting Response objects.
respOrganized	0x00000001	This Meeting object belongs to the organizer.
respTentative	0x00000002	This value on the attendee's Meeting object indicates that the attendee has tentatively accepted the Meeting Request object.
respAccepted	0x00000003	This value on the attendee's Meeting object indicates that the attendee has accepted the Meeting Request object.
respDeclined	0x00000004	This value on the attendee's Meeting object indicates that the attendee has declined the Meeting Request object.
respNotResponded	0x00000005	This value on the attendee's Meeting object indicates that the attendee has not yet responded. This value is on the Meeting Request object, Meeting Update object, and **Meeting Cancellation object**.

2.2.1.12 PidLidRecurring Property

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

The **PidLidRecurring** property ([MS-OXPROPS] section 2.216) specifies whether the object represents a recurring series. A value of TRUE indicates that the object represents a recurring series. A value of FALSE or the absence of this property indicates that the object represents either a

single instance or an exception (including an **orphan instance**). Note the difference between this property and the **PidLidIsRecurring** property (section 2.2.1.13).

2.2.1.13 PidLidIsRecurring Property

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

The **PidLidIsRecurring** property ([MS-OXPROPS] section 2.156) specifies whether the object is associated with a recurring series. A value of TRUE indicates that the object represents either a recurring series or an exception (including an orphan instance). A value of FALSE or the absence of this property<7> indicates that the object represents a single instance. Note the difference between this property and the **PidLidRecurring** property (section 2.2.1.12).

2.2.1.14 PidLidClipStart Property

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

For single-instance Calendar objects, the **PidLidClipStart** property ([MS-OXPROPS] section 2.60) specifies the start date and time of the event in UTC. For a recurring series, this property specifies midnight in the user's machine time zone, on the date of the first instance, then is persisted in UTC. This property is not required.

2.2.1.15 PidLidClipEnd Property

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

For single-instance Calendar objects, the **PidLidClipEnd** property ([MS-OXPROPS] section 2.59) specifies the end date and time of the event in UTC. For a recurring series, this property specifies midnight in the user's machine time zone, on the date of the last instance of the recurring series, then is persisted in UTC, unless the recurring series has no end, in which case the value MUST be "31 August 4500, 11:59 P.M" This property is not required.

2.2.1.16 PidLidAllAttendeesString Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidAllAttendeesString** property ([MS-OXPROPS] section 2.5) specifies a list of all the attendees except for the organizer, including **Resource objects** and **unsendable attendees**. The value for each attendee is the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) of the attendee's **Address Book object**. Separate entries are delimited by a semicolon followed by a space. This property is not required.

2.2.1.17 PidLidToAttendeesString Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidToAttendeesString** property ([MS-OXPROPS] section 2.343) contains a list of all the **sendable attendees** who are also **required attendees**. The value for each attendee is the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) of the attendee's Address Book object. Separate entries are delimited by a semicolon followed by a space. This property is not required.

2.2.1.18 PidLidCcAttendeesString Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidCcAttendeesString** property ([MS-OXPROPS] section 2.50) contains a list of all the sendable attendees who are also **optional attendees**. The value for each attendee is the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) of the attendee's Address Book object. Separate entries are delimited by a semicolon followed by a space. This property is not required.

2.2.1.19 PidLidNonSendableTo Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidNonSendableTo** property ([MS-OXPROPS] section 2.179) contains a list of all the unsendable attendees who are also required attendees. The value for each attendee is the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) of the attendee's Address Book object. Separate entries are delimited by a semicolon followed by a space. This property is not required.

2.2.1.20 PidLidNonSendableCc Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidNonSendableCc** property ([MS-OXPROPS] section 2.179) contains a list of all the unsendable attendees who are also optional attendees. The value for each attendee is the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) of the attendee's Address Book object. Separate entries are delimited by a semicolon followed by a space. This property is not required.

2.2.1.21 PidLidNonSendableBcc Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidNonSendableBcc** property ([MS-OXPROPS] section 2.177) contains a list of all the unsendable attendees who are also Resource objects. The value for each attendee is the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) of the attendee's Address Book object. Separate entries are delimited by a semicolon followed by a space. This property is not required.

2.2.1.22 PidLidNonSendToTrackStatus Property

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

The **PidLidNonSendToTrackStatus** property ([MS-OXPROPS] section 2.182) contains the value from the response table, as specified in section 2.2.1.11, for each attendee listed in the **PidLidNonSendableTo** property (section 2.2.1.19). This property is required only when the **PidLidNonSendableTo** property is set. The number of values in this property MUST equal the number of values in the **PidLidNonSendableTo** property. Each **PtypInteger32** value ([MS-OXCDATA] section 2.11.1) in this property corresponds to the attendee in the **PidLidNonSendableTo** property at the same index.

2.2.1.23 PidLidNonSendCcTrackStatus Property

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

The **PidLidNonSendCcTrackStatus** property ([MS-OXPROPS] section 2.181) contains the value from the response table, as specified in section 2.2.1.11, for each attendee listed in the **PidLidNonSendableCc** property (section 2.2.1.20). This property is required only when the

PidLidNonSendableCc property is set. The number of values in this property MUST equal the number of values in the **PidLidNonSendableCc** property. Each **PtypInteger32** value ([MS-OXCDATA] section 2.11.1) in this property corresponds to the attendee in the **PidLidNonSendableCc** property at the same index.

2.2.1.24 PidLidNonSendBccTrackStatus Property

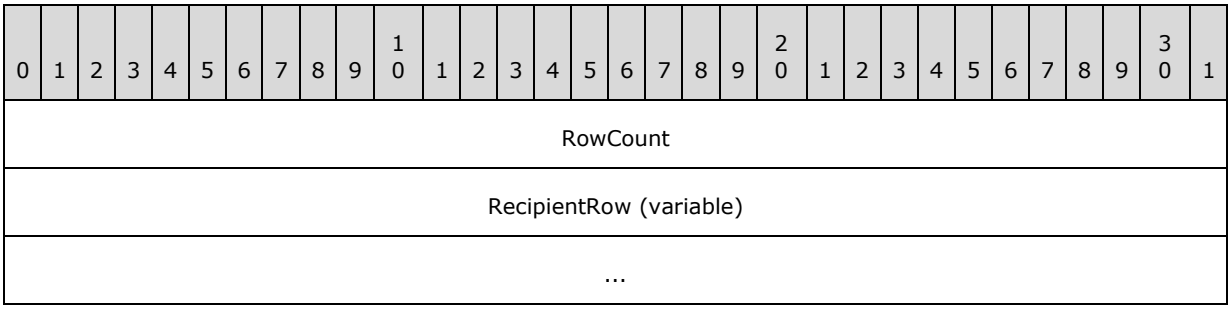
Type: **PtypMultipleInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidLidNonSendBccTrackStatus** property ([MS-OXPROPS] section 2.181) contains the value from the response table, as specified in section 2.2.1.11, for each attendee listed in the **PidLidNonSendableBcc** property (section 2.2.1.21). This property is required only when the **PidLidNonSendableBcc** property is set. The number of values in this property MUST equal the number of values in the **PidLidNonSendableBcc** property. Each **PtypInteger32** value ([MS-OXCDATA] section 2.11.1) in this property corresponds to the attendee in the **PidLidNonSendableBcc** property at the same index.

2.2.1.25 PidLidAppointmentUnsendableRecipients Property

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

The **PidLidAppointmentUnsendableRecipients** property ([MS-OXPROPS] section 2.35) contains a list of unsendable attendees. This property is not required but SHOULD be set.<8> It has the following format.



RowCount (4 bytes): An integer that specifies the number of structures in the **RecipientRow** field.

RecipientRow (variable): An array of **RecipientRow** structures, as specified in [MS-OXCDATA] section 2.8.3. Each structure specifies an unsendable attendee. The **RowCount** field specifies the number of structures contained in this field.

For details about properties that can be set on **RecipientRow** structures for Calendar objects and meeting-related objects, see section 2.2.4.10.

2.2.1.26 PidLidAppointmentNotAllowPropose Property

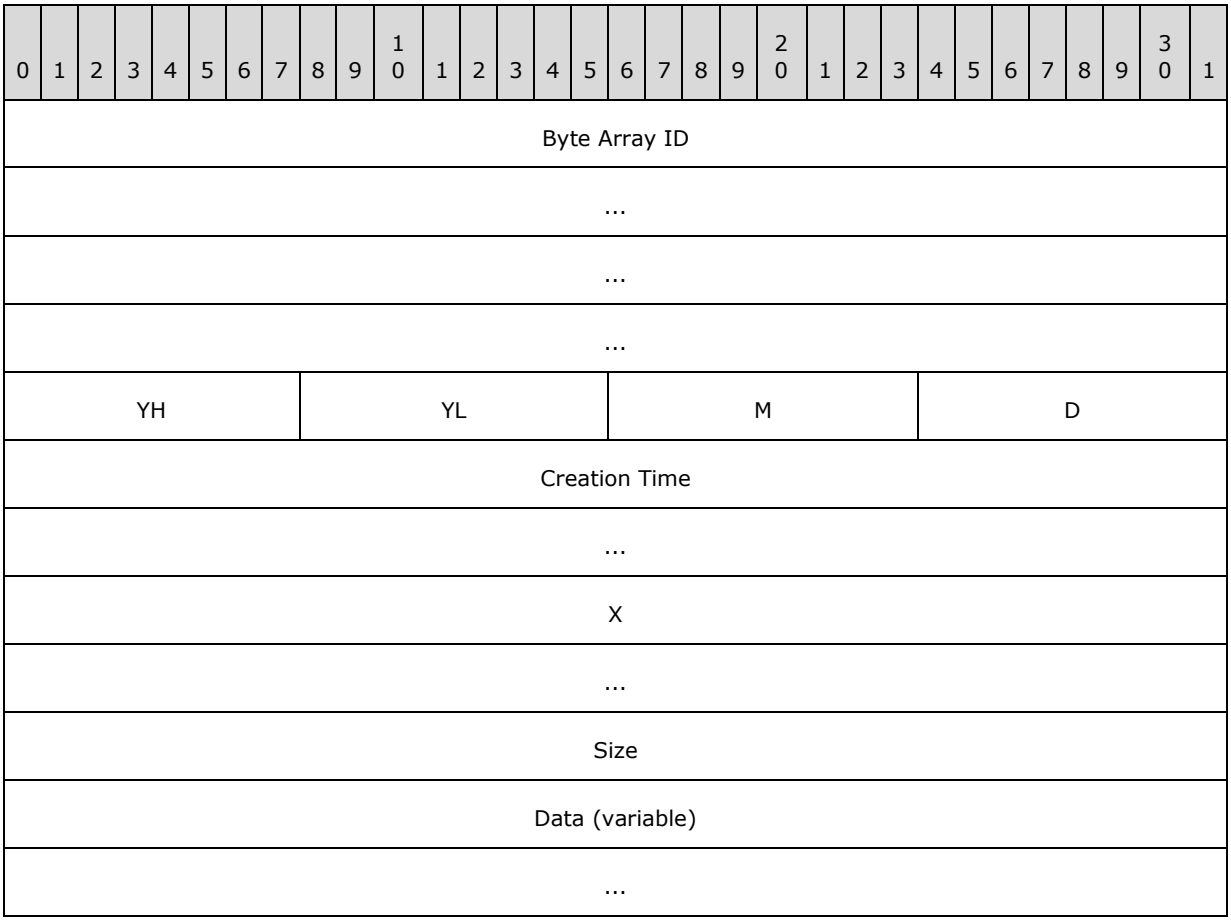
Type: **PtypBoolean** ([MS-OXCDATA] section 2.11.1)

A value of TRUE for the **PidLidAppointmentNotAllowPropose** property ([MS-OXPROPS] section 2.17) indicates that attendees are not allowed to propose a new date and/or time for the meeting. A value of FALSE or the absence of this property indicates that the attendees are allowed to propose a new date and/or time. This property is meaningful only on Meeting objects, Meeting Request objects, and Meeting Update objects.

2.2.1.27 PidLidGlobalObjectId Property

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

The **PidLidGlobalObjectId** property ([MS-OXPROPS] section 2.142) specifies the unique identifier of the Calendar object. After it is set for a Calendar object, the value of this property MUST NOT change. The fields in this **binary large object (BLOB)** are specified in the following table. All fields have little-endian byte order.



Byte Array ID (16 bytes): An array of 16 bytes identifying this BLOB as a Global Object ID. The byte array MUST be as follows: 0x04, 0x00, 0x00, 0x00, 0x82, 0x00, 0xE0, 0x00, 0x74, 0xC5, 0xB7, 0x10, 0x1A, 0x82, 0xE0, 0x08.

YH (1 byte): The high-ordered byte of the 2-byte year from the **PidLidExceptionReplaceTime** property (section 2.2.10.2.5) if the object represents an exception; otherwise, zero.

YL (1 byte): The low-ordered byte of the 2-byte year from the **PidLidExceptionReplaceTime** property if the object represents an exception; otherwise, zero.

M (1 byte): The month from the **PidLidExceptionReplaceTime** property if the object represents an exception; otherwise, zero. If it represents an exception, the value MUST be one of those listed in the following table.

Value	Meaning
0x01	January
0x02	February
0x03	March
0x04	April
0x05	May
0x06	June
0x07	July
0x08	August
0x09	September
0x0A	October
0x0B	November
0x0C	December

D (1 byte): The day of the month from the **PidLidExceptionReplaceTime** property if the object represents an exception; otherwise, zero.

Creation Time (8 bytes): A **FILETIME** structure ([MS-DTYP]) that specifies the date and time when this Global Object ID was generated.

X (8 bytes): Reserved, MUST be all zeros.

Size (4 bytes): This field specifies the size of the **Data** field.

Data (variable): An array of bytes that ensures the uniqueness of the Global Object ID among all Calendar objects in all **mailboxes**.

2.2.1.28 PidLidCleanGlobalObjectId Property

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

Contains the value of the **PidLidGlobalObjectId** property (section 2.2.1.27) for an object that represents an **Exception object** to a recurring series, where the Year, Month, and Day fields are all zero.

The format of the **PidLidCleanGlobalObjectId** property ([MS-OXPROPS] section 2.57) is the same as that of the **PidLidGlobalObjectId** property. The value of this property MUST be equal to the value of **PidLidGlobalObjectId**, except the **YH**, **YL**, **M**, and **D** fields MUST all be zero. All objects that refer to an instance of a recurring series (including an orphan instance), as well as the recurring series itself, will have the same value for this property.

2.2.1.29 PidTagOwnerAppointmentId Property

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

The **PidTagOwnerAppointmentId** property ([MS-OXPROPS] section 2.845) specifies a quasi-unique value among all Calendar objects in a user's mailbox. The value of this property can assist a client or server in finding a Calendar object but is not guaranteed to be unique among all objects.<9> This property is not required on objects.

2.2.1.30 PidTagStartDate Property

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

The **PidTagStartDate** property ([MS-OXPROPS] section 2.1013) SHOULD be set, and when set, it MUST be equal to the value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5).

2.2.1.31 PidTagEndDate Property

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

The **PidTagEndDate** property ([MS-OXPROPS] section 2.673) SHOULD be set, and when set, it MUST be equal to the value of the **PidLidAppointmentEndWhole** property (section 2.2.1.6).

2.2.1.32 PidLidCommonStart Property

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

The **PidLidCommonStart** property ([MS-OXPROPS] section 2.63) represents the start date and time of an event.

The value of this property MUST be equal to the value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5).

2.2.1.33 PidLidCommonEnd Property

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

The **PidLidCommonEnd** property ([MS-OXPROPS] section 2.62) represents the end date and time of an event.

The value of this property MUST be equal to the value of the **PidLidAppointmentEndWhole** property (section 2.2.1.6).

2.2.1.34 PidLidOwnerCriticalChange Property

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

The **PidLidOwnerCriticalChange** property ([MS-OXPROPS] section 2.199) specifies the date and time at which a Meeting Request object was sent by the organizer. The value is specified in UTC.

2.2.1.35 PidLidIsException Property

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

A value of TRUE for the **PidLidIsException** property ([MS-OXPROPS] section 2.155) indicates that the object represents an exception (including an orphan instance). A value of FALSE indicates that the object represents a recurring series or a **single-instance object**. The absence of this property for any object indicates a value of FALSE except for the **Exception Embedded Message object**, which assumes a value of TRUE.

2.2.1.36 PidTagResponseRequested Property

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

When the value of the **PidTagResponseRequested** property ([MS-OXOMSG] section 2.2.1.46) is FALSE, Meeting Response objects are not sent to the organizer. When the value of this property is TRUE and the client or server automatically responds (for more details, see sections 2.2.12.2, 2.2.12.3, and 2.2.12.4), a Meeting Response object is sent to the organizer. Otherwise, when the value is TRUE, the client or server can send a Meeting Response object but SHOULD prompt the user first to verify that the user wants a response sent.

2.2.1.37 PidTagReplyRequested Property

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

The **PidTagReplyRequested** property ([MS-OXPROPS] section 2.906) specifies whether the organizer requests a reply from the attendees.

For Calendar objects, this property MUST have the same value as the **PidTagResponseRequested** property ([MS-OXOMSG] section 2.2.1.46).

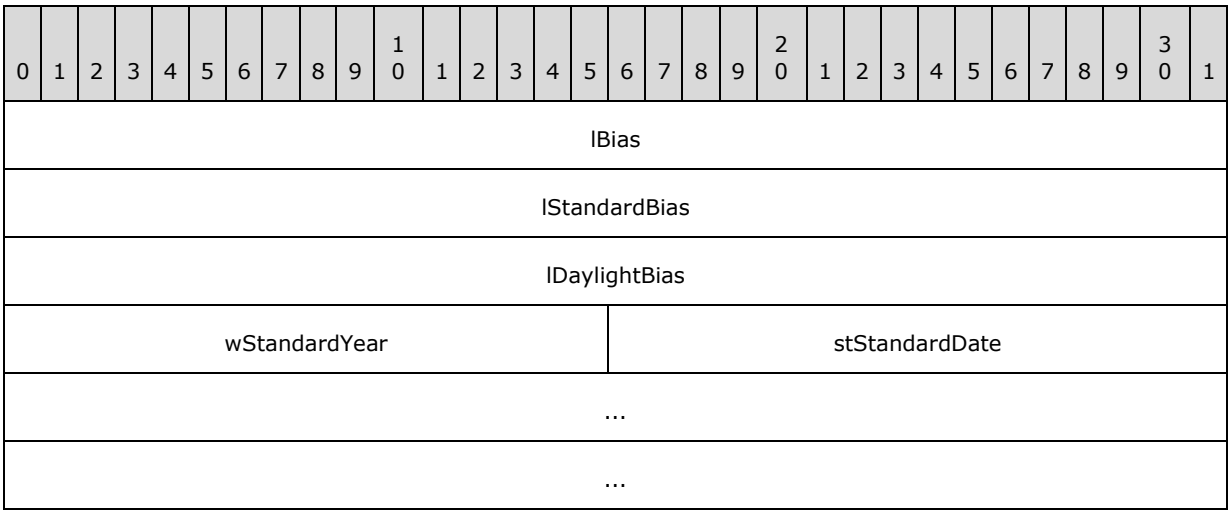
2.2.1.38 Best Body Properties

Best body properties, when stored or retrieved, are specified by using the best body algorithm, as specified in [MS-OXBBODY] section 2.1.3.1, and contain the contents of the Calendar objects or meeting-related objects. For transport, objects that are specified by the Appointment and Meeting Object Protocol SHOULD use the **Rich Text Format (RTF)** compression format, as specified in [MS-OXRTFCP] section 2.1.3.1.

2.2.1.39 PidLidTimeZoneStruct Property

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

The **PidLidTimeZoneStruct** property ([MS-OXPROPS] section 2.342) is set on a recurring series to specify time zone information. This property specifies how to convert time fields between local time and UTC. The fields in this BLOB are encoded in little-endian byte order.



...	
...	wDaylightYear
stDaylightDate	
...	
...	
...	

lBias (4 bytes): The time zone's offset in minutes from UTC.

lStandardBias (4 bytes): The offset in minutes from the value of the **lBias** field during standard time.

lDaylightBias (4 bytes): The offset in minutes from the value of the **lBias** field during daylight saving time.

wStandardYear (2 bytes): This field matches the **stStandardDate's wYear** member.

stStandardDate (16 bytes): A **SYSTEMTIME** structure, as specified in [MS-DTYP]. This field contains the date and local time that indicate when to begin using the value specified in the **lStandardBias** field.

If the time zone does not support daylight saving time, the **wMonth** member in the **SYSTEMTIME** structure MUST be zero (0). If the **wYear** member is not zero (0), the date is interpreted as an absolute date that only occurs once. If the **wYear** member is zero (0), the date is interpreted as a relative date that occurs yearly. The **wHour** and **wMinute** members are set to the transition time; the **wDayOfWeek** member is set to the appropriate weekday, and the **wDay** member is set to indicate the occurrence of the day of the week within the month (1 to 5, where 5 indicates the final occurrence during the month if that day of the week does not occur 5 times).

wDaylightYear (2 bytes): This field is equal to the value of the **stDaylightDate's wYear** field.

stDaylightDate (16 bytes): A **SYSTEMTIME** structure. This field contains the date and local time that indicate when to begin using the value specified in the **lDaylightBias** field. This field has the same format and constraints as the **stStandardDate** field.

2.2.1.40 PidLidTimeZoneDescription Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidTimeZoneDescription** property ([MS-OXPROPS] section 2.341) specifies a human-readable description of the time zone that is represented by the data in the **PidLidTimeZoneStruct** property (section 2.2.1.39).

2.2.1.41 PidLidAppointmentTimeZoneDefinitionRecur Property

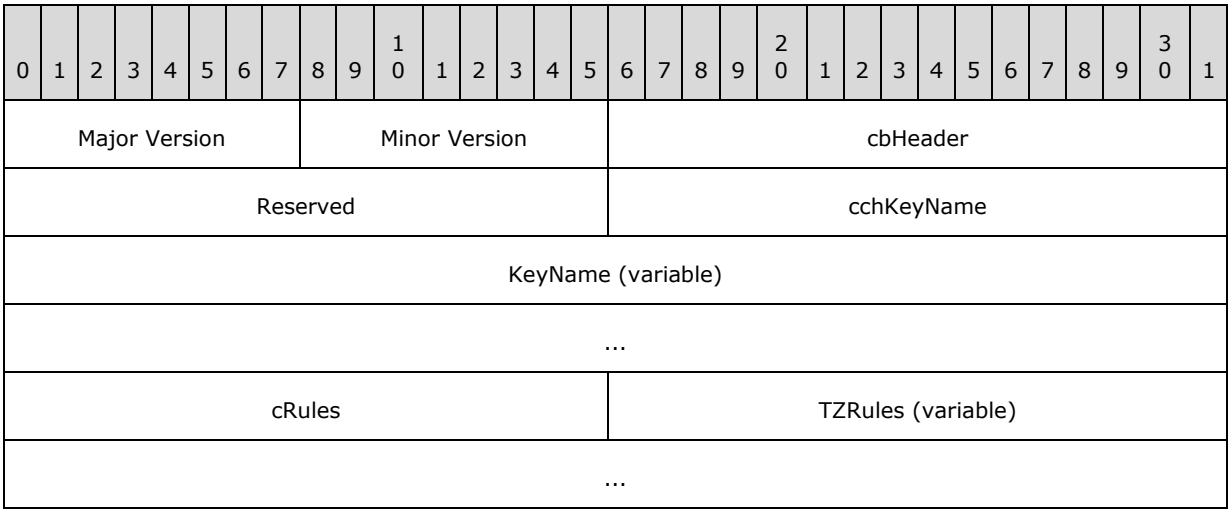
Type: **PtypBinary** ([MS-OXCDATA] section 2.11.1)

The **PidLidAppointmentTimeZoneDefinitionRecur** property ([MS-OXPROPS] section 2.33) contains time zone information that specifies how to convert the meeting date and time on a recurring series to and from UTC.

The **PidLidAppointmentTimeZoneDefinitionRecur** property contains one **TZRule** structure, as specified in section 2.2.1.41.1, that is marked with the **TZRULE_FLAG_EFFECTIVE_TZREG** flag, which specifies the effective time zone **rule**.

If the effective **TZRule** structure's **lBias**, **lStandardBias**, **lDaylightBias**, **stStandardDate**, and **stDaylightDate** fields are not equal to the corresponding fields in the **PidLidTimeZoneStruct** property (section 2.2.1.39), the **PidLidAppointmentTimeZoneDefinitionRecur** and **PidLidTimeZoneStruct** properties are considered inconsistent. If the **PidLidAppointmentTimeZoneDefinitionRecur** property is not set or is inconsistent with the associated **PidLidTimeZoneStruct** structure, the values in the **PidLidTimeZoneStruct** property are used to determine the effective time zone rule.<10>

The fields in this structure are encoded in little-endian byte order.



Major Version (1 byte): This field is set to 0x02.

Minor Version (1 byte): This field is set to 0x01.

cbHeader (2 bytes): An integer that specifies the number of bytes contained in the **Reserved, cchKeyName, KeyName**, and **cRules** fields.

Reserved (2 bytes): This field MUST be set to 0x0002.

cchKeyName (2 bytes): An integer that specifies the number of characters in the **KeyName** field.

KeyName (variable): A **Unicode** string that identifies the associated time zone. The string is not localized but instead is set to the unique name of the desired time zone. This string has a maximum length of 260 characters, and it is not null-terminated.

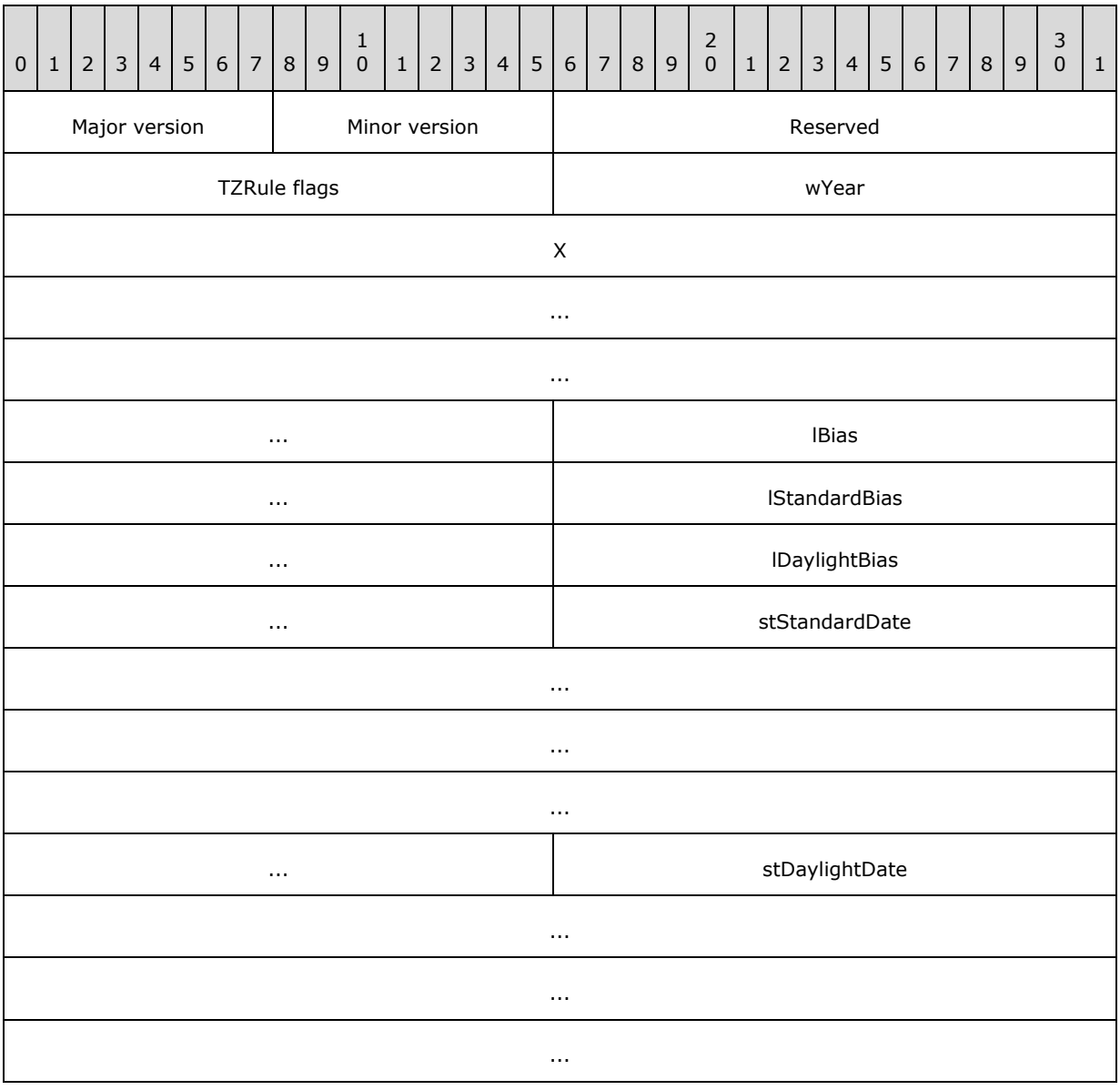
cRules (2 bytes): An integer that specifies the number of **TZRule** structures in the **TZRules** field. The minimum is 1; the maximum is 1024.

TZRules (variable): An array of **TZRule** structures as specified in section 2.2.1.41.1. Each **TZRule** structure contains information that specifies a time zone, including the time zone's offset from UTC and when and how it observes daylight saving time. If more than one time zone rule (4) is specified, rules (4) are sorted in ascending order by the **wYear** field. The **TZRule** structures are not aligned to 32-bit boundaries. Each **TZRule** structure starts at the next byte after the previous **TZRule** structure ends. For details on the **TZRule** structure, see section 2.2.1.41.1 (represented in little-endian byte order).

2.2.1.41.1 TZRule Structure

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

Each **TZRule** structure is represented as shown in the following diagram.



...

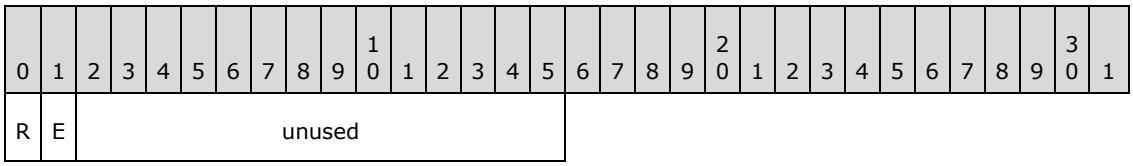
Major version (1 byte): This field is set to 0x02.

Minor version (1 byte): This field is set to 0x01.

Reserved (2 bytes): This field MUST be set to 0x003E.

TZRule flags (2 bytes): This field contains individual bit flags that specify information about this **TZRule** structure, represented here in little-endian byte order.

If the **TZRULE** property associated with this **TZRULE flags** field specifies the time zone rule (4) that will be used to convert to and from UTC, both of the flags specified in the following diagram are set (for example, the value is 0x0003). If this is not the active time zone rule (4), neither of these flags are set. These flags are set on exactly one **TZRule** structure that is contained in the associated property, and the flags for all other rules (4) MUST be set to zero (0).



R - TZRULE_FLAG_RECUR_CURRENT_TZREG (1 bit): This flag specifies that this rule (4) is associated with a recurring series.

E - TZRULE_FLAG_EFFECTIVE_TZREG (1 bit): This flag specifies that this rule (4) is the effective rule (4).

unused (14 bits): These bits are not used. MUST be zero and MUST be ignored.

wYear (2 bytes): A property that specifies the year in which this rule (4) is scheduled to take effect. A rule (4) will remain in effect from January 1 of its **wYear** field value until January 1 of the next rule's (4) **wYear** field value. If no rules (4) exist for subsequent years, this rule (4) will remain in effect indefinitely.

X (14 bytes): This field is unused; MUST be all zeros.

lBias (4 bytes): This field specifies the time zone's offset in minutes from UTC.

lStandardBias (4 bytes): This field specifies the offset in minutes from the value stored in the **lBias** field during standard time.

lDaylightBias (4 bytes): This field specifies the offset in minutes from **lBias** during daylight saving time.

stStandardDate (16 bytes): A **SYSTEMTIME** structure ([MS-DTYP]). This field contains the date and local time to begin using the value of the **lStandardBias** field.

If the time zone does not support daylight saving time, the **wMonth** field in the **SYSTEMTIME** structure MUST be zero. If the **wYear** field in the **SYSTEMTIME** structure is not zero, the date is interpreted as an absolute date that only occurs once. If the **wYear** field is zero, the date is interpreted as a relative date that occurs yearly. The **wHour** and **wMinute** fields are set to the transition time, the **wDayOfWeek** field is set to the appropriate weekday, and the **wDay** field is set to indicate the occurrence of the day of the

week within the month (1 to 5, where 5 indicates the final occurrence during the month if that day of the week does not occur 5 times).

stDaylightDate (16 bytes): A **SYSTEMTIME** structure containing the date and local time that specifies when to begin using the value stored in the **lDaylightBias** field. This property has the same format and constraints as the **stStandardDate** field.

2.2.1.42 PidLidAppointmentTimeZoneDefinitionStartDisplay Property

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

The **PidLidAppointmentTimeZoneDefinitionStartDisplay** property ([MS-OXPROPS] section 2.34) specifies time zone information that indicates the time zone of the **PidLidAppointmentStartWhole** property (section 2.2.1.5). The value of this property is used to convert the start date and time from UTC to this time zone for display purposes. The fields in this BLOB are encoded as specified in section 2.2.1.41, with one exception: For each **TZRule** structure specified by this property, the **R** flag in the **TZRule flags** field is not set (for example, if the **TZRule** structure is the effective rule (4), the value of the field **TZRule flags** field is **TZRULE_FLAG_EFFECTIVE_TZREG** (0x0002); otherwise, it will be 0x0000).

2.2.1.43 PidLidAppointmentTimeZoneDefinitionEndDisplay Property

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

The **PidLidAppointmentTimeZoneDefinitionEndDisplay** property ([MS-OXPROPS] section 2.32) specifies time zone information that indicates the time zone of the **PidLidAppointmentEndWhole** property (section 2.2.1.6). The format, constraints, and computation of this property are the same as specified for the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property (section 2.2.1.42).

2.2.1.44 PidLidAppointmentRecur Property

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

The **PidLidAppointmentRecur** property ([MS-OXPROPS] section 2.22) specifies the dates and times when a recurring series occurs by using one of the recurrence patterns and ranges specified in this section. The value of this property also contains information about both modified and deleted exceptions and information such as dates, subject, location, and other properties of exceptions. The binary data in this property for **Recurring Calendar objects** is stored as the **AppointmentRecurrencePattern** structure specified in section 2.2.1.44.5. This property MUST NOT exist on single-instance Calendar objects.

The following are some limitations of recurrences:

- Multiple instances cannot start on the same day.
- Occurrences cannot overlap; specifically, an exception that modifies the start date of an instance in the recurring series can occur only on a date that is sometime after the end of the prior instance and before the start of the next instance in the recurring series. The same is true if the prior or next instance in the recurring series is an exception. Note that calculating exception overlap is an implementation-specific choice.<11>

The schedule of a recurring series is determined by its recurrence pattern and range. This section specifies the types of **recurrence ranges** and recurrence patterns that are supported by this protocol.

Recurrence Range

The recurrence range identifies how long the event will continue. This protocol supports three ranges:

- Ends after a specific number of occurrences
- Ends by a given date
- Continues indefinitely

Recurrence Pattern

The recurrence pattern determines the frequency of an event. The **RecurrencePattern** structure (section 2.2.1.44.1) is also used to define **recurring tasks**, as specified in [MS-OXOTASK] section 2.2.2.2.15.

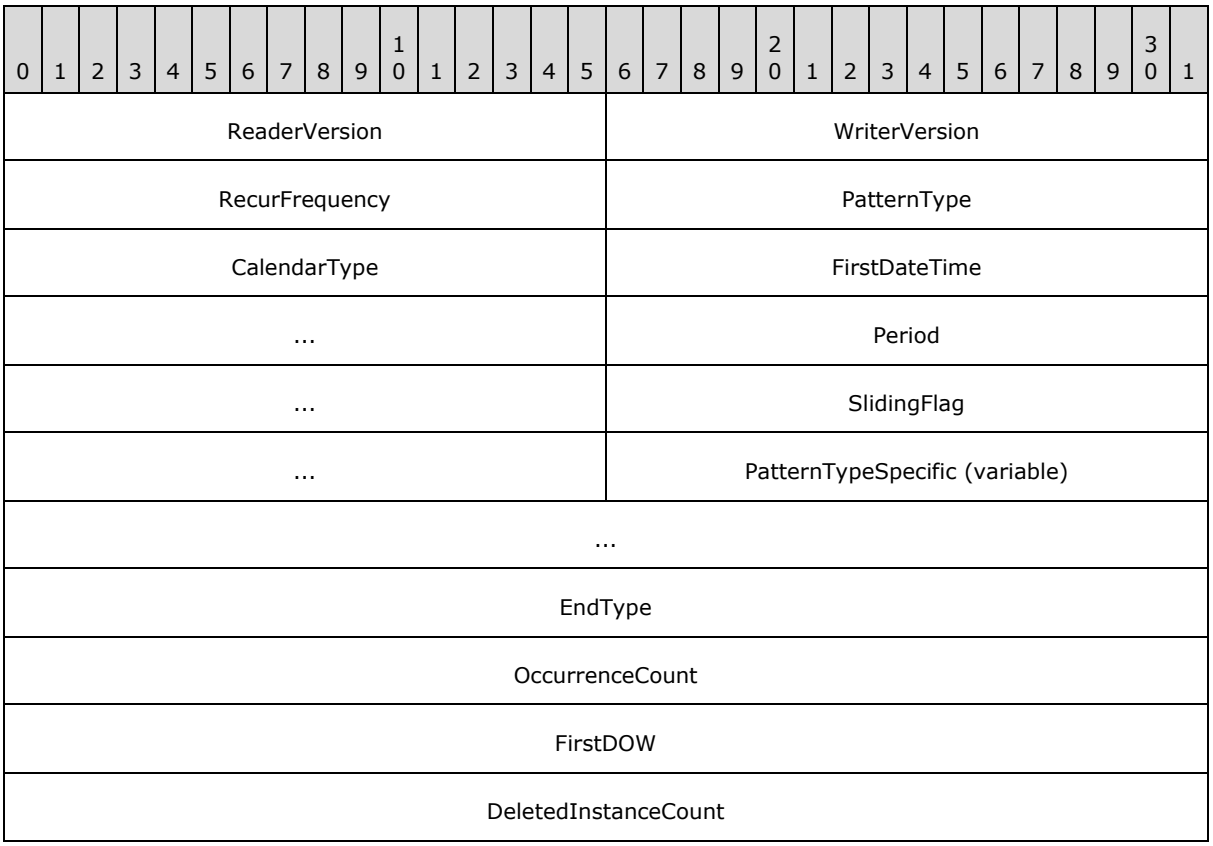
The following table lists the types of recurrences that are supported by this protocol.

Recurrence type	Description	Example
Daily recurrence	Schedules events according to one of the following patterns: ▪ Every *n* number of days. ▪ Every weekday.	An event that repeats every three days, starting on Monday, April 30, 2007, and continuing through Friday, June 8, 2007.
Weekly recurrence	Schedules events according to the following pattern: ▪ Every *n* weeks on one or more particular days of the week.	An event repeats every two weeks, on Tuesdays, starting on Monday, April 30, 2007, and ending after five occurrences.
Monthly recurrence	Schedules events according to one of the following patterns: ▪ On the *n*th day of every month. ▪ On a specific day of the week on the first, second, third, fourth, or last week of every month (for example, the first Tuesday of the month).	An event that repeats on the fourth of every month, effective Monday, April 30, 2007, without an end date.
Every *n* months recurrence	A combination of the monthly and weekly patterns. An every *n* months pattern can schedule events according to one of the following patterns: ▪ On the *m*th day every *n* months. ▪ On any day of the week on the first, second, third, fourth, or last week every *n* months (for example, the third Thursday of the month).	An event that occurs on the last Thursday of every two months, effective March 12, 2007, with an end date of December 31, 2007.
Month end	Schedules events to repeat on the last day of	An event that repeats on the last day

Recurrence type	Description	Example
recurrence	every *n* months.	of every month, effective Monday, April 30, 2007, without an end date.
Yearly recurrence	Schedules events according to one of the following patterns: ▪ On the *m*th day of the *n*th month, of every year. ▪ On any day of the week on the first, second, third, fourth, or last week of the *n*th month, of every year. The yearly recurrence pattern is based on a 12-month interval, and therefore uses the monthly recurrence parameters to represent all the yearly recurrences.	A birthday that occurs every June 22, and is an all-day event.

2.2.1.44.1 RecurrencePattern Structure

The **RecurrencePattern** structure specifies a recurrence pattern. The fields of this structure are stored in little-endian byte order.



DeletedInstanceDates (variable)
...
ModifiedInstanceCount
ModifiedInstanceDates (variable)
...
StartDate
EndDate

ReaderVersion (2 bytes): This field MUST be set to 0x3004.

WriterVersion (2 bytes): This field MUST be set to 0x3004.

RecurFrequency (2 bytes): An integer that specifies the frequency of the recurring series. Valid values are listed in the following table.

RecurFrequency	Value
Daily	0x200A
Weekly	0x200B
Monthly	0x200C
Yearly	0x200D

PatternType (2 bytes): An integer that specifies the type of recurrence pattern. The valid recurrence pattern types are listed in the following table.

Name	Value	Meaning
Day	0x0000	The event has a daily recurrence.
Week	0x0001	The event has a weekly recurrence.
Month	0x0002	The event has a monthly recurrence.
MonthEnd	0x0004	The event has a month-end recurrence.<12>
MonthNth	0x0003	The event has an every *n*th month pattern.
HjMonth	0x000A	The event has a monthly recurrence in the Hijri calendar. For this value in the **PatternType** field, the value of the **CalendarType** field MUST be set to 0x0000.<13>
HjMonthNth	0x000B	The event has an every nth month pattern in the Hijri calendar. For this value in the **PatternType** field, the value of the **CalendarType** field MUST be set to 0x0000.

Name	Value	Meaning
HjMonthEnd	0x000C	The event has a month end recurrence in the Hijri calendar. For this value in the **PatternType** field, the value of the **CalendarType** field MUST be set to 0x0000.

CalendarType (2 bytes): An integer that specifies the type of calendar that is used. The acceptable values for the calendar type are listed in the following table.

Name	Value	Meaning
Default	0x0000	The default value for the calendar type is Gregorian. If the value of the **PatternType** field is HjMonth, HjMonthNth, or HjMonthEnd and the value of the **CalendarType** field is Default, this recurrence uses the Hijri calendar.
CAL_GREGORIAN	0x0001	Gregorian (localized) calendar
CAL_GREGORIAN_US	0x0002	Gregorian (U.S.) calendar
CAL_JAPAN	0x0003	Japanese Emperor era calendar
CAL_TAIWAN	0x0004	Taiwan calendar
CAL_KOREA	0x0005	Korean Tangun era calendar
CAL_HIJRI	0x0006	Hijri (Arabic Lunar) calendar
CAL_THAI	0x0007	Thai calendar
CAL_HEBREW	0x0008	Hebrew lunar calendar
CAL_GREGORIAN_ME_FRENCH	0x0009	Gregorian Middle East French calendar
CAL_GREGORIAN_ARABIC	0x000A	Gregorian Arabic calendar
CAL_GREGORIAN_XLIT_ENGLISH	0x000B	Gregorian transliterated English calendar
CAL_GREGORIAN_XLIT_FRENCH	0x000C	Gregorian transliterated French calendar
CAL_LUNAR_JAPANESE	0x000E	Japanese lunar calendar
CAL_CHINESE_LUNAR	0x000F	Chinese lunar calendar
CAL_SAKA	0x0010	Saka era calendar
CAL_LUNAR_ETO_CHN	0x0011	Lunar ETO Chinese calendar
CAL_LUNAR_ETO_KOR	0x0012	Lunar ETO Korean calendar
CAL_LUNAR_ROKUYOU	0x0013	Lunar Rokuyou calendar
CAL_LUNAR_KOREAN	0x0014	Korean lunar calendar
CAL_UMALQURA	0x0017	Um Al Qura calendar

FirstDateTime (4 bytes): An integer that specifies the first ever day, week, or month of a recurring series, dating back to a reference date, which is January 1, 1601, for a Gregorian

calendar. The value and its meaning depend on the value of the **RecurFrequency** field. The value of the **FirstDateTime** field is used to determine the valid dates of a recurring series, as specified in section 2.2.1.44.1.2.

The value and meaning of the **FirstDateTime** field for each type of recurrence frequency are specified in the following table. For details about how the value is calculated, see section 2.2.1.44.1.1.

Frequency	Value and Meaning
Daily	The number of minutes between midnight January 1, 1601, and the first ever day on which the event would occur.
Weekly	The number of minutes between midnight January 1, 1601, and the first day of the first ever week in which the event would occur.
Monthly or Yearly	The number of minutes between midnight January 1, 1601, and the first day of the first ever month in which the event would occur.

Period (4 bytes): An integer that specifies the interval at which the meeting pattern specified in **PatternTypeSpecific** field repeats. The **Period** value MUST be between 1 and the maximum recurrence interval, which is 999 days for daily recurrences, 99 weeks for weekly recurrences, and 99 months for monthly recurrences. The following table lists the values for this field based on the recurrence frequency, which is specified in the **RecurFrequency** field.

Frequency	Value
Daily recurrence	The period is stored as the minutes in whole number of days. For example, to define a recurrence that occurs every two days, the **Period** field is set to 0x00000B40, which equals 2880 minutes, or two days.
Weekly recurrence	The period is stored in weeks. For example, if the **Period** field is set to 0x00000002, the meeting occurs every two weeks.
Monthly or yearly recurrence	The period is stored in months. If the recurrence is a yearly recurrence, The **Period** field MUST be set to 12.

SlidingFlag (4 bytes): This field is only used for scheduling tasks; otherwise the value MUST be zero (0). For more details about sliding tasks, see [MS-OXOTASK] section 3.1.4.6.2.

PatternTypeSpecific (variable): A structure that specifies the details of the recurrence pattern. The structure varies according to the value of the **PatternType** field, as specified in sections 2.2.1.44.1.3, 2.2.1.44.1.4, 2.2.1.44.1.5, and 2.2.1.44.1.6.

EndType (4 bytes): An integer that specifies the ending type for the recurrence. This field MUST be set to one of the values listed in the following table.

Recurrence range type	Value
End after date	0x00002021
End after *N* occurrences	0x00002022
Never end	SHOULD be 0x00002023 but can be 0xFFFFFFFF

OccurrenceCount (4 bytes): An integer that specifies the number of occurrences in a recurrence.

When the **EndType** of the pattern is "End after date", this value always has to be computed. Although the value of this field is always set, its value has no meaning on a recurring series that has no end date. This value can be set to 0x0000000A for a recurring series with no end date.<14>

FirstDOW (4 bytes): An integer that specifies the day on which the calendar week begins. The default value is Sunday (0x00000000). This field MUST be set to one of the values listed in the following table.

Day	Value
Sunday	0x00000000
Monday	0x00000001
Tuesday	0x00000002
Wednesday	0x00000003
Thursday	0x00000004
Friday	0x00000005
Saturday	0x00000006

DeletedInstanceCount (4 bytes): An integer that specifies the number of elements in the **DeletedInstanceDates** field.

DeletedInstanceDates (variable): An array of dates, each of which is the original instance date of either a deleted instance or a modified instance for this recurrence. The number of dates contained in this array is specified by the **DeletedInstanceCount** field. Each date is stored as the number of minutes between midnight, January 1, 1601, and midnight of the specified day, in the time zone specified by the **PidLidTimeZoneStruct** property (section 2.2.1.39). The dates are ordered from earliest to latest.

The array contains exactly one element for each deleted instance, and every deleted instance is represented. The array also contains an entry for every modified instance. The array SHOULD NOT contain duplicate entries. Deleted instances for which there is no corresponding value in the **ModifiedInstanceDates** field imply that they have been completely removed from the pattern.

ModifiedInstanceCount (4 bytes): An integer that specifies the number of elements in the **ModifiedInstanceDates** field. The value of this field MUST be less than or equal to the value of the **DeletedInstanceCount** field.

ModifiedInstanceDates (variable): An array of dates, each of which is the date of a modified instance. The number of dates contained in this array is specified by the **ModifiedInstanceCount** field. Each date is stored as the number of minutes between midnight, January 1, 1601, and midnight of the specified day, in the time zone specified by the **PidLidTimeZoneStruct** property (section 2.2.1.39). The dates are ordered from earliest to latest.

The array contains exactly one element for each modified instance, and every modified instance is represented. The array SHOULD NOT contain duplicate entries. Every modified

instance also has to have an entry in the **DeletedInstanceDates** field with the original instance date.

StartDate (4 bytes): An integer that specifies the date of the first occurrence. The value is the number of minutes between midnight, January 1, 1601, and midnight of the date of the first occurrence.

EndDate (4 bytes): An integer that specifies the ending date for the recurrence. The value is the number of minutes between midnight, January 1, 1601, and midnight of the date of the last occurrence. When the value of the **EndType** field is 0x00002022 (end after *n* occurrences), this value is calculated based on the number of occurrences If the recurrence does not have an end date, the value of the **EndDate** field MUST be set to 0x5AE980DF.

2.2.1.44.1.1 Calculating the Value of the FirstDateTime Field

This section provides details about how to calculate the value of the **FirstDateTime** field of the **RecurrencePattern** structure, specified in section 2.2.1.44.1, for a daily, weekly, monthly, or yearly recurring series. The value of **FirstDateTime** is in minutes measured from a reference time, which is midnight January 1, 1601 for a Gregorian calendar, to the date that is being specified.

The values of the **StartDate**, **Period**, and **FirstDOW** fields of the **RecurrencePattern** structure are used in the calculations.

DAILY

For a daily recurring series, the value of the **FirstDateTime** field is equal to the value of the **StartDate** field modulo the value of the **Period** field. The value of the **FirstDateTime** field for a daily recurring series will always be a value between 0 (zero) and (period – 1440). The calculation of the value is summarized by the following formula:

```
FirstDateTime = StartDate % Period
```

WEEKLY

For a weekly recurring series, the value of the **FirstDateTime** field is calculated as follows:

1. Determine the date of the beginning of the week that contains the date indicated by the **StartDate** field. In other words, this date is that of the first day of the week in which the first event of the recurring series occurs. This date depends on the value of the **FirstDOW** field, which specifies the day that begins a calendar week. This dependency is illustrated in the figure and example following this procedure.
2. Calculate the number of minutes between midnight, January 1, 1601, and midnight of the date that was determined in step 1.
3. Calculate the number of minutes between recurrences. This value is the number of minutes per week, 10080, multiplied by the value of the **Period** field.
4. The value of the **FirstDateTime** field is equal to the result from step 2 modulo the result from step 3.

The calculation of the value of the **FirstDateTime** field is summarized by the following formula:

```
FirstDateTime = (result from step 2) % (Period * 10080)
```

The following figure and example show how the date of the first event week, which is determined in step 1, depends on the value of the **FirstDOW** field.

Calendar A

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14

Calendar B

Wed	Thu	Fri	Sat	Sun	Mon	Tue
4	5	6	7	8	9	10
11	12	13	14	15	16	17

Figure 1: Date of the event week depends on the day that begins a calendar week

If the value of the **FirstDOW** field indicates that weeks begin on Sunday (0x00000000) and the value of the **StartDate** field indicates April 12, 2007, which falls on Thursday, as the date of the first event of the recurring series, then the date of the first event week is April 8, 2007, as shown in Calendar A in the preceding figure. However, if the value of the **FirstDOW** field indicates that weeks begin on Wednesday (0x00000003), the date of the first event week is April 11, 2007, as shown in Calendar B in the preceding figure.

MONTHLY or YEARLY

For a monthly recurring series or a yearly recurring series, the value of the **FirstDateTime** field is calculated as follows:

1. Determine the date of the beginning of the month that contains the date indicated by the **StartDate** field. In other words, this date is that of the first day of the month containing the first event of the recurring series.

2. Calculate the number of calendar months between midnight of January 1, 1601, and midnight of the date that was determined in step 1.

3. Calculate the offset of months within the year of 1601. This value is equal to the result from step 2 modulo the value of the **Period** field. The calculation is summarized by the following formula:

```
offset of months within the year of 1601 = (result from step 2) % Period
```

4. Calculate the number corresponding to the month that contains the first event within the year of 1601 by adding one to the result from step 3. The calculation of this value is summarized by the following formula:

```
month within the year of 1601 = ((result from step 2) % Period) + 1
```

5. The value of the **FirstDateTime** field is equal the number of minutes between midnight of the first day of the month that was determined in step 4 and midnight January 1, 1601.

2.2.1.44.1.2 Finding Valid Recurrence Dates

The process of finding valid dates for any recurring series involves the calculation of an offset, which is the value to be subtracted from an input value (day, week, or month) to find a valid value. An input value is valid if it is congruent with the date specified by the **FirstDateTime** field. An offset of 0 indicates that the input value is congruent and, therefore, is valid for the recurring series.

A week or a month is specified by the date on which the week or month begins. A valid week or a valid month contains the event dates. For weekly and monthly recurring series, the offset is used to determine only valid weeks or valid months. The value of the **PatternTypeSpecific** field of the **RecurrencePattern** structure, specified in section 2.2.1.44.1, is used to determine which days within the valid week or the valid month are valid event dates.

The actual formulas used to determine valid days, weeks, or months, depend on the type of recurrence pattern (daily, weekly, or monthly/yearly) being applied, but they generally involve the following variables:

- *InputValue*: The day, week, or month to be examined.

  For daily/weekly recurrences, the input value is in minutes, measured from the reference time, which is midnight January 1, 1601, for a Gregorian calendar.

  For monthly recurrences, the input value is in months, measured from the reference time.

- *FirstDateTime*: The value of the **FirstDateTime** field, as specified in section 2.2.1.44.1.

- *Period*: The value of the **Period** field, as specified in section 2.2.1.44.1.

Daily Recurrence

For a daily recurrence pattern, a valid day needs to be determined. A valid day is any day on which the event occurs. The offset for a daily recurrence pattern is the number of minutes to be subtracted from an input day to find a valid day and is calculated as follows.

```
offset = (InputValue - FirstDateTime) % Period
```

where *InputValue* is the day to be examined, expressed as the number of minutes elapsed from midnight January 1, 1601, to that day.

The offset will vary depending on the period (every *x* days). The minimum period is 1 day, so the offset will always be a multiple of 1440 (number of minutes in a day) or zero if the input day is valid. If the input day is not valid, the offset is used as follows to find a valid day.

```
Previous valid day = InputValue - offset

Next valid day = InputValue - offset + Period
```

For example, given the following days (in minutes, measured from the reference time, which is midnight January 1, 1601, for a Gregorian calendar), and a recurrence pattern that starts on Day 1.

Day 0 = 0

Day 1 = 1440 minutes

Day 2 = 2880 minutes

Day 3 = 4320 minutes

...

It can be seen that an "Every 1 day" (period is 1440 * 1 = 1440) recurrence pattern is uninteresting; the offset will always be 0 (zero), which indicates that every day is a valid instance. Now consider an "Every 3 days" recurrence pattern (period is 1440 * 3 = 4320) with Day 4 (5760

minutes) as the start date. In this case, the value of the **FirstDateTime** field is 1440 and valid instances are 4, 7, 10, 13, and so forth. If Day 9 (12960) is the input day, applying the formula results in the following evaluation, which indicates whether Day 9 is a valid instance for this recurrence pattern.

```
offset = (12960 – 1440) % 4320
       = 2880
```

The offset is not zero, so Day 9 is not a valid instance. The offset of 2880 minutes, or 2 days, is used to make the adjustment. Substituting the values into the formulas results in the following evaluations.

```
Previous valid day = 12960 – 2880
                   = 10080

Next valid day = 12960 – 2880 + 4320
               = 14400
```

The resulting values indicate that Day 7 (10080 minutes) is the previous valid instance and Day 10 (14400 minutes) is the next valid instance.

Weekly Recurrence

For a weekly recurrence pattern, a valid week needs to be determined and then the value of the **PatternTypeSpecific** field, as specified in section 2.2.1.44.1.4, is used to determine the valid day or days within that week. A valid week is any week in which the event occurs. The offset for a weekly recurrence pattern is the number of minutes to be subtracted from an input week to find a valid week and is calculated as follows:

```
offset = (InputValue – FirstDateTime) % (Period * 10080)
```

where *InputValue* is the week to be examined, expressed as the number of minutes elapsed from midnight January 1, 1601, to the beginning of that week. Note that the date of the beginning of the week depends on which of the seven days of the week is designated to be the start of a calendar week. The day on which a calendar week begins is specified by the **FirstDOW** field with Sunday (0x00000000) being the default.

The offset will vary depending on the period (every *x* weeks). The minimum period is 1 week, so the offset will always be a multiple of 10080 (number of minutes in a week) or zero if the input week is valid. If the input week is not valid, the offset is used as follows to find a valid week:

```
Previous valid week = InputValue – offset

Next valid week = InputValue – offset + (Period * 10080)
```

For example, consider the recurrence pattern "Every 3 weeks" with Thursday, February 8, 1601, being the start date and Sunday being the beginning of a calendar week. In this case, the value of the **FirstDateTime** field is 18720 and valid weeks are the weeks of February 4, February 25, March 18, April 8, and so forth. If the input week is the week of March 11, which is Day 69 (99360 minutes), applying the formula results in the following evaluation, which indicates whether the week of March 11 is a valid week.

```
offset = (99360 – 18720) % (3 * 10080)
       = 20160
```

The offset is not zero, so the week of March 11 is not a valid week. The offset of 20160 minutes, or 2 weeks, is used to make the adjustment. Substituting the values into the formulas results in the following evaluations:

```
Previous valid week = 99360 – 20160
                    = 79200

Next valid week = 99360 – 20160 + (3 * 10080)
                = 109440
```

The resulting values indicate that the week of February 25 (79200 minutes, Day 55) is the previous valid week and the week of March 18 (14400 minutes, Day 76) is the next valid week. To find the valid event date within the valid week, advance to Thursday, March 1, for the week of February 25; advance to Thursday, March 22, for the week of March 18.

For a more complex recurrence pattern, additional steps might be needed to find the event dates, but the formulas for determining the offset, the previous valid week, and the next valid week are the same.

Monthly or Yearly Recurrence

For a monthly or yearly recurrence pattern, a valid month needs to be determined and then the value of the **PatternTypeSpecific** field, as specified in section 2.2.1.44.1.5, is used to determine the valid day or days within that month. A valid month is any month in which the event occurs. The offset for monthly/yearly recurrence pattern is the number of months to be subtracted from an input month to find a valid month and is calculated as follows:

```
offset = (InputValue – FDTmonth) % Period
```

where *InputValue* is the month to be examined, expressed as the number of months elapsed from midnight January 1, 1601, to the beginning of that month and *FDTmonth* is the number of months elapsed from midnight January 1, 1601, to the date specified by the **FirstDateTime** field.

The offset will vary depending on the period (every *x* months). The minimum period is 1 month and the period is always 12 for yearly recurrences. A yearly recurrence pattern is just a monthly pattern that occurs every 12 months. The offset is zero if the input month is valid. If the input month is not valid, the offset is used as follows to find a valid month:

```
Previous valid month = InputValue – offset

Next valid month = InputValue – offset + Period
```

For example, consider the recurrence pattern "Every 5 months on the 19th of the month" with April 19, 2008, being the start date. In this case, the value of the **FirstDateTime** field is 84960, corresponding to March 1, 1601 (the very first valid month for this recurrence pattern, dating back to January 1, 1601). There are 2 months between midnight January 1, 1601, and March 1, 1601, so the value of *FTDmonth* is 2. The valid months are April and September of 2008, February, July, and December of 2009, and so forth. If the input month is November 2009, which is 4906 months from

midnight January 1, 1601, applying the formula results in the following evaluation, which indicates whether November 2009 is a valid month.

```
offset = (4906 – 2) % 5
       = 4
```

The offset is not zero, so November 2009 is not a valid month. The offset of 4 months is used to make the adjustment. Substituting the values into the formulas results in the following evaluations:

```
Previous valid month = 4906 – 4
                     = 4902

Next valid month = 4906 – 4 + 5
                 = 4907
```

The resulting values indicate that July 2009 is the previous valid month and December 2009 is the next valid month. To find the valid event date within the valid month, advance to 19th of the month.

For a more complex recurrence pattern, additional steps might be needed to find the valid event dates, but the formulas for determining the offset, the previous valid month, and the next valid month are the same. A non-Gregorian calendar presents additional challenges, such as leap months.

2.2.1.44.1.3 PatternTypeSpecific Day

For a daily recurrence pattern (value of the **PatternType** field is 0x0000), the **PatternTypeSpecific** field has no value and is zero bytes. In other words, the value of the **PatternTypeSpecific** field is not included in the BLOB when the value of the **PatternType** field is 0x0000.

2.2.1.44.1.4 PatternTypeSpecific Week

For a weekly recurrence pattern (value of the **PatternType** field is 0x0001), the structure of the **PatternTypeSpecific** field is as follows.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
X	Sa	F	Th	W	Tu	M	Su	unused																							

X (1 bit): This bit is not used. MUST be zero and MUST be ignored.

Sa (1 bit): (0x00000040) The event occurs on Saturday.

F (1 bit): (0x00000020) The event occurs on Friday.

Th (1 bit): (0x00000010) The event occurs on Thursday.

W (1 bit): (0x00000008) The event occurs on Wednesday.

Tu (1 bit): (0x00000004) The event occurs on Tuesday.

M (1 bit): (0x00000002) The event occurs on Monday.

Su (1 bit): (0x00000001) The event occurs on Sunday.

unused (3 bytes): These bits are not used. MUST be zero and MUST be ignored.

The day on which a calendar week begins is used in combination with the value of the **PatternTypeSpecific** field to determine the event dates. For example, consider the pattern "Every 2 weeks on Monday, Tuesday, and Friday, starting in week 2." If Wednesday is designated as the beginning of the calendar week according to the value of the **FirstDOW** field of the **RecurrencePattern** structure, the Monday, Tuesday, and Friday event dates in a given week are not the same as they would be if the calendar week begins on Sunday. The following figure might make this a little bit easier to understand.

	Calendar A						
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Week 1:	1	2	3	4	5	6	7
Week 2:	8	9	10	11	12	13	14
Week 3:	15	16	17	18	19	20	21
Week 4:	22	23	24	25	26	27	28

Calendar B						
Wed	Thu	Fri	Sat	Sun	Mon	Tue
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

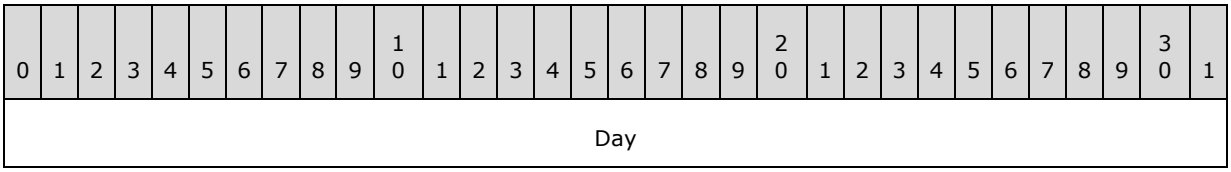
Figure 2: Event dates depend on the first day of the week

Assuming a pattern "Every 2 weeks on Monday, Tuesday, and Friday, starting in week 2", Calendar A in the preceding figure shows Monday, Tuesday, and Friday event dates when Sunday is designated as the beginning of a calendar week, and Calendar B shows the event dates when Wednesday is designated as the beginning of a calendar week.

If the calendar week begins on Sunday, the valid dates would be the 9th, 10th, 13th, 23rd, 24th, and 27th of the month, but if the calendar week begins on Wednesday, the valid dates would be the 13th, 16th, 17th, 27th, 30th, and 31st of the month. The day on which calendar weeks begin makes a huge difference. When applying the weekly recurrence pattern, all days within the pattern need to be in the same week.

2.2.1.44.1.5 PatternTypeSpecific Month

The value is little-endian byte order. For a Month, MonthEnd, HjMonth, or HjMonthEnd recurrence pattern (value of the **PatternType** field is 0x002, 0x004, 0x00A, or 0x00C, respectively), the structure of the **PatternTypeSpecific** field is as follows.



0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
Day																															

Day (4 bytes): The day of the month on which the recurrence falls.

2.2.1.44.1.6 PatternTypeSpecific MonthNth

For the MonthNth or HjMonthNth recurrence pattern (value of the **PatternType** field is 0x0003 or 0x000B, respectively), the structure of the **PatternTypeSpecific** field is as follows.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
X	Sa	F	Th	W	Tu	M	Su	unused																							
N																															

X (1 bit): These bits are not used. MUST be zero and MUST be ignored.

Sa (1 bit): The event occurs on Saturday.

F (1 bit): The event occurs on Friday.

Th (1 bit): The event occurs on Thursday.

W (1 bit): The event occurs on Wednesday.

Tu (1 bit): The event occurs on Tuesday.

M (1 bit): The event occurs on Monday.

Su (1 bit): The event occurs on Sunday.

unused (3 bytes): These bits are not used. MUST be zero and MUST be ignored.

Nth Weekday of month: (bits M, Tu, W, Th, F are set)

Nth Weekend of month: (bits Sa, Su are set)

N (4 bytes): The occurrence of the recurrence's days in each month in which the recurrence falls. It MUST be equal to one of the values listed in the following table.

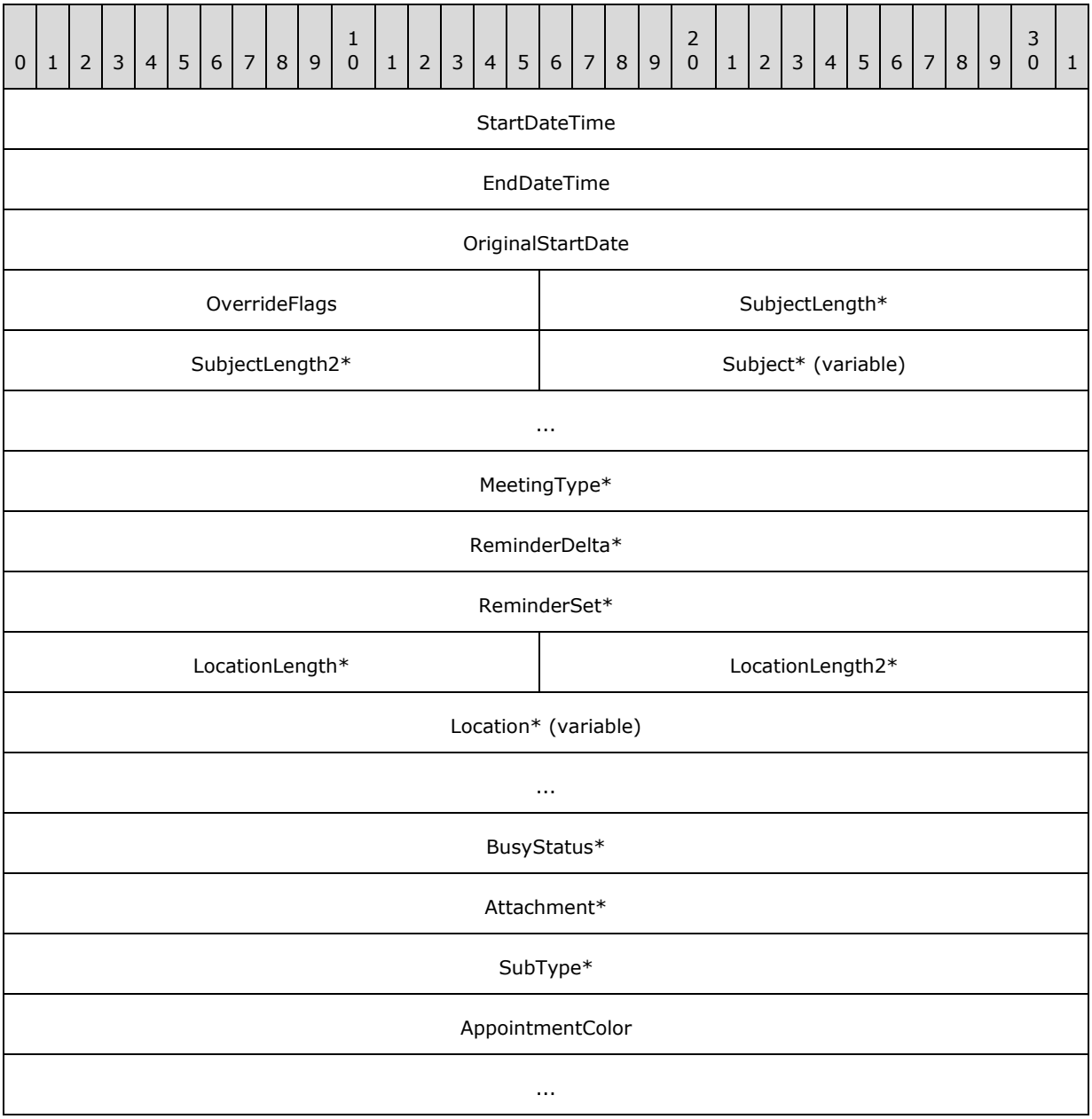
Name	Value	Meaning
First	0x00000001	The recurrence falls on the first occurrence of the days specified in every month.
Second	0x00000002	The recurrence falls on the second occurrence of the days specified in every month.
Third	0x00000003	The recurrence falls on the third occurrence of the days specified in every month.
Fourth	0x00000004	The recurrence falls on the fourth occurrence of the days specified in every month.
Last	0x00000005	The recurrence falls on the last occurrence of the days specified in every month.

For example:

- If an event occurs on the last weekday of every two months, the two fields of the **PatternTypeSpecific** field are set to 0x0000003E and 0x00000005.

- If an event occurs on the first weekday of every two months, the two fields of the **PatternTypeSpecific** field are set to 0x0000003E and 0x00000001.
- If an event occurs on the last weekend of every month, the two fields of the **PatternTypeSpecific** field are set to 0x00000041and 0x00000005.
- If an event occurs on the first weekend of every month, the two fields of the **PatternTypeSpecific** field are set to 0x00000041 and 0x00000001.

2.2.1.44.2 ExceptionInfo Structure



StartDateTime (4 bytes): The start time of the exception in local time in minutes since midnight, January 1, 1601.

EndDateTime (4 bytes): The end time of the exception in local time in minutes since midnight, January 1, 1601.

OriginalStartDate (4 bytes): The original starting time of the exception in local time in minutes since midnight, January 1, 1601.

OverrideFlags (2 bytes): A bit field that specifies what data in the **ExceptionInfo** structure has a value different from the recurring series. The valid flags for this field are summarized in the following table.

Flag	Value	Meaning
ARO_SUBJECT	0x0001	Indicates that the **Subject**, **SubjectLength**, and **SubjectLength2** fields are present.
ARO_MEETINGTYPE	0x0002	Indicates that the **MeetingType** field is present.
ARO_REMINDERDELTA	0x0004	Indicates that the **ReminderDelta** field is present.
ARO_REMINDER	0x0008	Indicates that the **ReminderSet** field is present.
ARO_LOCATION	0x0010	Indicates that the **Location**, **LocationLength**, and **LocationLength2** fields are present.
ARO_BUSYSTATUS	0x0020	Indicates that the **BusyStatus** field is present.
ARO_ATTACHMENT	0x0040	Indicates that the attachment field is present.
ARO_SUBTYPE	0x0080	Indicates that the **SubType** field is present.
ARO_APPTCOLOR	0x0100	Indicates that the **AppointmentColor** field is present.
ARO_EXCEPTIONAL_BODY	0x0200	Indicates that the Exception Embedded Message object has the **PidTagRtfCompressed** property ([MS-OXCMSG] section 2.2.1.56.4) set on it.

SubjectLength* (2 bytes): The value of this field is equal to the number of bytes of the **Subject** field plus 1. *This field is present only if the **ARO_SUBJECT** flag is set in the **OverrideFlags** field.

SubjectLength2* (2 bytes): The number of bytes of the **Subject** field. *This field is present only when the **ARO_SUBJECT** flag is set in the **OverrideFlags** field.

Subject* (variable): A non-null-terminated, non-Unicode string that is the value of the **PidTagNormalizedSubject** property ([MS-OXCMSG] section 2.2.1.10) in the Exception Embedded Message object. *This field is present only when the **ARO_SUBJECT** flag is set in the **OverrideFlags** field.

MeetingType* (4 bytes): The value of the **PidLidAppointmentStateFlags** property (section 2.2.1.10) in the Exception Embedded Message object. *This field is present only when the **ARO_MEETINGTYPE** flag is set in the **OverrideFlags** field.

ReminderDelta* (4 bytes): The value for the **PidLidReminderDelta** property ([MS-OXORMDR] section 2.2.1.3) in the Exception Embedded Message object. *This field is present only when the **ARO_REMINDERDELTA** flag is set in the **OverrideFlags** field.

ReminderSet* (4 bytes): The value for the **PidLidReminderSet** property ([MS-OXORMDR] section 2.2.1.1) in the Exception Embedded Message object. *This field is present only when the **ARO_REMINDER** flag is set in the **OverrideFlags** field.

LocationLength* (2 bytes): The number of bytes of the **Location** field plus 1. *This field is present only when the **ARO_LOCATION** flag is set in the **OverrideFlags** field.

LocationLength2* (2 bytes): The number of bytes of the **Location** field. *This field is present only when the **ARO_LOCATION** flag is set in the **OverrideFlags** field.

Location* (variable): A non-Unicode string that is the value of the **PidLidLocation** property (section 2.2.1.4) in the Exception Embedded Message object. *This field is present only when the **ARO_LOCATION** flag is set in the **OverrideFlags** field.

BusyStatus* (4 bytes): The value for the **PidLidBusyStatus** property (section 2.2.1.2) in the Exception Embedded Message object. For possible values, see section 2.2.1.2. *This field is present only when the **ARO_BUSYSTATUS** flag is set in the **OverrideFlags** field.

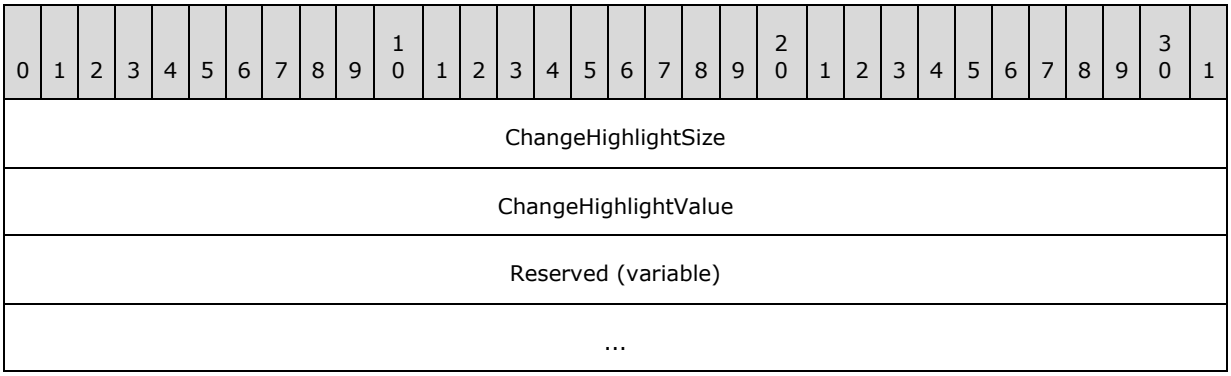
Attachment* (4 bytes): The value of this field specifies whether the Exception Embedded Message object contains attachments. The value will be 0x00000001 if attachments are present, and 0x00000000 otherwise. *This field is present only when the **ARO_ATTACHMENTS** flag is set in the **OverrideFlags** field.

SubType* (4 bytes): The value for the **PidLidAppointmentSubType** property (section 2.2.1.9) in the Exception Embedded Message object. For possible values, see section 2.2.1.9. *This field is present only when the **ARO_SUBTYPE** flag is set in the **OverrideFlags** field.

AppointmentColor (4 bytes): The value for the **PidLidAppointmentColor** property (section 2.2.1.50) in the Exception Embedded Message object. For possible values, see section 2.2.1.50. *This field is present only when the **ARO_APPTCOLOR** flag is set in the **OverrideFlags** field.

2.2.1.44.3 ChangeHighlight Structure

The **ChangeHighlight** structure is present only when the value of the **WriterVersion2** field in the associated **AppointmenRecurrencePattern** structure specified in section 2.2.1.44.5 is greater than or equal to 0x00003009.



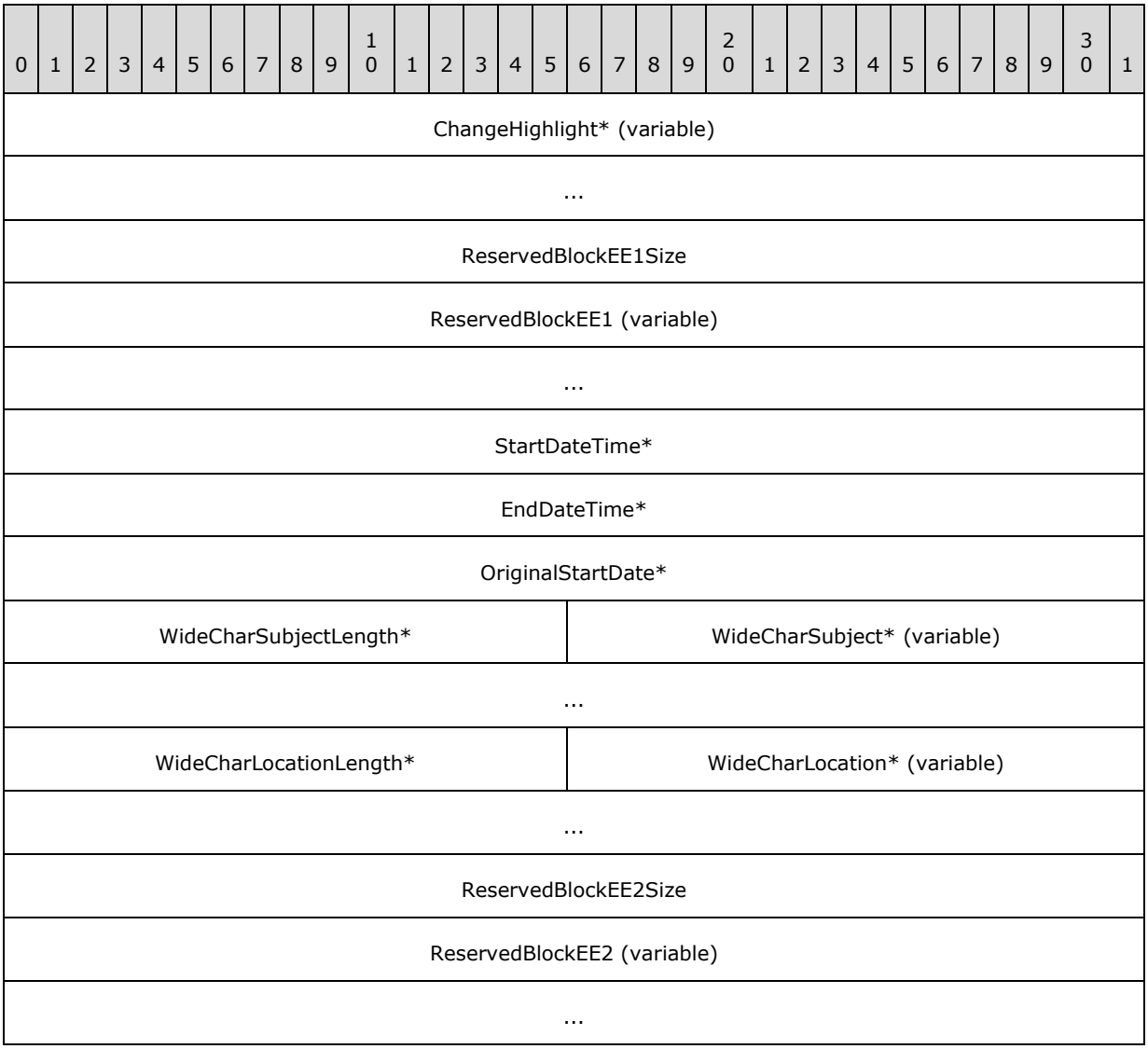
ChangeHighlightSize (4 bytes): The size of the **ChangeHighlightValue** and **Reserved** fields combined.

ChangeHighlightValue (4 bytes): The value of the **PidLidChangeHighlight** property (section 2.2.6.2) in the Exception Embedded Message object.

Reserved (variable): Reserved.<15> This field is reserved for future enhancements and is not used. This field is not read or written to.

2.2.1.44.4 ExtendedException Structure

There is one **ExtendedException** structure per **ExceptionInfo** structure, as specified in section 2.2.1.44.2, and each one MUST be in the same order as its corresponding **ExceptionInfo** structure.



ChangeHighlight* (variable): The value of the **PidLidChangeHighlight** property (section 2.2.6.2) in the Exception object. *This field is present only when the **WriterVersion2** field in the associated **AppointmentRecurrencePattern** structure specified in section 2.2.1.44.5 is greater than or equal to 0x00003009.

ReservedBlockEE1Size (4 bytes): The size of the **ReservedBlockEE1** field. This field MUST be set to zero.

ReservedBlockEE1 (variable): This field is reserved.

StartDateTime* (4 bytes): The start time of the exception in local time in minutes since midnight, January 1, 1601. *This field is present only when the **ARO_SUBJECT** flag or the **ARO_LOCATION** flag is set in the **OverrideFlags** field of the **ExtendedException** structure's associated **ExceptionInfo** structure. For details, see the description of the **OverrideFlags** field in section 2.2.1.44.2.

EndDateTime* (4 bytes): The end time of the exception in local time in minutes since midnight, January 1, 1601. *This field is present only when the **ARO_SUBJECT** flag or the **ARO_LOCATION** flag is set in the **OverrideFlags** field of the **ExtendedException** structure's associated **ExceptionInfo** structure. For details, see the description of the **OverrideFlags** field in section 2.2.1.44.2.

OriginalStartDate* (4 bytes): The original start date of the exception in local time in minutes since midnight, January 1, 1601. *This field is present only when the **ARO_SUBJECT** flag or the **ARO_LOCATION** flag is set in the **OverrideFlags** field of the **ExtendedException** structure's associated **ExceptionInfo** structure. For details, see the description of the **OverrideFlags** field in section 2.2.1.44.2.

WideCharSubjectLength* (2 bytes): The count of Unicode characters in the **WideCharSubject** field. *This field is present only when the **ARO_SUBJECT** flag is set in the **OverrideFlags** field of the **ExtendedException** structure's associated **ExceptionInfo** structure. For details, see the description of the **OverrideFlags** field in section 2.2.1.44.2.

WideCharSubject* (variable): The Unicode string value for the exception's **PidTagNormalizedSubject** property ([MS-OXCMSG] section 2.2.1.10). Note that the **WideCharSubject** field is not null-terminated. *This field is present only when the **ARO_SUBJECT** flag is set in the **OverrideFlags** field of the **ExtendedException** structure's associated **ExceptionInfo** structure. For details, see the description of the **OverrideFlags** field in section 2.2.1.44.2.

WideCharLocationLength* (2 bytes): The number of Unicode characters in the **WideCharLocation** field. *This field is present only when the **ARO_LOCATION** flag is set in the **OverrideFlags** field of the **ExtendedException** structure's associated **ExceptionInfo** structure. For details, see the description of the **OverrideFlags** field in section 2.2.1.44.2.

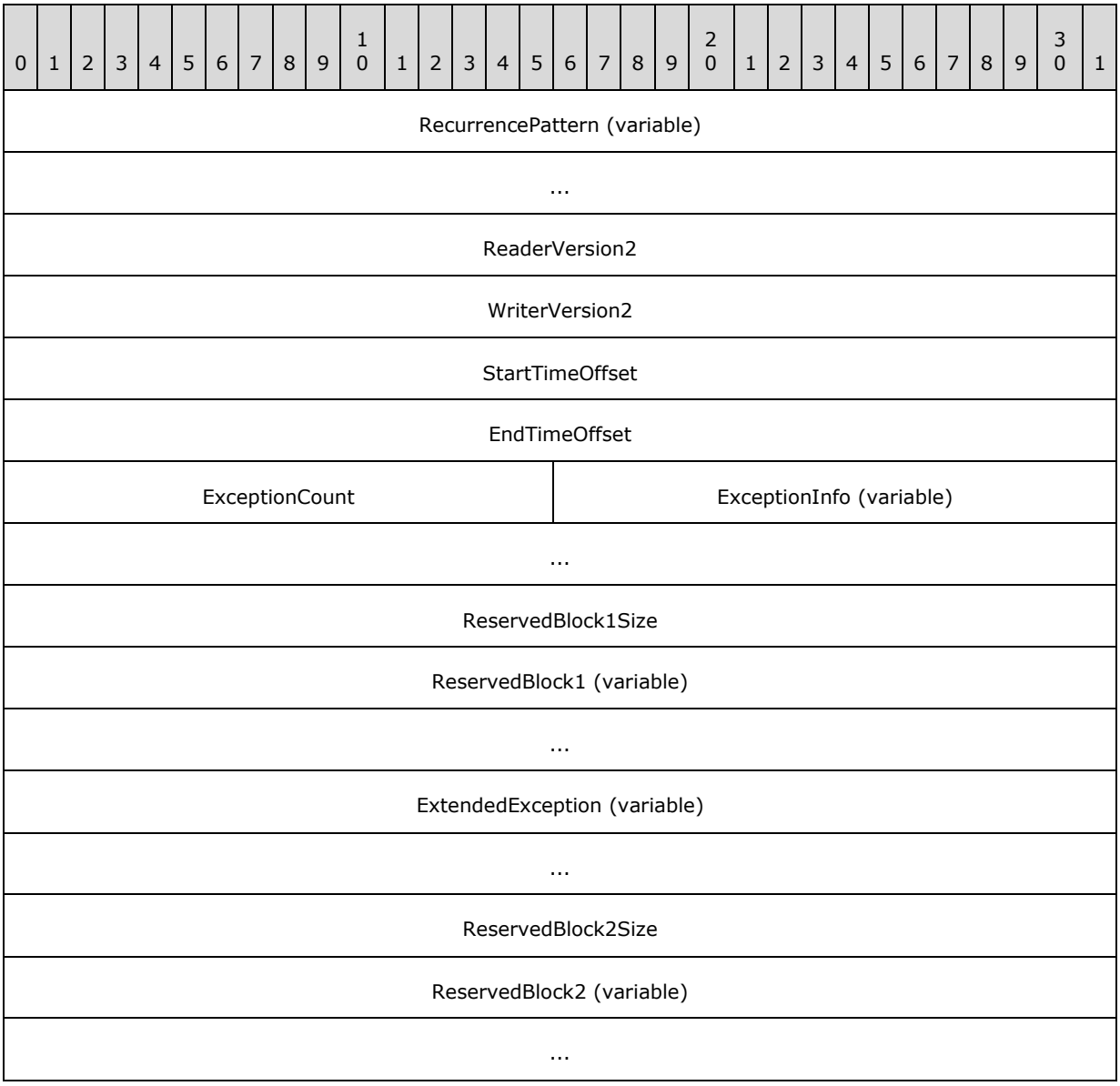
WideCharLocation* (variable): The Unicode string value for the **PidLidLocation** property (section 2.2.1.4) in the Exception Embedded Message object. Note that the **WideCharLocation** field is not null-terminated. *This field is present only when the **ARO_LOCATION** flag is set in the **OverrideFlags** field of the **ExtendedException** structure's associated **ExceptionInfo** structure. For details, see the description of the **OverrideFlags** field in section 2.2.1.44.2.

ReservedBlockEE2Size (4 bytes): The size of the **ReservedBlockEE2** field that follows. *This field is present only when the **ARO_SUBJECT** flag or the **ARO_LOCATION** flag is set in the **OverrideFlags** field of the **ExtendedException** structure's associated **ExceptionInfo** structure. For details, see the description of the **OverrideFlags** field in section 2.2.1.44.2.

ReservedBlockEE2 (variable): Reserved. This field MUST NOT be read from or written to. *This field is present only when the **ARO_SUBJECT** flag or the **ARO_LOCATION** flag is set in the **OverrideFlags** field of the **ExtendedException** structure's associated **ExceptionInfo** structure. For details, see the description of the **OverrideFlags** field in section 2.2.1.44.2.

2.2.1.44.5 AppointmentRecurrencePattern Structure

The **AppointmentRecurrencePattern** structure specifies a recurrence pattern for a Calendar object, including information about exception property values. The fields of this structure are stored in little-endian byte order.



RecurrencePattern (variable): This field is a **RecurrencePattern** structure, as specified in section 2.2.1.44.1, that defines the recurrences.

ReaderVersion2 (4 bytes): This value MUST be set to 0x00003006.

WriterVersion2 (4 bytes): This value SHOULD<16> be set to 0x00003009 but can be set to 0x00003008. The value of this field affects the format of the **ExtendedException** field, as specified in section 2.2.1.44.4.

StartTimeOffset (4 bytes): The number of minutes, since midnight, after which each occurrence starts. For example, the value for midnight is 0 (zero) and the value for 12:00 P.M. is 720.

EndTimeOffset (4 bytes): The number of minutes, since midnight, after which each occurrence ends. For example, the value for midnight is 0 (zero) and the value for 12:00 P.M. is 720.

ExceptionCount (2 bytes): An integer that specifies the number of **ExceptionInfo** structures contained in the **ExceptionInfo** field and the number of **ExtendedException** structures contained in the **ExtendedException** field. The value of this field MUST be the same as the value of the **ModifiedInstanceCount** field in the associated **ReccurencePattern** structure, as specified in section 2.2.1.44.1.

ExceptionInfo (variable): An array of **ExceptionInfo** structures. The number of structures in this field is specified by the **ExceptionCount** field. For details about the **ExceptionInfo** structure, see section 2.2.1.44.2.

ReservedBlock1Size (4 bytes): The size, in bytes, of the **ReservedBlock1** field. MUST be zero.

ReservedBlock1 (variable): This field is reserved and MUST be zero-length.

ExtendedException (variable): An array of **ExtendedException** structures. The number of structures in this field is specified by the **ExceptionCount** field. For details about the **ExtendedException** structure, see section 2.2.1.44.4.

ReservedBlock2Size (4 bytes): The size, in bytes, of the **ReservedBlock2** field. MUST be zero.

ReservedBlock2 (variable): This field is reserved and MUST be zero-length.

2.2.1.45 PidLidRecurrenceType Property

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

The **PidLidRecurrenceType** property ([MS-OXPROPS] section 2.215) specifies the recurrence type of the recurring series by using one of the values listed in the following table.

Recurrence type	Value	Meaning
rectypeNone	0x00000000	A single-instance appointment
rectypeDaily	0x00000001	A daily recurrence pattern
rectypeWeekly	0x00000002	A weekly recurrence pattern
rectypeMonthly	0x00000003	A monthly recurrence pattern
rectypeYearly	0x00000004	A yearly recurrence pattern

The use of the **PidLidRecurrenceType** property is optional.

2.2.1.46 PidLidRecurrencePattern Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidRecurrencePattern** property ([MS-OXPROPS] section 2.214) specifies a description of the recurrence pattern of the Calendar object. This property is not required, but if set, it is set to a description of the recurrence specified by the **PidLidAppointmentRecur** property (section 2.2.1.44).

2.2.1.47 PidLidLinkedTaskItems Property

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

The **PidLidLinkedTaskItems** property ([MS-OXPROPS] section 2.158) specifies a list of **PidTagEntryId** properties ([MS-OXCPERM] section 2.2.4) of **Task objects** related to the Calendar object that are set by a client.<17> This property is not required.

2.2.1.48 PidLidMeetingWorkspaceUrl Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidMeetingWorkspaceUrl** property ([MS-OXPROPS] section 2.171) specifies the **URL** of the **Meeting Workspace**, as specified in [MS-MEETS], that is associated with a Calendar object. This property is not required.

2.2.1.49 PidTagIconIndex Property

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

The value of the **PidTagIconIndex** property ([MS-OXPROPS] section 2.726) indicates that an icon is used with the object. It SHOULD be set to one of the values in the following table. A value of -1 means that the property is not set to a specific value and it is up to the client to determine the correct icon to display for this item.

Icon	Value	Used by
Single-instance appointment	0x00000400	Appointment object
Recurring appointment	0x00000401	Appointment object
Single-instance meeting	0x00000402	Meeting object
Recurring meeting	0x00000403	Meeting object
Meeting request/full update	0x00000404	Meeting Request object, Meeting Update object
Accept meeting request	0x00000405	Meeting Response object
Decline meeting request	0x00000406	Meeting Response object
Tentatively accept meeting request	0x00000407	Meeting Response object
Meeting cancellation	0x00000408	Meeting Cancellation object
Meeting update/Informational update	0x00000409	Meeting Update object
Forward notification	0x0000040B	Meeting Forward Notification object

2.2.1.50 PidLidAppointmentColor Property

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

The **PidLidAppointmentColor** property ([MS-OXPROPS] section 2.9) specifies the color to be used when displaying a Calendar object. The client SHOULD set this property for backward compatibility with older clients. The valid values for this property are listed in the following table.

Value	Color
0x00000000	None
0x00000001	Red
0x00000002	Blue
0x00000003	Green
0x00000004	Grey
0x00000005	Orange
0x00000006	Cyan
0x00000007	Olive
0x00000008	Purple
0x00000009	Teal
0x0000000A	Yellow

2.2.1.51 Deprecated Properties

The properties defined in sections 2.2.1.51.1 through 2.2.1.51.9 are deprecated. If nonzero or non-NULL, clients SHOULD set their value to zero or an empty string (as appropriate).<18> If the **PidLidConferencingCheck** property (section 2.2.1.51.2) is set to FALSE, all the properties in this section are ignored. These properties are to be set only on Calendar objects and meeting-related objects.

2.2.1.51.1 PidLidAutoStartCheck Property

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

The **PidLidAutoStartCheck** property ([MS-OXPROPS] section 2.41) specifies whether to automatically start the conferencing application when a **reminder** for the meeting fires.

If the associated object is a Calendar object, the client SHOULD set the **PidLidAutoStartCheck** property to FALSE.

When set to TRUE, this property indicates that the conferencing application can start. A value of FALSE indicates that either this property doesn't apply to its associated object or the conferencing application is not to start automatically.

2.2.1.51.2 PidLidConferencingCheck Property

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

When set to TRUE (0x00000001), the **PidLidConferencingCheck** property ([MS-OXPROPS] section 2.65) indicates that the associated meeting is one of the following types:

- "Windows Media Services"

- "Windows NetMeeting"
- "Exchange Conferencing"

If this property is set, **PidLidConferencingType** (section 2.2.1.51.3) is also to be set. This property is set to TRUE only on Meeting objects or meeting-related objects.

2.2.1.51.3 PidLidConferencingType Property

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

The **PidLidConferencingType** ([MS-OXPROPS] section 2.66) property specifies the type of the meeting. The value of this property MUST be set to one of the values listed in the following table.

Type of meeting	Value
Windows Netmeeting	0x00000000
Windows Media Services	0x00000001
Exchange Conferencing	0x00000002

2.2.1.51.4 PidLidDirectory Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidDirectory** property ([MS-OXPROPS] section 2.94) specifies the directory server to be used with NetMeeting.

2.2.1.51.5 PidLidAllowExternalCheck Property

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

The **PidLidAllowExternalCheck** property ([MS-OXPROPS] section 2.6) is deprecated.<19> The value of this property MAY<20> be set to TRUE.

2.2.1.51.6 PidLidOrganizerAlias Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidOrganizerAlias** property ([MS-OXPROPS] section 2.195) specifies the e-mail address of the organizer.

2.2.1.51.7 PidLidCollaborateDoc Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidCollaborateDoc** property ([MS-OXPROPS] section 2.61) specifies the document to be launched when the user joins the meeting. This property is valid only when the **PidLidConferencingType** property (section 2.2.1.51.3) has the value 0x00000000.

2.2.1.51.8 PidLidNetShowUrl Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidNetShowUrl** property ([MS-OXPROPS] section 2.175) specifies the URL to be launched when the user joins the meeting. This property is valid only when the **PidLidConferencingType** property (section 2.2.1.51.3) has the value 0x00000001 or 0x00000002.

For meetings that have 0x00000001 as the value of the **PidLidConferencingType** property, the URL is supplied by a user. For meetings that have 0x00000002 as the value of the **PidLidConferencingType** property, the URL is generated as follows:

- For each **blind carbon copy (Bcc) recipient** of a Meeting Request object, open the associated folder of the Calendar folder in the recipient's (1) mailbox.
- Find the message for which the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) has a value of **EXCH_CONFERENCE**. If the message is not found, move on to the next Bcc recipient. If the message is found, open it and get its **PidTagLocation** property ([MS-OXOCNTC] section 2.2.1.10.5).
- Append the value of the **PidLidGlobalObjectId** property (section 2.2.1.27) of the Meeting object encoded with **base64 encoding**.
- Append the string "&p=" followed by the value of the **PidLidOnlinePassword** property (section 2.2.1.51.9).
- Finally, convert the string to Unicode.

If there are multiple Exchange Conferencing mailboxes in the **Bcc** field, the value that is calculated by using the last mailbox is used.

2.2.1.51.9 PidLidOnlinePassword Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidOnlinePassword** property ([MS-OXPROPS] section 2.193) specifies the password for a meeting on which the **PidLidConferencingType** property (section 2.2.1.51.3) has the value 0x00000002. If set, this string is a maximum of 255 characters, not including the terminating null character.

2.2.2 Calendar Object

Properties that are specific to Calendar objects, (which include Appointment objects and Meeting objects) are specified in sections 2.2.2.1 through 2.2.2.4. Unless otherwise specified, these properties are to always exist. Note that Calendar objects can also have the following reminder-related properties, as specified in [MS-OXORMDR]:

- **PidLidReminderSet** ([MS-OXORMDR] section 2.2.1.1)
- **PidLidReminderSignalTime** ([MS-OXORMDR] section 2.2.1.2)
- **PidLidReminderDelta** ([MS-OXORMDR] section 2.2.1.3)
- **PidLidReminderTime** ([MS-OXORMDR] section 2.2.1.4)
- **PidLidReminderOverride** ([MS-OXORMDR] section 2.2.1.5)
- **PidLidReminderPlaySound** ([MS-OXORMDR] section 2.2.1.6)
- **PidLidReminderFileParameter** ([MS-OXORMDR] section 2.2.1.7).

2.2.2.1 PidTagMessageClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) MUST be "IPM.Appointment" or be prefixed with "IPM.Appointment.".

2.2.2.2 PidLidSideEffects Property

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

The possible flag values of the **PidLidSideEffects** property are specified in [MS-OXCMSG] section 2.2.1.16. All Calendar objects SHOULD<21> include the following flags:

- **seOpenToDelete**
- **seOpenToCopy**
- **seOpenToMove**
- **seCoerceToInbox**
- **seOpenForCtxMenu**

2.2.2.3 PidLidFExceptionalAttendees Property

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

A value of TRUE for the **PidLidFExceptionalAttendees** property ([MS-OXPROPS] section 2.130) indicates that it is a Recurring Calendar object with one or more exceptions and that at least one of the Exception Embedded Message objects has at least one **RecipientRow** structure, as specified in [MS-OXCDATA] section 2.8.3. A value of FALSE or the absence of this property indicates that either the Calendar object has no exceptions or none of the Exception Embedded Message objects has **RecipientRow** structures.

Note that an Appointment object cannot have attendees but can have an **Attachment object** that is an Exception object, and an Exception object can be turned into a meeting that has attendees. Therefore, this property can be used to indicate that an exception to an appointment has attendees, even though the appointment (series) does not.

This value SHOULD NOT be set for any Calendar object other than that of the organizer's.

2.2.2.4 PidLidClientIntent Property

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

The **PidLidClientIntent** property ([MS-OXPROPS] section 2.58) indicates what actions a user has taken on a Meeting object.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	B	C	D	E	F	G	H	I	J	K	L	M	unused																		

A - ciManager (1 bit): The user is the owner of the Meeting object's Calendar folder. If this bit is set, the **ciDelegate** bit SHOULD NOT be set.

B - ciDelegate (1 bit): The user is a delegate acting on a Meeting object in a delegator's Calendar folder. If this bit is set, the **ciManager** bit SHOULD NOT be set.

C - ciDeletedWithNoResponse (1 bit): The user deleted the Meeting object with no response sent to the organizer.

D - ciDeletedExceptionWithNoResponse (1 bit): The user deleted an exception to a recurring series with no response sent to the organizer.

E - ciRespondedTentative (1 bit): The user tentatively accepted the meeting request.

F - ciRespondedAccept (1 bit): The user accepted the meeting request.

G - ciRespondedDecline (1 bit): The user declined the meeting request.

H - ciModifiedStartTime (1 bit): The user modified the start time.

I - ciModifiedEndTime (1 bit): The user modified the end time.

J - ciModifiedLocation (1 bit): The user changed the location of the meeting.

K - ciRespondedExceptionDecline (1 bit): The user declined an exception to a recurring series.

L - ciCanceled (1 bit): The user canceled a meeting request.

M - ciExceptionCanceled (1 bit): The user canceled an exception to a recurring series.

unused (19 bits): These bits are unused, MUST be zero and MUST be ignored.

2.2.3 Appointment Object

There are no additional properties specific to Appointment objects not already specified for Calendar objects.

2.2.4 Meeting Object

The properties that are specific to Meeting objects are specified in sections 2.2.4.1 through 2.2.4.10.7. These properties have no meaning for Appointment objects. Unless otherwise specified, these properties are to always exist.

2.2.4.1 PidLidAppointmentSequenceTime Property

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

The value of the **PidLidAppointmentSequenceTime** property ([MS-OXPROPS] section 2.26) on the organizer's Meeting object indicates the date and time at which the **PidLidAppointmentSequence** property (section 2.2.1.1) was last modified. The value is specified in UTC.

2.2.4.2 PidLidAppointmentLastSequence Property

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

The value of the **PidLidAppointmentLastSequence** property ([MS-OXPROPS] section 2.15) indicates to the organizer the last sequence number (2) that was sent to any attendee. For details

about when and how a client increments the sequence number (2), see section 3.1.5.4. This property has no meaning for an attendee.

2.2.4.3 PidLidAppointmentReplyTime Property

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

The value of the **PidLidAppointmentReplyTime** property ([MS-OXPROPS] section 2.24) on the attendee's Meeting object specifies the date and time at which the attendee responded to a received Meeting Request object or Meeting Update object. The value is specified in UTC.

2.2.4.4 PidLidFInvited Property

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

The **PidLidFInvited** property ([MS-OXPROPS] section 2.135) indicates whether invitations have been sent for the meeting that this Meeting object represents. A value of FALSE or the absence of this property indicates that a Meeting Request object has never been sent. A value of TRUE indicates that a Meeting Request object has been sent. After this value is set to TRUE on a Meeting object, it MUST NOT be changed.

2.2.4.5 PidLidAppointmentReplyName Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidAppointmentReplyName** property ([MS-OXPROPS] section 2.23) on the attendee's Meeting object specifies the user who last replied to the meeting request or **meeting update**. This property is set only for a delegator when a delegate responded. The value is equal to the **PidTagMailboxOwnerName** property ([MS-OXCSTOR] section 2.2.2.1) for the delegate's **message store**. This property has no meaning for the organizer.

2.2.4.6 PidLidAppointmentProposalNumber Property

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

The **PidLidAppointmentProposalNumber** property ([MS-OXPROPS] section 2.18) specifies the number of attendees who have sent **counter proposals** that have not been accepted or rejected by the organizer.

2.2.4.7 PidLidAppointmentCounterProposal Property

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

When set to TRUE (0x00000001), the **PidLidAppointmentCounterProposal** property ([MS-OXPROPS] section 2.10) indicates to the organizer that there are counter proposals that have not been accepted or rejected (by the organizer). This property has no meaning for an attendee.

2.2.4.8 PidLidAutoFillLocation Property

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

A value of TRUE for the **PidLidAutoFillLocation** property ([MS-OXPROPS] section 2.38) on the organizer's Meeting object indicates that the value of the **PidLidLocation** property (section 2.2.1.4) is set to the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) from the

RecipientRow structure, as specified in [MS-OXCDATA] section 2.8.3, that represents a Resource object.

A value of FALSE or the absence of this property indicates that the value of the **PidLidLocation** property is not automatically set.

When set, the **PidLidLocation** property SHOULD be set to the first sendable resource that is added to the meeting, or if none of the resources are sendable, the value SHOULD be set to the first unsendable resource added to the meeting.

2.2.4.9 PidLidOriginalStoreEntryId Property

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

The **PidLidOriginalStoreEntryId** property ([MS-OXPROPS] section 2.196) specifies the **EntryID** of the delegator's message store. This property SHOULD be set on Meeting objects that have been created or updated by a delegate.

The format for the **PidLidOriginalStoreEntryId** property is the same as that for the **PidTagStoreEntryId** property ([MS-OXCMSG] section 2.2.1.44).

2.2.4.10 RecipientRow Properties

A Meeting object has one **RecipientRow** structure, as specified in [MS-OXCDATA] section 2.8.3, for each sendable attendee.

In addition, a **RecipientRow** structure can exist for the organizer of the Meeting object. Unsendable attendees do not have a corresponding **RecipientRow** structure but SHOULD have a **RecipientRow** structure in the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25).

The Appointment and Meeting Object Protocol specifies properties that can be set in the **RecipientProperties** field of **RecipientRow** structures, as specified in [MS-OXCDATA] section 2.8.3.2. These properties are listed in the sections 2.2.4.10.1 through 2.2.4.10.7.

2.2.4.10.1 PidTagRecipientFlags Property

The **PidTagRecipientFlags** property ([MS-OXPROPS] section 2.890) specifies a bit field that describes the recipient (2) status. This property is not required. The following individual flags can be set.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
A	B	C		D	E	F	G	H	I	unused																					

A - recipSendable (1 bit): The recipient (2) is a sendable attendee. This flag is used only in the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25).

B - recipOrganizer (1 bit): The **RecipientRow** structure ([MS-OXCDATA] section 2.8.3) on which this flag is set represents the meeting organizer.

C -- unused (2 bits): These bits are unused, MUST be zero and MUST be ignored.

D - recipExceptionalResponse (1 bit): This flag indicates that the attendee gave a response for the exception, as specified in section 2.2.1.44.2, on which the associated **RecipientRow** structure resides. This flag is used only in a **RecipientRow** structure of an Exception Embedded Message object of the organizer's Meeting object.

E - recipExceptionalDeleted (1 bit): This flag indicates that although the **RecipientRow** structure exists, it is treated as if the corresponding recipient (2) does not exist. This flag is used only in a **RecipientRow** structure of an Exception Embedded Message object of the organizer's Meeting object.

F - reserved (1 bit): This flag is reserved and MUST NOT be set.<22>

G - reserved (1 bit): This flag is reserved and MUST NOT be set.<23>

H - recipOriginal (1 bit): This flag indicates that the recipient (2) is an original attendee. This flag is used only in the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25).

I - reserved (1 bit): This flag is reserved and MUST NOT be set.<24>

unused (22 bits): These bits are unused, MUST be zero, and MUST be ignored.

2.2.4.10.2 PidTagRecipientTrackStatus Property

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

The value of the **PidTagRecipientTrackStatus** property ([MS-OXPROPS] section 2.896) indicates the response status that is returned by the attendee. If this value is not set, it is assumed to be "respNone" (0x00000000). If set, it MUST be one of the following, as specified in section 2.2.1.11:

- respNone
- respAccepted (0x00000003)
- respDeclined (0x00000004)
- respTentative (0x00000002)

2.2.4.10.3 PidTagRecipientTrackStatusTime Property

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

The **PidTagRecipientTrackStatusTime** property ([MS-OXPROPS] section 2.897) indicates the date and time at which the attendee responded. The value is specified in UTC.

2.2.4.10.4 PidTagRecipientProposed Property

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

A value of TRUE for the **PidTagRecipientProposed** property ([MS-OXPROPS] section 2.892) indicates that the attendee proposed a new date and/or time. A value of FALSE or the absence of this property means either that the attendee did not yet respond or that the most recent response from the attendee did not propose a new date or time. This value cannot be TRUE for attendees in a recurring series.

2.2.4.10.5 PidTagRecipientProposedStartTime Property

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

When the value of the **PidTagRecipientProposed** property (section 2.2.4.10.4) is set to TRUE, the value of the **PidTagRecipientProposedStartTime** property ([MS-OXPROPS] section 2.894) indicates the value requested by the attendee to set as the value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5) for the single-instance Meeting object or Exception object.

2.2.4.10.6 PidTagRecipientProposedEndTime Property

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

When the value of the **PidTagRecipientProposed** property (section 2.2.4.10.4) is set to TRUE, the value of the **PidTagRecipientProposedEndTime** property ([MS-OXPROPS] section 2.893) indicates the value requested by the attendee to set as the value of the **PidLidAppointmentEndWhole** property (section 2.2.1.6) for the single-instance Meeting object or Exception object.

2.2.4.10.7 Recipient Type Property

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

This property is specified in [MS-OXCMSG]. The appropriate value is set as the recipient type for each **RecipientRow** structure, as specified in [MS-OXCDATA] section 2.8.3, in the Meeting object. The appropriate values for the recipient type are listed in the following table.

Attendee type	Recipient type value
Organizer	0x01
Sendable, required attendee	0x01
Sendable, optional attendee	0x02
Sendable, Resource object	0x03 (only on the Meeting object in the organizer's Calendar folder)

2.2.5 Meeting-Related Objects

Properties that are specific to meeting-related objects are specified in sections 2.2.5.1 through 2.2.5.7. These include Meeting Request objects, Meeting Update objects, Meeting Cancellation objects, Meeting Response objects, and **Meeting Forward Notification objects**. Unless otherwise specified, these properties MUST exist.

2.2.5.1 PidLidSideEffects Property

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

The possible flag values of the **PidLidSideEffects** property are specified in [MS-OXCMSG] section 2.2.1.16. All Meeting Request objects are to always include the following flags:

- **seOpenToDelete** (0x00000001)
- **seOpenToCopy** (0x00000020)

- **seOpenToMove** (0x00000040)
- **seCannotUndoDelete** (0x00000400)
- **seCannotUndoCopy** (0x00000800)
- **seCannotUndoMove** (0x00001000)

2.2.5.2 PidLidAttendeeCriticalChange Property

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

The value of the **PidLidAttendeeCriticalChange** property ([MS-OXPROPS] section 2.37) specifies the date and time at which the meeting-related object was sent. The value is specified in UTC.<25>

2.2.5.3 PidLidWhere Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidLidWhere** property ([MS-OXPROPS] section 2.353) SHOULD be the same as the value of the **PidLidLocation** property (section 2.2.1.4) from the associated Meeting object.<26>

2.2.5.4 PidLidServerProcessed Property

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

A value of TRUE for the **PidLidServerProcessed** property ([MS-OXPROPS] section 2.232) indicates that the Meeting Request object or Meeting Update object has been processed.

2.2.5.5 PidLidServerProcessingActions Property

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

The **PidLidServerProcessingActions** property ([MS-OXPROPS] section 2.233) indicates what processing actions have been taken on the Meeting Request object or Meeting Update object. The following flags can be set.

Flag name	Value
cpsDelegatorWantsCopy	0x00000002
cpsCreatedOnPrincipal	0x00000010
cpsUpdatedCalItem	0x00000080
cpsCopiedOldProperties	0x00000100
cpsSendAutoResponse	0x00000400
cpsRevivedException	0x00000800
cpsProcessedMeetingForwardNotification	0x00001000

2.2.5.6 PidLidTimeZone Property

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

The value of the **PidLidTimeZone** property ([MS-OXPROPS] section 2.340) specifies information about the time zone of a recurring meeting. This property is read only when the **PidLidAppointmentRecur** property (section 2.2.1.44) is not set, but the value of the **PidLidIsRecurring** property (section 2.2.1.13) is set to TRUE and the value of the **PidLidIsException** property (section 2.2.1.35) is set to FALSE. The lower two bytes specify an index into a table that contains time zone information. From the upper two bytes, only the highest bit is read. If that bit is set, the time zone referenced will not observe daylight saving time; otherwise, the daylight saving time dates listed in the following table will be used.

Index	Standard offset from UTC+12 (international date line) in minutes	Standard date {wMonth, wDayOfWeek, wDay, wHour}	Daylight date {wMonth, wDayOfWeek, wDay, wHour}
0	0	N/A	N/A
1	12*60	{10, 0, 5, 2}	{3, 0, 5, 1}
2	11*60	{9, 0, 5, 2}	{3, 0, 5, 1}
3	11*60	{10, 0, 5, 3}	{3, 0, 5, 2}
4	11*60	{10, 0, 5, 3}	{3, 0, 5, 2}
5	10*60	{9, 0, 5, 1}	{3, 0, 5, 0}
6	11*60	{9, 0, 5, 1}	{3, 0, 5, 0}
7	10*60	{10, 0, 5, 4}	{3, 0, 5, 3}
8	15*60	{2, 0, 2, 2}	{10, 0, 3, 2}
9	16*60	{11, 0, 1, 2}	{3, 0, 2, 2}
10	17*60	{11, 0, 1, 2}	{3, 0, 2, 2}
11	18*60	{11, 0, 1, 2}	{3, 0, 2, 2}
12	19*60	{11, 0, 1, 2}	{3, 0, 2, 2}
13	20*60	{11, 0, 1, 2}	{3, 0, 2, 2}
14	21*60	{11, 0, 1, 2}	{3, 0, 2, 2}
15	22*60	N/A	N/A
16	23*60	N/A	N/A
17	0*60	{4, 0, 1, 3}	{9, 0, 5, 2}
18	2*60	{3, 0, 5, 3}	{10, 0, 5, 2}
19	(2*60)+30	{3, 0, 5, 3}	{10, 0, 5, 2}
20	3*60	N/A	N/A
21	4*60	N/A	N/A
22	5*60	N/A	N/A

Index	Standard offset from UTC+12 (international date line) in minutes	Standard date {wMonth, wDayOfWeek, wDay, wHour}	Daylight date {wMonth, wDayOfWeek, wDay, wHour}
23	(6*60)+30	N/A	N/A
24	8*60	N/A	N/A
25	(8*60)+30	{9, 2, 4, 2}	{3, 0, 1, 2}
26	9*60	N/A	N/A
27	10*60	{9, 0, 3, 2}	{3, 5, 5, 2}
28	(15*60)+30	{11, 0, 1, 0}	{3, 0, 2, 0}
29	13*60	{10, 0, 5, 1}	{3, 0, 5, 0}
30	14*60	{10, 0, 5, 1}	{3, 0, 5, 0}
31	12*60	N/A	N/A
32	15*60	N/A	N/A
33	16*60	N/A	N/A
34	17*60	N/A	N/A
35	17*60	N/A	N/A
36	18*60	N/A	N/A
37	18*60	{10, 0, 5, 2}	{4, 0, 1, 2}
38	19*60	N/A	N/A
39	24*60	N/A	N/A
40	0*60	N/A	N/A
41	1*60	N/A	N/A
42	2*60	{3, 0, 5, 2}	{10, 0, 1, 2}
43	2*60	N/A	N/A
44	(2*60)+30	N/A	N/A
45	4*60	{9, 0, 2, 2}	{4, 0, 2, 2}
46	6*60	N/A	N/A
47	7*60	N/A	N/A
48	(7*60)+30	N/A	N/A
49	10*60	{9, 4, 5, 2}	{5, 5, 1, 2}
50	10*60	N/A	N/A

Index	Standard offset from UTC+12 (international date line) in minutes	Standard date {wMonth, wDayOfWeek, wDay, wHour}	Daylight date {wMonth, wDayOfWeek, wDay, wHour}
51	9*60	{10, 0, 5, 1}	{3, 0, 5, 0}
52	2*60	{3, 0, 5, 2}	{8, 0, 5, 2}
53	2*60	{4, 0, 1, 3}	{10, 0, 5, 2}
54	(2*60)+30	{4, 0, 1, 3}	{10, 0, 5, 2}
55	2*60	{4, 0, 1, 3}	{10, 0, 1, 2}
56	16*60	{3, 6, 2, 23}	{10, 6, 2, 23}
57	4*60	{3, 0, 5, 3}	{10, 0, 5, 2}
58	19*60	{10, 0, 5, 2}	{4, 0, 1, 2}
59	20*60	{10, 0, 5, 2}	{4, 0, 1, 2}

The Standard date and Daylight date columns specify a date in the following format: {**wMonth**, **wDayOfWeek**, **wDay**, **wHour**}

The **wMonth** values are interpreted as shown in the following table.

Value	Meaning
1	January
2	February
3	March
4	April
5	May
6	June
7	July
8	August
9	September
10	October
11	November
12	December

The **wDayOfWeek** values are interpreted as shown in the following table.

Value	Meaning
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

wDay: Indicates the occurrence of the day of the week within the month (1 to 5, where 5 indicates the final occurrence during the month if that day of the week does not occur 5 times).

wHour: Indicates the hour at which the transition will occur in local time. The member ranges in value from 0 (zero) (12:00 A.M.) to 23 (11:00 P.M.).

If daylight saving time is observed, during the daylight time period, an additional -60 offset is added to the standard offset.

2.2.5.7 PidTagProcessed Property

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

The value of the **PidTagProcessed** property ([MS-OXPROPS] section 2.861) indicates whether a client has processed a meeting-related object. The **PidTagProcessed** property is left unset until processing is completed, and then it is set to TRUE (0x01).

2.2.6 Meeting Request/Update Object

The properties that are specific to Meeting Request objects and Meeting Update objects are specified in sections 2.2.6.1 through 2.2.6.12. Unless otherwise specified, these properties are to always exist.

2.2.6.1 PidTagMessageClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) MUST be "IPM.Schedule.Meeting.Request" or MUST be prefixed with "IPM.Schedule.Meeting.Request.".

2.2.6.2 PidLidChangeHighlight Property

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

The **PidLidChangeHighlight** property ([MS-OXPROPS] section 2.51) specifies a bit field that indicates how the Meeting object has changed.<27> This property is not required. The individual flags that can be set are as follows.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	B	C	D	E	F	G	H	I	J	K	L	M	unused																		N

A - BIT_CH_START (1 bit): The **PidLidAppointmentStartWhole** property (section 2.2.1.5) has changed.

B - BIT_CH_END (1 bit): The **PidLidAppointmentEndWhole** property (section 2.2.1.6) has changed.

C - BIT_CH_RECUR (1 bit): The recurrence pattern has changed. For details about recurrence patterns, see the section 2.2.1.44.

D - BIT_CH_LOCATION (1 bit): The **PidLidLocation** property (section 2.2.1.4) has changed.

E - BIT_CH_SUBJECT (1 bit): The **PidTagNormalizedSubject** property ([MS-OXCMSG] section 2.2.1.10) has changed.

F - BIT_CH_REQATT (1 bit): One or more required attendees were added.

G - BIT_CH_OPTATT (1 bit): One or more optional attendees were added.

H - BIT_CH_BODY (1 bit): The body was modified.

I - unused (1 bit): These bits are not used. MUST be zero and MUST be ignored.

J - BIT_CH_RESPONSE (1 bit): Either the **PidTagResponseRequested** property ([MS-OXOMSG] section 2.2.1.46) or the **PidTagReplyRequested** property ([MS-OXOMSG] section 2.2.1.45) has changed.

K - BIT_CH_ALLOWPROPOSE (1 bit): The **PidLidAppointmentNotAllowPropose** property (section 2.2.1.26) has changed.

L - Deprecated (1 bit): This flag is deprecated. This value is neither read nor written to.

M - Reserved (1 bit): This flag is reserved and MUST NOT be set.

unused (18 bits): These bits are not used. MUST be zero and MUST be ignored.

N - Reserved (1 bit): This flag is reserved and MUST NOT be set.

2.2.6.3 PidLidForwardInstance Property

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

A value of TRUE for the **PidLidForwardInstance** property ([MS-OXPROPS] section 2.138) indicates that the Meeting Request object represents an exception to a recurring series, and it was forwarded (even when forwarded by the organizer) rather than being an invitation sent by the organizer. A value of FALSE for this property indicates that the Meeting Request object is not a forwarded instance. This property is not required, read, or written to.<28>

2.2.6.4 PidLidIntendedBusyStatus Property

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

The **PidLidIntendedBusyStatus** property ([MS-OXPROPS] section 2.151) specifies the value of the **PidLidBusyStatus** property (section 2.2.1.2) on the Meeting object in the organizer's calendar at the time the Meeting Request object or Meeting Update object was sent. The allowable values of this property are the same as those for the **PidLidBusyStatus** property.

2.2.6.5 PidLidMeetingType Property

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

The **PidLidMeetingType** property ([MS-OXPROPS] section 2.170) indicates the type of Meeting Request object or Meeting Update object. The value of this property MUST be set to one of those listed in the following table.

Request type	Value	Meaning
mtgEmpty	0x00000000	Unspecified.
mtgRequest	0x00000001	The meeting request is the initial request.
mtgFull	0x00010000	The property is set to this value if one of the following applies: Attendees were added. The meeting was cancelled and the organizer is uncancelling it. **If PidLidChangeHighlight** property (section 2.2.6.2) has **BIT_CH_START**, **BIT_CH_END**, **or BIT_CH_RECUR** set
mtgInfo	0x00020000	An **informational update** was made to the meeting and it is not one of the conditions listed in **mtgFull**.
mtgOutOfDate	0x00080000	A newer Meeting Request object or Meeting Update object was received after this one. For more details, see section 3.1.5.2.
mtgDelegatorCopy	0x00100000	Set on the delegator's copy when a delegate will handle meeting-related objects. For more details, see section 3.1.4.7.2.1.

2.2.6.6 PidLidAppointmentMessageClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidAppointmentMessageClass** property ([MS-OXPROPS] section 2.16) indicates the value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) of the Meeting object that is to be generated from the Meeting Request object. The value of the **PidLidAppointmentMessageClass** property MUST either be "IPM.Appointment" or be prefixed with "IPM.Appointment.". This property is not required.

2.2.6.7 PidLidOldLocation Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidLidOldLocation** property ([MS-OXPROPS] section 2.189) indicates the original value of the **PidLidLocation** property (section 2.2.1.4) before a meeting update.<29> This property is not required.

2.2.6.8 PidLidOldWhenStartWhole Property

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

The **PidLidOldWhenStartWhole** property ([MS-OXPROPS] section 2.192) indicates the original value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5) before a meeting update.<30> This property is not required.

2.2.6.9 PidLidOldWhenEndWhole Property

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

The **PidLidOldWhenEndWhole** property ([MS-OXPROPS] section 2.191) indicates the original value of the **PidLidAppointmentEndWhole** property (section 2.2.1.6) before a meeting update.<31> This property is not required.

2.2.6.10 Attachments

A Meeting Request object or Meeting Update object represents a single-instance object, a recurring series, or an exception. A Meeting Request object or a Meeting Update object for a recurring series cannot include any **Exception Attachment objects**. A separate Meeting Request object or Meeting Update object is to be sent for each exception, even when attendees are invited to both the recurring series and the exceptions.

2.2.6.11 PidLidCalendarType Property

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

When the Meeting Request object represents a recurring series or an exception, the **PidLidCalendarType** property ([MS-OXPROPS] section 2.48) is the value of the **CalendarType** field from the **PidLidAppointmentRecur** property (section 2.2.1.44). Otherwise, this property is not set and is assumed to be zero.

2.2.6.12 Best Body Properties

The body of a Meeting Request object is a copy of the body of the Meeting object or Exception Embedded Message object to which it refers, optionally preceded by downlevel text. The term "downlevel text" refers to extra text that can be added into the body of a Meeting Request object before a copy of the Meeting object body, so that a client that receives the Meeting Request object but does not understand its format will still show the meeting details. Downlevel text is to be separated from the copied Meeting object body with a delimiter, and then the delimiter is to be followed by two blank lines. The delimiters that are used are listed in the following table. Note that adding downlevel text, and what that text is, is an implementation-specific choice. Clients can use the values of the **PidLidAppointmentStartWhole** (section 2.2.1.5), **PidLidAppointmentEndWhole** (section 2.2.1.6), and **PidLidLocation** (section 2.2.1.4) properties as the downlevel text.<32>

PidLidCalendarType	Delimiter
CAL_HIJRI	+=+=+=+=+=+=+=+=+=+
CAL_HEBREW	+=+=+=+=+=+=+=+=+=+
CAL_THAI	+=+=+=+=+=+=+=+=+=+
CAL_LUNAR_KOREAN	+=+=+=+=+=+=+=+=+=+
CAL_LUNAR_JAPANESE	+=+=+=+=+=+=+=+=+=+
CAL_CHINESE_LUNAR	+=+=+=+=+=+=+=+=+=+

PidLidCalendarType	Delimiter
CAL_SAKA	+=+=+=+=+=+=+=+=+=+
CAL_GREGORIAN	*~*~*~*~*~*~*~*~*~*
Any other value	*~*~*~*~*~*~*~*~*~*

2.2.7 Meeting Response Object

A Meeting Response object takes the form of one of three types: accept, tentatively accept, or decline. The properties specified in sections 2.2.7.1 through 2.2.7.8 apply to all response types, except where individually noted. Unless otherwise specified, these properties are to always exist.

2.2.7.1 PidTagMessageClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) MUST begin with "IPM.Schedule.Meeting.Resp" and MUST be appended with either ".Pos", ".Tent", or ".Neg", indicating accept, tentatively accept, or decline, respectively.

2.2.7.2 PidTagSubjectPrefix Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagSubjectPrefix** property ([MS-OXCMSG] section 2.2.1.9) is a localized string that contains an implementation-dependent response to a meeting request to accept, tentatively accept, decline, or propose a new time for a meeting.

For example, if localized in English, this property can be set to values such as, "Accepted", "Tentative", "Declined", or "New Time Proposed".

2.2.7.3 PidLidAppointmentProposedStartWhole Property

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

The **PidLidAppointmentProposedStartWhole** property ([MS-OXPROPS] section 2.21) specifies the proposed value for the **PidLidAppointmentStartWhole** property (section 2.2.1.5) for a counter proposal. This value is specified in UTC.

2.2.7.4 PidLidAppointmentProposedEndWhole Property

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

The **PidLidAppointmentProposedEndWhole** property ([MS-OXPROPS] section 2.20) specifies the proposed value for the **PidLidAppointmentEndWhole** property (section 2.2.1.6) for a counter proposal. This value is specified in UTC.

2.2.7.5 PidLidAppointmentProposedDuration Property

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

The **PidLidAppointmentProposedDuration** property ([MS-OXPROPS] section 2.19) indicates the proposed value for the **PidLidAppointmentDuration** property (section 2.2.1.7) for a counter

proposal. If set, it is equal to the number of minutes between the value of the **PidLidAppointmentProposedStartWhole** property (section 2.2.7.3) and the value of the **PidLidAppointmentProposedEndWhole** property (section 2.2.7.4).

2.2.7.6 PidLidAppointmentCounterProposal Property

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

A value of TRUE for the **PidLidAppointmentCounterProposal** property ([MS-OXPROPS] section 2.10) indicates that the Meeting Response object is a counter proposal.

2.2.7.7 PidLidIsSilent Property

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

A value of TRUE for the **PidLidIsSilent** property ([MS-OXPROPS] section 2.157) indicates that the user did not include any text in the body of the Meeting Response object.

2.2.7.8 PidLidPromptSendUpdate Property

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

A value of TRUE for the **PidLidPromptSendUpdate** property ([MS-OXPROPS] section 2.212) indicates that the Meeting Response object was out-of-date when it was received.

2.2.8 Meeting Cancellation Object

The properties that are specific to Meeting Cancellation objects are specified in section 2.2.8.1 through section 2.2.8.6. Unless otherwise specified, these properties are to always exist.

2.2.8.1 PidTagMessageClass Property

Type: **PtypString8** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) MUST be "IPM.Schedule.Meeting.Canceled".

2.2.8.2 PidTagSubjectPrefix Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagSubjectPrefix** property ([MS-OXCMSG] section 2.2.1.9) contains an implementation-dependent localized message that indicates that the meeting was canceled. For example, in English, this property can be set to "Canceled".

2.2.8.3 PidLidIntendedBusyStatus Property

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

The value of the **PidLidIntendedBusyStatus** property (section 2.2.6.4) MUST be set to "olFree" (0x00000000).

2.2.8.4 PidLidResponseStatus Property

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

The value of the **PidLidResponseStatus** property (section 2.2.1.11) MUST be set to "respNotResponded" (0x00000005).

2.2.8.5 PidLidBusyStatus Property

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

The value of the **PidLidBusyStatus** property (section 2.2.1.2) MUST be set to "olFree" (0x00000000).

2.2.8.6 PidLidMeetingType Property

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

The **PidLidMeetingType** property (section 2.2.6.5) indicates the type of Meeting Cancellation object. The value of this property MUST be set to one listed in the following table.

Request type	Value	Description
mtgEmpty	0x00000000	Unspecified.
mtgDelegatorCopy	0x00100000	This is set on the delegator's copy when a delegate will handle meeting-related objects. For more details, see section 3.1.4.7.2.1.

2.2.9 Meeting Forward Notification Object

The properties that are specific to Meeting Forward Notification objects are specified in sections 2.2.9.1 through 2.2.9.4. Unless otherwise specified, these properties MUST exist.

2.2.9.1 PidTagMessageClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) MUST be "IPM.Schedule.Meeting.Notification.Forward".

2.2.9.2 PidTagSubjectPrefix Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagSubjectPrefix** property ([MS-OXCMSG] section 2.2.1.9) MUST be a localized string that indicates that the object is a Meeting Forward Notification object.

2.2.9.3 PidLidForwardNotificationRecipients Property

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

The **PidLidForwardNotificationRecipients** property ([MS-OXPROPS] section 2.139) contains a list of **RecipientRow** structures, as specified in [MS-OXCDATA] section 2.8.3, that indicate the recipients (2) of a meeting forward. For the format of this property, see section 2.2.1.25.

2.2.9.4 PidLidPromptSendUpdate Property

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

A value of TRUE for the **PidLidPromptSendUpdate** property (section 2.2.7.8) indicates that the Meeting Forward Notification object was out-of-date when it was received.

2.2.10 Exceptions

An exception specifies changes to an instance of a recurring series. Two objects define an exception:

- The Exception Attachment object holds attachment-related information. One Exception Attachment object SHOULD exist for each instance listed in the **ModifiedInstanceDates** field of the **PidLidAppointmentRecur** property (section 2.2.1.44) on the Calendar object. Note that there are circumstances in which the number of Exception Attachment objects will not match the number of values in the **ModifiedInstanceDates** field of the **PidLidAppointmentRecur** property. For example, when an Exception Attachment object cannot be found in the set of attachments, a client or server can create it. In some cases, this erroneously leads to multiple Exception Attachment objects for an instance.
- The Exception Embedded Message object contains the modifications to an instance. One Exception Embedded Message object MUST exist for each Exception Attachment object.

The properties that are specific to the Exception Attachment object that make up the exception are specified in sections 2.2.10.1.1 through 2.2.10.1.6. The properties that are specific to the Exception Embedded Message object that make up the exception are specified in sections 2.2.10.2.1 through 2.2.10.2.7. Unless otherwise specified, these properties are to always exist.

2.2.10.1 Exception Attachment Object

The Exception Attachment object MUST have the properties listed in sections 2.2.10.1.1 through 2.2.10.1.6.

2.2.10.1.1 PidTagAttachmentHidden Property

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

The value of the **PidTagAttachmentHidden** property ([MS-OXCMSG] section 2.2.2.24) MUST be TRUE.

2.2.10.1.2 PidTagAttachmentFlags Property

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

The value of the **PidTagAttachmentFlags** property ([MS-OXCMSG] section 2.2.2.23) MUST include the **afException** flag (0x00000002).

2.2.10.1.3 PidTagAttachMethod Property

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

The value of the **PidTagAttachMethod** property ([MS-OXCMSG] section 2.2.2.9) MUST be **afEmbeddedMessage** (0x00000005), which indicates that the exception data in the **PidTagAttachDataObject** property ([MS-OXCMSG] section 2.2.2.8) is an **Embedded Message object**.

2.2.10.1.4 PidTagExceptionStartTime Property

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

The value of the **PidTagExceptionStartTime** property ([MS-OXPROPS] section 2.677) indicates the start date and time of the exception in the local time zone of the computer when the exception is created.

This property is informational and cannot be relied on for critical information because if a user changes the client computer's time zone after this property is written, the value of this property will no longer match what is expected by the client.

2.2.10.1.5 PidTagExceptionEndTime Property

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

The value of the **PidTagExceptionEndTime** property ([MS-OXPROPS] section 2.675) indicates the end date and time of the exception in the local time zone of the computer when the exception is created.

This property is informational and cannot be relied on for critical information because if a user changes the client computer's time zone after this property is written, the value of this property will no longer match what is expected by the client.

2.2.10.1.6 PidTagExceptionReplaceTime Property

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

The value of the **PidTagExceptionReplaceTime** property ([MS-OXPROPS] section 2.676) indicates the original date and time at which the instance in the recurrence pattern would have occurred if it were not an exception. This value is specified in UTC.<33>

2.2.10.2 Exception Embedded Message Object

The data stored in the Embedded Message object that is represented by the **PidTagAttachDataObject** property ([MS-OXCMSG] section 2.2.2.8) contains properties that are specific to the exception. Any property that is not set on the Exception Embedded Message object is obtained from the recurrence series. The following properties SHOULD NOT be set on an Exception Embedded Message object; if they are set, they are not used by the client or server:

- **PidLidAppointmentLastSequence** (section 2.2.4.2)
- **PidLidMeetingWorkspaceUrl** (section 2.2.1.48)
- **PidLidContacts** ([MS-OXCMSG] section 2.2.1.57.2)
- **PidTagSensitivity** ([MS-OXCMSG] section 2.2.1.13)
- **PidLidPrivate** ([MS-OXCMSG] section 2.2.1.15)
- **PidNameKeywords** ([MS-OXCMSG] section 2.2.1.17)

The properties that are specific to the Exception Embedded Message object are specified in sections 2.2.10.2.1 through 2.2.10.2.7.

2.2.10.2.1 PidTagMessageClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) MUST be "IPM.OLE.CLASS.{00061055-0000-0000-C000-000000000046}".

2.2.10.2.2 Best Body Properties

If the value of the **PidLidFExceptionalBody** property (section 2.2.10.2.6) is FALSE, body properties SHOULD NOT be written to the Exception Embedded Message object. When the value of the **PidLidFExceptionalBody** property is TRUE (0x00000001), body properties are part of the Exception Embedded Message object even when blank and follow the same rules as the Best body properties for a Calendar object, as specified in section 2.2.1.38.

2.2.10.2.3 PidLidAppointmentStartWhole Property

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

The **PidLidAppointmentStartWhole** property (section 2.2.1.5) MUST exist on an Exception Embedded Message object, even if the exception has the same start date and time as the instance in the recurring series to which it corresponds. **PidLidAppointmentStartWhole** contains the start date and time of the exception and is specified in UTC.

2.2.10.2.4 PidLidAppointmentEndWhole Property

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

The **PidLidAppointmentEndWhole** property (section 2.2.1.6) MUST exist on an Exception object, even if the exception has the same end date and time as the instance in the recurring series to which it corresponds. **PidLidAppointmentEndWhole** contains the end date and time of the exception and is specified in UTC.

2.2.10.2.5 PidLidExceptionReplaceTime Property

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

The **PidLidExceptionReplaceTime** property ([MS-OXPROPS] section 2.117) specifies the date and time within the recurrence pattern that the exception will replace. The value is specified in UTC. This property allows the Exception Attachment object to be found for a particular instance.

2.2.10.2.6 PidLidFExceptionalBody Property

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

A value of TRUE for the **PidLidFExceptionalBody** property ([MS-OXPROPS] section 2.131) indicates that the Exception Embedded Message object has a body that differs from the Recurring Calendar object. If the value of this property is TRUE, the Exception Embedded Message object MUST have a body. If the value of this property is FALSE, or if the property does not exist, a client or server obtains the body from the Recurring Calendar object.

2.2.10.2.7 PidLidFInvited Property

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

The value of the **PidLidFInvited** property ([MS-OXPROPS] section 2.135) for an Exception Embedded Message object takes the same meaning as specified in section 2.2.4.4. If a meeting request has been sent for an exception but not for the recurring series, the value of this property on the Recurring Calendar object will still be FALSE, but the value on the Exception Embedded Message object will be TRUE.

2.2.11 Calendar Folder

For a folder to be treated as a Calendar folder, unless otherwise specified, it MUST have the properties specified in sections 2.2.11.1 and 2.2.11.2. When creating Calendar objects, the client or server SHOULD create them in the **Calendar special folder**. Note that an end user can create calendar items in any Calendar folder. However, **free/busy status** information is calculated only from the Calendar special folder.

2.2.11.1 PidTagContainerClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The value of the **PidTagContainerClass** property ([MS-OXPROPS] section 2.633) for all Calendar folders MUST be set to "IPF.Appointment".

2.2.11.2 PidTagDefaultPostMessageClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

If the **PidTagDefaultPostMessageClass** property ([MS-OXPROPS] section 2.651) is set on a Calendar folder, the value MUST either contain "IPM.Appointment" or begin with "IPM.Appointment".

2.2.12 Delegate Information Object

The properties that are set on the **Delegate Information object**, as specified in [MS-OXODLGT], are specified in sections 2.2.12.1 through 2.2.12.5.

2.2.12.1 PidTagFreeBusyCountMonths Property

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

The **PidTagFreeBusyCountMonths** property ([MS-OXPROPS] section 2.694) is used to calculate the start and end dates of the range of free/busy status data to be published to the **public folders**, as specified in [MS-OXOPFFB]. The value of this property MUST be greater than or equal to 0x00000000 and less than or equal to 0x00000024. This property is not required.

2.2.12.2 PidTagScheduleInfoAutoAcceptAppointments Property

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

A value of TRUE for the **PidTagScheduleInfoAutoAcceptAppointments** property ([MS-OXPROPS] section 2.956) indicates that a client or server SHOULD automatically respond to all meeting requests for the attendee or Resource object. The response MUST be acceptance, unless an additional constraint specified by the **PidTagScheduleInfoDisallowRecurringAppts** (section 2.2.12.3) or **PidTagScheduleInfoDisallowOverlappingAppts** property (section 2.2.12.4) is met. A value of FALSE or the absence of this property indicates that a client or server does not automatically accept meeting requests. This property is not required.

2.2.12.3 PidTagScheduleInfoDisallowRecurringAppts Property

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

The **PidTagScheduleInfoDisallowRecurringAppts** property ([MS-OXPROPS] section 2.963) is only meaningful when the value of the **PidTagScheduleInfoAutoAcceptAppointments** property (section 2.2.12.2) is TRUE. A value of TRUE indicates that when automatically responding to

meeting requests, a client or server declines Meeting Request objects that represent a recurring series. A value of FALSE or the absence of this property indicates that recurring meetings are accepted. This property is not required.

2.2.12.4 PidTagScheduleInfoDisallowOverlappingAppts Property

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

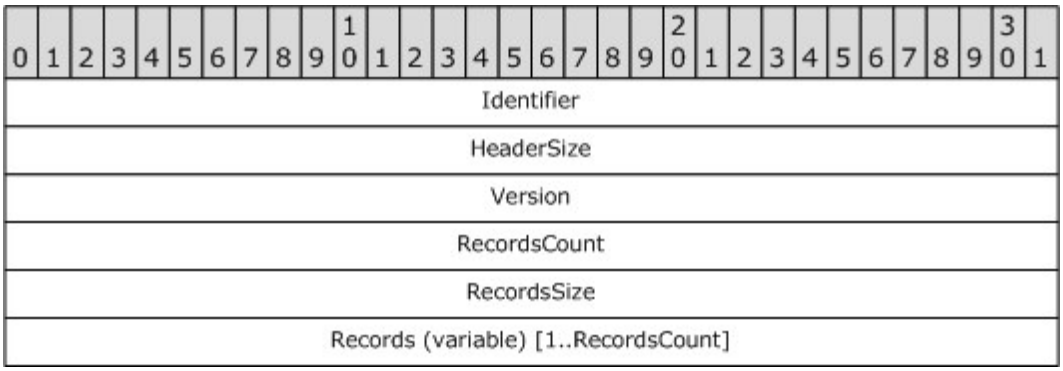
The **PidTagScheduleInfoDisallowOverlappingAppts** property ([MS-OXPROPS] section 2.962) is only meaningful when the value of the **PidTagScheduleInfoAutoAcceptAppointments** property (section 2.2.12.2) is TRUE. A value of TRUE indicates that when automatically responding to meeting requests, a client or server declines instances that overlap with previously scheduled events. A value of FALSE or the absence of this property indicates that overlapping instances are accepted. This property is not required.

2.2.12.5 PidTagScheduleInfoAppointmentTombstone Property

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

The **PidTagScheduleInfoAppointmentTombstone** property ([MS-OXPROPS] section 2.955) in a delegator's Delegate Information object contains a list of **tombstones**. This property is not required. If this property does not exist when a meeting is declined by the delegator or the delegate, it MUST be created.

This property has the following structure, where the fields are stored in little-endian byte order.



Identifier (4 bytes): This field MUST have a value of 0xBEDEAFCD.

HeaderSize (4 bytes): This field MUST have a value of 0x00000014.

Version (4 bytes): This field MUST have a value of 0x00000003.

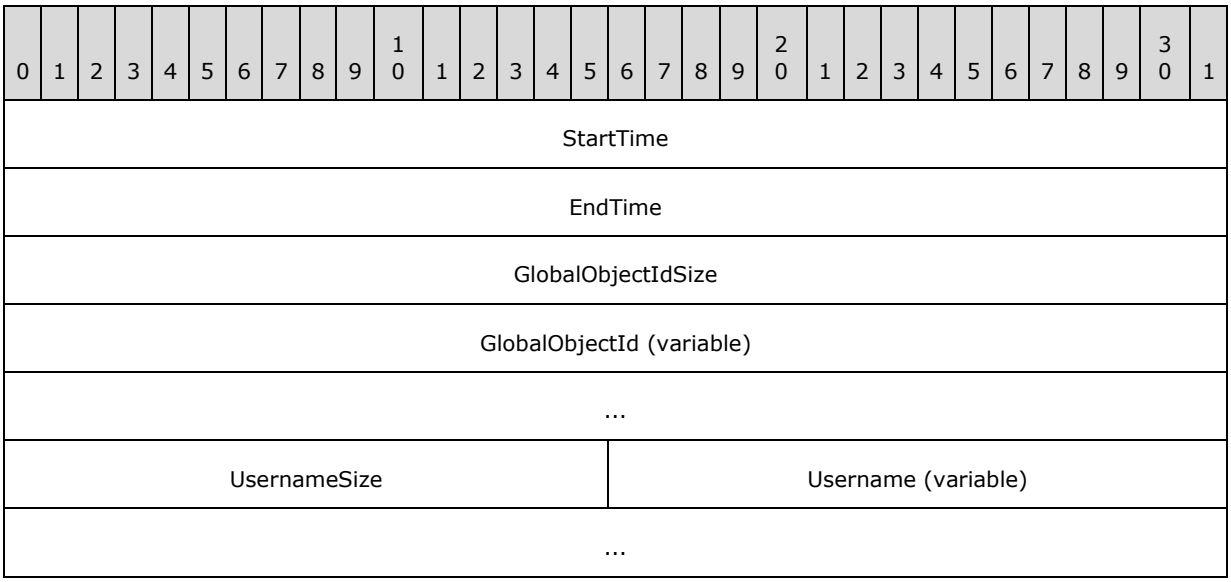
RecordsCount (4 bytes): The number of structures contained in the **Records** field.

RecordsSize (4 bytes): This field MUST have a value of 0x00000014.

Records (variable): An array of **Record** structures (section 2.2.12.5.1), each of which specifies a tombstone.

2.2.12.5.1 Record Structure

The **Record** structure specifies a tombstone within the **Records** field of the **PidTagScheduleInfoAppointmentTombstone** property (section 2.2.12.5).



StartTime (4 bytes): The start time of the Meeting object in minutes since midnight, January 1, 1601, UTC.

EndTime (4 bytes): The end time of the Meeting object in minutes since midnight, January 1, 1601, UTC.

GlobalObjectIdSize (4 bytes): The size, in bytes, of the **GlobalObjectId** field.

GlobalObjectId (variable): The value of the **PidLidGlobalObjectId** property (section 2.2.1.27) of the meeting that this record represents.

UsernameSize (2 bytes): The size, in bytes, of the **Username** field.

Username (variable): A non-Unicode string. The value of the **Username** field is equal to the value of the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) of the Address Book object of the user who added the tombstone.

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.

The following abstract data model (ADM) types are defined in this section:

Mailbox

Calendar

Appointment Item

Meeting Item

3.1.1.1 Per Mailbox

Mailboxes are represented by the **Mailbox** ADM type. The following ADM objects are maintained for each **Mailbox** ADM type:

Mailbox.Calendar: An abstract representation of a calendar.

3.1.1.2 Per Calendar

Calendars are represented by the **Calendar** ADM type. The following ADM objects are maintained for each **Calendar** ADM type:

Calendar.AppointmentItem: A Calendar object that represents a scheduled event for a single user.

Calendar.MeetingItem: A Calendar object that represents a scheduled event with two or more attendees.

3.1.1.3 Per Appointment Item

A Calendar object that represents a scheduled event for a single user is the **AppointmentItem** ADM type. The following ADM objects are maintained for each **AppointmentItem** ADM type:

AppointmentItem.StartDate: The day and time on which the appointment represented by the **AppointmentItem** begins.

AppointmentItem.EndDate: The day and time at which the appointment represented by the **AppointmentItem** is slated to finish.

AppointmentItem.Recurrence: Defines whether the calendar item repeats, or whether it is a one-time recurrence.

3.1.1.4 Per Meeting Item

A Calendar object that represents a scheduled event with two or more attendees is the **MeetingItem** ADM type. A **MeetingItem** object can have all of the same ADM objects as an **AppointmentItem** ADM type. The following additional ADM objects are also maintained for each **MeetingItem** ADM type:

MeetingItem.Attendees: A list of people invited to the meeting.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Creating a Calendar Object

To create a Calendar object, a client calls the **RopCreateMessage** **ROP** ([MS-OXCROPS] section 2.2.6.2), as specified in [MS-OXCMSG] section 3.1.4.2, and creates a Calendar object in a Calendar folder.

Although Appointment objects can be created in any Calendar folder, Meeting objects SHOULD only be created in the Calendar special folder, as specified in [MS-OXOSFLD] section 2.2.1. If a user creates a Meeting object in another Calendar folder, the client can prompt the user and offer to create a clone of the meeting in the Calendar special folder at the time of creation. All Calendar objects MUST have all the required properties, as specified in sections 2.2.1 and 2.2.2. A Meeting object MUST also have the required properties, as specified in section 2.2.4.

3.1.4.2 Converting an Appointment Object to a Meeting Object

To change an Appointment object into a Meeting object, the client sets the **asfMeeting** bit to 1 in the **PidLidAppointmentStateFlags** property (section 2.2.1.10). As long as a meeting request has not been sent for the Meeting object (according to the **PidLidFInvited** property (section 2.2.4.4)), the client can set the **asfMeeting** bit to 0 (zero), reverting the Meeting object back to an Appointment object. However, after a meeting request is sent out, the **asfMeeting** bit MUST remain set to 1 on the Meeting object. In other words, the Meeting object MUST NOT revert to an Appointment object, even if all attendees are later removed.

3.1.4.3 Copying a Calendar Object

To copy a Calendar object, the client creates a new Calendar object in the target folder by using the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2) and then copies all properties from the original object onto the new Calendar object, with the exception of the following properties. Note that a copy of a Calendar object is a static copy of the original. When the source object is a meeting, the new copy will not be updated with any future changes made by the organizer.

The following properties MUST NOT be copied onto the new object:

- **PidLidAppointmentColor** ([MS-OXPROPS] section 2.9)
- **PidLidGlobalObjectId** (section 2.2.1.27)

- **PidLidCleanGlobalObjectId** (section 2.2.1.28)
- **PidLidMeetingWorkspaceUrl** (section 2.2.1.48)

These four properties are set according to the descriptions in the sections referenced as though the Calendar object was just created.

If the Calendar object to be copied is a Meeting object, the following actions MUST be taken by the client:

- The **auxApptFlagCopied** flag is added to the value of the **PidLidAppointmentAuxiliaryFlags** property (section 2.2.1.3) on the new object.
- The **asfReceived** flag SHOULD be added to the value of the **PidLidAppointmentStateFlags** property (section 2.2.1.10) on the new object.<34>

In addition:

- The value of the **PidLidFInvited** property (section 2.2.4.4) on the new object MUST be set to FALSE.
- The value of the **PidTagOwnerAppointmentId** property (section 2.2.1.29) on the new object MUST be set to 0x00000000.
- The **RecipientRow** structures ([MS-OXCDATA] section 2.8.3) SHOULD be copied onto the new object.<35>
- The **PidLidResponseStatus** property (section 2.2.1.11) SHOULD<36> be set to "respNotResponded" (0x00000005).
- The **PidTagSubjectPrefix** property ([MS-OXCMSG] section 2.2.1.9) SHOULD<37> be set to a localized string indicating the meeting is a copy.

3.1.4.3.1 Copying a Calendar Object When The Source Object Is an Exception

When the source object is an exception, the client uses the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2) to create a new Calendar object. The client follows the same requirements for the new object, as already specified for copying a Calendar object. Furthermore, all properties that are not set on the Exception Embedded Message object, as specified in section 2.2.10.2, but that are set on the Recurring Calendar object are to be copied onto the new object. In addition, the following actions MUST be taken by the client:

- The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) MUST be reset to "IPM.Appointment" on the new object.
- In addition to those already specified in section 3.1.4.3, the following properties MUST NOT be copied onto the new object:
  - **PidLidAppointmentRecur** (section 2.2.1.44)
  - **PidLidRecurrenceType** (section 2.2.1.45)
  - **PidLidRecurrencePattern** (section 2.2.1.46)
  - **PidLidTimeZoneStruct** (section 2.2.1.39)
  - **PidLidTimeZoneDescription** (section 2.2.1.40)

▪**PidLidFExceptionalAttendees** (section 2.2.2.3)

- The value of the **PidLidClipStart** property (section 2.2.1.14) MUST be set to the value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5).
- The value of the **PidLidClipEnd** (section 2.2.1.15) property MUST be set to the value of the **PidLidAppointmentEndWhole** (section 2.2.1.6) property.
- The value of the **PidTagIconIndex** property (section 2.2.1.49) SHOULD be set to 0x00000400 if the Exception Attachment object was attached to an Appointment object or to 0x00000402 if the Exception Attachment object was attached to a Meeting object.
- The value of the **PidLidRecurring** property (section 2.2.1.12) MUST be set to FALSE.
- When copying the **RecipientRow** structures ([MS-OXCDATA] section 2.8.3), the client copies them from the Exception Embedded Message object and not from the Recurring Calendar object.

3.1.4.4 Creating an Appointment Object When the Source is Not a Calendar Object

When the source object is not a Calendar object, the client creates a new Appointment object by using the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2), and after copying all properties from the source object, ensures that all required properties, as specified in sections 2.2.1 and 2.2.2, exist on the new Appointment object.

3.1.4.5 Deleting a Meeting Object

To delete a Meeting object, the client calls the **RopDeleteMessages** ROP ([MS-OXCROPS] section 2.2.4.11) on the relevant Message object.

When the user deletes a Meeting object, the client SHOULD<38> send a Meeting Cancellation object to all attendees, as specified in section 3.1.4.9.1.

3.1.4.6 Expanding a Recurrence

To enumerate the instances of the recurring series between the dates specified by the values of the **StartDate** and **EndDate** fields, a client uses the **RecurrencePattern** structure specified in section 2.2.1.44.1. The client excludes every instance that occurs on a date in the **DeletedInstanceDates** field and includes every date in the list in the **ModifiedInstanceDates** field. Note that the **ModifiedInstanceDates** field contains only the date on which the exception will occur and not its exact time. To get specific start and end dates and times for a given exception, the client uses the values from the **StartDateTime** and **EndDateTime** fields of the **ExceptionInfo** structure specified in section 2.2.1.44.2.

3.1.4.6.1 Finding an Exception

To find an exception, the client examines the **AppointmentRecurrencePattern** structure specified in section 2.2.1.44.5, which specifies deleted instances and modified instances. Every modified instance is associated with an Exception Attachment object, as specified in section 2.2.10. For each modified instance in the **RecurrencePattern** structure, there is a matching **ExceptionInfo** structure, as specified in section 2.2.1.44.2. The **StartDateTime** property is stored in the time zone represented by the **PidLidTimeZoneStruct** property (section 2.2.1.39) that is stored on the Recurring Calendar object. To find the Exception Attachment object that corresponds to a modified instance, the **StartDateTime** field of the **ExceptionInfo** structure of that modified instance is matched to the **PidLidAppointmentStartWhole** property (section 2.2.1.5) of the Exception Embedded Message object. The value of the **StartDateTime** field is converted to UTC by using the **PidLidTimeZoneStruct** property. This date and time SHOULD match the value

PidLidAppointmentStartWhole property of exactly one Exception Embedded Message object. If an Exception Attachment object cannot be found, the client creates a new one.

3.1.4.6.2 Creating an Exception

To create a new exception that replaces an instance of the recurring series, the client modifies the value of the **PidLidAppointmentRecur** property (section 2.2.1.44) as follows:

- The exception's new start date is added to the array in the **ModifiedInstanceDates** field in the **RecurrencePattern** structure, as specified in section 2.2.1.44.1.
- The value of the **ModifiedInstanceCount** field of the **RecurrencePattern** structure is incremented.
- The original start date is added to the array in the **DeletedInstanceDate** field and the value of the **DeletedInstanceCount** field is incremented.
- The new and original start dates are in the time zone specified by the **PidLidTimeZoneStruct** property (section 2.2.1.39).
- The **ExceptionInfo** structure, as specified in section 2.2.1.44.2, is added to the **recurrence BLOB**. Note that the original start date and the new start date can be the same if the date was not modified in the exception.

The client also adds an Exception Attachment object, as specified in section 2.2.10.1, and an Exception Embedded Message object, as specified in section 2.2.10.2, each with properties specified in section 2.2.10, and adds any overridden properties to the Exception Embedded Message object. The value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5) of the Exception Embedded Message object is specified in UTC and is the UTC equivalent of the date and time added to the **StartDateTime** field in the **ExceptionInfo** structure, as specified in section 2.2.1.44.2. The client also copies the **RecipientRow** structures, as specified in [MS-OXCDATA] section 2.8.3, from the Meeting object to the Exception Embedded Message object.

3.1.4.6.3 Deleting an Instance of a Recurring Series

To delete a single occurrence of a recurring series that is not a previously modified instance, the client increments the value of the **DeletedInstanceCount** field in the **RecurrencePattern** structure, as specified in section 2.2.1.44.1, and adds the date of the instance being deleted to the array in the **DeletedInstanceDates** field.

3.1.4.6.4 Deleting an Exception

To delete an exception, the client removes the instance being deleted from the array in the **ModifiedInstanceDate** field of the **RecurrencePattern** structure, as specified in section 2.2.1.44.1, and decrements the value of the **ModifiedInstanceCount** field. The client also deletes the associated Exception Attachment object.

3.1.4.7 Meeting Requests

3.1.4.7.1 Sending a Meeting Request

To inform attendees of an event, the organizer or delegate of the organizer sends a meeting request. To send the meeting request, the client creates a new Meeting Request object by using the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2), sets the Meeting Request object

properties as follows, and then sends the Meeting Request object by using the **RopSubmitMessage** ROP ([MS-OXCROPS] section 2.2.7.1).

The client copies all properties specified in section 2.2.1 from the Meeting object to the Meeting Request object, adds all required properties as specified in section 2.2.6, and then sets the following on the Meeting Request object:

- The value of the **PidLidAppointmentSequence** property (section 2.2.1.1) to zero.
- The **asfReceived** and **asfMeeting** bits on the **PidLidAppointmentStateFlags** property (section 2.2.1.10).
- The value of the **PidLidResponseStatus** property (section 2.2.1.11) to "respNotResponded" (0x00000005).
- The value of the **PidLidIntendedBusyStatus** property (section 2.2.6.4) equal to the value of the **PidLidBusyStatus** property (section 2.2.1.2) from the Meeting object.
- The value of the **PidLidBusyStatus** property to "olTentative" (0x00000001).
- The value of the **PidLidFExceptionalAttendees** property (section 2.2.2.3) to FALSE.
- The value of the **PidLidFExceptionalBody** property (section 2.2.10.2.6) to FALSE.
- The value of the **PidLidIsRecurring** property (section 2.2.1.13).
- The value of the **PidLidRecurring** property (section 2.2.1.12).
- The value of the **PidLidCalendarType** property (section 2.2.6.11), if the Meeting Request object represents a recurring series.
- The value of the **PidLidWhere** property (section 2.2.5.3) equal to the value of the **PidLidLocation** property (section 2.2.1.4) from the Meeting object.
- The value of the **PidLidAttendeeCriticalChange** property (section 2.2.5.2) to the current date and time in UTC.
- The value of the **PidLidMeetingType** property (section 2.2.6.5) to "mtgRequest" (0x00000001).
- The value of the **PidLidAllAttendeesString** property (section 2.2.1.16).
- The value of the **PidLidToAttendeesString** property (section 2.2.1.17).
- The value of the **PidLidCcAttendeesString** property (section 2.2.1.18).
- The value of the **PidTagStartDate** property (section 2.2.1.30).
- The value of the **PidTagEndDate** property (section 2.2.1.31).

The **PidTagProcessed** property (section 2.2.5.7) is not set.

The following optional steps SHOULD also be taken:

- If the user has not modified the value of the **PidLidReminderDelta** property ([MS-OXORMDR] section 2.2.1.3) on the Meeting Request object from its default value (as defined by the client), the value of this property SHOULD be set to 0x5AE980E1.
- The client SHOULD prepend downlevel text to the message body, as specified in 2.2.6.12.

After successfully sending a Meeting Request object, the client modifies the Meeting object in the organizer's Calendar folder in the following ways:

- Set the value of the **PidLidFInvited** property (section 2.2.4.4) to TRUE.
- Set the value of the **PidLidToAttendeesString** property equal to the value that was set on the Meeting Request object.
- Set the value of the **PidLidCcAttendeesString** property equal to the value that was set on the Meeting Request object.
- Set the value of **PidLidOwnerCriticalChange** property (section 2.2.1.34) equal to the value that was set on the Meeting Request object.

3.1.4.7.1.1 Using Direct Booking

The term "direct booking" refers to the action of creating a Meeting object directly on the Calendar folder of an attendee instead of sending a Meeting Request object to the attendee<39>. The decision whether to attempt to direct book any sendable attendees is an implementation choice; however, a client MAY<40> attempt to direct book any sendable attendee as long as the following two **conditions** exist:

- The value of the **PidTagScheduleInfoAutoAcceptAppointments** property (section 2.2.12.2) in the attendee's Delegate Information object is set to TRUE. Note that this requires public folders to be enabled on the server.
- The organizer has permission to write to the attendee's Calendar special folder, as specified in [MS-OXCPERM].

The client fails the direct booking action and does not send a Meeting Request object to any attendees if either of the following occurs:

- The value of the **PidTagScheduleInfoDisallowRecurringAppts** property (section 2.2.12.3) in the attendee's Delegate Information object is set to TRUE and the Meeting Request object represents a recurring series (see section 2.2.12.2).
- The value of the **PidTagScheduleInfoDisallowOverlappingAppts** property (section 2.2.12.4) in the attendee's Delegate Information object is set to TRUE and there is a meeting conflict during the date and time specified on the Meeting Request object. For details about how to determine whether a conflict exists, see section 3.1.4.11.

To direct book an attendee, the client takes the following actions:

- Create the Meeting object on the attendee's Calendar special folder, as specified in section 3.1.4.7.2.2 and then modify the Meeting object as if the attendee had accepted it, as specified in section 3.1.4.8.1. A Meeting Response object MUST NOT be sent to the organizer.
- Publish updated free/busy status information to the Resource object's Delegate Information object.
- Set the value of the **PidTagRecipientTrackStatus** property (section 2.2.4.10.2) to "respAccepted" (0x00000003) on the **RecipientRow** structure ([MS-OXCDATA] section 2.8.3) that represents the attendee on the organizer's Meeting object.
- Set the value of the **PidTagRecipientTrackStatusTime** property (section 2.2.4.10.3) to the current date and time on the **RecipientRow** structure that represents the attendee in the organizer's Meeting object.

- If the Meeting Request object represents an exception, set the **recipExceptionalResponse** bit to 1 in the **PidTagRecipientFlags** property (section 2.2.4.10.1) on the **RecipientRow** structure that represents the attendee in the organizer's Meeting object.
- Remove the **RecipientRow** structure that represents the attendee from the Meeting Request object so that it will not be sent to the attendee.

3.1.4.7.2 Receiving a Meeting Request

After receiving a Meeting Request object, the client checks whether the Calendar object is eligible for update, as specified in section 3.1.4.7.2.1, to determine whether to create a Meeting object in the user's Calendar special folder by using the information in the Meeting Request object. If the client determines that the Meeting object is to be created, it creates the object as specified in section 3.1.4.7.2.2. If the **PiAutoProcess** setting (as specified in [MS-OXOCFG] section 2.2.5.1.1) in the **calendar options dictionary** is set to FALSE, the client SHOULD NOT<41> immediately create the Meeting object but instead wait until the client UI indicates that the user has viewed the meeting request. A client that does not support the calendar options dictionary can have its own defined mechanism for allowing the user to decide whether Meeting objects will be automatically created upon receipt of a Meeting Request object.

If the client does create the Meeting object, the client creates it according to the rules specified in section 3.1.4.7.2.1.

3.1.4.7.2.1 Automatically Creating a Meeting Object

When a delegator receives a Meeting Request object, the client follows the sequencing rules described in section 3.1.5.6 before automatically creating a Meeting object.

If any one of the following conditions are met, the client does not automatically create the Meeting object:

- The Meeting Request object is located in the **Sent Items folder** or the **Outbox folder**, as specified in [MS-OXOSFLD].
- The value of the **PidTagProcessed** property (section 2.2.5.7) on the Meeting Request object is set to TRUE.
- The Meeting Request object is intended for the delegator and a tombstone exists, as specified in section 2.2.12.5, indicating that another user has already declined the meeting.

The client MAY<42> skip automatic creation of the Meeting object if the value of the **PidLidServerProcessed** property (section 2.2.5.4) on the Meeting Request object is set to TRUE and the **PidLidServerProcessingActions** property (section 2.2.5.5) either does not exist or has the bit of the **PidLidReminderDelta** property set to 1. If the client skips automatic creation of the Meeting object, it MUST NOT set the **PidTagProcessed** property on the Meeting Request object.

3.1.4.7.2.2 Creating the Meeting Object

Before creating the Meeting object, the client searches for a Calendar object that matches the Meeting Request object, as specified in section 3.1.5.1, and does not create a new Meeting object if a match is found. Otherwise, the client creates a new Meeting object and copies all the properties specified in section 2.2.1 from the Meeting Request object onto it and adds all the required properties specified in section 2.2.4. The client MAY change the value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) on the new Meeting object to the value of the **PidLidAppointmentMessageClass** property (section 2.2.6.6) from the Meeting Request object. In addition, the client sets the following properties on the Meeting object:

- The value of the **PidLidResponseStatus** property (section 2.2.1.11) to "respNotResponded" (0x00000005).
- The value of the **PidLidBusyStatus** property (section 2.2.1.2) to "olTentative" (0x00000001), unless the value of the **PidLidIntendedBusyStatus** property (section 2.2.6.4) is "olFree" (0x00000000), in which case it MUST be set to "olFree".
- If the value of the **PidLidReminderDelta** property ([MS-OXORMDR] section 2.2.1.3) in the Meeting Request object is set to 0x5AE980E1, change the value of the newly created Meeting object's **PidLidReminderDelta** property to its default value (as defined by the client), and then recalculate the value of the **PidLidReminderSignalTime** property ([MS-OXORMDR] section 2.2.1.2), as specified in [MS-OXORMDR].
- If the value of the **PidLidReminderSet** property ([MS-OXORMDR] section 2.2.1.1) is FALSE and the value of the **PidLidAppointmentSubType** property (section 2.2.1.9) is FALSE (that is, the meeting is not an all-day event), then the client SHOULD change the value of the **PidLidReminderSet** property to TRUE, set the **PidLidReminderDelta** property to its default value (as defined by the client), and recalculate the value of the **PidLidReminderSignalTime** property.<43>
- The client SHOULD copy the value of the **PidLidAppointmentAuxiliaryFlags** property (section 2.2.1.3) from the Meeting Request object to the new Meeting object.
- The client SHOULD remove the downlevel text, as specified in section 2.2.6.12, from the body.
- The client SHOULD<44> set the value of the **PidLidAppointmentReplyName** property (section 2.2.4.5) to a null string.
- The client SHOULD<45> copy the **RecipientRow** structures, as specified in [MS-OXCDATA] section 2.8.3, in the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25) from the Meeting Request object to the **RecipientRow** structures of the Meeting object. For each **RecipientRow** structure copied, if the **recipOriginal** bit is set in the **PidTagRecipientFlags** property (section 2.2.4.10.1) of the **RecipientRow** structure, then the client MUST set the **recipSendable** bit in the **PidTagRecipientFlags** property.
- The client MUST NOT copy the **PidLidAppointmentUnsendableRecipients** property from the Meeting Request object to the Meeting object.
- If the **PidLidAppointmentUnsendableRecipients** property is not set on the Meeting Request object, or if the client did not copy the **RecipientRow** structures in the **PidLidAppointmentUnsendableRecipients** property of the Meeting Request object to the Meeting object, then the client creates a **RecipientRow** structure for each recipient (2) listed in the **PidLidNonSendableTo** (section 2.2.1.19), **PidLidNonSendableCc** (section 2.2.1.20), and **PidLidNonSendableBcc** (section 2.2.1.21) properties. The client sets the recipient type (as specified in section 2.2.4.10.7) for each **RecipientRow** structure added as specified in section 2.2.1.19, 2.2.1.20, and 2.2.1.21.
- The client sets the **PidLidNonSendableTo**, **PidLidNonSendableCc**, and **PidLidNonSendableBcc** properties to a NULL string on the Meeting object.

If the Meeting Request object represents a recurring series and the Meeting object is newly created, the client searches the folder for orphan instances of the meeting by matching the **PidLidCleanGlobalObjectId** property (section 2.2.1.28) with that of the new Meeting object. The client then converts any orphan instances that are found into exceptions and deletes the orphan instances. For each converted exception, the client SHOULD<46> copy the value of the **PidLidBusyStatus** property from the orphan instance to the **BusyStatus** field of the associated **ExceptionInfo** structure and set the **ARO_BUSYSTATUS** flag as specified in section 2.2.1.44.2.

Finally, after creating the Meeting object, the client:

- SHOULD set the value of its **PidTagProcessed** property (section 2.2.5.7) to TRUE, unless it is in a public folder, in which case this property is not set.
- MAY<47> set the **PidLidServerProcessed** property (section 2.2.5.4) on the Meeting Request object to TRUE. If setting the **PidLidServerProcessed** property, the client either sets both the **cpsCreatedOnPrincipal** and **cpsUpdatedCalItem** bits of the **PidLidServerProcessingActions** property (section 2.2.5.5) on the Meeting Request object or leaves the **PidLidServerProcessingActions** property unset.

3.1.4.7.2.3 Sending an Auto Response

After creating the Meeting object, the client can automatically send a Meeting Response object to the organizer if the value of the **PidTagScheduleInfoAutoAcceptAppointments** property (section 2.2.12.2) in the organizer's Delegate Information object is nonzero. When sending the Meeting Response object, the client does so as specified in section 3.1.4.8. If the client chooses to automatically respond to Meeting Request objects, it also adheres to the requirements of the **PidTagScheduleInfoDisallowRecurringAppts** (section 2.2.12.3) and **PidTagScheduleInfoDisallowOverlappingAppts** (section 2.2.12.4) properties, accepting or declining meetings as appropriate.

The client MAY<48> skip automatic sending of Meeting Response objects to the organizer if the **PidLidServerProcessed** property (section 2.2.5.4) of the Meeting Request object is set to TRUE and the **cpsSendAutoResponse** bit of the **PidLidServerProcessingActions** property (section 2.2.5.5) is set to 1. If the client automatically responds to the Meeting Request object, it MAY<49> set the **cpsSendAutoResponse** bit of the **PidLidServerProcessingActions** property to 1.

When the client is acting for the delegate and the client supports sending automatic responses, it uses the values defined for the delegator and not for the delegate when determining whether to automatically respond to Meeting Request objects on behalf of the delegator.

3.1.4.7.3 Sending a Meeting Update

The organizer or delegate of the organizer sends an update to inform attendees of changes to an event that has already been sent out (according to the **PidLidFInvited** property (section 2.2.4.4) on the Meeting object). To do so, the client creates and submits a Meeting Update object, following the same rules as sending a Meeting Request object, as specified in section 3.1.4.7.1, with differences as explained in this section.

If the value of the **PidLidLocation** property (section 2.2.1.4) was modified by the user on the Meeting object, the client SHOULD set the value of the **PidLidOldLocation** property (section 2.2.6.7) on the Meeting Update object to the old value. Similarly, if the value of the **PidLidAppointmentStartWhole** (section 2.2.1.5) and/or **PidLidAppointmentEndWhole** (section 2.2.1.6) properties were modified by the user on the Meeting object, the client SHOULD set the value of the **PidLidOldWhenStartWhole** (section 2.2.6.8) and **PidLidOldWhenEndWhole** (section 2.2.6.9) properties to the old values respectively.<50>

The client modifies the sequence number (2) as specified in section 3.1.5.4.

3.1.4.7.3.1 Detecting a Significant Change to the Meeting Object

Certain constraints result when a **significant change** is made to a Meeting object. A significant change to a Meeting object includes any of the following conditions:

- The value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5) is changed.

- The value of the **PidLidAppointmentEndWhole** property (section 2.2.1.6) is changed.
- The recurrence pattern, as defined in the **PidLidAppointmentRecur** property (section 2.2.1.44), was added, modified, or removed.

If a significant change is made to the Meeting object, the value of the **PidLidMeetingType** property (section 2.2.6.5) MUST be set to **mtgFull** (0x00010000). Otherwise, the value of this property SHOULD<51> be set to **mtgInfo** (0x00020000).

3.1.4.7.3.2 Clearing Previous Responses

If the Meeting object is set to request responses (according to the **PidTagResponseRequested** property ([MS-OXOMSG] section 2.2.1.46)), and a significant change, as specified in section 3.1.4.7.3.1, has been made, the client SHOULD clear all tallied responses that have been previously received from attendees. The client SHOULD NOT clear the tallied responses if a significant change has not been made or if the Meeting object is not set to request responses.<52>

To clear the tallied responses, the client sets the value of the **PidTagRecipientTrackStatus** property (section 2.2.4.10.2) to "respNone" (0x00000000) in each **RecipientRow** structure, as specified in [MS-OXCDATA] section 2.8.3, of the Meeting object, as well as for any **RecipientRow** structures in the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25) and any recipients (2) listed in the **PidLidNonSendToTrackStatus** (section 2.2.1.22), **PidLidNonSendCcTrackStatus** (section 2.2.1.23), and **PidLidNonSendBccTrackStatus** (section 2.2.1.24) properties. The client also can set the value of the **PidTagRecipientTrackStatusTime** property (section 2.2.4.10.3) in each **RecipientRow** structure to an invalid date (for example, 12:18 A.M. 23 October 1602). Changing this value is not required.

3.1.4.7.3.3 Adding Attendees to a Meeting

When the organizer adds a new attendee to a recurring series or single-instance meeting, the client adds the attendee to the Meeting object's **RecipientRow** structures, as specified in [MS-OXCDATA] section 2.8.3, and sets the properties as specified in section 2.2.4.10.

When the organizer adds a new attendee to an exception of a recurring series, the client adds a recipient row for the attendee to the Exception Embedded Message object, as specified in section 2.2.10.2. If the attendee already existed in the exception but the **recipExceptionalDeleted** bit of the **PidTagRecipientFlags** property (section 2.2.4.10.1) of the attendee's **RecipientRow** structure was set, then the client resets this bit to 0.

3.1.4.7.3.4 Sending Updates to New Attendees Only

When a significant change, as specified in section 3.1.4.7.3.1, has not been made and the user has added attendees, the client MAY send the Meeting Update object to only the new attendees or prompt the user asking whether the user wants to send the update to all recipients (2) or only to added or removed recipients (2). The client SHOULD<53> treat an attendee as a new attendee if the value of the **recipSendable** bit of the attendee's **PidTagRecipientFlags** property (section 2.2.4.10.1) has changed from 0 to 1. When sending a Meeting Update object to only new attendees, the client SHOULD<54> add all other attendees (for example, those not receiving the Meeting Update object) into the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25) on the Meeting Update object. For each attendee added to the **PidLidAppointmentUnsendableRecipients** property, the client sets the **recipOriginal** bit of the **PidTagRecipientFlags** property (section 2.2.4.10.1) of the attendee's **RecipientRow** structure, as specified in [MS-OXCDATA] section 2.8.3, if the **recipSendable** bit is set, and sets the **recipSendable** bit to 0.

3.1.4.7.3.5 Updating a Recurring Series

After a Meeting Update object is sent for a recurring series that has modified exceptions and the recurrence pattern has not changed, the client sends a Meeting Update object for each modified exception (according to the **PidLidAppointmentStartWhole** property (section 2.2.1.5) on the Exception Embedded Message object, as specified in section 2.2.10.2) for which the start date and time has not yet passed. The Meeting Update object for each exception conforms to the specifications in section 2.2.6. Before sending a Meeting Update object for each exception, the client SHOULD<55> send a Meeting Cancellation object for that exception to each attendee included in the recurring series that is not included in the exception. If the attendee exists in the **RecipientRow** structures ([MS-OXCDATA] section 2.8.3) of the Exception object and the **recipExceptionalDeleted** bit of the **PidTagRecipientFlags** property (section 2.2.4.10.1) of the attendee's **RecipientRow** structure is set, then the client treats the attendee as not included in the exception. If the series has deleted exceptions, the client sends a Meeting Cancellation object for each deleted exception for which (according to the **DeletedInstanceDates** field of the **PidLidAppointmentRecur** property (section 2.2.1.44) of the Meeting object) the start date and time has not yet passed. The Meeting Cancellation object for each exception conforms to the specifications in section 2.2.6. If the recurrence pattern has changed, the client SHOULD<56> send out Meeting Cancellation objects for each exception whose start date and time (according to the **PidLidAppointmentStartWhole** property (section 2.2.1.5) on the Exception Embedded Message object) has not yet passed to every attendee of the exception and removes every exception from the **PidLidAppointmentRecur** property and every Exception Attachment object.

After a Meeting Update object is sent to a partial attendee list as defined in section 3.1.4.7.3.4 for a recurring series that has exceptions, the client SHOULD send a Meeting Request object for each exception whose start date and time (according to the **PidLidAppointmentStartWhole** property on the Exception Embedded Message object) has not yet passed to every attendee of the exception that is in the Partial attendee List.

3.1.4.7.4 Receiving a Meeting Update

After receiving a Meeting Update object, the client determines whether to update the Meeting object in the user's Calendar special folder with the information in the Meeting Update object, as specified in section 3.1.4.7.4.1. If the client determines that the Meeting object is to be updated, it does so as specified in section 3.1.4.7.4.2. If the **PiAutoProcess** setting, as specified in [MS-OXOCFG] section 2.2.5.1.1, in the calendar options dictionary is set to FALSE, the client SHOULD NOT<57> immediately update the Meeting object but wait until the user views the Meeting Update object. A client that does not support the calendar options dictionary MAY have its own defined mechanism for allowing the user to decide whether Meeting objects will be automatically updated upon receipt of a Meeting Update object.

3.1.4.7.4.1 Skipping Automatic Updating of the Meeting Object

When a delegator receives a Meeting Update object, the client follows the sequencing rules described in section 3.1.5.6 before deciding to automatically update the Meeting object.

If any one of the following conditions is met, the client does not automatically update the Meeting object:

- The Meeting Update object is located in the Sent Items folder, as specified in [MS-OXOSFLD] section 2.2, or the Outbox folder, as specified in [MS-OXOSFLD] section 2.2.
- The value of the **PidTagProcessed** property (section 2.2.5.7) on the Meeting Update object is set to TRUE.

- The Meeting Update object is intended for the delegator and a tombstone exists, as specified in section 2.2.12.5, indicating that another user has already declined the meeting.

The client MAY<58> skip automatic updating of the Meeting object if the value of the **PidLidServerProcessed** property (section 2.2.5.4) on the Meeting Update object is set to TRUE, and the **PidLidServerProcessingActions** property (section 2.2.5.5) either does not exist or has the **cpsUpdatedCalItem** bit of this property is set. If the client skips automatic updating of the Meeting object, it MUST NOT set the **PidTagProcessed** property on the Meeting Update object.

3.1.4.7.4.2 Updating the Meeting Object

To update a Meeting object, a client first searches for a Calendar object that matches the Meeting Update object, as specified in section 3.1.5.1, and verifies the following (taking the appropriate actions as specified):

- If the Meeting Update object represents an exception, and the recurring series is found in the Calendar folder but the exception was previously deleted from the recurring series, then the client re-creates the exception, as specified in section 3.1.4.6.2 (unless the **cpsRevivedException** bit of the **PidLidServerProcessingActions** property (section 2.2.5.5) of the Meeting Request object is set and the value of the **PidLidServerProcessed** property (section 2.2.5.4) is set to TRUE, in which case the client MAY<59> skip re-creation of the exception). After re-creating the exception, the client can set the **cpsRevivedException** bit of the **PidLidServerProcessingActions** property of the Meeting Request object to TRUE.
- If the Meeting object is not found in the Calendar folder, as specified in section 3.1.5.1, then the client SHOULD change the value of the **PidLidMeetingType** property (section 2.2.6.5) on the Meeting Update object to **mtgRequest** (0x00000001) and then follow the steps for receiving a new Meeting Request object, as specified in section 3.1.4.7.2.
- If the user is the organizer of the meeting, the client does not update the Calendar object with the information from the Meeting Update object.
- If the Meeting Update object is out-of-date, as specified in section 3.1.5.2, the client SHOULD change the value of the **PidLidMeetingType** property on the Meeting Update object to **mtgOutofDate** (0x00080000) and does not update the Meeting object. Similarly, if the Meeting Update object is not newer than the Meeting object, as specified in section 3.1.5.3, the client does not update the Meeting object.

After verifying that the Meeting object is eligible for update, the client SHOULD<60> do the following:

- Copy the value of the **PidLidLocation** property (section 2.2.1.4) from the Meeting object to the **PidLidOldLocation** property (section 2.2.6.7) on the Meeting Request object.
- Copy the value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5) from the Meeting object to the **PidLidOldWhenStartWhole** property (section 2.2.6.8) on the Meeting Request object.
- Copy the value of the **PidLidAppointmentEndWhole** property (section 2.2.1.6) from the Meeting object to the **PidLidOldWhenEndWhole** property (section 2.2.6.9) on the Meeting Request object.

The client MAY<61> skip these actions if the **cpsCopiedOldProperties** bit of the **PidLidServerProcessingActions** property of the Meeting Update object is set and the **PidLidServerProcessed** property is set to TRUE.

After completing these actions the client can set the **cpsCopiedOldProperties** bit of the **PidLidServerProcessingActions** property of the Meeting Update object.

Next, the client copies all the properties specified in section 2.2.1 from the Meeting Update object onto the Meeting object. The client also adds all required properties specified in section 2.2.4; however, the client SHOULD comply with the following exceptions:

- If the value of the **PidTagSensitivity** property ([MS-OXCMSG] section 2.2.1.13) on the Meeting object is set to private, it SHOULD<62> remain so, even if this is not the value of the property on the Meeting Update object.
- The downlevel text SHOULD be removed from the body as specified in section 2.2.6.12.

If the user has not yet responded to the original Meeting Request object, as reflected in the value of the **PidLidResponseStatus** property (section 2.2.1.11) on the Meeting object, the client MUST set the value of the **PidLidMeetingType** property on the Meeting Update object to **mtgFull** (0x00010000) and the value of the **PidTagIconIndex** property (section 2.2.1.49) on the Meeting Update object to "Meeting request/full update" (0x00000404) if these properties have any other values.

If the Meeting Update object does not include a significant change, as specified in section 3.1.4.7.3.1, and the attendee had already responded to the original Meeting Request object, the client SHOULD NOT<63> change the value of the **PidLidResponseStatus** property on the Meeting object. Regardless of whether the attendee had previously responded, if the Meeting Update object represents an update with a significant change, the client sets the following properties on the Meeting object so that it looks as if the attendee has not yet responded:

- The value of the **PidLidResponseStatus** property to "respNotResponded" (0x00000005).
- The value of the **PidLidBusyStatus** property (section 2.2.1.2) to "olTentative" (0x00000001), unless the value of the **PidLidIntendedBusyStatus** property (section 2.2.6.4) is "olFree" (0x00000000), in which case it is set to "olFree".

The client follows the same steps to Auto Respond to a Meeting Update object as is specified for a Meeting Request object in section 3.1.4.7.2.3.

Finally, after updating the Meeting object, the client:

- SHOULD set the value of the **PidTagProcessed** property (section 2.2.5.7) to TRUE, unless the object is in a public folder, in which case this property is not set.
- Can set the **PidLidServerProcessed** property on the Meeting Request object to TRUE. If the Meeting Request object's **PidLidServerProcessed** property is set, the client either MUST also set the **cpsUpdatedCalItem** bit of the **PidLidServerProcessingActions** property on the Meeting Request object or MUST leave both properties unset.

3.1.4.7.5 Forwarding a Meeting Request

To forward a Meeting Request object, either from the organizer or from an attendee who received it, the client creates a new Meeting Request object and copies all the properties from the original Meeting Request object onto the new object. The client then makes the following additional changes on the new object:

- Set the value of the **PidLidAttendeeCriticalChange** property (section 2.2.5.2) to the current date and time, in UTC.

- Set the value of the **PidLidResponseStatus** property (section 2.2.1.11) to "respNotResponded" (0x00000005).
- Set the value of the **PidLidBusyStatus** property (section 2.2.1.2) to "olTentative" (0x00000001), unless the value of the **PidLidIntendedBusyStatus** property (section 2.2.6.4) is "olFree" (0x00000000), in which case the **PidLidBusyStatus** property is set to "olFree".
- Ensure that the **asfMeeting** and **asfReceived** bits are set in the **PidLidAppointmentStateFlags** property (section 2.2.1.10).
- Reset the value of the **PidLidAllAttendeesString** (section 2.2.1.16), **PidLidToAttendeesString** (section 2.2.1.17), and **PidLidCcAttendeesString** (section 2.2.1.18) properties to a blank string.
- Set the value of the **PidTagSenderName** property ([MS-OXOMSG] section 2.2.1.51) to the value of the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) of the Address Book object of the forwarding user.
- Set the value of the **PidTagSenderEntryId** property ([MS-OXOMSG] section 2.2.1.50) to the value of the EntryID of the Address Book object of the forwarding user.
- Set the value of the **PidTagSenderSearchKey** property ([MS-OXOMSG] section 2.2.1.52) to the value of the **search key** of the Address Book object of the forwarding user.
- Set the value of the **PidTagSentRepresentingName** property ([MS-OXOMSG] section 2.2.1.57) to the value of the **PidTagDisplayName** property of the Address Book object of the organizer.
- Set the value of the **PidTagSentRepresentingEntryId** property ([MS-OXOMSG] section 2.2.1.56) to the value of the EntryID of the Address Book object of the organizer.
- Set the value of the **PidTagSentRepresentingSearchKey** property ([MS-OXOMSG] section 2.2.1.58) to the value of the search key of the Address Book object of the organizer.
- If the Meeting Request object represents an exception to a recurring series, set the value of the **PidLidForwardInstance** property (section 2.2.6.3) to TRUE.
- Set the value of the **PidLidChangeHighlight** property (section 2.2.6.2) to 0x00000000.
- Set the **auxApptFlagForwarded** bit in the **PidLidAppointmentAuxiliaryFlags** property (section 2.2.1.3).
- SHOULD<64> set the value of the **PidLidMeetingType** property (section 2.2.6.5) to **mtgRequest** (0x00000001).

The client SHOULD copy all the **RecipientRow** structures, as specified in [MS-OXCDATA] section 2.8.3, from the original Meeting Request object to the **PidLidAppointmentUnsendableRecipients**<65> property (section 2.2.1.25) of the new object. The client MUST NOT copy the **RecipientRow** structures from the original Meeting Request object to the **RecipientRow** structures on the new object. The client can set the **auxApptFlagForceMtgResponse** bit in the **PidLidAppointmentAuxiliaryFlags** property. The **PidTagProcessed** property (section 2.2.5.7) MUST NOT be set.

When a Meeting Request object is forwarded, the client MAY send a Meeting Forward Notification object to the organizer in the manner specified in section 3.1.4.10.1.

3.1.4.7.5.1 Forwarding a Recurring Series

After a Meeting Request object is forwarded for a recurring series that has exceptions, the client SHOULD<66> forward each exception whose start date and time (according to the **PidLidAppointmentStartWhole** property (section 2.2.1.5) on the Exception Embedded Message object, as specified in section 2.2.10.2) has not yet passed, as specified in 3.1.4.7.5.

3.1.4.8 Sending Meeting Responses

3.1.4.8.1 Accepting a Meeting

When the attendee or a delegate of the attendee accepts a Meeting Request object, the client ensures that the Meeting object has been created in the attendee's Calendar special folder, as specified in section 3.1.4.7.2.2. Similarly, when the attendee or delegate of the attendee accepts a Meeting Update object, the client ensures that the Meeting object has been updated in the attendee's Calendar special folder, as specified in section 3.1.4.7.4.2, unless the Meeting Update object is out-of-date, as specified in section 3.1.5.2, in which case the client does not modify the Meeting object or send a Meeting Response object.

After creating or updating the Meeting object, all changes made to the Meeting object in the attendee's Calendar special folder MUST be atomic; for example, by creating a copy of the object, applying the changes to the copy, and then deleting the original Meeting object. The client MUST make the following changes to the Meeting object:

- Set the value of the **PidLidBusyStatus** property (section 2.2.1.2) equal to the value of the **PidLidIntendedBusyStatus** property (section 2.2.6.4) from the Meeting Request object.
- Set the value of the **PidLidResponseStatus** property (section 2.2.1.11) to "respAccepted" (0x00000003).
- Set the value of the **PidLidAppointmentReplyTime** property (section 2.2.4.3) to the current date and time.
- If it is the delegate that is responding, set the value of the **PidLidAppointmentReplyName** property (section 2.2.4.5) equal to the value of the **PidTagMailboxOwnerName** property ([MS-OXCSTOR] section 2.2.2.1) from the message store. If the delegate is not the one who is responding, the **PidLidAppointmentReplyName** property is not set.

The client can prompt a user to send a Meeting Response object back to the organizer, as specified in 3.1.4.8.4.

When sending a Meeting Response object to the organizer, the client performs the additional step of setting the **ciRespondedAccept** bit of the **PidLidClientIntent** property (section 2.2.2.4) of the Meeting object.

3.1.4.8.2 Tentatively Accepting a Meeting

When the attendee or a delegate of the attendee tentatively accepts a Meeting Request object, the client follows the process specified in section 3.1.4.8.1, except that when updating the Meeting object, the following substitutions are made:

- Set the value of the **PidLidBusyStatus** property (section 2.2.1.2) to "olTentative" (0x00000001), unless the value of the **PidLidIntendedBusyStatus** property (section 2.2.6.4) is "olFree" (0x00000000), in which case it MUST be set to "olFree".

- Set the value of the **PidLidResponseStatus** property (section 2.2.1.11) to "respTentative" (0x00000002).

When sending a Meeting Response object to the organizer, the client also sets the **ciRespondedTentative** bit of the **PidLidClientIntent** property (section 2.2.2.4) of the Meeting object.

3.1.4.8.3 Declining a Meeting

When the attendee or a delegate of the attendee declines a Meeting Request object, the client ensures that the Meeting object has been created in the attendee's Calendar special folder, as specified in section 3.1.4.7.2.2. Similarly, when the attendee or delegate of the attendee declines a Meeting Update object, the client ensures that the Meeting object has been updated in the attendee's Calendar special folder, as specified in section 3.1.4.7.4.2, unless the Meeting Update object is out-of-date, as specified in section 3.1.5.2, in which case the client MUST NOT modify the Meeting object and MUST NOT send a Meeting Response object.

After creating or updating the Meeting object, the client applies the following changes to the Meeting object in the attendee's Calendar special folder:

- If the value of the **PidLidReminderSet** property ([MS-OXORMDR] section 2.2.1.1) is set to TRUE, the Meeting object is not a recurring series, and the **signal time** has passed, set the value of the **PidLidReminderSet** property to FALSE.
- Set the value of the **PidLidResponseStatus** property (section 2.2.1.11) to "respDeclined" (0x00000004).
- Set the value of the **PidLidAppointmentReplyTime** property (section 2.2.4.3) to the current date and time.
- If the delegate is responding, set the value of the **PidLidAppointmentReplyName** property (section 2.2.4.5) equal to the value of the **PidTagMailboxOwnerName** property ([MS-OXCSTOR] section 2.2.2.1) from the message store. If the delegate is not the one who is responding, the **PidLidAppointmentReplyName** property is not set.
- If the delegate is acting on behalf of the delegator, the client SHOULD set the value of the **PidLidOriginalStoreEntryId** property (section 2.2.4.9) to the EntryID of the delegator's message store.

The following additional actions are performed by the client:

- If the Meeting Request object or Meeting Update object represents either a recurring series or single-instance meeting, the client removes the Meeting object from the attendee's calendar, either by moving the Meeting object to the **Deleted Items folder**, as specified in [MS-OXOSFLD] section 2.2, or by permanently deleting the object.
- If the Meeting Request object or Meeting Update object represents an exception to a recurring series, the client removes the Exception Attachment object from the recurring series, as specified in section 3.1.4.6.4.
- If the delegator or a delegate acting on behalf of the delegator declines a meeting, a tombstone SHOULD be added to the **PidTagScheduleInfoAppointmentTombstone** property (section 2.2.12.5) on the delegator's Delegate Information object, as specified in section 3.1.5.6.

The client can send a Meeting Response object back to the organizer, as specified in section 3.1.4.8.4.

When sending a Meeting Response object to the organizer, the client performs the following additional actions:

- Set the **ciRespondedDecline** bit of the **PidLidClientIntent** property (section 2.2.2.4) of the Meeting object.
- If the Meeting object represents an exception to a recurring series, then set the **ciRespondedExceptionDecline** bit of the **PidLidClientIntent** property of the Meeting object that represents the recurring series for the exception.

When not sending a Meeting Response object, the client performs the following additional actions:

- Set the **ciDeletedWithNoResponse** bit of the **PidLidClientIntent** property of the Meeting object.
- If the Meeting object represents an exception to a recurring series, then set the **ciDeletedExceptionWithNoResponse** bit of the **PidLidClientIntent** property of the Meeting object that represents the recurring series for the exception.

3.1.4.8.4 Sending a Meeting Response

After choosing a response, an attendee or a delegate of the attendee sends a Meeting Response object to inform the organizer of the attendee's choice. The client SHOULD NOT send a Meeting Response object if one of the following conditions is true:

- The attendee is also the meeting organizer.<67>
- The value of the **PidTagResponseRequested** property ([MS-OXOMSG] section 2.2.1.46) on the Meeting Request object is set to FALSE. (An example of why this property might be set to FALSE is the case in which a very large number of attendees are invited to a meeting and the organizer does not want her **Inbox folder** flooded with replies.)

If the following condition is true, the client can require sending a Meeting Response object to the organizer:

- The **auxApptFlagForceMtgResponse** bit is set on the **PidLidAppointmentAuxiliaryFlags** property (section 2.2.1.3) of the Meeting object (which came from the Meeting Request object or Meeting Update object).

Beyond these constraints, the client can send a Meeting Response object to the organizer to inform the organizer of the attendee's choice. To do so, the client creates and submits a new Meeting Response object. The client then copies the following properties from the Meeting object to the Meeting Response object:<68>

- **PidLidLocation** (section 2.2.1.4)
- **PidLidWhere** (section 2.2.5.3)
- **PidLidAppointmentSequence** (section 2.2.1.1)
- **PidLidOwnerCriticalChange** (section 2.2.1.34)
- **PidTagStartDate** (section 2.2.1.30)
- **PidTagEndDate** (section 2.2.1.31)
- **PidLidAppointmentStartWhole** (section 2.2.1.5)

- **PidLidAppointmentEndWhole** (section 2.2.1.6)
- **PidLidGlobalObjectId** (section 2.2.1.27)
- **PidLidIsException** (section 2.2.1.35)
- **PidTagOwnerAppointmentId** (section 2.2.1.29)
- **PidTagSensitivity** ([MS-OXCMSG] section 2.2.1.13)

In addition to these properties, if the Meeting Response object represents a recurring series, the client MUST copy the following properties from the Meeting object:<69>

- **PidLidTimeZoneStruct** (section 2.2.1.39)
- **PidLidAppointmentRecur** (section 2.2.1.44)
- **PidLidAppointmentTimeZoneDefinitionRecur** (section 2.2.1.41)
- **PidLidIsRecurring** (section 2.2.1.13)
- **PidLidTimeZone** (section 2.2.5.6)
- **PidLidTimeZoneDescription** (section 2.2.1.40)

The client MUST also set the following on the Meeting Response object:

- The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) as specified in section 2.2.7.1.
- The value of the **PidTagIconIndex** property as specified in section 2.2.1.49.
- The value of the **PidLidAttendeeCriticalChange** property (section 2.2.5.2) to the current date and time.
- The value of the **PidTagSubjectPrefix** property ([MS-OXCMSG] section 2.2.1.9) to indicate the response type.
- Increment the value of the **PidTagConversationIndex** property, as specified in [MS-OXOMSG] section 2.2.1.3.
- The value of the **PidTagSentRepresentingName** property ([MS-OXOMSG] section 2.2.1.57) to the value of the **PidTagMailboxOwnerName** property ([MS-OXCSTOR] section 2.2.2.1) from the user's mailbox (for example, a delegate acting on behalf of the delegator would write the name of the delegate).
- The value of the **PidTagSentRepresentingEntryId** property ([MS-OXOMSG] section 2.2.1.56) to the value of the **PidTagMailboxOwnerEntryId** property ([MS-OXCSTOR] section 2.2.2.1) from the user's mailbox.
- The value of the **PidLidIsSilent** property (section 2.2.7.7) to TRUE if the user did not write any text in the body of the response.

3.1.4.8.4.1 Proposing a New Time

Along with the response, whether Accept, Tentatively Accept, or Decline, the attendee or a delegate of the attendee can request that the organizer change the meeting date and/or time. The client

MUST NOT allow the attendee or delegate of the attendee to propose a new date or time in the following cases:

- The attendee is the organizer.
- The value of the **PidLidAppointmentNotAllowPropose** property (section 2.2.1.26) on the Meeting Request object is set to TRUE.
- The Meeting Request object represents a recurring series. (However, the attendee can propose a new date and/or time for a single instance of a recurring series.)

To propose the new date and/or time, the client sets the following properties on the Meeting Response object:

- The value of the **PidTagSubjectPrefix** property ([MS-OXCMSG] section 2.2.1.9) to propose a new date and/or time.
- The value of the **PidLidAppointmentCounterProposal** property (section 2.2.4.7) to TRUE.
- The value of the **PidLidAppointmentProposedStartWhole** property (section 2.2.7.3) to the proposed new start date and time, in UTC.
- The value of the **PidLidAppointmentProposedEndWhole** property (section 2.2.7.4) to the proposed new end date and time, in UTC.
- The value of the **PidLidAppointmentProposedDuration** property (section 2.2.7.5) to the proposed new duration, in minutes.

In addition to the previous information, when proposing a new date and/or time, the client MUST NOT set the value of the **PidLidIsSilent** property (section 2.2.7.7) to TRUE, even if the attendee does not edit the body of the response.

3.1.4.8.5 Receiving a Meeting Response

After receiving a Meeting Response object, the client determines, as specified in section 3.1.4.8.5.1, whether to record the attendee's response on the Meeting object in the organizer's Calendar special folder. If the client determines that the attendee's response needs to be recorded, it records the response as specified in section 3.1.4.8.5.2. If the **PiAutoProcess** setting, as specified in [MS-OXOCFG] section 2.2.5.1.1, in the calendar options dictionary is set to FALSE, the client SHOULD NOT<70> immediately record the response but instead wait until the user views the Meeting Response object. A client that does not support the calendar options dictionary MAY have its own defined mechanism for allowing the user to decide whether meeting responses will be automatically recorded upon receipt of a Meeting Response object.

3.1.4.8.5.1 Deciding to Record the Response

If any one of the following conditions is met, the client does not record the response for the attendee on the organizer's Meeting object:

- The Meeting Response object is located in the Sent Items folder or the Outbox folder, as specified in [MS-OXOSFLD] section 2.2.
- The value of the **PidTagProcessed** property (section 2.2.5.7) on the Meeting Response object is set to TRUE.

The client SHOULD NOT<71> record the response for the attendee when the value of the **PidLidServerProcessed** property (section 2.2.5.4) on the Meeting Response object is set to TRUE

and the **PidLidServerProcessingActions** property (section 2.2.5.5) either does not exist or has the **cpsUpdatedCalItem** bit of this property set.<72>

3.1.4.8.5.2 Recording the Response

Once a client determines that a response needs to be recorded on the Meeting object, it finds the Calendar object, as specified in section 3.1.5.1, and checks whether the Meeting Response object represents an exception to a recurring series. If it does, and the recurring series is found in the calendar but there is no Exception Attachment object for this instance, the client verifies the following:

- If the instance was previously deleted from the recurring series on the organizer's Meeting object, the client MUST NOT re-create the Exception Attachment object on the organizer's Meeting object just to record the response. Instead, the response is discarded.
- If the instance exists on the organizer's Meeting object but is not an exception, the Exception Attachment object is created on the organizer's Meeting object so that the response can be recorded.

If the Meeting Response object is out-of-date, as specified in section 3.1.5.2, the client SHOULD<73> set the value of the **PidLidPromptSendUpdate** property (section 2.2.7.8) on the Meeting Response object to TRUE and SHOULD<74> verify that a **RecipientRow** structure, as specified in [MS-OXCDATA] section 2.8.3, exists for the attendee, but the response MUST NOT be recorded.

To verify that a **RecipientRow** structure exists for the attendee, the client needs to find the **RecipientRow** structure that corresponds to the attendee in the organizer's Meeting object. If the client cannot find a **RecipientRow** structure for the attendee, it SHOULD add a **RecipientRow** structure for the attendee as an optional attendee unless the Meeting Response object is out-of-date, in which case the attendee SHOULD NOT be added as an optional attendee. If a **RecipientRow** structure for the attendee already exists, and the value of the **PidTagRecipientTrackStatusTime** property (section 2.2.4.10.3) from the **RecipientRow** structure is a time that is later than the value of the **PidLidAttendeeCriticalChange** property (section 2.2.5.2) on the Meeting Response object, the response from the Meeting Response object is not recorded.<75>

To record the response, the client sets the following on the **RecipientRow** structure:

- The value of the **PidTagRecipientTrackStatus** property (section 2.2.4.10.2) to the appropriate value from the response table specified in section 2.2.1.11, according to the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) on the Meeting Response object.

PidTagMessageClass value	PidTagRecipientTrackStatus value
"IPM.Schedule.Meeting.Resp.Pos"	respAccepted (0x00000003)
"IPM.Schedule.Meeting.Resp.Tent"	respTentative (0x00000002)
"IPM.Schedule.Meeting.Resp.Neg"	respDeclined (0x00000004)

- The value of the **PidTagRecipientTrackStatusTime** property (section 2.2.4.10.3) to the value of the **PidLidAttendeeCriticalChange** property from the Meeting Response object.<76>
- The **recipExceptionalResponse** bit in the **PidTagRecipientFlags** property (section 2.2.4.10.1), if the Meeting Response object represents an exception to a recurring series.

Regardless of whether the Meeting Response object proposes a new date or time, additional properties might need to be set. For more details about proposals for a new date or time, see section 3.1.4.8.5.3.

After recording the response, the client SHOULD<77> delete the Meeting Response object if the value of the **PidLidIsSilent** property (section 2.2.7.7) is set to TRUE and the **piAutoDeleteReceipts** value in the calendar options dictionary, as specified in [MS-OXOCFG] section 2.2.5.1.1, is set to TRUE. A client that does not support the calendar options dictionary MAY have its own defined mechanism for allowing the user to decide whether to automatically delete Meeting Response objects on which the **PidLidIsSilent** property is set to TRUE.

3.1.4.8.5.3 Handling New Date/Time Proposals

When the value of the **PidLidAppointmentCounterProposal** property (section 2.2.4.7) on the Meeting Response object is set to TRUE, the attendee is proposing a new date and/or time. When this is the case, the client takes the following additional actions:

- Set the value of the **PidTagRecipientProposed** property (section 2.2.4.10.4) to TRUE in the **RecipientRow** structure, as specified in [MS-OXCDATA] section 2.8.3, for the attendee.
- Set the value of the **PidTagRecipientProposedStartTime** property (section 2.2.4.10.5) in the **RecipientRow** structure for the attendee equal to the value of the **PidLidAppointmentProposedStartWhole** property (section 2.2.7.3) from the Meeting Response object.
- Set the value of the **PidTagRecipientProposedEndTime** property (section 2.2.4.10.6) in the **RecipientRow** structure for the attendee equal to the value of the **PidLidAppointmentProposedEndWhole** property (section 2.2.7.4) from the Meeting Response object.
- Set the value of the **PidLidAppointmentCounterProposal** property on the organizer's Meeting object to TRUE.
- If it is the first time this attendee has proposed a new date or time, increment the value of the **PidLidAppointmentProposalNumber** property (section 2.2.4.6) on the organizer's Meeting object by 0x00000001. If this property did not previously exist on the organizer's Meeting object, it MUST be set to the value of 0x00000001.

If a Meeting Response object is received without a proposal for a new date or time and an attendee previously proposed a new date or time (for example, the value of the **PidTagRecipientProposed** property (section 2.2.4.10.4) in the **RecipientRow** structure for the attendee is already set to TRUE), and the new Meeting Response object does not have the **PidLidAppointmentCounterProposal** property set to TRUE, the client takes the following actions to undo the previous counter proposal:

- Set the value of the **PidTagRecipientProposed** property (section 2.2.4.10.4) to FALSE in the **RecipientRow** structure for the attendee.
- Decrement the value of the **PidLidAppointmentProposalNumber** property on the organizer's Meeting object by 1.
- If the value of the **PidLidAppointmentProposalNumber** property becomes zero (meaning no other attendees have proposed a new date or time), set the value of the **PidLidAppointmentCounterProposal** property on the organizer's Meeting object to FALSE.

When an organizer accepts a counter proposal, an updated meeting request is sent, as specified in section 3.1.4.7.3.

3.1.4.9 Meeting Cancellations

3.1.4.9.1 Sending a Meeting Cancellation

The organizer or delegate of the organizer sends a Meeting Cancellation object to inform attendees that they no longer need to attend the event. To send a Meeting Cancellation object, the client creates and submits a new Meeting Cancellation object. The client then copies all properties from the Meeting object to the Meeting Cancellation object, with the exception/addition of those specified in section 2.2.8.

The client modifies the sequence number (2), as specified in section 2.2.8.

The client sets the following on the Meeting Cancellation object:

- All the bits in the value of the **PidLidAppointmentStateFlags** property (section 2.2.1.10) that are set in this value on the Meeting object, and the **asfReceived** and **asfCanceled** bits.
- The value of the **PidLidResponseStatus** property (section 2.2.1.11) to "respNotResponded" (0x00000005).
- The value of the **PidLidIntendedBusyStatus** property (section 2.2.6.4) to "olFree" (0x00000000).
- The value of the **PidLidBusyStatus** property (section 2.2.1.2) to "olFree".
- The value of the **PidLidFExceptionalAttendees** property (section 2.2.2.3) to FALSE.
- The value of the **PidLidFExceptionalBody** property (section 2.2.10.2.6) to FALSE.
- The **PidTagProcessed** property (section 2.2.5.7) MUST NOT be set.
- The value of the **PidTagSubjectPrefix** property ([MS-OXCMSG] section 2.2.1.9), as specified in section 2.2.8.1.

The following optional properties are also set on the Meeting Cancellation object:

- **PidLidAllAttendeesString** (section 2.2.1.16).
- **PidLidToAttendeesString** (section 2.2.1.17).
- **PidLidCcAttendeesString** (section 2.2.1.18).
- **PidTagStartDate** (section 2.2.1.30).
- **PidTagEndDate** (section 2.2.1.31).

If the user has not modified the value of the **PidLidReminderDelta** property ([MS-OXORMDR] section 2.2.1.3) from its default value (as defined by the client), the value of this property SHOULD be set to the **LONG** value ([MS-DTYP]) 0x5AE980E1.

After successfully sending a Meeting Cancellation object, the client modifies the Meeting object in the organizer's Calendar folder:

- Set the value of the **PidLidToAttendeesString** property equal to the value that was set on the Meeting Cancellation object.
- Set the value of the **PidLidCcAttendeesString** property equal to the value that was set on the Meeting Cancellation object.

- Set the **ciCanceled** bit of the **PidLidClientIntent** property (section 2.2.2.4) of the Meeting object.
- If the Meeting object represents an exception to a recurring series, set the **ciExceptionCanceled** bit of the **PidLidClientIntent** property of the Meeting object that represents the recurring series for the exception.

3.1.4.9.1.1 Partial Attendee List

When the organizer or delegate of the organizer removes attendees from the Meeting object, the client sends a Meeting Cancellation object to the attendees that were removed but does not send a Meeting Cancellation object to any other attendees. If the organizer or delegate has changed the value of the **recipSendable** bit of the **PidTagRecipientFlags** property (section 2.2.4.10.1) of any attendees to 0, the client SHOULD<78> send a cancellation to those attendees.

When sending a cancellation for a recurring series, the client removes the recipient rows corresponding to the attendees receiving cancellations from the Meeting object's recipient rows.

When sending a cancellation for an exception to a recurring series that is not a deleted exception, the client sets the **recipExceptionalDeleted** bit of the **PidTagRecipientFlags** property (section 2.2.4.10.1) for each recipient row of the Exception Embedded Message object (as specified in section 2.2.10.2) corresponding to the attendee receiving the cancellation.

3.1.4.9.1.2 Cancelling a Recurring Series

After a Meeting Cancellation object is sent to all attendees for a recurring series that has exceptions, the client sends a Meeting Cancellation object for each exception whose start date and time (according to the **PidLidAppointmentStartWhole** property (section 2.2.1.5) on the Exception Embedded Message object, as specified in section 2.2.10.2) has not yet passed. The Meeting Cancellation object for each exception conforms to the specifications in section 2.2.8.

If the series has deleted exceptions, the client SHOULD NOT<79> send a Meeting Cancellation object for each deleted exception for which the start date and time (according to the **DeletedInstanceDates** field of the **PidLidAppointmentRecur** property (section 2.2.1.44) of the Meeting object) have not yet passed.

After a Meeting Cancellation object is sent to a partial attendee list, as specified in section 3.1.4.9.1.1, the client SHOULD<80> send a meeting cancellation for each exception whose start date and time has not yet passed to every attendee of the exception that is also in the partial attendee list. If sending a meeting cancellation for an exception, the client sets the **recipExceptionalDeleted** bit of the **PidTagRecipientFlags** property (section 2.2.4.10.1) for each removed attendee.

3.1.4.9.2 Receiving a Meeting Cancellation

After receiving a Meeting Cancellation object, the client determines, as specified in section 3.1.4.9.2.1, whether to update the Meeting object in the user's Calendar special folder with the information in the Meeting Cancellation object. If the client determines that the Meeting object needs to be updated, it updates the object as specified in section 3.1.4.9.2.2. If the **PiAutoProcess** setting, as specified in [MS-OXOCFG] section 2.2.5.1.1, in the calendar options dictionary is set to 0 (zero), the client SHOULD NOT<81> immediately update the Meeting object but wait until the user views the Meeting Cancellation object. A client that does not support the calendar options dictionary MAY have its own defined mechanism for allowing the user to decide whether Meeting objects will be automatically updated upon receipt of a Meeting Cancellation object.

3.1.4.9.2.1 Deciding to Update a Meeting Object

When a delegator receives a Meeting Cancellation object, the client MUST follow the sequencing rules described in section 3.1.5.6 before automatically updating the Meeting object.

If any one of the following conditions is met, the client does not automatically update the Meeting object:

- The Meeting Cancellation object is located in the Sent Items folder or the Outbox folder, as specified in [MS-OXOSFLD] section 2.2.
- The value of the **PidTagProcessed** property (section 2.2.5.7) on the Meeting Cancellation object is set to TRUE.
- The client MAY<82> skip automatic updating of the Meeting object if the value of the **PidLidServerProcessed** property (section 2.2.5.4) on the Meeting Cancellation object is set to TRUE and the Meeting Cancellation object's **PidLidServerProcessingActions** property (section 2.2.5.5) either does not exist or the **cpsUpdatedCalItem** bit of this property is set. If the client does not automatically update the Meeting object, it MUST NOT set the **PidTagProcessed** property on the Meeting Cancellation object.

If the client determines that the Meeting object is to be updated, it first tries to find the Calendar object, as specified in section 3.1.5.1. If the Meeting Cancellation object represents an exception to a recurring series, and the recurring series was found in the calendar but the exception was previously deleted from the recurring series, the client SHOULD NOT<83> re-create the Exception Attachment object and the Exception Embedded Message object, as specified in section 2.2.10.2, on the recurring Meeting object. If the Meeting object was not found at all, the client SHOULD NOT<84> re-create it.

If the Meeting Cancellation object is out-of-date, as specified in section 3.1.5.2, the client SHOULD change the value of the **PidLidMeetingType** property (section 2.2.6.5) on the Meeting Cancellation object to **mtgOutofDate** (0x00080000) but does not update the Meeting object. Similarly, if the Meeting Cancellation object is not newer than the Meeting object, as specified in section 3.1.5.3, the client does not update the Meeting object.

3.1.4.9.2.2 Updating the Meeting Object

To update the Meeting object, the client copies all the properties specified in 2.2.1 from the Meeting Update object onto the Meeting object.

After updating the Meeting object, the client SHOULD set the value of the **PidTagProcessed** property (section 2.2.5.7) to TRUE, unless the object is in a public folder, in which case this property is not set.<85>

3.1.4.10 Meeting Forward Notifications

3.1.4.10.1 Sending a Meeting Forward Notification

When a Meeting Request object is forwarded, as specified in section 3.1.4.7.5, the client can send a Meeting Forward Notification object to the organizer. The client does not send a Meeting Forward Notification object if one of the following conditions is true:

- The **PidTagAddressType** property ([MS-OXOABK] section 2.2.3.13) of the organizer's Address Book object is not equal to "EX".

- The **PidTagAddressType** property of the organizer's Address Book object is equal to "EX", but the **PidLidGlobalObjectId** property (section 2.2.1.27) is of type **ThirdPartyGlobalId**, as specified in [MS-OXCICAL] section 2.1.3.1.1.20.26.
- The version number returned by the server in either the **EcDoConnectEx** method, as specified in [MS-OXCRPC], or the **X-ServerApplication** header of the **Connect** request type response,<86> as specified in [MS-OXCMAPIHTTP], is greater than or equal to 8.0.0.0.

The client SHOULD NOT send a Meeting Forward Notification object if the following condition is true:

- The **asfReceived** bit of the **PidLidAppointmentStateFlags** property (section 2.2.1.10) of the corresponding Meeting object is not set.

To notify the organizer of the new attendees, the client creates and submits a new Meeting Forward Notification object. The client MUST copy the following properties from the Meeting object to the Meeting Forward Notification object:<87>

- **PidNameSubject** ([MS-OXODOC] section 2.2.1.2)
- **PidLidLocation** (section 2.2.1.4)
- **PidLidWhere** (section 2.2.5.3)
- **PidLidAppointmentSequence** (section 2.2.1.1)
- **PidLidOwnerCriticalChange** (section 2.2.1.34)
- **PidTagStartDate** (section 2.2.1.30)
- **PidTagEndDate** (section 2.2.1.31)
- **PidLidAppointmentStartWhole** (section 2.2.1.5)
- **PidLidAppointmentEndWhole** (section 2.2.1.6)
- **PidLidGlobalObjectId** (section 2.2.1.27)
- **PidLidCleanGlobalObjectId** (section 2.2.1.28)
- **PidLidIsException** (section 2.2.1.35)
- **PidTagOwnerAppointmentId** (section 2.2.1.29)
- **PidTagSensitivity** ([MS-OXCMSG] section 2.2.1.13)
- **PidTagResponseRequested** ([MS-OXOMSG] section 2.2.1.46)

In addition to these properties, if the forwarded Meeting Request object represents a recurring series, the client copies the following properties from the Meeting object to the Meeting Forward Notification object:<88>

- **PidLidTimeZoneStruct** (section 2.2.1.39)
- **PidLidAppointmentRecur** (section 2.2.1.44)
- **PidLidAppointmentTimeZoneDefinitionRecur** (section 2.2.1.41)
- **PidLidIsRecurring** (section 2.2.1.13)

- **PidLidTimeZone** (section 2.2.5.6)
- **PidLidTimeZoneDescription** (section 2.2.1.40)

The client MUST also set the following on the Meeting Forward Notification object:

- The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) as specified in section 2.2.9.1.
- The value of the **PidTagIconIndex** property as specified in section 2.2.1.49.
- The value of the **PidLidAttendeeCriticalChange** property (section 2.2.5.2) to the current date and time.
- The value of the **PidTagSubjectPrefix** property ([MS-OXCMSG] section 2.2.1.9) as specified in section 2.2.9.2.
- Increment the value of the **PidTagConversationIndex** property as specified in [MS-OXOMSG] section 2.2.1.3.
- The value of the **PidTagSentRepresentingName** property ([MS-OXOMSG] section 2.2.1.57) to the value of the **PidTagMailboxOwnerName** property ([MS-OXCSTOR] section 2.2.2.1) from the user's mailbox (for example, a delegate acting on behalf of the delegator would write the name of the delegate).
- The value of the **PidTagSentRepresentingEntryId** property ([MS-OXOMSG] section 2.2.1.56) to the value of the **PidTagMailboxOwnerEntryId** property ([MS-OXCSTOR] section 2.2.2.1) from the user's mailbox.

In addition, the client copies each **RecipientRow** structure, as specified in [MS-OXCDATA] section 2.8.3, with the **recipSendable** bit set in the **PidTagRecipientFlags** property (section 2.2.4.10.1) from the forwarded Meeting Request object's **RecipientRow** structures to the **PidLidForwardNotificationRecipients** property (section 2.2.9.3) on the Meeting Forward Notification object.

3.1.4.10.2 Receiving a Meeting Forward Notification

After receiving a Meeting Forward Notification object, the client determines, as specified in section 3.1.4.8.5.1, whether to add the attendees included in the Meeting Forward Notification object to the Meeting object. If the client adds the attendees to the Meeting object, it MUST do so as specified in section 3.1.4.10.2.2. If the **PiAutoProcess** setting, as specified in [MS-OXOCFG] section 2.2.5.1.1, in the calendar options dictionary is set to 0 (zero), then the client SHOULD NOT<89> immediately add the forwarded attendees to the Meeting object but instead wait until the user views the Meeting Forward Notification object. A client that does not support the calendar options dictionary can have its own defined mechanism for allowing the user to decide whether forwarded attendees will be copied to the Meeting object upon receipt of a Meeting Forward Notification object.

3.1.4.10.2.1 Deciding to Add the Forwarded Attendees to the Meeting Object

If any one of the following conditions is met, the client MUST NOT record the attendee on the organizer's Meeting object:

- The Meeting Forward Notification object is located in the Sent Items folder or the Outbox folder, as specified in [MS-OXOSFLD] section 2.2.
- The value of the **PidTagProcessed** property (section 2.2.5.7) on the Meeting Forward Notification object is set to TRUE.

- The **cpsProcessedMeetingForwardNotification** bit of the **PidLidServerProcessingActions** property (section 2.2.5.5) of the Meeting Forward Notification object is set to 1.

3.1.4.10.2.2 Adding the Forwarded Attendees to the Meeting Object

If the client is adding the forwarded attendees to the Meeting object, it MUST find the Calendar object, as specified in section 3.1.5.1. If the Meeting Forward Notification object represents an exception to a recurring series and the recurring series was found in the calendar but it does not have an Exception Attachment object for this instance, one of two actions might need to be taken:

- If the instance was previously deleted from the recurring series on the organizer's Meeting object, the client MUST NOT re-create the Exception Attachment object on the organizer's Meeting object just to add the attendee.
- If the instance exists on the organizer's Meeting object but is not an exception, the Exception Attachment object MUST be created on the organizer's Meeting object so that the response can be recorded.

To add the forwarded attendees to the Meeting object, the client MUST copy each **RecipientRow** structure, as specified in [MS-OXCDATA] section 2.8.3, in the **PidLidForwardNotificationRecipients** property (section 2.2.9.3) of the Meeting Forward Notification object to the **RecipientRow** structures of the Meeting object if and only if the following conditions are met:

- The value of the **RecipientRow** structure's recipient type is not 0x03.
- The recipient (2) already exists in the Meeting object's **RecipientRow** structures according to the value of the **PidTagEntryId** property ([MS-OXCPERM] section 2.2.4).

If the client copies a **RecipientRow** structure and the recipient type of the **RecipientRow** structure is 0x01, the client MUST set the recipient type of the corresponding **RecipientRow** structure on the Meeting object to 0x02.

If the Meeting Forward Notification object is out-of-date as specified in section 3.1.5.2, the client sets the value of the **PidLidPromptSendUpdate** property (section 2.2.7.8) to TRUE.

After copying the forwarded attendees to the Meeting object, the client MUST set either the **PidTagProcessed** property (section 2.2.5.7) or the **PidLidServerProcessed** property (section 2.2.5.4) of the Meeting Forward Notification object to TRUE. If the client sets the **PidLidServerProcessed** property, the client MUST set the **cpsProcessedMeetingForwardNotification** bit of the **PidLidServerProcessingActions** property (section 2.2.5.5) of the Meeting Forward Notification object to 1.

3.1.4.11 Determining Meeting Conflicts

To determine whether a meeting conflicts with another meeting, follow these steps:

- Build a list of meetings that are in the range. Determine the range by using the start and end date and time of the given meeting as the start and end of the range. Any meeting for which the end date and time is greater than or equal to the start date and time of the given meeting and the start date and time is less than or equal to the end date and time of the given meeting is considered to be in conflict.
- Expand any recurring meetings. For details about how to expand recurring meetings, see section 3.1.4.6. If multiple instances or exceptions fall into the range, each of them is considered as a single-instance meeting for the purpose of this algorithm.

If the size of the list is greater than or equal to 1, the given meeting is considered to be in conflict.

3.1.4.12 Modifying a Meeting Object as an Attendee

If the user is modifying a Meeting object, and the **asfReceived** flag of the **PidLidAppointmentStateFlags** property (section 2.2.1.10) of the Meeting object is set, then the client takes the following additional actions:

- If the user is modifying the **PidLidAppointmentStartWhole** property (section 2.2.1.5), the client SHOULD set the **ciModifiedStartTime** flag of the **PidLidClientIntent** property (section 2.2.2.4) on the Meeting object.
- If the user is modifying the **PidLidAppointmentEndWhole** property (section 2.2.1.6), the client SHOULD set the **ciModifiedEndTime** flag of the **PidLidClientIntent** property on the Meeting object.
- If the user is modifying the **PidLidLocation** property (section 2.2.1.4), the client SHOULD set the **ciModifiedLocation** flag of the **PidLidClientIntent** property on the Meeting object.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Finding the Calendar Object

Several actions require finding the Calendar object to which a meeting-related object is referring. To find Calendar objects, the client searches in the Calendar special folder of the mailbox that the event was intended for. This is typically the mailbox of the user who is logged on, but for the delegate, the client searches the delegator's folder for objects received on behalf of the delegator.

To look for the object, the client first looks for a Calendar object for which the **PidLidGlobalObjectId** property (section 2.2.1.27) matches the value of the **PidLidCleanGlobalObjectId** property (section 2.2.1.28) of the meeting-related object.

If the action is being applied to an exception to a recurring series, the following additional operations are required, depending on whether a matching recurring series object was found:

- If a recurring series object was found, the client attempts to find the Exception Attachment object within a Calendar object by comparing the value of the **PidLidExceptionReplaceTime** property (section 2.2.10.2.5) from the meeting-related object with either the **PidTagExceptionReplaceTime** property (section 2.2.10.1.6) on the Exception Attachment object or the **PidLidExceptionReplaceTime** property on the Exception Embedded Message object, as specified in section 2.2.10.2. Note that the **PidTagExceptionReplaceTime** property will not always be present on the Exception Attachment object. In the case where the Exception Attachment object cannot be found, the client can create a new one.
- If the recurring series object was not found, the client looks for a recurring series object for which the **PidLidGlobalObjectId** property matches the value of the **PidLidGlobalObjectId** property of the meeting-related object. This would be the case, for example, if a user has been invited only to an exception to a recurring series.

3.1.5.2 Out-of-Date Meetings

A Meeting Request object or Meeting Update object becomes out-of-date when a more recent version is received and processed. A Meeting Response object is out-of-date when the attendee responds to an older Meeting Request object or Meeting Update object, instead of the most current Meeting Update object.

This section specifies how the client can determine whether the Meeting Request object or Meeting Response object is out-of-date. If one of the following conditions is true, the Meeting Request object or Meeting Response object is considered to be out-of-date:

- The value of the **PidLidMeetingType** property (section 2.2.6.5) on the Meeting Request object is set to mtgOutofDate.
- The sequence number (2) of the Meeting object is greater than that of the Meeting Request object or Meeting Response object.
- The sequence number (2) of the Meeting object is the same as that of the Meeting Request object or Meeting Response object, but the value of the **PidLidOwnerCriticalChange** property (section 2.2.1.34) on the Meeting Request object or Meeting Response object is earlier than the request time of the Meeting object, where the request time is determined as shown in the following table.

Recipient	Property that specifies request time
Organizer	**PidLidAppointmentSequenceTime** (section 2.2.4.1)*
Attendees	**PidLidOwnerCriticalChange**

*If **PidLidAppointmentSequenceTime** (section 2.2.4.1) does not exist on the Organizer's item, the request time is the value of the associated object's **PidLidOwnerCriticalChange** property.

The value of the **PidLidAttendeeCriticalChange** property (section 2.2.5.2) on the Meeting Response object is less than the value of the **PidTagRecipientTrackStatusTime** property (section 2.2.4.10.3) on the **RecipientRow** structure ([MS-OXCDATA] section 2.8.3) of the organizer's Meeting object that represents the attendee.

3.1.5.3 Newer Meetings

A Meeting Request object or Meeting Cancellation object is considered to be from a newer version of the organizer's Meeting object than the Meeting object on the attendee's calendar if one of the following conditions is true:

- The sequence number (2) on the Meeting Request object or Meeting Cancellation object is greater than the sequence number (2) on the Meeting object.
- The sequence number (2) on the Meeting Request object or Meeting Cancellation object equals the sequence number (2) on the Meeting object, but the value of the **PidLidOwnerCriticalChange** property (section 2.2.1.34) on the Meeting Request object or Meeting Cancellation object is greater than that of the Meeting object.

3.1.5.4 Incrementing the Sequence Number

When sending a Meeting Update object or Meeting Cancellation object, the client increments the sequence number (2) except when sending a Meeting Cancellation object for a deleted exception after sending a Meeting Update object for a recurring series (see section 3.1.4.7.5).<90>

If not incrementing the sequence number (2), the client sets the value of the **PidLidAppointmentSequence** property (section 2.2.1.1) on the Meeting Update object or Meeting Cancellation object equal to the value of the **PidLidAppointmentLastSequence** property (section 2.2.4.2) of the Meeting object.

When incrementing the sequence number (2), the client sets the sequence number (2) of the Meeting Update object or Meeting Cancellation object to a value greater than the sequence number (2) that was set on any previous Meeting Request object, Meeting Cancellation object, or Meeting Update object. The client selects the greater of **PidLidAppointmentLastSequence** and **PidLidAppointmentSequence** properties from the Meeting object, and increments that value by 1, which results in the new sequence number (2). The client sets the new sequence number (2) as the value of both the **PidLidAppointmentLastSequence** property on the Meeting object and the **PidLidAppointmentSequence** property on the Meeting Request object or the Meeting Cancellation object.

If the Meeting Update object or Meeting Cancellation object is being sent to all attendees of the meeting, the client MUST set the new sequence number (2) as the value of the **PidLidAppointmentSequence** property of the Meeting object and MUST set the **PidLidAppointmentSequenceTime** property (section 2.2.4.1) as the value of the **PidLidOwnerCriticalChange** property (section 2.2.1.34).

If the Meeting Update object or Meeting Cancellation object is not being sent to all attendees of the meeting, the client SHOULD NOT modify the **PidLidAppointmentSequence** property of the Meeting object but SHOULD verify that the **PidLidAppointmentSequenceTime** property exists on the Meeting object. In the case that the **PidLidAppointmentSequenceTime** property does not exist on the Meeting object, the client sets it to the original value of the **PidLidOwnerCriticalChange** property from the Meeting object (the value before the Meeting Update object or Meeting Cancellation object was created).

3.1.5.5 Time Display Adjustments

In some cases, the client needs to adjust the way in which it interprets the **PidLidAppointmentStartWhole** (section 2.2.1.5), **PidLidAppointmentEndWhole** (section 2.2.1.6), and **PidLidReminderSignalTime** ([MS-OXORMDR] section 2.2.1.2) properties. Instead of interpreting these time properties as UTC values, a different process is followed for **floating appointments**, as specified in section 3.1.5.5.1, and **time zone updates**, as specified in section 3.1.5.5.2.

3.1.5.5.1 Data Interpretation for Floating Appointments

The client SHOULD<91> interpret an object as a floating appointment if both of the following conditions are met:

- The value of the **PidLidAppointmentSubType** property (section 2.2.1.9) is TRUE.
- The **asfMeeting** bit in the **PidLidAppointmentStateFlags** property (section 2.2.1.10) is set to 0.

To correctly interpret the floating appointment, the client MUST use the **TZRule** structure that is marked with the **TZRULE_FLAG_EFFECTIVE_TZREG** flag in the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property (section 2.2.1.42) to convert the values of the **PidLidAppointmentStartWhole** (section 2.2.1.5) and **PidLidAppointmentEndWhole** (section 2.2.1.6) properties from UTC to the time zone described by the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property. The client MUST interpret these two time properties at this calculated time regardless of any additional time zone considerations. When performing these calculations, the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property is used for all time properties, including the **PidLidAppointmentEndWhole** property (section 2.2.1.6).

3.1.5.5.2 Data Interpretation for Time Zone Updates

The **TZRule** structure that is marked with the **TZRULE_FLAG_EFFECTIVE_TZREG** flag in the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property (section 2.2.1.42) indicates the **TZRule** structure with which the Appointment object's times were converted to UTC time when the object was created. In some cases, the time zone rule (4) that is in effect for the given time zone will be updated after the object is created.

When the client detects that the time zone rule (4) for the time zone specified by the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property has been updated, the client SHOULD<92> continue to interpret the **PidLidAppointmentStartWhole** (section 2.2.1.5) and **PidLidAppointmentEndWhole** (section 2.2.1.6) properties so that the values occur at the same time that was specified when the object was created. For example, if a user creates an Appointment object to begin at 2:00 P.M. on April 1 in a time zone that has a -8 offset from UTC, the **PidLidAppointmentStartWhole** property would have been saved as 10:00 P.M. UTC. If after creating this object, the time zone specified in the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property is updated such that on April 1 the time zone's offset from UTC is now -7, the object's start time continues to be interpreted as 2:00 P.M. when the value of the **PidLidAppointmentStartWhole** property is converted to that same time zone. The client can detect and perform these calculations using the data specified in the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property. When performing these calculations, the value of the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property is to be used for all time properties, including the **PidLidAppointmentEndWhole** property.

If the object's times are being converted to a time zone that is different from the time zone specified by the **PidLidAppointmentTimeZoneDefinitionStartDisplay** property, the client first converts the **PidLidAppointmentStartWhole** and **PidLidAppointmentEndWhole** properties from UTC to the time zone specified by the effective **TZRule** structure, and then uses the updated time zone rule (4) to convert to an updated UTC time before converting the time to another time zone.

3.1.5.6 Delegator Wants Copy

A value of TRUE for the **PidTagScheduleInfoDelegatorWantsInfo** property ([MS-OXODLGT] section 2.2.2.2.2) on the delegator's Delegate Information object indicates that the delegator only wants to be notified of meetings without taking action on them. When the delegator receives a Meeting Request object or Meeting Cancellation object, the client SHOULD<93> check the value of the **PidTagScheduleInfoDelegatorWantsInfo** property to see whether it is set to TRUE unless one or more of the following conditions are true:

- The value of the **PidLidMeetingType** property (section 2.2.6.5) is **mtgDelegatorCopy** or **mtgOutOfDate**.
- The value of the **PidLidServerProcessed** property (section 2.2.5.4) on the meeting-related object is TRUE and the value of the **cpsDelegatorWantsCopy** bit of the **PidLidServerProcessingActions** property (section 2.2.5.5) on the meeting-related object is set.
- The value of the **PidTagSensitivity** property ([MS-OXCMSG] section 2.2.1.13) is set to "private".

If none of the above conditions is true and the client finds that the value of the **PidTagScheduleInfoDelegatorWantsInfo** property is set to TRUE, the client MUST change the value of the **PidLidMeetingType** property to **mtgDelegatorCopy** and change the value of the **PidTagIconIndex** property (section 2.2.1.49) to 0x00000409.

After checking whether the **PidTagScheduleInfoDelegatorWantsInfo** property is set to TRUE, the client MAY<94> set the **cpsDelegatorWantsCopy** bit of the **PidLidServerProcessingActions** property on the meeting-related object.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

3.2.1 Abstract Data Model

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

The server uses the same ADM as the client, as specified in section 3.1.1. This data model is based on the data model for the Messaging and Attachment Object Protocol, as specified in [MS-OXCMSG] section 3.2.1.

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

This protocol adheres to the messaging processing events and sequencing rules specified in [MS-OXCMSG] and [MS-OXCFOLD].

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Examples of Properties

4.1.1 Recurrence BLOB Examples

Several examples of the **PidLidAppointmentRecur** (section 2.2.1.44) recurrence BLOB are included in sections 4.1.1.1 through 4.1.1.6. The data for the fields of the recurrence BLOB are stored in little-endian byte ordering.

4.1.1.1 Weekly Recurrence BLOB Without Exceptions

The following example shows the binary recurrence data for an appointment that has the following characteristics:

- Beginning on Monday, March 26, 2007, occurs every Monday, Thursday, and Friday from 10:00 A.M. to 10:30 A.M.
- The recurrence ends after 12 occurrences.

The following is the recurrence BLOB.

```
043004300B2001000000C021000001000000000000003200000022200000C000000000000000000000000000000008
020BC0C20ADBC0C06300000093000005802000076020000000000000000000000000
```

The following table lists the content of the recurrence BLOB.

Field name	Type	Size	Example	Description
ReaderVersion	**WORD** ([MS-DTYP])	2	04 30	This field indicates version 0x3004.
WriterVersion	**WORD**	2	04 30	This field indicates version 0x3004.
RecurFrequency	**WORD**	2	0b 20	The pattern of the recurrence is weekly.
PatternType	**WORD**	2	01 00	The pattern type is Week (0x0001).
CalendarType	**WORD**	2	00 00	The calendar type is Gregorian (0x0000).
FirstDateTime	**ULONG** ([MS-DTYP])	4	c0 21 00 00	The number of minutes offset from the reference date, January 1, 1601, is 8640 (0x000021C0), which corresponds to the first ever week of January 7, 1601. See the calculation steps following this table.
Period	**ULONG**	4	01 00 00 00	The recurrence occurs every week (0x0001).
SlidingFlag	**ULONG**	4	00 00 00 00	This field is only used for scheduling tasks. Otherwise, the value can only be zero).
PatternTypeSpecific	BYTE array	Varies	32 00 00 00	The recurring appointment occurs on Monday, Thursday, and Friday. The value is determined by adding together the

Field name	Type	Size	Example	Description
				binary value of the decimal day mask (Sunday = 2^0 = 1, Monday = 2^1 = 2, Tuesday = 2^2 = 4, and so on). Monday (0x00000002) + Thursday (0x0000010) + Friday (0x00000020) = 0x00000032
EndType	**ULONG**	4	22 20 00 00	End after *n* occurrences. (0x00002022)
OccurrenceCount	**ULONG**	4	0C 00 00 00	The recurrence ends after 12 occurrences. 12 decimal value = 0x0C hexadecimal value.
FirstDOW	**ULONG**	4	00 00 00 00	The first day of the week on the calendar is Sunday (the default value).
DeletedInstanceCount	**ULONG**	4	00 00 00 00	There are no deleted instances.
ModifiedInstanceCount	**ULONG**	4	00 00 00 00	There are no modified instances.
StartDate	**ULONG**	4	80 20 BC 0C	The start date of the recurrence given in minutes since midnight January 1, 1601, is 213,655,680 (0x0CBC2080), which corresponds to March 26, 2007, 12:00:00 A.M.
EndDate	**ULONG**	4	20 AD BC 0C	The end date of the recurrence given in minutes since midnight January 1, 1601, corresponds to April 20, 2007, 12:00:00 A.M.
ReaderVersion2	**ULONG**	4	06 30 00 00	This field indicates version 0x00003006.
WriterVersion2	**ULONG**	4	09 30 00 00	This field indicates version 0x00003009.
StartTimeOffset	**ULONG**	4	58 02 00 00	The hexadecimal start time of the recurrence is 0x00000258, which corresponds to 600 in decimal; 600 minutes is 10 hours, which is 10:00 A.M.
EndTimeOffset	**ULONG**	4	76 02 00 00	The hexadecimal end time of the recurrence is 0x000000276, which corresponds to 630 minutes, which is 10:30 A.M.
ExceptionCount	**WORD**	2	00 00	There are no exceptions in this recurrence BLOB.
ReservedBlock1Size	**ULONG**	4	00 00 00 00	There is no data in the reserved block.
ReservedBlock2Size	**ULONG**	4	00 00 00 00	There is no data in the reserved block.

The steps for calculating the value of the **FirstDateTime** field for a weekly recurrence, as specified in section 2.2.1.44.1.1, are applied to this example as follows:

1. The first day of the week that contains the start date: March 25, 2007
2. The number of minutes between midnight, March 25, 2007, and midnight, January 1, 1601: 213,654,240
3. The number of minutes between recurrences: 1 × 10080 = 10080
4. The value of the **FirstDateTime** field: 213,654,240 % 10080 = 8640 (0x000021C0)

4.1.1.2 Weekly Recurrence BLOB with Exceptions

The following example shows the binary recurrence data for a meeting request.

The meeting request is the same as the request that is used in section 4.1.1.1 (occurs every Monday, Thursday, and Friday from 10:00 A.M. to 10:30 A.M., ends after 12 occurrences), but in this example, the following information has been changed in the exception:

- The subject has been changed from "Simple Recurrence" to "Simple Recurrence with exceptions".
- The location has been changed from 34/4639 to 34/4141.
- The start date and time has been modified from Monday 4/16/2007 10:00 A.M. to Monday 4/16/2007 11:00 A.M.
- The end date and time has been modified from Monday 4/16/2007 10:30 A.M. to Monday 4/16/2007 11:30 A.M.

The following is the recurrence BLOB.

```
043004300B2001000000C0210000010000000000000032000000222000000C0000000000000001000000A096BC0C0
1000000A096BC0C8020BC0C20ADBC0C0630000009300000580200007602000001003499BC0C5299BC0CF898BC0C11
00220021005369 6D706C65205265637572656E6365207769746820657863657074696F6E730800070033342F343
134310000000004000000000000000000003499BC0C5299BC0CF898BC0C2100530069006D0070006C0065002000
5200650063007500720072006500 6E006300650020007700690074006800200065007800630065007000740069006
F006E007300070033003400 2F00340031003400310000000000000000
```

Size: 0x0106 bytes

The content of the modified recurrence BLOB is listed in the following table.

Field name	Type	Size	Example	Description
ReaderVersion	**WORD** ([MS-DTYP])	2	04 30	This field indicates version 0x3004.
WriterVersion	**WORD**	2	04 30	This field indicates version 0x3004.
RecurFrequency	**WORD**	2	0b 20	The pattern of the recurrence is weekly.
PatternType	**WORD**	2	01 00	The pattern type is Week (0x0001).
CalendarType	**WORD**	2	00 00	The calendar type is Gregorian (0x0000).

Field name	Type	Size	Example	Description
FirstDateTime	**ULONG** ([MS-DTYP])	4	c0 21 00 00	The number of minutes offset from the reference date, January 1, 1601, is 8640 (0x000021C0), which corresponds to the first ever week of January 7, 1601. See the calculation steps following this table.
Period	**ULONG**	4	01 00 00 00	The recurrence occurs every week (0x0001).
SlidingFlag	**ULONG**	4	00 00 00 00	This field is only used for scheduling tasks. Otherwise, the value can only be 0 (zero).
PatternTypeSpecific	**BYTE** array	Varies	32 00 00 00	The recurring appointment occurs on Monday, Thursday, and Friday. The value is determined by adding together the binary value of the decimal day mask (Sunday = 2^0 = 1, Monday = 2^1 = 2, Tuesday = 2^2 = 4, and so on). Monday (0x00000002) + Thursday (0x0000010) + Friday (0x00000020) = 0x00000032
EndType	**ULONG**	4	22 20 00 00	Ends after *n* occurrences. (0x00002022)
OccurrenceCount	**ULONG**	4	0C 00 00 00	The recurrence ends after 12 occurrences. 12 decimal value = 0x0C hexadecimal value.
FirstDOW	**ULONG**	4	00 00 00 00	The first day of the week on the calendar is Sunday (the default value).
DeletedInstanceCount	**ULONG**	4	01 00 00 00	There is one deleted instance.
DeletedInstanceDate	**ULONG**	4	A0 96 BC 0C	The date of the deleted or modified instance is 4/16/2007 at 12:00:00 A.M.
ModifiedInstanceCount	**ULONG**	4	01 00 00 00	There is one modified instance.
ModifiedInstanceDate	**ULONG**	4	A0 96 BC 0C	The date of the modified instance is 4/16/2007 at 12:00:00 A.M.
StartDate	**ULONG**	4	80 20 BC 0C	The start date of the recurrence given in minutes since midnight January 1, 1601, is 213,655,680 (0x0CBC2080), which corresponds to 3/26/2007, 12:00:00 A.M.
EndDate	**ULONG**	4	20 AD BC 0C	The end date of the recurrence given in minutes since midnight January 1, 1601, corresponds to 4/20/2007, 12:00:00 A.M.

Field name	Type	Size	Example	Description
ReaderVersion2	**ULONG**	4	06 30 00 00	This field indicates version 0x00003006.
WriterVersion2	**ULONG**	4	09 30 00 00	This field indicates version 0x00003009.
StartTimeOffset	**ULONG**	4	58 02 00 00	The hexadecimal start time of the recurrence is 0x00000258, which corresponds to 600 in decimal. 600 minutes is 10 hours, which is 10:00 A.M.
EndTimeOffset	**ULONG**	4	76 02 00 00	The hexadecimal end time of the recurrence is 0x000000276, which corresponds to 630 minutes, which is 10:30 A.M.
ExceptionCount	**WORD**	2	01 00	One exception.
ExceptionInfo structure block				
StartDateTime	**ULONG**	4	34 99 BC 0C	The start date and time of the exception is 4/16/2007 at 11:00:00 A.M.
EndDateTime	**ULONG**	4	52 99 BC 0C	The end date and time of the exception is 4/16/2007 at 11:30:00 A.M.
OriginalStartTime	**ULONG**	4	F8 98 BC 0C	The original start date and time of the modified occurrence was 4/16/2007 at 10:00:00 A.M.
OverrideFlags	**WORD**	2	11 00	A value of 0x0011 indicates that two override flags are present: **ARO_SUBJECT** (0x0001) and **ARO_LOCATION** (0x0010).
SubjectLength	**WORD**	2	22 00	The length of the subject including a null terminator is 34 characters.
SubjectLength2	**WORD**	2	21 00	The length of the subject is 33 characters.
Subject	**BYTE** array	Varies	53 69 6D 70 6C 65 20 52 65 63 75 72 72 65 6E 63 65 20 77 69 74 68 20 65 78 63 65 70 74 69 6F 6E	"Simple Recurrence with exceptions"

Field name	Type	Size	Example	Description
			73	
LocationLength	**WORD**	2	08 00	The length of the location string including a null terminator is 8 characters.
LocationLength2	**WORD**	2	07 00	The length of the location string is 7 characters.
Location	**BYTE** array	Varies	33 34 2F 34 31 34 31	The modified location is "34/4141".
ReservedBlock1Size	**ULONG**	4	00 00 00 00	There is no data in this **skip block**.
ExtendedException structure block				
ChangeHighlight	**BYTE** array	Varies	04 00 00 00 00 00 00 00	The size of the **ChangeHighlight** structure is 4. The value of the **PidLidChangeHighlight** property (section 2.2.6.2) is zero for this exception.
ReservedBlockEE1Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.
StartTime	**ULONG**	4	34 99 BC 0C	The start time of the recurrence is 4/16/2007 at 11:00:00 A.M.
EndTime	**ULONG**	4	52 99 BC 0C	The end time of the recurrence is 4/16/2007 at 11:30:00 A.M.
OriginalStartTime	**ULONG**	4	F8 98 BC 0C	The original start date and time of the recurrence was 4/16/2007 at 10:00:00 A.M.
WideCharSubjectLength	**WORD**	2	21 00	The length of the Unicode subject string is 33 characters.
WideCharSubject	**BYTE** array	Varies	53 00 69 00 6D 00 70 00 6C 00 65 00 20 00 52 00 65 00 63 00 75 00 72 00 72 00 65 00 6E 00 63 00 65 00 20	The modified Unicode subject is "Simple Recurrence with exceptions".

Field name	Type	Size	Example	Description
			00 77 00 69 00 74 00 68 00 20 00 65 00 78 00 63 00 65 00 70 00 74 00 69 00 6F 00 6E 00 73 00	
WideCharLocationLength	**WORD**	2	07 00	The Unicode location string is 7 characters.
WideCharLocation	**BYTE** array	Varies	33 00 34 00 2F 00 34 00 31 00 34 00 31 00	The modified Unicode location is: "34/4141".
ReservedBlockEE2Size	**ULONG**	4	00 00 00 00	No data in this skip block.
ReservedBlock2Size	**ULONG**	4	00 00 00 00	No data in this skip block.

The steps for calculating the value of the **FirstDateTime** field for a weekly recurrence, as specified in section 2.2.1.44.1.1, are applied to this example as follows:

1. The first day of the week that contains the start date: March 25, 2007
2. The number of minutes between midnight, March 25, 2007, and midnight, January 1, 1601: 213,654,240
3. The number of minutes between recurrences: 1 × 10080 = 10080
4. The value of the **FirstDateTime** field: 213,654,240 % 10080 = 8640 (0x000021C0)

4.1.1.3 Daily Recurrence BLOB with Exceptions

The following example shows the binary recurrence data for an appointment that has the following characteristics:

- Occurs every three days, effective 4/7/2011 until 5/4/2011 from 8:00 A.M. to 8:30 A.M.
- The instances on 4/19/2011 and 4/22/2011 were deleted.

The following is the recurrence BLOB.

```
043004300A2000000000A0050000E0100000000000000212000000A0000000000000000002000000A0C1DC0C80D2DC0C0
0000000207EDC0C0016DD0C0630000009300000E0010000FE0100000000000000000000000000
```

Size: 0x0054 bytes

The content of the modified recurrence BLOB is listed in the following table.

Field name	Type	Size	Example	Description
ReaderVersion	**WORD** ([MS-DTYP])	2	04 30	This field indicates version 0x3004.
WriterVersion	**WORD**	2	04 30	This field indicates version 0x3004.
RecurFrequency	**WORD**	2	0A 20	The pattern of the recurrence is daily.
PatternType	**WORD**	2	00 00	The pattern type is Day (0x0000).
CalendarType	**WORD**	2	00 00	The calendar type is Gregorian (0x0000).
FirstDateTime	**ULONG** ([MS-DTYP])	4	A0 05 00 00	The number of minutes offset from the reference date, January 1, 1601, is 1440 (0x000005A0), which corresponds to the first ever day, January 2, 1601. Applying the formula for calculating the value of the **FirstDateTime** field for a daily recurrence, as specified in section 2.2.1.44.1.1: 215,776,800 % 4320 = 1440
Period	**ULONG**	4	E0 10 00 00	The recurrence occurs every 4320 minutes (3 days).
SlidingFlag	**ULONG**	4	00 00 00 00	This field is only used for scheduling tasks. Otherwise, the value can only be zero.
EndType	**ULONG**	4	21 20 00 00	Ends after an end date. (0x00002021)
OccurrenceCount	**ULONG**	4	0C 00 00 00	Not used.
FirstDOW	**ULONG**	4	00 00 00 00	The first day of the week on the calendar is Sunday (the default value).
DeletedInstanceCount	**ULONG**	4	02 00 00 00	There are two deleted instances.
DeletedInstanceDate	**ULONG**	4	A0 C1 DC 0C	The date of the deleted instance is 4/19/2007.
DeletedInstanceDate	**ULONG**	4	80 D2 DC 0C	The date of the deleted instance is 4/22/2007.
ModifiedInstanceCount	**ULONG**	4	00 00 00 00	There are no modified instances.

Field name	Type	Size	Example	Description
StartDate	**ULONG**	4	20 7E DC 0C	The start date of the recurrence, given in minutes since midnight January 1, 1601, is 215,776,800 (0x0CDC7E20), which corresponds to 4/7/2011.
EndDate	**ULONG**	4	00 16 DD 0C	The end date of the recurrence, given in minutes since midnight January 1, 1601, is 215,815,680 (0x0CDD1600), which corresponds to 5/4/2011.
ReaderVersion2	**ULONG**	4	06 30 00 00	This field indicates version 0x00003006.
WriterVersion2	**ULONG**	4	09 30 00 00	This field indicates version 0x00003009.
StartTimeOffset	**ULONG**	4	E0 01 00 00	The appointment's start time is 480 minutes past midnight, or 8:00 AM.
EndTimeOffset	**ULONG**	4	FE 01 00 00	The appointment's end time is 510 minutes past midnight, or 8:30 AM.
ExceptionCount	**WORD**	2	00 00	No modified exceptions.
ReservedBlock1Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.
ReservedBlock2Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.

4.1.1.4 N-Monthly Recurrence BLOB with Exceptions

The following example shows the binary recurrence data for an appointment that has the following characteristics:

- Occurs every third weekend, every 3 months from 2:00 PM to 5:00 P.M., starting on 2/9/2008 and ending after 10 occurrences.
- The instance on 5/10/2008 is moved to 5/11/2008.
- The location of the instance on 8/9/2008 is changed to "new location".

The following is the recurrence BLOB for this recurrence.

```
043004300C200300000060AE00000300000000000000041000000030000002220000000A000000000000000020000006
028C50C4028C70C02000000002EC50C4028C70C8028C30C6027D50C0630000009300000048030000FC030000020048
31C50CFC31C50CA82BC50C0000882BC70C3C2CC70C882BC70C10000D000C006E6577206C6F636174696F6E0000000
0040000000000000000000000040000000000000000000000882BC70C3C2CC70C882BC70C0C006E00650077002000
6C006F006300610074006900 6F006E000000000000000000
```

Size: 0x00D2 bytes

The following table lists the content of the modified recurrence BLOB.

Field name	Type	Size	Example	Description
ReaderVersion	**WORD** ([MS-DTYP])	2	04 30	This field indicates version 0x3004.
WriterVersion	**WORD**	2	04 30	This field indicates version 0x3004.
RecurFrequency	**WORD**	2	0C 20	The pattern of the recurrence is monthly.
PatternType	**WORD**	2	03 00	The pattern type is MonthNth (0x0003).
CalendarType	**WORD**	2	00 00	The calendar type is Gregorian (0x0000).
FirstDateTime	**ULONG** ([MS-DTYP])	4	60 AE 00 00	The number of minutes offset from the reference date, January 1, 1601, is 44640 (0x0000AE60), which corresponds to the first day of the first ever month of February 1601. See the calculation steps following this table.
Period	**ULONG**	4	03 00 00 00	The recurrence occurs every three months.
SlidingFlag	**ULONG**	4	00 00 00 00	This field is only used for scheduling tasks. Otherwise, the value can only be zero.
PatternTypeSpecific	**BYTE** array	Varies	41 00 00 00 03 00 00 00	The recurring appointment occurs on Saturday (0x00000040) and Sunday (0x00000001). The appointment occurs on the third occurrence of these days (0x00000003).
EndType	**ULONG**	4	22 20 00 00	End after *n* occurrences. (0x00002022).
OccurrenceCount	**ULONG**	4	0A 00 00 00	The recurrence ends after 10 occurrences.
FirstDOW	**ULONG**	4	00 00 00 00	The first day of the week on the calendar is Sunday (the default value).
DeletedInstanceCount	**ULONG**	4	02 00 00 00	There are two deleted instances.
DeletedInstanceDate	**ULONG**	4	60 28 C5 0C	The date of the deleted instance is 5/10/2008.
DeletedInstanceDate	**ULONG**	4	40 28 C7 0C	The date of the deleted instance is 8/9/2008.
ModifiedInstanceCount	**ULONG**	4	02 00 00 00	There are two modified instances.
ModifiedInstanceDate	**ULONG**	4	00 2E C5 0C	The date of the modified instance is 5/11/2008.

Field name	Type	Size	Example	Description
ModifiedInstanceDate	**ULONG**	4	40 28 C7 0C	The date of the modified instance is 8/9/2008.
StartDate	**ULONG**	4	80 28 C3 0C	The start date of the recurrence is 2/9/2008.
EndDate	**ULONG**	4	60 27 D5 0C	The end date of the recurrence is 5/8/2010.
ReaderVersion2	**ULONG**	4	06 30 00 00	This field indicates version 0x00003006.
WriterVersion2	**ULONG**	4	09 30 00 00	This field indicates version 0x00003009.
StartTimeOffset	**ULONG**	4	48 03 00 00	The appointment's start time is 840 minutes past midnight, or 2:00 P.M.
EndTimeOffset	**ULONG**	4	FC 03 00 00	The appointment's end time is 1020 minutes past midnight, or 5:00 P.M.
ExceptionCount	**WORD**	2	02 00	Two exceptions.
ExceptionInfo structure block for exception 1 follows				
StartDateTime	**ULONG**	4	48 31 C5 0C	The start date and time of the exception is 5/11/2008 2:00 P.M.
EndDateTime	**ULONG**	4	FC 31 C5 0C	The end time of the exception is 5/11/2008 5:00 P.M.
OriginalStartTime	**ULONG**	4	A8 2B C5 0C	The original start date and time of the occurrence was 5/10/2008 2:00 P.M.
OverrideFlags	**WORD**	2	00 00	None.
ExceptionInfo structure block for exception 2 follows				
StartDateTime	**ULONG**	4	88 2B C7 0C	The start date and time of the exception is 8/9/2008 2:00 P.M.
EndDateTime	**ULONG**	4	3C 2C C7 0C	The end date and time of the exception is 8/9/2008 5:00 P.M.
OriginalStartTime	**ULONG**	4	88 2B C7 0C	The original start date and time of the occurrence was 8/9/2008 2:00 P.M.
OverrideFlags	**WORD**	2	10 00	**ARO_LOCATION** (0x00000010). The location has been modified.
LocationLength	**WORD**	2	0D 00	The length of the location string, including a terminating null character, is 13.
LocationLength2	**WORD**	2	0C 00	The length of the location string is 12.

Field name	Type	Size	Example	Description
Location	**BYTE** array	Varies	6E 65 77 20 6C 6F 63 61 74 69 6F 6E	"new location"
ReservedBlock1Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.
ExtendedException block for exception 1 follows				
ChangeHighlight	**BYTE** array	Varies	04 00 00 00 00 00 00 00	The size of the **ChangeHighlight** field is 4. The value of the **PidLidChangeHighlight** property (section 2.2.6.2) is zero for this exception.
ReservedBlockEE1Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.
ExtendedException structure block for exception 2 follows				
ChangeHighlight	**BYTE** array	Varies	04 00 00 00 00 00 00 00	The size of the **ChangeHighlight** field is 4. The value of the **PidLidChangeHighlight** property is zero for this exception.
ReservedBlockEE1Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.
StartDateTime	**ULONG**	4	88 2B C7 0C	The start date and time of the exception is 8/9/2008 2:00 P.M.
EndDateTime	**ULONG**	4	3C 2C C7 0C	The end date and time of the exception is 8/9/2008 5:00 P.M.
OriginalStartTime	**ULONG**	4	88 2B C7 0C	The original start date and time of the occurrence was 8/9/2008 2:00 P.M.
WideCharLocationLength	**WORD**	2	0C 00	The length of the exception's Unicode location is 12 characters.
WideCharLocation	**BYTE** array	Varies	6E 00 65 00 77 00 20 00 6C 00 6F 00 63 00 61 00 74 00 69 00 6F 00 6E 00	"new location" in Unicode.
ReservedBlockEE2Size	**ULONG**	4	00 00 00 00	No data in this skip block.
ReservedBlock2Size	**ULONG**	4	00 00 00 00	No data in this skip block.

The steps for calculating the value of the **FirstDateTime** field for a monthly recurrence, as specified in section 2.2.1.44.1.1, are applied to this example as follows:

1. The first day of the month that contains the start date: February 1, 2008
2. The number of calendar months between midnight, February 1, 2008, and midnight, January 1, 1601: 4885
3. The offset of months within the year 1601: 4885 % 3 = 1

4. The number corresponding to the first recurrence month within the year 1601: 1 + 1 = 2

5. The number of minutes between midnight, February 1, 1601, and midnight, January 1, 1601: 44640 (0x0000AE60).

4.1.1.5 Yearly Recurrence BLOB with Exceptions

The following example shows the binary recurrence data for an appointment that has the following characteristics:

- Occurs every April 19, effective 4/19/2011, from 8:00 A.M. to 8:30 A.M.
- Move the instance on 4/19/2012 to 4/21/2012.

The following is the recurrence BLOB for this recurrence.

```
043004300D200200000040FA01000C0000000000000001300000023200000A0000000000000000100000060CCE40C0
1000000A0D7E40CA0C1DC0CDF80E95A0630000009300000E0010000FE010000010080D9E40C9ED9E40C40CEE40C00
0000000000040000000000000000000000000000000000
```

Size: 0x0072 bytes

The content of the modified recurrence BLOB is listed in the following table.

Field name	Type	Size	Example	Description
ReaderVersion	**WORD** ([MS-DTYP])	2	04 30	This field indicates version 0x3004.
WriterVersion	**WORD**	2	04 30	This field indicates version 0x3004.
RecurFrequency	**WORD**	2	0D 20	The pattern of the recurrence is yearly.
PatternType	**WORD**	2	02 00	The pattern type is Month (0x0002).
CalendarType	**WORD**	2	00 00	The calendar type is Gregorian.
FirstDateTime	**ULONG** ([MS-DTYP])	4	40 FA 01 00	The number of minutes offset from the reference date, January 1, 1601, is 129,600 (0x0001FA40), which corresponds to the first day of the first ever month of April, 1601. See the calculation steps following this table.
Period	**ULONG**	4	0C 00 00 00	The recurrence occurs every 12 months.
SlidingFlag	**ULONG**	4	00 00 00 00	This field is only used for scheduling tasks. Otherwise, the value can only be 0 (zero).
PatternTypeSpecific	**BYTE** array	Varies	13 00 00 00	The recurrence falls on the 19th of the month.
EndType	**ULONG**	4	23 20 00 00	Never ends. (0x00002023)
OccurrenceCount	**ULONG**	4	0A 00 00	Not used.

Field name	Type	Size	Example	Description
			00	
FirstDOW	**ULONG**	4	00 00 00 00	The first day of the week on the calendar is Sunday (the default value).
DeletedInstanceCount	**ULONG**	4	01 00 00 00	There is one deleted instance.
DeletedInstanceDate	**ULONG**	4	60 CC E4 0C	The date of the deleted instance is 4/19/2012.
ModifiedInstanceCount	**ULONG**	4	01 00 00 00	There is one modified instance.
ModifiedInstanceDate	**ULONG**	4	A0 D7 E4 0C	The date of the modified instance is 4/21/2012.
StartDate	**ULONG**	4	A0 C1 DC 0C	The start date of the recurrence is 4/19/2011.
EndDate	**ULONG**	4	DF 80 E9 5A	The end date of the recurrence is never. (12/31/4500)
ReaderVersion2	**ULONG**	4	06 30 00 00	This field indicates version 0x00003006.
WriterVersion2	**ULONG**	4	09 30 00 00	This field indicates version 0x00003009.
StartTimeOffset	**ULONG**	4	E0 01 00 00	The appointment's start time is 480 minutes past midnight, or 8:00 A.M.
EndTimeOffset	**ULONG**	4	FE 01 00 00	The appointment's end time is 510 minutes past midnight, or 8:30 A.M.
ExceptionCount	**WORD**	2	01 00	One exception.
ExceptionInfo structure block for exception 1:				
StartDateTime	**ULONG**	4	80 D9 E4 0C	The start date and time of the exception is 4/21/2012, 8:00 A.M.
EndDateTime	**ULONG**	4	9E D9 E4 0C	The end date and time of the exception is 4/21/2012, 8:30 A.M.
OriginalStartTime	**ULONG**	4	40 CE E4 0C	The original start date and time of the occurrence was 4/19/2012, 8:00 A.M.
OverrideFlags	**WORD**	2	00 00	None.
ReservedBlock1Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.
ExtendedException structure block for exception 1:				
ChangeHighlight	**Byte** array	Varies	04 00 00 00 00 00 00 00	The size of the **ChangeHighlight** field is 4. The value of the **PidLidChangeHighlight** property (section 2.2.6.2) is zero for this exception.

Field name	Type	Size	Example	Description
ReservedBlockEE1Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.
ReservedBlock2Size	**ULONG**	4	00 00 00 00	No data in this skip block.

The steps for calculating the value of the **FirstDateTime** field for a yearly recurrence, as specified in section 2.2.1.44.1.1, are applied to this example as follows:

1. The first day of the month that contains the start date: April 1, 2011
2. The number of calendar months between midnight, April 1, 2011, and midnight, January 1, 1601: 4923
3. The offset of months within the year 1601: 4923 % 12 = 3
4. The number corresponding to the first recurrence month within the year 1601: 3 + 1 = 4
5. The number of minutes between midnight, April 1, 1601, and midnight, January 1, 1601: 129,600 (0x0001FA40).

4.1.1.6 Yearly Hebrew Lunar Recurrence BLOB with Exceptions

The following example shows the binary recurrence data for an appointment that has the following characteristics:

- Occurs every year on ג ו ס י נ starting ג' ו ס י נ ס ש ת "ח from 8:00 A.M. to 8:30 A.M.
- Change the busy status of the second instance to "tentative", make the reminder fire 60 minutes before the appointment, and change the body text.

The following is the recurrence BLOB for this recurrence.

```
043004300D20020080080750A000C000000000000000300000232000000A000000000000000010000000207EDC0C0
1000000207EDC0C6074C40CDF80E95A06300000093000000E0010000FE0100000100080DC0C1E80DC0C0080DC0C24
023C000000010000000000000004000000000000000000000000000000000
```

Size: 0x007A bytes

The content of the modified recurrence BLOB is in the following table.

Field name	Type	Size	Example	Description
ReaderVersion	**WORD** ([MS-DTYP])	2	04 30	This field indicates version 0x3004.
WriterVersion	**WORD**	2	04 30	This field indicates version 0x3004.
RecurFrequency	**WORD**	2	0D 20	The pattern of the recurrence is yearly.
PatternType	**WORD**	2	02 00	The pattern type is Month (0x0002).
CalendarType	**WORD**	2	08 00	The calendar type is CAL_HEBREW (0x0008).

Field name	Type	Size	Example	Description
FirstDateTime	**ULONG** ([MS-DTYP])	4	80 75 0A 00	The number of minutes offset from the reference date, January 1, 1601, is 685,440 (0x000A7580), which corresponds to the date of the first ever lunar month: April 22, 1602. See the calculation steps following this table.
Period	**ULONG**	4	0C 00 00 00	The recurrence occurs every 12 months.
SlidingFlag	**ULONG**	4	00 00 00 00	This field is only used for scheduling tasks. Otherwise the value can only be 0 (zero).
PatternTypeSpecific	**BYTE** array	Varies	03 00 00 00	The recurrence falls on the third day of the month (in the Hebrew lunar calendar).
EndType	**ULONG**	4	23 20 00 00	Never ends. (0x00002023).
OccurrenceCount	**ULONG**	4	0A 00 00 00	Not used.
FirstDOW	**ULONG**	4	00 00 00 00	The first day of the week on the calendar is Sunday (the default value).
DeletedInstanceCount	**ULONG**	4	01 00 00 00	There is one deleted instance.
DeletedInstanceDate	**ULONG**	4	20 7E DC 0C	The date of the deleted instance is 4/7/2011.
ModifiedInstanceCount	**ULONG**	4	01 00 00 00	There is one modified instance.
ModifiedInstanceDate	**ULONG**	4	20 7E DC 0C	The date of the modified instance is 4/7/2011.
StartDate	**ULONG**	4	60 74 C4 0C	The start date of the recurrence is 4/8/2008.
EndDate	**ULONG**	4	DF 80 E9 5A	The end date of the recurrence is never. (12/31/4500).
ReaderVersion2	**ULONG**	4	06 30 00 00	This field indicates version 0x00003006.
WriterVersion2	**ULONG**	4	09 30 00 00	This field indicates version 0x00003009.
StartTimeOffset	**ULONG**	4	E0 01 00 00	The appointment's start time is 480 minutes past midnight, or 8:00 A.M.
EndTimeOffset	**ULONG**	4	FE 01 00 00	The appointment's end time is 510 minutes past midnight, or 8:30 A.M.
ExceptionCount	**WORD**	2	01 00	One exception.
ExceptionInfo structure block:				

Field name	Type	Size	Example	Description
StartDateTime	**ULONG**	4	00 80 DC 0C	The start date and time of the exception is 4/7/2011 8:00 A.M.
EndDateTime	**ULONG**	4	1E 80 DC 0C	The end date and time of the exception is 4/7/2011 at 8:30 A.M.
OriginalStartTime	**ULONG**	4	00 80 DC 0C	The original start date and time of the occurrence was 4/7/2011 at 8:00 A.M.
OverrideFlags	**WORD**	2	24 02	A value of 0x0224 indicates that the following flags are set to 1 in this property: **ARO_BUSYSTATUS** | **ARO_REMINDERDELTA** | **ARO_EXCEPTIONAL_BODY**
ReminderDelta	**ULONG**	4	3C 00 00 00	The exception's value for the **PidLidReminderDelta** property ([MS-OXORMDR] section 2.2.1.3) is 60 (0x0000003C).
BusyStatus	**ULONG**	4	01 00 00 00	The exception's value for the **PidLidBusyStatus** property (section 2.2.1.2) is 1.
ReservedBlock1Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.
ExtendedException structure block:				
ChangeHighlight	**BYTE** array	Varies	04 00 00 00 00 00 00 00	The size of the **ChangeHighlight** field is 4. The value of the **PidLidChangeHighlight** property (section 2.2.6.2) is zero for this exception.
ReservedBlockEE1Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.
ReservedBlock2Size	**ULONG**	4	00 00 00 00	There is no data in this skip block.

The steps for calculating the value of the **FirstDateTime** field for a yearly recurrence, as specified in section 2.2.1.44.1.1, are applied to this example as follows with adjustments made for the Hebrew lunar calendar:

1. The first day of the month that contains the start date: April 6, 2008 (in Gregorian calendar)
2. The number of calendar months between midnight of April 6, 2008, and midnight, September 27, 1601, of the Gregorian calendar: 4879
3. The offset of Hebrew lunar months: 4879 % 12 = 7
4. The date of the first recurrence nearest September 27, 1601, of the Gregorian calendar: Add 7 months to the date September 27, 1601, to get the date April 22, 1602.
5. The number of minutes between midnight, April 22, 1602, and midnight, January 1, 1601: 685,440 (0x000A7580)

4.1.2 Global Object ID Examples

This section includes examples of the **PidLidGlobalObjectId** (section 2.2.1.27) and **PidLidCleanGlobalObjectId** (section 2.2.1.28) BLOB properties that refer to an exception to a recurring series. The data for the fields of the Global Object ID BLOB are stored in little-endian byte order, unless otherwise specified.

4.1.2.1 PidLidGlobalObjectId

The following is the value of the **PidLidGlobalObjectId** property (section 2.2.1.27) for an object that represents an exception to a recurring series. The instance that is represented by the exception was moved from March 25, 2008, to March 26, 2008.

```
cb: 56
lpb:
040000008200E00074C5B7101A82E00807D803195025D461E473C80100000000000000000100000002A5844B3A444F
74A9C246C60886F116B
```

The content of the **PidLidGlobalObjectId** property is in the following table.

Field name	Type	Size	Sample	Description
Byte Array ID	**BYTE** array	16	04 00 00 00 82 00 E0 00 74 C5 B7 10 1A 82 E0 08	This byte array identifies the BLOB as a Global Object ID.
Year (YH + YL)	**WORD** ([MS-DTYP])	2	07 D8	The original year of the instance represented by the exception. This value is in **big-endian** format instead of little-endian format and comprises the high order and low order bytes of the 2-byte year from the **PidLidExceptionReplaceTime** property (section 2.2.10.2.5). 0x07D8 (2008)
M	**BYTE**	1	03	The original month of the instance represented by the exception. 0x03 (March)
D	**BYTE**	1	19	The original day of the instance represented by the exception. 0x19 (25)
Creation Time	**PtypTime** ([MS-OXCDATA] section 2.11.1)	8	50 25 D4 61 E4 73 C8 01	2008/02/20 17:16:51
X	**BYTE** array	8	00 00	Reserved.

Field name	Type	Size	Sample	Description
			00 00 00 00 00 00	
Size	**LONG** [MS-DTYP]	4	10 00 00 00	The length of the **Data** field. 0x00000010 (16) bytes
Data	BYTE array	16	2A 58 44 B3 A4 44 F7 4A 9C 24 6C 60 88 6F 11 6B	The data that uniquely identifies this Meeting object.

4.1.2.2 PidLidCleanGlobalObjectId

The following is the value of the **PidLidCleanGlobalObjectId** property (section 2.2.1.28) for the exception from the example described in section 4.1.2.1. The only difference between these two properties is that in the clean version, the **Year**, **Month**, and **Day** fields are all 0 (zero).

```
cb: 56
lpb:
040000008200E00074C5B7101A82E008000000000005025D461E473C8010000000000000000000100000002A5844B3A444F
74A9C246C60886F116B
```

4.1.3 Downlevel Text for Meeting Request Body

A Meeting Request object can have extra body text with the date, time, and location to help clients that do not understand the format, as specified in section 2.2.6.12. The following is sample text from the body of a Meeting object.

```
Paulo,

This Friday I feel like eating out. How about we hit our old favorite?

- Jim
```

Figure 3: Extra body text: Client understands format

The following shows how the body of the Meeting Request object might look to a client that does not understand the Meeting Request object format.

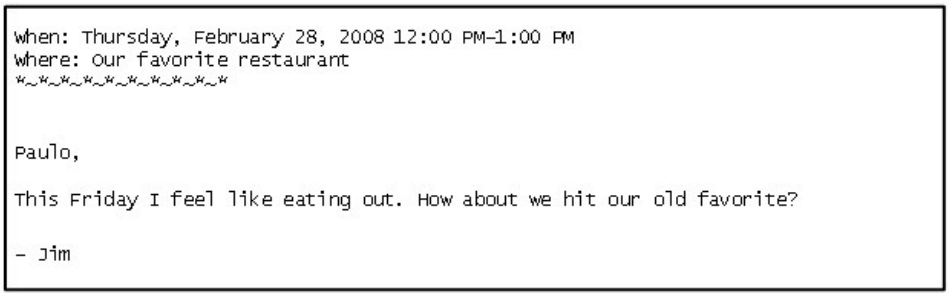


Figure 4: Extra body text: Client does not understand format

4.1.4 PidLidAppointmentTimeZoneDefinitionRecur BLOB

The following is an example of a **PidLidAppointmentTimeZoneDefinitionRecur** property (section 2.2.1.41) BLOB.

```
cb: 184 (0x000000B8)
lpb:
02013000020015005000610063006900660069006300200053007400610 06E00640061007200640020005400690 06
D00650002000 2013E000000D607000000000000000000000000000000E001000000000000C4FFFFFF00000A00000005
000200000000000000000004000000010002000000000000000 2013E000200D707000000000000000000000000000
0E001000000000000C4FFFFFF00000B000000010002000000000000000000300000002000200000000000000
```

The content of this **PidLidAppointmentTimeZoneDefinitionRecur** property BLOB.

Field name	Type	Size	Example	Description
Major Version	**BYTE** ([MS-DTYP])	1	02	This field indicates version 0x02.
Minor Version	**BYTE**	1	01	This field indicates version 0x01.
cbHeader	**WORD** ([MS-DTYP])	2	30 00	Header contains 48 bytes.
Reserved	**WORD**	2	02 00	This value is always 2.
cchKeyName	**WORD**	2	15 00	The **KeyName** field has a length of 21 **Unicode** characters.
KeyName	Unicode **String**, not terminated	Varies	50 00 61 00 63 00 69 00 66 00 69 00 63 00 20 00 53 00 74 00 61 00 6E 00 64 00 61	"Pacific Standard Time"

Field name	Type	Size	Example	Description
			00 72 00 64 00 20 00 54 00 69 00 6D 00 65 00	
cRules	**WORD**	2	02 00	There will be two TZRule structures.
(Beginning of first **TZRule** structure)				
Major Version	**BYTE**	1	02	This field indicates version 0x02.
Minor Version	**BYTE**	1	01	This field indicates version 0x01.
Reserved	**WORD**	2	3E 00	This value is always 2.
TZRule flags	**WORD**	2	00 00	This rule (4) is not marked as the effective rule (4).
wYear	**WORD**	2	D6 07	This rule (4) applies beginning January 1, 2006.
X	**BYTE** array	14	00 00 00 00 00 00 00 00 00 00 00 00 00 00	Can only be a **Byte** array of all zeros.
lBias	**LONG** ([MS-DTYP])	4	E0 01 00 00	This rule (4) has a standard bias of 480 minutes from UTC.
lStandardBias	**LONG**	4	00 00 00 00	No additional bias during standard time.
lDaylightBias	**LONG**	4	C4 FF FF FF	Daylight offset of -60 from the standard bias during daylight time.
stStandardDate	**SYSTEMTIME**	16	00 00 0A 00 00 00 05 00 02 00 00 00 00 00 00 00	This indicates the following **SYSTEMTIME** (in decimal): wYear: 0 wMonth: 10 wDayOfWeek: 0 wDay: 5 wHour: 2 wMinute: 0 wSecond: 0 wMilliseconds: 0 This means that the time zone will transition to standard time on the last

Field name	Type	Size	Example	Description
				Sunday of October at 2:00 A.M.
stDaylightDate	**SYSTEMTIME**	16	00 00 04 00 00 00 01 00 02 00 00 00 00 00 00 00	This indicates the following **SYSTEMTIME** (in decimal format): wYear: 0 wMonth: 4 wDayOfWeek: 0 wDay: 1 wHour: 2 wMinute: 0 wSecond: 0 wMilliseconds: 0 This means that the time zone will transition to daylight time on the first Sunday of April at 2:00 A.M.
(Beginning of second **TZRule** structure)				
Major Version	**BYTE**	1	02	This field indicates version 0x02.
Minor Version	**BYTE**	1	01	This field indicates version 0x01.
Reserved	**WORD**	2	3E 00	This value is always 2.
TZRule flags	**WORD**	2	02 00	The TZRULE_FLAG_EFFECTIVE_TZREG flag is set to indicate that this rule (4) is the effective rule (4).
wYear	**WORD**	2	D7 07	This rule (4) applies beginning January 1, 2007.
X	**BYTE** array	14	00 00 00 00 00 00 00 00 00 00 00 00 00 00	Can only be a **BYTE** array of all zeros.
lBias	**LONG**	4	E0 01 00 00	This rule (4) has a standard bias of 480 minutes from UTC.
lStandardBias	**LONG**	4	00 00 00 00	No additional offset during standard time.
lDaylightBias	**LONG**	4	C4 FF FF FF	Offset of -60 from the standard bias during daylight time.
stStandardDate	**SYSTEMTIME**	16	00 00 0B 00 00 00 01 00 02 00 00 00 00 00 00	This indicates the following **SYSTEMTIME** (in decimal): wYear: 0 wMonth: 11 wDayOfWeek: 0 wDay: 1

Field name	Type	Size	Example	Description
			00	wHour: 2 wMinute: 0 wSecond: 0 wMilliseconds: 0 This means that the time zone will transition to standard time on the first Sunday of November at 2:00 A.M.
stDaylightDate	**SYSTEMTIME**	16	00 00 03 00 00 00 02 00 02 00 00 00 00 00 00 00	This indicates the following **SYSTEMTIME** (in decimal format): wYear: 0 wMonth: 3 wDayOfWeek: 0 wDay: 2 wHour: 2 wMinute: 0 wSecond: 0 wMilliseconds: 0 This means that the time zone will transition to daylight time on the second Sunday of March at 2:00 A.M.

4.1.5 PidLidTimeZoneStruct

The following is an example of a value for the **PidLidTimeZoneStruct** property (section 2.2.1.39).

```
cb: 48 (0x00000030)
lpb:
E001000000000000C4FFFFFF000000000B0000000100020000000000000000000000030000000200020000000000
000
```

The following table lists the content of the **PidLidTimeZoneStruct** property BLOB.

Field name	Type	Size	Example	Description
lBias	**LONG** ([MS-DTYP])	4	E0 01 00 00	This rule (4) has a standard bias of 480 minutes from UTC.
lStandardBias	**LONG**	4	00 00 00 00	No additional offset during standard time.
lDaylightBias	**LONG**	4	C4 FF FF FF	Offset of -60 from the standard bias during daylight time.
wStandardYear	**WORD** ([MS-DTYP])	2	00 00	No year is specified, which indicates that the rule (4) is a relative rule (4).
stStandardDate	**SYSTEMTIME**	16	00 00 0B 00 00 00 01 00 02 00 00	This indicates the following **SYSTEMTIME** (in decimal format): wYear: 0 wMonth: 11

Field name	Type	Size	Example	Description
			00 00 00 00 00	wDayOfWeek: 0 wDay: 1 wHour: 2 wMinute: 0 wSecond: 0 wMilliseconds: 0 This means that the time zone will transition to standard time on the first Sunday of November at 2:00 A.M.
wDaylightYear	**WORD**	2	00 00	No year is specified, which indicates that the rule (4) is a relative rule (4).
stDaylightDate	**SYSTEMTIME**	16	00 00 03 00 00 00 02 00 02 00 00 00 00 00 00 00	This indicates the following **SYSTEMTIME** (in decimal): wYear: 0 wMonth: 3 wDayOfWeek: 0 wDay: 2 wHour: 2 wMinute: 0 wSecond: 0 wMilliseconds: 0 This means that the time zone will transition to daylight time on the second Sunday of March at 2:00 A.M.

4.1.6 Sample of PidLidTimeZone

A value for the **PidLidTimeZone** property (section 2.2.5.6) equal to 13 would indicate that the time zone has an offset from UTC+12 of 20 * 60 minutes, or 1200 minutes from UTC+12. This time zone has a daylight saving Standard Date of {11, 0, 1, 2}, equivalent to the first Sunday of November at 2:00 A.M. It has a Daylight Date of {3, 0, 2, 2}, equivalent to the second Sunday of March at 2:00 A.M.

4.2 Examples of Objects

Before manipulating an object, the client needs to ask the server to map from **property names** to **property IDs**, using the **RopGetPropertyIdsFromNames** remote operation (ROP) ([MS-OXCROPS] section 2.2.8.1). The following properties are referenced in the examples that follow.

Property	Property set GUID	Property ID
PidLidAppointmentSequence (section 2.2.1.1)	{ 00062002-0000-0000-c000-000000000046}	0x8201
PidLidAppointmentSequenceTime (section 2.2.4.1)	{ 00062002-0000-0000-c000-000000000046}	0x8202
PidLidChangeHighlight (section 2.2.6.2)	{ 00062002-0000-0000-	0x8204

Property	Property set GUID	Property ID
	c000-000000000046}	
PidLidBusyStatus (section 2.2.1.2)	{ 00062002-0000-0000-c000-000000000046}	0x8205
PidLidAppointmentAuxiliaryFlags (section 2.2.1.3)	{ 00062002-0000-0000-c000-000000000046}	0x8207
PidLidLocation (section 2.2.1.4)	{ 00062002-0000-0000-c000-000000000046}	0x8208
PidLidAppointmentStartWhole (section 2.2.1.5)	{ 00062002-0000-0000-c000-000000000046}	0x820D
PidLidAppointmentEndWhole (section 2.2.1.6)	{ 00062002-0000-0000-c000-000000000046}	0x820E
PidLidAppointmentDuration (section 2.2.1.7)	{ 00062002-0000-0000-c000-000000000046}	0x8213
PidLidAppointmentColor ([MS-OXPROPS] section 2.9)	{ 00062002-0000-0000-c000-000000000046}	0x8214
PidLidAppointmentSubType (section 2.2.1.9)	{ 00062002-0000-0000-c000-000000000046}	0x8215
PidLidAppointmentRecur (section 2.2.1.44)	{ 00062002-0000-0000-c000-000000000046}	0x8216
PidLidAppointmentStateFlags (section 2.2.1.10)	{ 00062002-0000-0000-c000-000000000046}	0x8217
PidLidResponseStatus (section 2.2.1.11)	{ 00062002-0000-0000-c000-000000000046}	0x8218
PidLidAppointmentReplyTime (section 2.2.4.3)	{ 00062002-0000-0000-c000-000000000046}	0x8220
PidLidRecurring (section 2.2.1.12)	{ 00062002-0000-0000-c000-000000000046}	0x8223
PidLidIntendedBusyStatus (section 2.2.6.4)	{ 00062002-0000-0000-c000-000000000046}	0x8224
PidLidFInvited (section 2.2.4.4)	{ 00062002-0000-0000-c000-000000000046}	0x8229
PidLidAppointmentReplyName (section 2.2.4.5)	{ 00062002-0000-0000-c000-000000000046}	0x8230
PidLidRecurrenceType (section 2.2.1.45)	{ 00062002-0000-0000-c000-000000000046}	0x8231
PidLidRecurrencePattern (section 2.2.1.46)	{ 00062002-0000-0000-c000-000000000046}	0x8232
PidLidTimeZoneStruct (section 2.2.1.39)	{ 00062002-0000-0000-c000-000000000046}	0x8233

Property	Property set GUID	Property ID
PidLidTimeZoneDescription (section 2.2.1.40)	{ 00062002-0000-0000-c000-000000000046}	0x8234
PidLidClipStart (section 2.2.1.14)	{ 00062002-0000-0000-c000-000000000046}	0x8235
PidLidClipEnd (section 2.2.1.15)	{ 00062002-0000-0000-c000-000000000046}	0x8236
PidLidAllAttendeesString (section 2.2.1.16)	{ 00062002-0000-0000-c000-000000000046}	0x8238
PidLidAutoFillLocation (section 2.2.4.8)	{ 00062002-0000-0000-c000-000000000046}	0x823A
PidLidToAttendeesString (section 2.2.1.17)	{ 00062002-0000-0000-c000-000000000046}	0x823B
PidLidCcAttendeesString (section 2.2.1.18)	{ 00062002-0000-0000-c000-000000000046}	0x823C
PidLidAppointmentNotAllowPropose (section 2.2.1.26)	{ 00062002-0000-0000-c000-000000000046}	0x825A
PidLidAppointmentTimeZoneDefinitionStartDisplay (section 2.2.1.42)	{ 00062002-0000-0000-c000-000000000046}	0x825E
PidLidAppointmentTimeZoneDefinitionEndDisplay (section 2.2.1.43)	{ 00062002-0000-0000-c000-000000000046}	0x825F
PidLidAppointmentTimeZoneDefinitionRecur (section 2.2.1.41)	{ 00062002-0000-0000-c000-000000000046}	0x8260
PidLidExceptionReplaceTime (section 2.2.10.2.5)	{ 00062002-0000-0000-c000-000000000046}	0x8228
PidLidFExceptionalAttendees (section 2.2.2.3)	{ 00062002-0000-0000-c000-000000000046}	0x822B
PidLidFExceptionalBody (section 2.2.10.2.6)	{ 00062002-0000-0000-c000-000000000046}	0x8206
PidLidReminderDelta ([MS-OXORMDR] section 2.2.1.3)	{ 00062008-0000-0000-c000-000000000046}	0x8501
PidLidReminderTime ([MS-OXORMDR] section 2.2.1.4)	{ 00062008-0000-0000-c000-000000000046}	0x8502
PidLidReminderSet ([MS-OXORMDR] section 2.2.1.1)	{ 00062008-0000-0000-c000-000000000046}	0x8503
PidLidReminderSignalTime ([MS-OXORMDR] section 2.2.1.2)	{ 00062008-0000-0000-c000-000000000046}	0x8504
PidLidPrivate ([MS-OXCMSG] section 2.2.1.15)	{ 00062008-0000-0000-c000-000000000046}	0x8506
PidLidSideEffects ([MS-OXCMSG] section 2.2.1.16)	{ 00062008-0000-0000-	0x8510

Property	Property set GUID	Property ID
	c000-000000000046}	
PidLidCommonStart ([MS-OXCMSG] section 2.2.1.18)	{ 00062008-0000-0000-c000-000000000046}	0x8516
PidLidCommonEnd ([MS-OXCMSG] section 2.2.1.19)	{ 00062008-0000-0000-c000-000000000046}	0x8517
PidLidAttendeeCriticalChange (section 2.2.5.2)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0001
PidLidWhere (section 2.2.5.3)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0002
PidLidGlobalObjectId (section 2.2.1.27)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0003
PidLidIsSilent (section 2.2.7.7)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0004
PidLidIsRecurring (section 2.2.1.13)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0005
PidLidIsException (section 2.2.1.35)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x000A
PidLidTimeZone (section 2.2.5.6)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x000C
PidLidOwnerCriticalChange (section 2.2.1.34)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x001A
PidLidCalendarType (section 2.2.6.11)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x001C
PidLidCleanGlobalObjectId (section 2.2.1.28)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0023
PidLidAppointmentMessageClass (section 2.2.6.6)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0024
PidLidMeetingType (section 2.2.6.5)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0026
PidLidOldLocation (section 2.2.6.7)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0028
PidLidOldWhenEndWhole (section 2.2.6.9)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x0029
PidLidOldWhenStartWhole (section 2.2.6.8)	{6ed8da90-450b-101b-98da-00aa003f1305}	0x002A

It is up to the server to keep track of, and return, the actual mapping. The following mapping values will be used in each of the examples in this section, as if the server had returned these values.

Property	Property ID
PidLidAppointmentSequence	0x81AF
PidLidAppointmentSequenceTime	0x82E7
PidLidChangeHighlight	0x82EC
PidLidBusyStatus	0x81B6
PidLidAppointmentAuxiliaryFlags	0x82D2
PidLidLocation	0x8009
PidLidAppointmentStartWhole	0x81B2
PidLidAppointmentEndWhole	0x81AC
PidLidAppointmentDuration	0x81A9
PidLidAppointmentColor	0x82CA
PidLidAppointmentSubType	0x8120
PidLidAppointmentRecur	0x81AD
PidLidAppointmentStateFlags	0x81B3
PidLidResponseStatus	0x8122
PidLidAppointmentReplyTime	0x8139
PidLidRecurring	0x81FD
PidLidIntendedBusyStatus	0x81E2
PidLidFInvited	0x81DA
PidLidAppointmentReplyName	0x81AE
PidLidRecurrenceType	0x81FE
PidLidRecurrencePattern	0x81FC
PidLidTimeZoneStruct	0x8214
PidLidTimeZoneDescription	0x8213
PidLidClipStart	0x81BA
PidLidClipEnd	0x81B9
PidLidAllAttendeesString	0x81A8
PidLidAutoFillLocation	0x82E8
PidLidToAttendeesString	0x82D9
PidLidCcAttendeesString	0x82DA
PidLidAppointmentNotAllowPropose	0x82D5

Property	Property ID
PidLidAppointmentTimeZoneDefinitionStartDisplay	0x83Aa8
PidLidAppointmentTimeZoneDefinitionEndDisplay	0x83A9
PidLidAppointmentTimeZoneDefinitionRecur	0x83AA
PidLidExceptionReplaceTime	0x83AC
PidLidFExceptionalAttendees	0x82D7
PidLidFExceptionalBody	0x82D8
PidLidReminderDelta	0x81FF
PidLidReminderTime	0x8005
PidLidReminderSet	0x8004
PidLidReminderSignalTime	0x8006
PidLidPrivate	0x82EF
PidLidSideEffects	0x8002
PidLidCommonStart	0x81BC
PidLidCommonEnd	0x81BB
PidLidAttendeeCriticalChange	0x81B5
PidLidWhere	0x8219
PidLidGlobalObjectId	0x81E0
PidLidIsSilent	0x81E6
PidLidIsRecurring	0x81E5
PidLidIsException	0x81E4
PidLidTimeZone	0x8212
PidLidOwnerCriticalChange	0x8128
PidLidCalendarType	0x81B7
PidLidCleanGlobalObjectId	0x81B8
PidLidAppointmentMessageClass	0x8311
PidLidMeetingType	0x8314
PidLidOldLocation	0x8316
PidLidOldWhenEndWhole	0x83CD
PidLidOldWhenStartWhole	0x83CC

4.2.1 Appointment Example

After making a dentist appointment for 10:00 A.M. (Pacific Daylight Time) on May 1, 2009, Minesh decides to set the information in her Calendar folder so that she will not forget about it. The appointment is an hour long, and she wants to be reminded about it half an hour before it happens. She wants to treat this as a private appointment, which indicates to a client to hide the details from other people. The following is a description of what a client might do to accomplish Minesh's intentions and the responses a server might return.

To create an Appointment object, the client uses the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2). The server returns a success code and a handle to a Message object.

The client then uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to transmit Minesh's data to the server. An example of the data that might be sent by the client is shown in the following table.

Property	Property ID	Property type	Value
PidTagMessageClass ([MS-OXCMSG] section 2.2.1.3)	0x001A	0x001F (**PtypString** ([MS-OXCDATA] section 2.11.1.2))	IPM.Appointment
PidTagIconIndex (section 2.2.1.49)	0x1080	0x0003 (**PtypInteger32** ([MS-OXCDATA] section 2.11.1))	0x00000400
PidTagSensitivity ([MS-OXCMSG] section 2.2.1.13)	0x0036	0x0003 (**PtypInteger32**)	0x00000002 (SENSITIIVITY_PRIVATE)
PidLidPrivate ([MS-OXCMSG] section 2.2.1.15)	0x82EF	0x000B (**PtypBoolean** ([MS-OXCDATA] section 2.11.1))	0x01 (TRUE)
PidLidSideEffects ([MS-OXCMSG] section 2.2.1.16)	0x8002	0x0003 (**PtypInteger32**)	0x00000171
PidLidCommonStart ([MS-OXCMSG] section 2.2.1.18)	0x81BC	0x0040 (**PtypTime** ([MS-OXCDATA] section 2.11.1))	0x01C9CA7E43442800 (2009/05/01 17:00:00.000)
PidLidCommonEnd ([MS-OXCMSG] section 2.2.1.19)	0x81BB	0x0040 (**PtypTime**)	0x01C9CA86A5089000 (2009/05/01 18:00:00.000)
PidLidReminderSet ([MS-OXORMDR] section 2.2.1.1)	0x8004	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidReminderDelta ([MS-OXORMDR] section 2.2.1.3)	0x81FF	0x0003 (**PtypInteger32**)	0x0000001E (30)

Property	Property ID	Property type	Value
PidLidReminderTime ([MS-OXORMDR] section 2.2.1.4)	0x8005	0x0040 (**PtypTime**)	0x01C9CA7E43442800 (2009/05/01 17:00:00.000)
PidLidReminderSignalTime ([MS-OXORMDR] section 2.2.1.2)	0x8006	0x0040 (**PtypTime**)	0x01C9CA7A1261F400 (2009/05/01 16:30:00.000)
PidLidBusyStatus (section 2.2.1.2)	0x81B6	0x0003 (**PtypInteger32**)	0x00000002 (olBusy)
PidLidLocation (section 2.2.1.4)	0x8009	0x001F (**PtypString**)	My Dentist's Office
PidLidAppointmentColor ([MS-OXPROPS] section 2.9)	0x82CA	0x0003 (**PtypInteger32**)	0x00000000
PidLidAppointmentStateFlags (section 2.2.1.10)	0x81B3	0x0003 (**PtypInteger32**)	0x00000000
PidLidAppointmentAuxiliaryFlags (section 2.2.1.3)	0x82D2	0x0003 (**PtypInteger32**)	0x00000000
PidLidAppointmentSubType (section 2.2.1.9)	0x8120	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidResponseStatus (section 2.2.1.11)	0x8122	0x0003 (**PtypInteger32**)	respNone (0x00000000)
PidLidFInvited (section 2.2.4.4)	0x81da	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidAppointmentDuration (section 2.2.1.7)	0x81a9	0x0003 (**PtypInteger32**)	0x0000003C (60)
PidLidAppointmentStartWhole (section 2.2.1.5)	0x81b2	0x0040 (**PtypTime**)	0x01C9CA7E43442800 (2009/05/01 17:00:00.000)
PidLidAppointmentEndWhole (section 2.2.1.6)	0x81ac	0x0040 (**PtypTime**)	0x01C9CA86A5089000 (2009/05/01 18:00:00.000)
PidLidClipStart (section 2.2.1.14)	0x81BA	0x0040 (**PtypTime**)	0x01C9CA7E43442800 (2009/05/01 17:00:00.000)
PidLidClipEnd (section 2.2.1.15)	0x81B9	0x0040 (**PtypTime**)	0x01C9CA86A5089000 (2009/05/01 18:00:00.000)

Property	Property ID	Property type	Value
PidLidRecurrenceType (section 2.2.1.45)	0x81FE	0x0003 (**PtypInteger32**)	0x00000000
PidLidRecurring (section 2.2.1.12)	0x81FD	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidTimeZoneDescription (section 2.2.1.40)	0x8213	0x001F (**PtypString**)	(GMT-08:00) Pacific Time (US & Canada)
PidLidAppointmentTimeZoneDefinitionStartDisplay (section 2.2.1.42)	0x83A8	0x0102 (**PtypBinary** ([MS-OXCDATA] section 2.11.1))	See paragraph marked *1 following this table.
PidLidAppointmentTimeZoneDefinitionEndDisplay (section 2.2.1.43)	0x83A9	0x0102 (**PtypBinary**)	See paragraph marked *1 following this table.
PidLidGlobalObjectId (section 2.2.1.27)	0x81E0	0x0102 (**PtypBinary**)	See paragraph marked *2 following this table
PidLidCleanGlobalObjectId (section 2.2.1.28)	0x81B8	0x0102 (**PtypBinary**)	See paragraph marked *2 following this table

*1 The start and end dates for this appointment are both set in the same time zone. For an example of this time zone definition BLOB, see section 4.1.4. The time zone data for this appointment is as follows:

```
cb: 184
lpb:
02013000020015005000610063006900660069006300200053007400610 06E00640061007200640020005400690 06
D006500020002013E000000D607000000000000000000000000000000E001000000000000C4FFFFFF00000A00000005
000200000000000000000004000000010002000000000000000002013E000200D707000000000000000000000000000000
0E001000000000000C4FFFFFF00000B000000010002000000000000000000030000000200020000000000000
```

*2 This appointment is a single instance, so the value of the **PidLidGlobalObjectId** and **PidLidCleanGlobalObjectId** properties are the same. For an example of the Global Object ID BLOB, see section 4.1.2. The value for this appointment is as follows:

```
cb: 56
lpb:
040000008200E00074C5B7101A82E008000000000020631F30F072C80100000000000000000010000000D97737CAB6762
A43BFF793851D08DB16
```

After setting all property values, the client can use the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) to commit the properties on the server. Without this, the newly created object will not be persisted. The server returns a success code that indicates that the data has been accepted.

Finally, the client uses the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3) to release the handle that the server had returned from the initial the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2).

4.2.2 Meeting Example

Mr. Glen John needs to set up a weekly half-hour meeting with a newly hired employee, named Mr. Dennis Saylor. Mr. John likes to have meetings with team members on Tuesdays, and he is available at 10:30 A.M. The following sections provide a description of what a client might do to accomplish these tasks and the responses a server might return.

4.2.2.1 Creating the Meeting

To create the Meeting object, the client uses the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2). The server returns a success code and a handle to a Message object.

The client then uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to transmit Mr. John's data to the server. An example of the data that might be sent by the client is shown in the following table.

Property	Property ID	Property type	Value
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	0x001F (**PtypString** ([MS-OXCDATA] section 2.11.1.2))	Weekly meeting
PidLidBusyStatus (section 2.2.1.2)	0x81B6	0x0003 (**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	0x00000002 (2)
PidLidAppointmentColor ([MS-OXPROPS] section 2.9)	0x82CA	0x0003 (**PtypInteger32**)	0x00000000 (0)
PidLidLocation (section 2.2.1.4)	0x8009	0x001F (**PtypString**)	Your Office
PidLidRecurring (section 2.2.1.12)	0x81FD	0x000B (**PtypBoolean** ([MS-OXCDATA] section 2.11.1))	0x01 (TRUE)
PidLidAppointmentStartWhole (section 2.2.1.5)	0x81B2	0x0040 (**PtypTime** ([MS-OXCDATA] section 2.11.1))	0x01C878A5984A4400 (2008/02/26 18:30:00.000)
PidLidAppointmentEndWhole (section 2.2.1.6)	0x81AC	0x0040 (**PtypTime**)	0x01C878A9C92C7800 (2008/02/26 19:00:00.000)
PidLidAppointmentDuration (section 2.2.1.7)	0x81A9	0x0003 (**PtypInteger32**)	0x0000001E (30)
PidLidAppointmentAuxiliaryFlags (section 2.2.1.3)	0x82D2	0x0003 (**PtypInteger32**)	0x00000000 (0)

Property	Property ID	Property type	Value
PidLidAppointmentSubType (section 2.2.1.9)	0x8120	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidAppointmentStateFlags (section 2.2.1.10)	0x81B3	0x0003 (**PtypInteger32**)	0x00000001 (1)
PidLidResponseStatus (section 2.2.1.11)	0x8122	0x0003 (**PtypInteger32**)	respOrganized (0x00000001)
PidLidAppointmentNotAllowPropose (section 2.2.1.26)	0x82D5	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidFInvited (section 2.2.4.4)	0x81DA	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidRecurrenceType (section 2.2.1.45)	0x81FE	0x0003 (**PtypInteger32**)	0x00000002 (2)
PidLidRecurrencePattern (section 2.2.1.46)	0x81FC	0x001F (**PtypString**)	Every Tuesday from 10:30 A.M. to 11:00 A.M.
PidLidTimeZoneDescription (section 2.2.1.40)	0x8213	0x001F (**PtypString**)	(GMT-08:00) Pacific Time (U.S. & Canada)
PidLidClipStart (section 2.2.1.14)	0x81BA	0x0040 (**PtypTime**)	0x01C8784D95BC0000 (2008/02/26 08:00:00.000)
PidLidClipEnd (section 2.2.1.15)	0x81B9	0x0040 (**PtypTime**)	0x0CB2E57949B47A00 (4500/08/31 23:59:00.000)
PidLidToAttendeesString (section 2.2.1.17)	0x82D9	0x001F (**PtypString**)	desaylor
PidLidAppointmentSequence (section 2.2.1.1)	0x81AF	0x0003 (**PtypInteger32**)	0x00000000 (0)
PidLidAutoFillLocation (section 2.2.4.8)	0x82E8	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidReminderDelta ([MS-OXORMDR] section 2.2.1.3)	0x81FF	0x0003 (**PtypInteger32**)	0x0000000F (15)
PidLidReminderTime ([MS-OXORMDR] section 2.2.1.4)	0x8005	0x0040 (**PtypTime**)	0x01C878A5984A4400 (2008/02/26 18:30:00.000)
PidLidReminderSignalTime ([MS-OXORMDR] section 2.2.1.2)	0x8006	0x0040 (**PtypTime**)	0x01C878A37FD92A00 (2008/02/26 18:15:00.000)

Property	Property ID	Property type	Value
PidLidCommonStart ([MS-OXCMSG] section 2.2.1.18)	0x81BC	0x0040 (**PtypTime**)	0x01C878A5984A4400 (2008/02/26 18:30:00.000)
PidLidCommonEnd ([MS-OXCMSG] section 2.2.1.19)	0x81BB	0x0040 (**PtypTime**)	0x01C878A9C92C7800 (2008/02/26 19:00:00.000)
PidLidReminderSet ([MS-OXORMDR] section 2.2.1.1)	0x8004	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidSideEffects ([MS-OXCMSG] section 2.2.1.16)	0x8002	0x0003 (**PtypInteger32**)	0x00000171 (369)
PidLidMeetingType (section 2.2.6.5)	0x8314	0x0003 (**PtypInteger32**)	0x00000001 (1)
PidTagMessageClass ([MS-OXCMSG] section 2.2.1.3)	0x001A	0x001F (**PtypString**)	IPM.Appointment
PidTagResponseRequested ([MS-OXPROPS] section 2.919	0x0063	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidTagIconIndex (section 2.2.1.49)	0x1080	0x0003 (**PtypInteger32**)	0x00000403 (1027)
PidLidTimeZoneStruct (section 2.2.1.39)	0x8214	0x0102 (**PtypBinary** ([MS-OXCDATA] section 2.11.1))	See paragraph marked *1 following this table.
PidLidAppointmentTimeZoneDefinitionRecur (section 2.2.1.41)	0x83AA	0x0102 (**PtypBinary**)	See paragraph marked *2 following this table.
PidLidAppointmentTimeZoneDefinitionStartDisplay (section 2.2.1.42)	0x83A8	0x0102 (**PtypBinary**)	See paragraph marked *3 following this table.
PidLidAppointmentTimeZoneDefinitionEndDisplay (section 2.2.1.43)	0x83A9	0x0102 (**PtypBinary**)	See paragraph marked *3 following this table.
PidLidGlobalObjectId (section 2.2.1.27)	0x81E0	0x0102 (**PtypBinary**)	See paragraph marked *4 following this table.
PidLidCleanGlobalObjectId (section 2.2.1.28)	0x81B8	0x0102 (**PtypBinary**)	See paragraph marked *4 following this table.
PidLidAppointmentRecur (section 2.2.1.44)	0x81AD	0x0102 (**PtypBinary**)	See paragraph marked *5 following this table.

Property	Property ID	Property type	Value
Best body properties	A body **stream** (1), the text of which was written by Mr. John, that indicates to Mr. Saylor the purpose of the meeting. For more information, see [MS-OXBBODY].		

*1 For an example of the **PidLidTimeZoneStruct** property BLOB, see section 4.1.5. The value for this Meeting object is as follows:

```
cb: 48

lpb:
E00100000000000C4FFFFFF000000000B0000000100020000000000000000000030000000200020000000000
000
```

*2 The date values for the **PidLidAppointmentTimeZoneDefinitionRecur** property for this appointment are both set in the same time zone. For an example of the time zone definition property BLOB, see section 4.1.4. The only difference between this BLOB and that in the **PidLidAppointmentTimeZoneDefinitionStartDisplay** and **PidLidAppointmentTimeZoneDefinitionEndDisplay** properties is that the **TZRULE_FLAG_RECUR_CURRENT_TZREG** flag is set in this BLOB. The value for this Meeting object is as follows:

```
cb: 184

lpb:
02013000020015005000610063006900660069006300200053007400610060E0064006100720064002000540069006
D006500020002013E000000D607000000000000000000000000000000E00100000000000C4FFFFFF00000A00000005
000200000000000000000000400000001000200000000000000002013E000300D707000000000000000000000000000000
0E00100000000000C4FFFFFF00000B000000010002000000000000000000030000000200020000000000000
```

*3 The start and end dates for this appointment are both set in the same time zone. The value for this Meeting object is as follows:

```
cb: 184
lpb:
02013000020015005000610063006900660069006300200053007400610060E0064006100720064002000540069006
D006500020002013E000000D607000000000000000000000000000000E00100000000000C4FFFFFF00000A00000005
000200000000000000000000400000001000200000000000000002013E000200D707000000000000000000000000000000
0E00100000000000C4FFFFFF00000B000000010002000000000000000000030000000200020000000000000
```

*4 This Meeting object is a recurring series, so the values of the **PidLidGlobalObjectId** property and the **PidLidCleanGlobalObjectId** property are the same. For an example of the Global Object ID BLOB, see section 4.1.2. The value for this Meeting object is as follows:

```
cb: 56
lpb:
040000008200E00074C5B7101A82E00800000000406FD661E473C80100000000000000000100000002A5844B3A444F
74A9C246C60886F116B
```

*5 Section 4.1.1.2 shows an example of the recurrence BLOB for a weekly recurring meeting. The value for this Meeting object is as follows:

```
cb: 80

lpb:
043004300B2001000000C021000001000000000000000400000023200000A000000000000000000000000000000002
088C30CDF80E95A0630000009300000760200009402000000000000000000000000
```

The client uses the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5) to add Dennis Saylor to the Meeting object, including the extra properties listed in the following table.

Property	Property ID	Property type	Value
PidTagRecipientFlags (section 2.2.4.10.1)	0x5FFD	0x0003 (**PtypInteger32**)	0x00000201 (513)
PidTagRecipientTrackStatus (section 2.2.4.10.2)	0x5FFF	0x0003 (**PtypInteger32**)	0x00000000 (0)

After setting all property values, the client can use the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) to commit the properties on the server. Without these properties, the newly created object will not be persisted. The server returns a success code that indicates that the data has been accepted.

4.2.2.2 Sending the Meeting Request

The client needs to use the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2) to create a new Meeting Request object in the Outbox folder, (as described in [MS-OXOSFLD] section 2.2, so that attendees can be notified of the event. The server returns a success code and a handle to a new Message object.

Next, the client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to set on this new Meeting Request object all the properties that were set on the Meeting object as described in section 4.2.2.1, except for the following:

- **PidLidBusyStatus** (section 2.2.1.2)
- **PidLidAppointmentStateFlags** (section 2.2.1.10)
- **PidLidResponseStatus** (section 2.2.1.11)
- **PidLidFInvited** (section 2.2.4.4)
- **PidLidAppointmentSequence** (section 2.2.1.1)
- **PidLidAutoFillLocation** (section 2.2.4.8)
- **PidLidReminderDelta** ([MS-OXORMDR] section 2.2.1.3)
- **PidLidReminderSignalTime** ([MS-OXORMDR] section 2.2.1.2)
- **PidLidSideEffects** ([MS-OXCMSG] section 2.2.1.16)

- **PidTagMessageClass** ([MS-OXCMSG] section 2.2.1.3)
- **PidTagIconIndex** (section 2.2.1.49)
- Best body properties

The values of the **PidLidReminderDelta** and **PidLidReminderSignalTime** **reminder properties** are not copied because the organizer kept the default reminder values. Instead, special values will be set on the Meeting Request object so that the receiving client uses default values that the attendee has defined.

In addition to the values that were already on the Meeting object, the client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to put the property values listed in the following table onto the Meeting Request object.

Property	Property ID	Property type	Value
PidTagMessageClass	0x001A	0x001F (**PtypString** ([MS-OXCDATA] section 2.11.1.2))	IPM.Schedule.Meeting.Request
PidTagIconIndex	0x1080	0x0003 (**PtypInteger32** ([MS-OXCDATA] section 2.11.1))	0xFFFFFFFF (-1)
PidTagStartDate (section 2.2.1.30)	0x0060	0x0040 (**PtypTime** ([MS-OXCDATA] section 2.11.1))	0x01C878A5984A4400 (2008/02/26 18:30:00.000)
PidTagEndDate (section 2.2.1.31)	0x0061	0x0040 (**PtypTime**)	0x01C878A9C92C7800 (2008/02/26 19:00:00.000)
PidTagOwnerAppointmentId (section 2.2.1.29)	0x0062	0x0003 (**PtypInteger32**)	0x4D9427D8 (1301555160)
PidLidBusyStatus	0x81B6	0x0003 (**PtypInteger32**)	0x00000001 (olTentative)
PidLidIntendedBusyStatus (section 2.2.6.4)	0x81E2	0x0003 (**PtypInteger32**)	0x00000002 (olBusy)
PidLidAppointmentStateFlags (section 2.2.1.10)	0x81B3	0x0003 (**PtypInteger32**)	0x00000003 (3)
PidLidResponseStatus	0x8122	0x0003 (**PtypInteger32**)	respNotResponded (0x00000005)
PidLidFInvited	0x81DA	0x000B (**PtypBoolean** ([MS-OXCDATA] section 2.11.1))	0x01 (TRUE)
PidLidAllAttendeesString (section 2.2.1.16)	0x81A8	0x001F (**PtypString**)	desaylor
PidLidAppointmentSequence	0x81AF	0x0003	0x00000000 (0)

Property	Property ID	Property type	Value
		(**PtypInteger32**)	If this had been an update, the sequence number (2) would have been incremented.
PidLidChangeHighlight (section 2.2.6.2)	0x82EC	0x0003 (**PtypInteger32**)	0x00000000 (0)
PidLidReminderDelta	0x81FF	0x0003 (**PtypInteger32**)	0x5AE980E1 (1525252321)
PidLidReminderSignalTime	0x8006	0x0040 (**PtypTime**)	0x01C878A5984A4400 (2008/02/26 18:30:00.000)
PidLidSideEffects	0x8002	0x0003 (**PtypInteger32**)	0x00001C61 (7265)
PidLidAttendeeCriticalChange (section 2.2.5.2)	0x81B5	0x0040 (**PtypTime**)	0x01C874276FF4F450 (2008/02/21 01:16:51.093)
PidLidWhere (section 2.2.5.3)	0x8219	0x001F (**PtypString**)	Your Office
PidLidAppointmentMessageClass (section 2.2.6.6)	0x8311	0x001F (**PtypString**)	IPM.Appointment
PidLidIsRecurring (section 2.2.1.13)	0x81E5	0x000B (**PtypBoolean** ([MS-OXCDATA] section 2.11.1))	0x01 (TRUE)
PidLidIsException (section 2.2.1.35)	0x81E4	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidTimeZone (section 2.2.5.6)	0x8212	0x0003 (**PtypInteger32**)	0x0000000D (13)
PidLidCalendarType (section 2.2.6.11)	0x81B7	0x0003 (**PtypInteger32**)	0x00000001 (1)
PidLidOwnerCriticalChange (section 2.2.1.34)	0x8128	0x0040 (**PtypTime**)	0x01C874276FF4F450 (2008/02/21 01:16:51.093)
Best body properties	A body stream (1), the text of which is the downlevel text, as specified in section 2.2.6.12, followed by a copy of the body text from the Meeting object.		

In addition to these properties, the client needs to use the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to add all properties that are required to send a Message object, as described in [MS-OXOMSG], to the Meeting Request object so that it can be delivered to the attendee. This client also needs to use the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5) to add a **RecipientRow** structure, as described in [MS-OXCDATA] section 2.8.3, for Mr. Saylor to the Meeting Request object.

After the Meeting Request object has been created and filled in, it will be sent instead of saved. The client uses the **RopSubmitMessage** ROP ([MS-OXCROPS] section 2.2.7.1) to send this Message object for transport.

After the server returns a success code from submission, the client makes the changes listed in the following table to the Meeting object on Mr. John's calendar by using the **RopSetProperties** ROP.

Property	Property ID	Property type	Value
PidLidFInvited	0x81DA	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidAppointmentSequence	0x81AF	0x0003 (**PtypInteger32**)	0x00000000 (0)
PidLidAppointmentSequenceTime (section 2.2.4.1)	0x82E7	0x0040 (**PtypTime**)	0x01C874276FF4F450 (2008/02/21 01:16:51.093)
PidLidAttendeeCriticalChange	0x81B5	0x0040 (**PtypTime**)	0x0CB34557A3DD4000 (4501/01/01 00:00:00.000)
PidLidOwnerCriticalChange	0x8128	0x0040 (**PtypTime**)	0x01C874276FF4F450 (2008/02/21 01:16:51.093)
PidTagOwnerAppointmentId	0x0062	0x0003 (**PtypInteger32**)	0x4D9427D8 (1301555160)

Finally, the client issues the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) to save these changes to the organizer's Meeting object and then releases both the meeting and Meeting Request objects by using the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3) for each.

4.2.2.3 Receiving the Meeting Request

After receiving the Meeting Request object, a client might tentatively add a Meeting object to the Calendar special folder in Mr. Saylor's mailbox.

To accomplish this task, the client uses **RopOpenMessage** ROP ([MS-OXCROPS] section 2.2.6.1) to obtain a handle to the Meeting Request object, and the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2) to create a Meeting object in the Calendar special folder. The server returns a handle to each of these objects, along with appropriate success codes.

Next, the client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to set, on this new Meeting object, all the properties that were set on the Meeting Request object as described in 4.2.2.2, except for the following:

- **PidTagMessageClass** ([MS-OXCMSG] section 2.2.1.3)
- **PidTagIconIndex** (section 2.2.1.49)
- **PidLidChangeHighlight** (section 2.2.6.2)
- **PidLidReminderDelta** ([MS-OXORMDR] section 2.2.1.3)
- **PidLidReminderSignalTime** ([MS-OXORMDR] section 2.2.1.2)
- **PidLidSideEffects** ([MS-OXCMSG] section 2.2.1.16)
- Best body properties

In addition to the values that were already on the Meeting object, the client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to put the property values listed in the following table onto the Meeting object.

Property	Property ID	Property type	Value
PidLidReminderDelta	0x81FF	0x0003 (**PtypInteger32**([MS-OXCDATA] section 2.11.1)	0x0000000F (15) The default value for this client, given that the value on the Meeting Request object was 0x5AE980E1.
PidLidReminderSignalTime	0x8006	0x0040 (**PtypTime** ([MS-OXCDATA] section 2.11.1)	0x01C878A37FD92A00 (2008/02/26 18:15:00.000)
PidTagMessageClass	0x001A	0x001F (**PtypString** ([MS-OXCDATA] section 2.11.1.2))	IPM.Appointment
PidTagIconIndex	0x1080	0x0003 (**PtypInteger32**)	0x00000403 (1027)
PidLidChangeHighlight	0x82EC	0x0003 (**PtypInteger32**)	0x00000E1F (3615)
PidLidSideEffects	0x8002	0x0003 (**PtypInteger32**)	0x00000171 (369)
Best body properties	The client can look for and remove the downlevel text, as described in section 2.2.6.12, before copying the text stream (1) onto the new Meeting object.		

The client needs to set the recipients (2) on the Meeting object by using the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5). The recipients (2) are obtained from the **RecipientRow** structures, as described in [MS-OXCDATA] section 2.8.3, of the Meeting Request object, as well as the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25). In addition, if the organizer (in this case, Mr. John) is not in the list of recipients (2), his information is obtained from the **PidTagSentRepresenting*** properties and added as a **RecipientRow** structure.

After setting all property values, the client can use the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) to commit the properties on the server. Without this, the newly created object will not be persisted. The server returns a success code that indicates that the data has been accepted.

The client sets the property shown in the following table on the Meeting Request object by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6), followed by the **RopSaveChangesMessage** ROP.

Property	Property ID	Property type	Value
PidTagProcessed (section 2.2.5.7)	0x7D01	0x000B (**PtypBoolean** ([MS-OXCDATA] section 2.11.1))	0x01 (TRUE)

Finally, the client uses the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3) to release the handle of the Meeting object and Meeting Request object.

4.2.2.4 Accepting the Meeting Request

After receiving the Meeting Request object, Mr. Dennis Saylor indicates he will attend the meeting with Mr. Glen John. The client needs to send a Meeting Response object back to Mr. John so that he knows that Mr. Saylor will attend.

To accomplish this task, the client uses the **RopOpenMessage** ROP ([MS-OXCROPS] section 2.2.6.1) to obtain a handle to the tentative Meeting object, and the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2) to create a Meeting object in the Calendar special folder. The server returns a handle to each of these objects, along with appropriate success codes.

The client uses the **RopCopyTo** ROP ([MS-OXCROPS] section 2.2.8.12) to copy all properties from the tentative Meeting object to the new Meeting object. The properties listed in the following table are then modified on the new Meeting object by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6).

Property	Property ID	Property type	Value
PidLidAppointmentMessageClass (section 2.2.6.6)	0x8311	0x001F (**PtypString** ([MS-OXCDATA] section 2.11.1.2))	IPM.Appointment
PidLidBusyStatus (section 2.2.1.2)	0x81B6	0x0003 (**PtypInteger32** ([MS-OXCDATA] section 2.11.1))	0x00000002 (olBusy)
PidLidResponseStatus (section 2.2.1.11)	0x8122	0x0003 (**PtypInteger32**)	respAccepted (0x00000003)
PidLidAppointmentReplyTime (section 2.2.4.3)	0x8139	0x0040 (**PtypTime** ([MS-OXCDATA] section 2.11.1))	0x01C87427BCCA9A00 (2008/02/21 01:19:00.000)
PidLidAppointmentReplyName (section 2.2.4.5)	0x81AE	0x001F (**PtypString**)	desaylor

The client uses the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) to persist the new Meeting object in Mr. Saylor's Calendar special folder. It releases a handle to the tentative Meeting object by using the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3), and then deletes the tentative Meeting object by using the **RopDeleteMessages** ROP ([MS-OXCROPS] section 2.2.4.11).

Now the client needs to respond to the organizer. It uses the **RopCreateMessage** ROP to create a new Meeting Response object in the Outbox folder. The server returns a success code and a handle to a new Message object.

The client uses the **RopGetPropertiesSpecific** ROP ([MS-OXCROPS] section 2.2.8.3) on the Meeting object and then uses the **RopSetProperties** ROP to copy, onto this new Meeting Response object, the value of the following properties that were on the Meeting object:

- **PidTagNormalizedSubject** ([MS-OXCMSG] section 2.2.1.10)
- **PidLidBusyStatus**
- **PidLidAppointmentColor** ([MS-OXPROPS] section 2.9)

- **PidLidLocation** (section 2.2.1.4)
- **PidLidRecurring** (section 2.2.1.12)
- **PidLidAppointmentStartWhole** (section 2.2.1.5)
- **PidLidAppointmentEndWhole** (section 2.2.1.6)
- **PidLidAppointmentTimeZoneDefinitionStartDisplay** (section 2.2.1.42)
- **PidLidAppointmentTimeZoneDefinitionEndDisplay** (section 2.2.1.43)
- **PidLidAppointmentDuration** (section 2.2.1.7)
- **PidLidAppointmentAuxiliaryFlags** (section 2.2.1.3)
- **PidLidAppointmentSubType** (section 2.2.1.9)
- **PidLidAppointmentRecur** (section 2.2.1.44)
- **PidLidRecurrenceType** (section 2.2.1.45)
- **PidLidRecurrencePattern** (section 2.2.1.46)
- **PidLidTimeZoneStruct** (section 2.2.1.39)
- **PidLidAppointmentTimeZoneDefinitionRecur** (section 2.2.1.41)
- **PidLidTimeZoneDescription** (section 2.2.1.40)
- **PidLidClipStart** (section 2.2.1.14)
- **PidLidClipEnd** (section 2.2.1.15)
- **PidLidAppointmentSequence** (section 2.2.1.1)
- **PidLidCommonStart** ([MS-OXCMSG] section 2.2.1.18)
- **PidLidCommonEnd** ([MS-OXCMSG] section 2.2.1.19)
- **PidLidWhere** (section 2.2.5.3)
- **PidLidGlobalObjectId** (section 2.2.1.27)
- **PidLidCleanGlobalObjectId** (section 2.2.1.28)
- **PidLidAppointmentMessageClass**
- **PidLidIsRecurring** (section 2.2.1.13)
- **PidLidIsException** (section 2.2.1.35)
- **PidLidTimeZone** (section 2.2.5.6)
- **PidLidCalendarType** (section 2.2.6.11)
- **PidLidOwnerCriticalChange** (section 2.2.1.34)
- **PidTagStartDate** (section 2.2.1.30)

- **PidTagEndDate** (section 2.2.1.31)
- **PidTagOwnerAppointmentId** (section 2.2.1.29)

In addition to the values that were already on the Meeting object, the client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to put the property values listed in the following table onto the Meeting Response object.

Property	Property ID	Property type	Value
PidTagMessageClass ([MS-OXCMSG] section 2.2.1.3)	0x001A	0x001F (**PtypString**)	IPM.Schedule.Meeting.Resp.Pos
PidTagSubjectPrefix ([MS-OXCMSG] section 2.2.1.9)	0x003D	0x001F (**PtypString**)	Accepted:
PidLidSideEffects ([MS-OXCMSG] section 2.2.1.16)	0x8002	0x0003 (**PtypInteger32**)	0x00001C61 (7265)
PidLidAttendeeCriticalChange (section 2.2.5.2)	0x81B5	0x0040 (**PtypTime**)	0x01C87427BF62AA00 (2008/02/21 01:19:04.352)
PidLidIsSilent (section 2.2.7.7)	0x81E6	0x000B (**PtypBoolean** ([MS-OXCDATA] section 2.11.1))	0x01 (TRUE)

The client adds the organizer by using the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5), and then sends the object via the **RopSubmitMessage** ROP ([MS-OXCROPS] section 2.2.7.1). After the server returns a success code from submission, the client releases both the Meeting object and the Meeting Response objects with a the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3) for each.

4.2.2.5 Receiving the Meeting Response

When Mr. John receives Mr. Saylor's response, the response can be recorded on the Meeting object in Mr. John's Calendar special folder.

To accomplish this task, the client issues the **RopOpenMessage** ROP ([MS-OXCROPS] section 2.2.6.1) to get a handle to the object, and the **RopGetPropertiesSpecific** ROP ([MS-OXCROPS] section 2.2.8.3) to get the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3). The server returns a handle to the Meeting Response object and the value for this property, which is "IPM.Schedule.Meeting.Resp.Pos".

After seeing that this is a Meeting Response object, the client issues the **RopOpenMessage** ROP for the Meeting object in the Calendar special folder. The server returns a handle for the Meeting object. The server also returns the set of **RecipientRow** structures, as described in [MS-OXCDATA] section 2.8.3, as a result of opening the object. These **RecipientRow** structures need to be stored in an in-memory recipient cache so that they can be manipulated and then later replace those on the Meeting object.

The client uses the **RopGetPropertiesSpecific** ROP ([MS-OXCROPS] section 2.2.8.3) to get the following properties from the Meeting Response object, the values of which are returned by the server:

- **PidTagSentRepresentingSearchKey** ([MS-OXOMSG] section 2.2.1.58)

- **PidTagSentRepresentingName** ([MS-OXOMSG] section 2.2.1.57)
- **PidTagSenderSearchKey** ([MS-OXOMSG] section 2.2.1.52)
- **PidTagSenderName** ([MS-OXOMSG] section 2.2.1.51)
- **PidLidAttendeeCriticalChange** (section 2.2.5.2)

If the **PidTagSentRepresentingSearchKey** and **PidTagSentRepresentingName** properties are available, these are used for searching for the **RecipientRow** structure. Otherwise, the **PidTagSenderSearchKey** and **PidTagSenderName** properties are used. The client looks through the **RecipientRow** structures, first attempting to find a **PidTagSearchKey** property ([MS-OXCPRPT] section 2.2.1.9) value that matches the value of the **PidTagSentRepresentingSearchKey** (or **PidTagSenderSearchKey**) property. If no match is found, then the client attempts to match the value of the **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) from the **RecipientRow** structure with the value of the **PidTagSentRepresentingName** (or **PidTagSenderName**) property.

If a **RecipientRow** structure is not found, a new one with its **PidTagRecipientType** property ([MS-OXOMSG] section 2.2.3.1) set to **MAPI_CC** is added to the in-memory recipient cache to represent this attendee. The extra properties that are added to the in-memory **RecipientRow** structure that represents this attendee are listed in the following table.

Property	Property ID	Property type	Value
PidTagRecipientTrackStatus (section 2.2.4.10.2)	0x5FFF	0x0003 (**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	respAccepted (0x00000003)
PidTagRecipientTrackStatusTime (section 2.2.4.10.3)	0x5FFB	0x0040 (**PtypTime** ([MS-OXCDATA] section 2.11.1)	0x01C87427BCCA9A00 (2008/02/21 01:19:00.000)

The value of the **PidLidAttendeeCriticalChange** property is rounded down to the nearest minute and then set as the value of the **PidTagRecipientTrackStatusTime** property.

The client uses the **RopRemoveAllRecipients** ROP ([MS-OXCROPS] section 2.2.6.4) to delete all the recipients (2) from the Meeting object and then uses the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5) to copy the in-memory recipient cache back onto the Message object.

The client sets the property listed in the following table on the Meeting Response object by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6), followed by the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3).

Property	Property ID	Property type	Value
PidTagProcessed (section 2.2.5.7)	0x7D01	0x000B (**PtypBoolean** ([MS-OXCDATA] section 2.11.1))	0x01 (TRUE)

Finally, the client releases both the Meeting object and the Meeting Response object by using the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3).

4.2.2.6 Creating and Sending the Exception

Mr. John will be out of the office one Tuesday and therefore wants to move that instance to a Wednesday. He creates an exception for this instance, adds some comments in the object body as to why it is being changed, and then sends a Meeting Update object to notify Mr. Saylor of the new date.

To accomplish this task, the client uses the **RopOpenMessage** ROP ([MS-OXCROPS] section 2.2.6.1) to open the Meeting object from Mr. John's Calendar special folder, to which the server returns a success code and a handle to the Meeting object.

The data for the exception is written to an Embedded Message object in an Attachment object on the Meeting object. A client first uses the **RopCreateAttachment** ROP ([MS-OXCROPS] section 2.2.6.13) to create the Attachment object. A server returns a success code and a handle to the new Attachment object. The property listed in the following table is set on the Attachment object.

Property type	Property	Property ID	Value
0x0003 (**PtypInteger32** ([MS-OXCDATA] section 2.11.1))	**PidTagAttachMethod** ([MS-OXCMSG] section 2.2.2.9)	0x3705	0x00000005 (ATTACH_EMBEDDED_MSG)

After setting the attachment method, the client uses the **RopOpenEmbeddedMessage** ROP ([MS-OXCROPS] section 2.2.6.16) with the **OpenModeFlag** field set to **TRUE** to request a new Embedded Message object from the Attachment object. The server returns a success code and a handle to the new Embedded Message object. The client then uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to set the properties listed in the following table on the Exception Embedded Message object, as described in section 2.2.10.2.

Property	Property ID	Property type	Value
PidTagMessageClass ([MS-OXCMSG] section 2.2.1.3)	0x001A	0x001F (**PtypString** ([MS-OXCDATA] section 2.11.1.2))	IPM.OLE.class.{00061055-0000-0000-C000-000000000046}
PidLidBusyStatus (section 2.2.1.2)	0x81B6	0x0003 (**PtypInteger32**)	0x00000002 (2)
PidLidAppointmentStartWhole (section 2.2.1.5)	0x81B2	0x0040 (**PtypTime** ([MS-OXCDATA] section 2.11.1))	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidAppointmentEndWhole (section 2.2.1.6)	0x81AC	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidLidAppointmentTimeZoneDefinitionStartDisplay (section 2.2.1.42)	0x83A8	0x0102 (**PtypBinary** ([MS-OXCDATA]	See the paragraph marked *1 that follows this table.

Property	Property ID	Property type	Value
		section 2.11.1))	
PidLidAppointmentTimeZoneDefinitionEndDisplay (section 2.2.1.43)	0x83A8	0x0102 (**PtypBinary**)	See the paragraph marked *1 that follows this table.
PidLidAppointmentDuration (section 2.2.1.7)	0x81A9	0x0003 (**PtypInteger32**)	0x0000001E (30)
PidLidAppointmentSubType (section 2.2.1.9)	0x8120	0x000B (**PtypBoolean** ([MS-OXCDATA] section 2.11.1))	0x00 (FALSE)
PidLidExceptionReplaceTime (section 2.2.10.2.5)	0x83AC	0x0040 (**PtypTime**)	0x01C88E9DDA16DC00 (2008/03/25 17:30:00.000)
PidLidFInvited (section 2.2.4.4)	0x81DA	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidFExceptionalBody (section 2.2.10.2.6)	0x82D8	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidClipStart (section 2.2.1.14)	0x81BA	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidClipEnd (section 2.2.1.15)	0x81B9	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidLidToAttendeesString (section 2.2.1.17)	0x82D9	0x001F (**PtypString**)	desaylor
PidLidReminderTime ([MS-OXORMDR] section 2.2.1.4)	0x8005	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidCommonStart ([MS-OXCMSG] section 2.2.1.18)	0x81BC	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidCommonEnd ([MS-OXCMSG] section 2.2.1.19)	0x81BB	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidLidOwnerCriticalChange (section 2.2.1.34)	0x8128	0x0040 (**PtypTime**)	0x01C874289289D700 (2008/02/21 01:24:58.608)
PidLidMeetingType (section 2.2.6.5)	0x8314	0x0003 (**PtypInteger32**)	0x00010000 (65536)
PidTagStartDate (section 2.2.1.30)	0x0060	0x0040	0x01C88E9DDA16DC00 (2008/03/25

Property	Property ID	Property type	Value
		(**PtypTime**)	17:30:00.000)
PidTagEndDate (section 2.2.1.31)	0x0061	0x0040 (**PtypTime**)	0x01C88EA20AF91000 (2008/03/25 18:00:00.000)
PidTagOwnerAppointmentId (section 2.2.1.29)	0x0062	0x0003 (**PtypInteger32**)	0x4D9427D8 (1301555160)
Best body properties	A body stream (1), the text of which was written by Mr. John. For more information about body streams (1), see [MS-OXBBODY].		

*1 The start and end dates for this appointment are both set in the same time zone. For a description of the time zone definition property BLOB, see section 4.1.4. The value for this exception (and is the same as the associated Meeting object) is as follows.

```
cb: 184
lpb:
02013000020015005000610063006900660069006300200053007400061006E00640061007200640020005400690060
D00650002000 2013E000000D607000000000000000000000000000000E001000000000000C4FFFFFF00000A00000005
000200000000000000000004000000010002000000000000000002013E000200D70700000000000000000000000000000
0E001000000000000C4FFFFFF00000B000000010002000000000000000000030000000200020000000000000
```

The client uses the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5) to add all the recipients (2) from the Meeting object onto the Exception Embedded Message object, as described in section 2.2.10.2, and then saves the new Exception Embedded Message object by using the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3), to which the server returns success codes.

The client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to set the properties listed in the following table on the Exception Attachment object (not the Exception Embedded Message object).

Property	Property ID	Property type	Value
PidTagExceptionStartTime (section 2.2.10.1.4)	0x7FFB	0x0040 (**PtypTime**)	0x01C88F2C5821C400 (2008/03/26 10:30:00.000)
PidTagExceptionEndTime (section 2.2.10.1.5)	0x7FFC	0x0040 (**PtypTime**)	0x01C88F308903F800 (2008/03/26 11:00:00.000)
PidTagExceptionReplaceTime (section 2.2.10.1.6)	0x7FF9	0x0040 (**PtypTime**)	0x01C88E9DDA16DC00 (2008/03/25 17:30:00.000)
PidTagAttachmentFlags ([MS-OXCMSG] section 2.2.2.23)	0x7FFD	0x0003 (**PtypInteger32**)	0x00000002 (**afException**)
PidTagAttachmentHidden ([MS-OXCMSG] section 2.2.2.24)	0x7FFE	0x000B (**PtypBoolean**)	0x01 (TRUE)

The client uses the **RopSaveChangesAttachment** ROP ([MS-OXCROPS] section 2.2.6.15) to save the changes to the Attachment object.

The client needs to use the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2) to create a new Meeting Request object in the Outbox folder so that attendees can be notified of the change. The server returns a success code and a handle to a new Message object.

Next, the client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to set the properties listed in the following table on this new Meeting Request object.

Property	Property ID	Property type	Value
PidTagMessageClass	0x001A	0x001F (**PtypString**)	IPM.Schedule.Meeting.Request
PidLidBusyStatus	0x81B6	0x0003 (**PtypInteger32**)	0x00000001 (1)
PidLidAppointmentColor ([MS-OXPROPS] section 2.9)	0x82CA	0x0003 (**PtypInteger32**)	0x00000000 (0)
PidLidIntendedBusyStatus (section 2.2.6.4)	0x81E2	0x0003 (**PtypInteger32**)	0x00000002 (2)
PidLidLocation (section 2.2.1.4)	0x8009	0x001F (**PtypString**)	Your Office
PidLidRecurring (section 2.2.1.12)	0x81FD	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidAppointmentStartWhole	0x81B2	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidAppointmentEndWhole	0x81AC	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidLidTimeZoneStruct (section 2.2.1.39)	0x8214	0x0102 (**PtypBinary**)	See the paragraph marked *1 that follows this table.
PidLidAppointmentTimeZoneDefinitionStartDisplay	0x83A8	0x0102 (**PtypBinary**)	See the paragraph marked *2 that follows this table.
PidLidAppointmentTimeZoneDefinitionEndDisplay	0x83A9	0x0102 (**PtypBinary**)	See the paragraph marked *2 that follows this table.
PidLidAppointmentTimeZoneDefinitionRecur (section 2.2.1.41)	0x83AA	0x0102 (**PtypBinary**)	See the paragraph marked *3 that follows this table.
PidLidAppointmentDuration	0x81A9	0x0003 (**PtypInteger32**)	0x0000001E (30)
PidLidAppointmentAuxiliaryFlags (section 2.2.1.3)	0x82D2	0x0003 (**PtypInteger3**	0x00000000 (0)

Property	Property ID	Property type	Value
		2)	
PidLidAppointmentSubType	0x8120	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidAppointmentStateFlags (section 2.2.1.10)	0x81B3	0x0003 (**PtypInteger32**)	0x00000003 (3)
PidLidResponseStatus (section 2.2.1.11)	0x8122	0x0003 (**PtypInteger32**)	respNotResponded (0x00000005)
PidLidAppointmentNotAllowPropose (section 2.2.1.26	0x82D5	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidFExceptionalAttendees (section 2.2.2.3)	0x82D7	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidFExceptionalBody	0x82D8	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidRecurrenceType (section 2.2.1.45)	0x81FE	0x0003 (**PtypInteger32**)	0x00000002 (2)
PidLidRecurrencePattern (section 2.2.1.46)	0x81FC	0x001F (**PtypString**)	Every Tuesday from 10:30 A.M. to 11:00 A.M.
PidLidTimeZoneDescription (section 2.2.1.40)	0x8213	0x001F (**PtypString**)	(GMT-08:00) Pacific Time (US & Canada)
PidLidClipStart	0x81BA	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidClipEnd	0x81B9	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidLidAllAttendeesString (section 2.2.1.16)	0x81A8	0x001F (**PtypString**)	desaylor
PidLidToAttendeesString	0x82D9	0x001F (**PtypString**)	desaylor
PidLidAppointmentSequence (section 2.2.1.1)	0x81AF	0x0003 (**PtypInteger32**)	0x00000000 (0)
PidLidAppointmentSequenceTime (section 2.2.4.1)	0x82E7	0x0040 (**PtypTime**)	0x01C874276FF4F450 (2008/02/21 01:16:51.093)
PidLidChangeHighlight (section 2.2.6.2)	0x82EC	0x0003 (**PtypInteger32**)	0x00000083 (131)

Property	Property ID	Property type	Value
PidLidReminderDelta ([MS-OXORMDR] section 2.2.1.3)	0x81FF	0x0003 (**PtypInteger32**)	0x5AE980E1 (1525252321)
PidLidReminderTime	0x8005	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidReminderSignalTime ([MS-OXORMDR] section 2.2.1.2)	0x8006	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidCommonStart	0x81BC	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidCommonEnd	0x81BB	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidLidReminderSet	0x8004	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidSideEffects ([MS-OXCMSG] section 2.2.1.16)	0x8002	0x0003 (**PtypInteger32**)	0x00001C61 (7265)
PidLidAttendeeCriticalChange (section 2.2.5.2)	0x81B5	0x0040 (**PtypTime**)	0x01C8742891F14080 (2008/02/21 01:24:57.608)
PidLidWhere (section 2.2.5.3)	0x8219	0x001F (**PtypString**)	Your Office
PidLidGlobalObjectId (section 2.2.1.27)	0x81E0	0x0102 (**PtypBinary**)	See the paragraph marked *4 that follows this table.
PidLidCleanGlobalObjectId (section 2.2.1.28)	0x81B8	0x0102 (**PtypBinary**)	See the paragraph marked *5 that follows this table.
PidLidAppointmentMessageClass (section 2.2.6.6)	0x8311	0x001F (**PtypString**)	IPM.Appointment
PidLidIsRecurring (section 2.2.1.13)	0x81E5	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidIsException (section 2.2.1.35)	0x81E4	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidTimeZone (section 2.2.5.6)	0x8212	0x0003 (**PtypInteger32**)	0x0000000D (13)
PidLidCalendarType (section 2.2.6.11)	0x81B7	0x0003 (**PtypInteger32**)	0x00000001 (1)
PidLidOwnerCriticalChange	0x8128	0x0040	0x01C874289289D700

Property	Property ID	Property type	Value
		(**PtypTime**)	(2008/02/21 01:24:58.608)
PidLidMeetingType	0x8314	0x0003 (**PtypInteger32**)	0x00010000 (65536)
PidLidOldLocation (section 2.2.6.7)	0x8316	0x001F (**PtypString**)	(NULL)
PidLidOldWhenStartWhole (section 2.2.6.8)	0x83CC	0x0040 (**PtypTime**)	0x01C88E9DDA16DC00 (2008/03/25 17:30:00.000)
PidLidOldWhenEndWhole (section 2.2.6.9)	0x83CD	0x0040 (**PtypTime**)	0x01C88EA20AF91000 (2008/03/25 18:00:00.000)
PidTagResponseRequested ([MS-OXOMSG])	0x0063	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidTagStartDate	0x0060	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidTagEndDate	0x0061	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidTagOwnerAppointmentId	0x0062	0x0003 (**PtypInteger32**)	0x4D9427D8
best body properties	A body stream (1), the text of which is the downlevel text, as specified in section 2.2.6.12, followed by a copy of the body text from the Exception Embedded Message object.		

*1 For a description of the **PidLidTimeZoneStruct** property BLOB, see section 4.1.5. The value for this Meeting Request object is as follows.

```
cb: 48

lpb:
E00100000000000C4FFFFFF000000000B00000001000200000000000000000000000300000002000200000000000
000
```

*2 The dates in the **PidLidAppointmentTimeZoneDefinitionRecur** property for this appointment are both set in the same time zone. For a description of the time zone definition property BLOB, see section 4.1.4. The only difference between this BLOB and that in the **PidLidAppointmentTimeZoneDefinitionStartDisplay** and **PidLidAppointmentTimeZoneDefinitionEndDisplay** properties is that the **TZRULE_FLAG_RECUR_CURRENT_TZREG** flag is set in this BLOB. The value for this Meeting Request object is as follows.

```
cb: 184

lpb:
020130000200150050006100630069006600690063002000530074006100 6E00640061007200640020005400690006
D006500020002013E000000D607000000000000000000000000000000E001000000000000C4FFFFFF00000A00000005
000200000000000000000000400000001000200000000000000002013E000300D707000000000000000000000000000000
0E001000000000000C4FFFFFF00000B000000010002000000000000000000030000000200020000000000000000
```

*3 The start and end dates for this appointment are both set in the same time zone. The value for this Meeting Request object is as follows.

```
cb: 184
lpb:
020130000200150050006100630069006600690063002000530074006100 6E00640061007200640020005400690006
D006500020002013E000000D607000000000000000000000000000000E001000000000000C4FFFFFF00000A00000005
000200000000000000000000400000001000200000000000000002013E000200D707000000000000000000000000000000
0E001000000000000C4FFFFFF00000B000000010002000000000000000000030000000200020000000000000000
```

*4 The value of the **PidLidGlobalObjectId** property for this Meeting Request object is as follows. For a description of the Global Object ID BLOB, see section 4.1.2.

```
cb: 56
lpb:
040000008200E00074C5B7101A82E00807D803195025D461E473C801000000000000000000100000002A5844B3A444F
74A9C246C60886F116B
```

*5 The value of the **PidLidCleanGlobalObjectId** for this Meeting Request object is as follows. This value is identical to the value of the **PidLidGlobalObjectId** property, except that the **Year**, **Month**, and **Day** fields are filled with zeros.

```
cb: 56
lpb:
040000008200E00074C5B7101A82E008000000005025D461E473C801000000000000000000100000002A5844B3A444F
74A9C246C60886F116B
```

In addition to these properties, the client needs to use the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to add all properties that are required to send a Message object, as described in [MS-OXOMSG], to the Meeting Request object so that it can be delivered to the attendee. This client also needs to use the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5) to add a **RecipientRow** structure, as described in [MS-OXCDATA] section 2.8.3, for Mr. Saylor to the Meeting Request object.

Now that the Meeting Request object has been created and filled in, it will be sent instead of saved. The client uses the **RopSubmitMessage** ROP ([MS-OXCROPS] section 2.2.7.1) to send this Message object for transport.

The client makes the changes listed in the following table to the Meeting object (the object that represents the recurring series) on Mr. John's calendar by using the **RopSetProperties** ROP.

Property	Property ID	Property type	Value
PidLidAppointmentRecur (section 2.2.1.44)	0x81AD	0x0102 (**PtypBinary**)	See the paragraph marked *1 that follows this table.
PidLidFExceptionalAttendees	0x82D7	0x000B (**PtypBoolean**)	0x01 (TRUE)

*1 The value of the **PidLidAppointmentRecur** property will include necessary information about this new exception. The new value for this Meeting object is as follows.

```
cb: 114

lpb:
043004300B2001000000C021000001000000000000000400000023200000A000000000000000010000000A025C40C0
1000000402BC40C2088C30CDF80E95A0630000009300000760200009402000001 00B62DC40CD42DC40C1628C40C00
020000000004000000000000000000000000000000
```

Finally, the client issues the **RopSaveChangesMessage** ROP to save the Meeting object that represents the recurring series, and then uses the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3) to release all handles (Embedded Message objects, Attachment objects, Meeting objects, and Meeting Request objects).

4.2.2.7 Accepting the Exception

After receiving the Meeting Update object, Mr. Dennis Saylor indicates that the change will still work with his schedule. The Calendar object in Mr. Saylor's Calendar folder needs to be updated, and a Meeting Response object needs to be sent back to Mr. John.

To accomplish this task, the client uses the **RopOpenMessage** ROP ([MS-OXCROPS] section 2.2.6.1) to open the Meeting Update object to which the server returns a success code and a handle. The client uses the **RopGetPropertiesSpecific** ROP ([MS-OXCROPS] section 2.2.8.3) to get at least the following properties: **PidTagOwnerAppointmentId** (section 2.2.1.29), **PidLidGlobalObjectId** (section 2.2.1.27), and **PidLidCleanGlobalObjectId** (section 2.2.1.28).

The client uses the **RopGetContentsTable** ROP ([MS-OXCROPS] section 2.2.4.14) to open the **contents table** of the Calendar special folder. The server returns a handle to the contents table. The client sets at least the following column set on the contents table by using the **RopSetColumns** ROP ([MS-OXCROPS] section 2.2.5.1):

- **PidTagMid** ([MS-OXCFXICS] section 2.2.1.2.1)
- **PidTagOwnerAppointmentId**
- **PidLidGlobalObjectId**

The Meeting Update object in this example has a value for the **PidTagOwnerAppointmentId** property, so the client uses the **RopSortTable** ROP ([MS-OXCROPS] section 2.2.5.2) to sort the contents table in ascending order of this property. The client then uses the **RopFindRow** ROP ([MS-OXCROPS] section 2.2.5.13) to find the first matching table row. The server returns a success code with the first matching row or returns an error code if a matching row was not found.

For each matching row, the client compares the value of the **PidLidCleanGlobalObjectId** property from the Meeting Update object with the value of the **PidLidGlobalObjectId** property in the row,

until a match is found. (If a match had not been found, a client would search for an orphan instance by trying to match the value of the **PidLidGlobalObjectId** (section 2.2.1.27) property from the Meeting Update object (because this Meeting Update object represents an exception). If an orphan instance is not found, a client would search for a matching row with the value of 0 (zero) for the **PidTagOwnerAppointmentId** property. If a matching recurring series or orphan exception still could not be found, then it would be assumed that the Meeting object does not exist in the folder and the Meeting Update object would be treated as a Meeting Request object.) After finding a matching row, the client issues the **RopOpenMessage** ROP by using the value of the **PidTagMid** property from that row to open the Meeting object, to which the server returns a success code and a handle.

Having obtained the recurring series, the client tries to find the Exception Attachment object. The client uses the **RopGetAttachmentTable** ROP ([MS-OXCROPS] section 2.2.6.17) to open the list of attachments. The client uses **RopSetColumns** to set at least the following columns on this table:

- **PidTagAttachMethod** ([MS-OXCMSG] section 2.2.2.9)
- **PidTagAttachmentFlags** ([MS-OXCMSG] section 2.2.2.23)
- **PidTagAttachNumber** ([MS-OXCMSG] section 2.2.2.6)
- **PidTagExceptionReplaceTime** (section 2.2.10.1.6)

The client uses **RopQueryRows** ([MS-OXCROPS] section 2.2.5.4) to loop through the rows in the **attachments table**, attempting to find the matching Exception Attachment object. If the value of the **PidTagAttachmentFlags** property in a row does not include the **afException** flag, the attachment does not represent an exception. To find the matching Exception Attachment object, the client uses the values of the **Day**, **Month**, and **Year** fields of the **PidLidGlobalObjectId** property on the Meeting Update object to compute the replace date and time and looks for an Exception Attachment object with a matching value. Note that if the Exception Attachment object has the **PidTagExceptionReplaceTime** property (section 2.2.10.1.6), the value of this property is compared with the computed replace time to determine whether the attachment is the matching exception. If the attachment does not have this property, the client needs to use the **RopOpenAttachment** ROP ([MS-OXCROPS] section 2.2.6.12), **RopOpenEmbeddedMessage** ([MS-OXCROPS] section 2.2.6.16), and **RopGetPropertiesSpecific** ([MS-OXCROPS] section 2.2.8.3) ROPs to get the **PidLidExceptionReplaceTime** property from the Exception Embedded Message object and match that value against the computed replace time.

In this example, an Exception Attachment object does not exist, so the client uses the **RopCreateAttachment** ROP ([MS-OXCROPS] section 2.2.6.13) to create a new one, to which the server returns a success code and a handle. The client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to set the property in the following table on the Attachment object.

Property	Property ID	Property type	Value
PidTagAttachMethod	0x3705	0x0003 (**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	0x00000005 (ATTACH_EMBEDDED_MSG)

After setting the attachment method, the client uses the **RopOpenEmbeddedMessage** ROP ([MS-OXCROPS] section 2.2.6.16) with the **OpenModeFlag** field set to **TRUE** to request a new Embedded Message object from the Attachment object. The server returns a success code and a handle to the new Embedded Message object. The client then uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to set the properties listed in the following table on the Exception Embedded Message object, as specified in section 2.2.10.2, as copied from the Meeting Request object:

Property	Property ID	Property type	Value
PidTagMessageClass ([MS-OXCMSG] section 2.2.1.3)	0x001A	0x001F (**PtypString** ([MS-OXCDATA] section 2.11.1.2))	IPM.OLE.class.{00061055-0000-0000-C000-000000000046}
PidTagSubjectPrefix ([MS-OXCMSG] section 2.2.1.9)	0x003D	0x001F (**PtypString**)	
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	0x001F (**PtypString**)	Weekly meeting
PidLidBusyStatus (section 2.2.1.2)	0x81B6	0x0003 (**PtypInteger32**)	0x00000001 (olTentative)
PidLidIntendedBusyStatus (section 2.2.6.4)	0x81E2	0x0003 (**PtypInteger32**)	0x00000002 (olBusy)
PidLidLocation (section 2.2.1.4)	0x8009	0x001F (**PtypString**)	Your Office
PidLidRecurring (section 2.2.1.12)	0x81FD	0x000B (**PtypBoolean** ([MS-OXCDATA] section 2.11.1))	0x01 (TRUE)
PidLidAppointmentStartWhole (section 2.2.1.5)	0x81B2	0x0040 (**PtypTime** ([MS-OXCDATA] section 2.11.1))	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidAppointmentEndWhole (section 2.2.1.6)	0x81AC	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidLidTimeZoneStruct (section 2.2.1.39)	0x8214	0x0102 (**PtypBinary** ([MS-OXCDATA] section 2.11.1))	See the paragraph marked *1 that follows this table.
PidLidAppointmentTimeZoneDefinitionStartDisplay (section 2.2.1.42)	0x83A8	0x0102 (**PtypBinary**)	See the paragraph marked *2 that follows this table.
PidLidAppointmentTimeZoneDefinitionEndDisplay (section 2.2.1.43)	0x83A9	0x0102 (**PtypBinary**)	See the paragraph marked *2 that follows this table.
PidLidAppointmentTimeZoneDefinitionRecur (section 2.2.1.41)	0x83AA	0x0102 (**PtypBinary**)	See the paragraph marked *3 that follows this table.

Property	Property ID	Property type	Value
PidLidAppointmentDuration (section 2.2.1.7)	0x81A9	0x0003 (**PtypInteger32**)	0x0000001E (30)
PidLidAppointmentAuxiliaryFlags (section 2.2.1.3)	0x82D2	0x0003 (**PtypInteger32**)	0x00000000 (0)
PidLidAppointmentSubType (section 2.2.1.9)	0x8120	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidAppointmentStateFlags (section 2.2.1.10)	0x81B3	0x0003 (**PtypInteger32**)	0x00000003 (3)
PidLidResponseStatus (section 2.2.1.11)	0x8122	0x0003 (**PtypInteger32**)	respNotResponded (0x00000005)
PidLidAppointmentNotAllowPropose (section 2.2.1.26	0x82D5	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidExceptionReplaceTime (section 2.2.10.2.5)	0x83AC	0x0040 (**PtypTime**)	0x01C88E9DDA16DC00 (2008/03/25 17:30:00.000)
PidLidFInvited (section 2.2.4.4)	0x81DA	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidFExceptionalAttendees (section 2.2.2.3)	0x82D7	0x000B (**PtypBoolean**)	0x00 (FALSE)
PidLidFExceptionalBody (section 2.2.10.2.6)	0x82D8	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidRecurrenceType (section 2.2.1.45)	0x81FE	0x0003 (**PtypInteger32**)	0x00000002 (2)
PidLidRecurrencePattern (section 2.2.1.46)	0x81FC	0x001F (**PtypString**)	Every Tuesday from 10:30 A.M. to 11:00 A.M.
PidLidTimeZoneDescription (section 2.2.1.40)	0x8213	0x001F (**PtypString**)	(GMT - 08:00) Pacific Time (U.S. & Canada)
PidLidClipStart (section 2.2.1.14)	0x81BA	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidClipEnd (section 2.2.1.15)	0x81B9	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidLidAllAttendeesString (section 2.2.1.16)	0x81A8	0x001F (**PtypString**)	desaylor
PidLidToAttendeesString (section 2.2.1.17)	0x82D9	0x001F	desaylor

Property	Property ID	Property type	Value
		(**PtypString**)	
PidLidAppointmentSequence (section 2.2.1.1)	0x81AF	0x0003 (**PtypInteger32**)	0x00000000 (0)
PidLidAppointmentSequenceTime (section 2.2.4.1)	0x82E7	0x0040 (**PtypTime**)	0x01C874276FF4F450 (2008/02/21 01:16:51.093)
PidLidChangeHighlight (section 2.2.6.2)	0x82EC	0x0003 (**PtypInteger32**)	0x00000083 (131)
PidLidReminderTime ([MS-OXORMDR] section 2.2.1.4)	0x8005	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidCommonStart ([MS-OXCMSG] section 2.2.1.18)	0x81BC	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidLidCommonEnd ([MS-OXCMSG] section 2.2.1.19)	0x81BB	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidLidAttendeeCriticalChange (section 2.2.5.2)	0x81B5	0x0040 (**PtypTime**)	0x01C8742891F14080 (2008/02/21 01:24:57.608)
PidLidWhere (section 2.2.5.3)	0x8219	0x001F (**PtypString**)	Your Office
PidLidGlobalObjectId	0x81E0	0x0102 (**PtypBinary**)	See the paragraph marked *4
PidLidCleanGlobalObjectId	0x81B8	0x0102 (**PtypBinary**)	See the paragraph marked *5
PidLidAppointmentMessageClass (section 2.2.6.6)	0x8311	0x001F (**PtypString**)	IPM.appointment
PidLidIsRecurring (section 2.2.1.13)	0x81E5	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidIsException (section 2.2.1.35)	0x81E4	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidLidTimeZone (section 2.2.5.6)	0x8212	0x0003 (**PtypInteger32**)	0x0000000D (13)
PidLidCalendarType (section 2.2.6.11)	0x81B7	0x0003 (**PtypInteger32**)	0x00000001 (CAL_GREGORIAN)
PidLidOwnerCriticalChange (section 2.2.1.34)	0x8128	0x0040 (**PtypTime**)	0x01C874289289D700 (2008/02/21

Property	Property ID	Property type	Value
			01:24:58.608)
PidLidMeetingType (section 2.2.6.5)	0x8314	0x0003 (**PtypInteger32**)	0x00010000 (65536)
PidLidOldLocation (section 2.2.6.7)	0x8316	0x001F (**PtypString**)	(null)
PidLidOldWhenStartWhole (section 2.2.6.8)	0x83CC	0x0040 (**PtypTime**)	0x01C88E9DDA16DC00 (2008/03/25 17:30:00.000)
PidLidOldWhenEndWhole (section 2.2.6.9)	0x83CD	0x0040 (**PtypTime**)	0x01C88EA20AF91000 (2008/03/25 18:00:00.000)
PidTagResponseRequested ([MS-OXOMSG] section 2.2.1.46)	0x0063	0x000B (**PtypBoolean**)	0x01 (TRUE)
PidTagStartDate (section 2.2.1.30)	0x0060	0x0040 (**PtypTime**)	0x01C88F6704809C00 (2008/03/26 17:30:00.000)
PidTagEndDate (section 2.2.1.31)	0x0061	0x0040 (**PtypTime**)	0x01C88F6B3562D000 (2008/03/26 18:00:00.000)
PidTagOwnerAppointmentId	0x0062	0x0003 (**PtypInteger32**)	0x4D9427D8
Best body properties	The client can look for and remove the downlevel text, as specified in section 2.2.6.12, before copying the text stream (1) onto the new Exception Embedded Message object.		

*1 For a description of the **PidLidTimeZoneStruct** property BLOB see section 4.1.5. The value for this Meeting Request object is as follows.

```
cb: 48

lpb:
E00100000000000C4FFFFFF000000000B00000001000200000000000000000000000300000002000200000000000
000
```

*2 The dates in the value of the **PidLidAppointmentTimeZoneDefinitionRecur** property for this appointment are both set in the same time zone. For a description of the time zone definition property BLOB, see section 4.1.4. The only difference between this BLOB and that in the **PidLidAppointmentTimeZoneDefinitionStartDisplay** and **PidLidAppointmentTimeZoneDefinitionEndDisplay** properties is that the **TZRULE_FLAG_RECUR_CURRENT_TZREG** flag is set in this BLOB. The value for this Meeting Request object is as follows.

```
cb: 184

lpb:
020130000200150050006100630069006600690063002000530074006100 6E00640061007200640020005400690006
D006500020002013E000000D607000000000000000000000000000000E001000000000000C4FFFFFF00000A00000005
0002000000000000000000004000000010002000000000000000002013E000300D707000000000000000000000000000000
0E001000000000000C4FFFFFF00000B000000010002000000000000000000030000000200020000000000000000
```

*3 The start and end dates for this appointment are both set in the same time zone. The value for this Meeting Request object is as follows.

```
cb: 184
lpb:
020130000200150050006100630069006600690063002000530074006100 6E00640061007200640020005400690006
D006500020002013E000000D607000000000000000000000000000000E001000000000000C4FFFFFF00000A00000005
0002000000000000000000004000000010002000000000000000002013E000200D707000000000000000000000000000000
0E001000000000000C4FFFFFF00000B000000010002000000000000000000030000000200020000000000000000
```

*4 The value of the **PidLidGlobalObjectId** property for this Meeting Request object is as follows. For a description of the Global Object ID BLOB see section 4.1.2.

```
cb: 56
lpb:
040000008200E00074C5B7101A82E00807D803195025D461E473C801000000000000000000100000002A5844B3A444F
74A9C246C60886F116B
```

*5 The value of the **PidLidCleanGlobalObjectId** property for this Meeting Request object is as follows. This value is identical to the value of the **PidLidGlobalObjectId** property except that the **Year**, **Month**, and **Day** fields are filled with zeros.

```
cb: 56
lpb:
040000008200E00074C5B7101A82E00800000000005025D461E473C801000000000000000000100000002A5844B3A444F
74A9C246C60886F116B
```

The client uses the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5) to set the recipients (2) on the Exception Embedded Message object. The recipients (2) are obtained from the **RecipientRow** structures, as described in [MS-OXCDATA] section 2.8.3, of the Meeting Request object, as well as the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25). In addition, if the organizer (in this case, Mr. John) is not in the list of recipients (1), his information is obtained from the **PidTagSentRepresentingSearchKey** ([MS-OXOMSG] section 2.2.1.58) and **PidTagSentRepresentingName** ([MS-OXOMSG] section 2.2.1.57) properties and added as a **RecipientRow** structure. The Exception Embedded Message object is saved by using the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3), to which the server returns a success code.

After saving the Exception Embedded Message object, the client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to set the properties listed in the following table on the Exception Attachment object (not the Exception Embedded Message object).

Property	Property ID	Property type	Value
PidTagExceptionStartTime (section 2.2.10.1.4)	0x7FFB	0x0040 (**PtypTime**)	0x01C88F2C5821C400 (2008/03/26 10:30:00.000)
PidTagExceptionEndTime (section 2.2.10.1.5)	0x7FFC	0x0040 (**PtypTime**)	0x01C88F308903F800 (2008/03/26 11:00:00.000)
PidTagExceptionReplaceTime	0x7FF9	0x0040 (**PtypTime**)	0x01C88E9DDA16DC00 (2008/03/25 17:30:00.000)
PidTagAttachmentFlags	0x7FFD	0x0003 (**PtypInteger32**)	0x00000002 (afException)
PidTagAttachmentHidden ([MS-OXCMSG] section 2.2.2.24)	0x7FFE	0x000B (**PtypBoolean**)	0x01 (TRUE)

The client uses the **RopSaveChangesAttachment** ROP ([MS-OXCROPS] section 2.2.6.15) to save the changes to the Attachment object.

Now that the exception has been created, the client makes the following change to the Meeting object (the object that represents the recurring series) on Mr. Saylor's calendar by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6).

Property	Property ID	Property type	Value
PidLidAppointmentRecur (section 2.2.1.44)	0x81AD	0x0102 (**PtypBinary**)	See the paragraph marked *1 that follows this table.

*1 The value of the **PidLidAppointmentRecur** property will include necessary information about this new exception. The new value for the attendee's Meeting object is as follows.

```
cb: 114

lpb:
043004300B2001000000C021000001000000000000000400000023200000A000000000000000010000000A025C40C0
1000000402BC40C2088C30CDF80E95A0630000009300000760200009402000001 00B62DC40CD42DC40C1628C40C00
020000000004000000000000000000000000000000
```

The client sets the following property on the Meeting Request object by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6), followed by the **RopSaveChangesMessage** ROP.

Property	Property ID	Property type	Value
PidTagProcessed (section 2.2.5.7)	0x7D01	0x000B (**PtypBoolean**)	0x01 (TRUE)

After processing the Meeting Request object, the client is now ready to act on the response. To start, the changes listed in the following table are made to the Exception Embedded Message object by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6).

Property	Property ID	Property type	Value
PidLidBusyStatus	0x81B6	0x0003 (**PtypInteger32**)	0x00000002 (2)
PidLidResponseStatus	0x8122	0x0003 (**PtypInteger32**)	respAccepted (0x00000003)
PidLidAppointmentReplyTime (section 2.2.4.3)	0x8139	0x0040 (**PtypTime**)	0x01C87428FEA81000 (2008/02/21 01:28:00.000)
PidLidAppointmentReplyName (section 2.2.4.5)	0x81AE	0x001F (**PtypString**)	desaylor

The client again saves the Exception Embedded Message object by using the **RopSaveChangesMessage** ROP and another **RopSaveChangesMessage** ROP to save the Meeting object that represents the recurring series, to which the server returns success codes.

The last thing the client needs to do is send a response to the organizer. The client creates a new Meeting Response object in the Outbox folder by using the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2), to which the server returns a success code and a handle. The client sets the following properties on this new Message object by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) using the values from the Exception Embedded Message object:

- **PidTagNormalizedSubject**
- **PidLidBusyStatus**
- **PidLidAppointmentColor** ([MS-OXPROPS] section 2.9)
- **PidLidLocation**
- **PidLidRecurring** (section 2.2.1.12)
- **PidLidAppointmentStartWhole**
- **PidLidAppointmentEndWhole**
- **PidLidAppointmentTimeZoneDefinitionStartDisplay**
- **PidLidAppointmentTimeZoneDefinitionEndDisplay**
- **PidLidAppointmentDuration**
- **PidLidAppointmentAuxiliaryFlags**
- **PidLidAppointmentSubType**
- **PidLidAppointmentRecur**
- **PidLidRecurrenceType**
- **PidLidRecurrencePattern**
- **PidLidTimeZoneStruct**
- **PidLidAppointmentTimeZoneDefinitionRecur**

- **PidLidTimeZoneDescription**
- **PidLidClipStart**
- **PidLidClipEnd**
- **PidLidAppointmentSequence**
- **PidLidCommonStart**
- **PidLidCommonEnd**
- **PidLidWhere**
- **PidLidGlobalObjectId**
- **PidLidCleanGlobalObjectId**
- **PidLidAppointmentMessageClass**
- **PidLidIsRecurring** (section 2.2.1.13)
- **PidLidIsException**
- **PidLidTimeZone**
- **PidLidCalendarType**
- **PidLidOwnerCriticalChange**
- **PidTagStartDate** (section 2.2.1.30)
- **PidTagEndDate** (section 2.2.1.31)
- **PidTagOwnerAppointmentId** (section 2.2.1.29)

In addition to these, the client uses the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to put the property values listed in the following table onto the Meeting Response object.

Property	Property ID	Property type	Value
PidTagMessageClass	0x001A	0x001F (**PtypString**)	IPM.Schedule.Meeting.Resp.Pos
PidTagSubjectPrefix	0x003D	0x001F (**PtypString**)	Accepted:
PidLidSideEffects ([MS-OXCMSG] section 2.2.1.16)	0x8002	0x0003 (**PtypInteger32**)	0x00001C61 (7265)
PidLidAttendeeCriticalChange	0x81B5	0x0040 (**PtypTime**)	0x01C874292153F290 (2008/02/21 01:28:58.169)
PidLidIsSilent (section 2.2.7.7)	0x81E6	0x000B (**PtypBoolean**)	0x01 (TRUE)

The client adds the organizer by using the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5) and then sends the object via the **RopSubmitMessage** ROP ([MS-OXCROPS] section

2.2.7.1). After the server returns a success code from submission, the client releases all objects, including the Embedded Message objects, Attachment objects, attachments table, Meeting objects, and Meeting Request objects, by using the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3) for each.

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to the Appointment and Meeting Object Protocol. General security considerations that pertain to the underlying **remote procedure call (RPC)**-based transport apply, as described in [MS-OXCROPS].

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

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

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

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

<1> Section 2.2: The following additional properties can be set on items described by the Appointment and Meeting Object Protocol for backward compatibility with Office Outlook 2003. These properties are not used by Office Outlook 2007, Outlook 2010, and Outlook 2013: **PidLidRequiredAttendees** ([MS-OXPROPS] section 2.229), **PidLidOptionalAttendees** ([MS-OXPROPS] section 2.194), **PidLidResourceAttendees** ([MS-OXPROPS] section 2.230), **PidLidDelegateMail** ([MS-OXPROPS] section 2.92), **PidLidTimeZone** (section 2.2.5.6), **PidLidStartRecurrenceDate** ([MS-OXPROPS] section 2.303), **PidLidStartRecurrenceTime** ([MS-OXPROPS] section 2.304), **PidLidEndRecurrenceDate** ([MS-OXPROPS] section 2.115), **PidLidEndRecurrenceTime** ([MS-OXPROPS] section 2.116), **PidLidDayInterval** ([MS-OXPROPS] section 2.90), **PidLidWeekInterval** ([MS-OXPROPS] section 2.352), **PidLidMonthInterval** ([MS-OXPROPS] section 2.172), **PidLidYearInterval** ([MS-OXPROPS] section 2.362), **PidLidMonthOfYearMask** ([MS-OXPROPS] section 2.174), and **PidLidRecurrenceType** (section 2.2.1.45).

<2> Section 2.2.1.2: Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not support the olWorkingElsewhere value. Office Outlook 2003, Office Outlook 2007, and Outlook 2010 interpret the olWorkingElsewhere value as olFree.

<3> Section 2.2.1.5: For compatibility with Office Outlook 2003, the **PidTagStartDate** property (section 2.2.1.30) needs to be set, and when set, it has to be equal to the value of the **PidLidAppointmentStartWhole** property ([MS-OXPROPS] section 2.29).

<4> Section 2.2.1.6: For compatibility with Office Outlook 2003, the **PidTagEndDate** property (section 2.2.1.31) needs to be set, and when set, it has to be equal to the value of the **PidLidAppointmentEndWhole** property ([MS-OXPROPS] section 2.14).

<5> Section 2.2.1.7: Exchange 2003 ignores the **PidLidAppointmentDuration** property ([MS-OXPROPS] section 2.11) and always computes the length of the event from the difference between the value of the **PidLidAppointmentEndWhole** property (section 2.2.1.6) and the value of the **PidLidAppointmentStartWhole** property (section 2.2.1.5).

<6> Section 2.2.1.8: If no value is specified in the **PidNameKeywords** property ([MS-OXCMSG] section 2.2.1.17), Office Outlook 2003 displays the Calendar object in the color specified by the **PidLidAppointmentColor** property (section 2.2.1.50).

<7> Section 2.2.1.13: Exchange 2003 does not read or write the **PidLidIsRecurring** property ([MS-OXPROPS] section 2.156).

<8> Section 2.2.1.25: Office Outlook 2003 instead uses the following properties to track unsendable attendees:

PidLidNonSendableTo (section 2.2.1.19)

PidLidNonSendableCc (section 2.2.1.20)

PidLidNonSendableBcc (section 2.2.1.21)

PidLidNonSendToTrackStatus (section 2.2.1.22)

PidLidNonSendCcTrackStatus (section 2.2.1.23)

PidLidNonSendBccTrackStatus (section 2.2.1.24)

<9> Section 2.2.1.29: When a Meeting object is created, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 set the value of the **PidTagOwnerAppointmentId** property ([MS-OXPROPS] section 2.845) to the number of minutes between the start time and midnight, January 1, 1601. When trying to find a Meeting object, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 sort the table according to the **PidTagOwnerAppointmentId** property, thus allowing increased performance in the search.

<10> Section 2.2.1.41: Office Outlook 2003 does not support the **PidLidAppointmentTimeZoneDefinitionRecur** property ([MS-OXPROPS] section 2.33).

<11> Section 2.2.1.44: Office Outlook 2003 and Office Outlook 2007 use the start time when calculating whether exceptions overlap.

<12> Section 2.2.1.44.1: This value can be read by Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 but is not used.

<13> Section 2.2.1.44.1: This value can be read by Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 but is not used.

<14> Section 2.2.1.44.1: Office Outlook 2003 and Office Outlook 2007 always write a default value of 0x0000000A for the **OccurrenceCount** field when the recurrence pattern has no end date.

<15> Section 2.2.1.44.3: The **Reserved** field is not present in Office Outlook 2003 or Office Outlook 2007.

<16> Section 2.2.1.44.5: Office Outlook 2003 uses version 0x00003008.

<17> Section 2.2.1.47: Office Outlook 2003 does not set the **PidLidLinkedTaskItems** property ([MS-OXPROPS] section 2.158).

<18> Section 2.2.1.51: Office Outlook 2003 reads and writes the properties in this section.

<19> Section 2.2.1.51.5: The **PidLidAllowExternalCheck** property ([MS-OXPROPS] section 2.6) is used for interoperability with NetMeeting.

<20> Section 2.2.1.51.5: Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, Exchange 2003, Exchange 2007, Exchange 2010, and Exchange 2013 do not require that the **PidLidAllowExternalCheck** property ([MS-OXPROPS] section 2.6) be set on a Calendar object.

<21> Section 2.2.2.2: Exchange 2003 includes only the **seCoerceToInbox** and **seOpenForCtxMenu** flags. Without all the flags, the client user interface will not always behave as expected when a Calendar object is moved, deleted, or copied or when a context menu is displayed for the object.

<22> Section 2.2.4.10.1: Office Outlook 2003 and Office Outlook 2007 use these reserved flags for internal information that does not affect the Appointment and Meeting Object Protocol. A server or client does not need to read these flags but needs to leave the values unchanged if they are set.

<23> Section 2.2.4.10.1: Office Outlook 2003 and Office Outlook 2007 use these reserved flags for internal information that does not affect the Appointment and Meeting Object Protocol. A server or client does not need to read these flags but needs to leave the values unchanged if they are set.

<24> Section 2.2.4.10.1: Office Outlook 2003 and Office Outlook 2007 use these reserved flags for internal information that does not affect the Appointment and Meeting Object Protocol. A server or client does not need to read these flags but needs to leave the values unchanged if they are set.

<25> Section 2.2.5.2: If the value of the **PidLidAttendeeCriticalChange** property is not specified, Exchange 2003 will use the last modified time as this value.

<26> Section 2.2.5.3: Exchange 2003 does not read or write the **PidLidWhere** property.

<27> Section 2.2.6.2: Office Outlook 2003 does not read or write the **PidLidChangeHighlight** property ([MS-OXPROPS] section 2.51).

<28> Section 2.2.6.3: Office Outlook 2003 reads and writes the **PidLidForwardInstance** property ([MS-OXPROPS] section 2.138).

<29> Section 2.2.6.7: Office Outlook 2003 does not read or write the **PidLidOldLocation** property ([MS-OXPROPS] section 2.189).

<30> Section 2.2.6.8: Office Outlook 2003 does not read or write the **PidLidOldWhenStartWhole** property ([MS-OXPROPS] section 2.192).

<31> Section 2.2.6.9: Office Outlook 2003 does not support the **PidLidOldWhenEndWhole** property ([MS-OXPROPS] section 2.191).

<32> Section 2.2.6.12: Exchange 2003, Exchange 2007, Exchange 2010, and Exchange 2013 do not add downlevel text.

<33> Section 2.2.10.1.6: Office Outlook 2003 and Office Outlook 2007 do not write the value for the **PidTagExceptionReplaceTime** property ([MS-OXPROPS] section 2.676).

<34> Section 3.1.4.3: Office Outlook 2003 does not set the **PidLidAppointmentStateFlags** property (section 2.2.1.10).

<35> Section 3.1.4.3: Office Outlook 2003 and Office Outlook 2007 sometimes do not copy the recipient list. If the **RecipientRow** structures ([MS-OXCDATA] section 2.8.3) from a Meeting object are not copied, the resulting snapshot will not show who was invited to the meeting at the time the copy was made.

<36> Section 3.1.4.3: Office Outlook 2003 does not set the **PidLidResponseStatus** property (section 2.2.1.11).

<37> Section 3.1.4.3: Office Outlook 2003 does not set the **PidTagSubjectPrefix** property ([MS-OXCMSG] section 2.2.1.9).

<38> Section 3.1.4.5: Office Outlook 2003 prompts the user with an option to delete without sending a cancellation.

<39> Section 3.1.4.7.1.1: Outlook 2013 does not support direct booking. By default, Outlook 2010 does not use direct booking, but it can be set up to use it.

<40> Section 3.1.4.7.1.1: Office Outlook 2003 and Office Outlook 2007 attempt direct booking only for Resource objects.

<41> Section 3.1.4.7.2: Office Outlook 2003 does not support the calendar options dictionary.

<42> Section 3.1.4.7.2.1: Office Outlook 2003 does not skip automatic creation of the Meeting object based on the values of these properties.

<43> Section 3.1.4.7.2.2: If the **PidLidReminderSet** property ([MS-OXORMDR] section 2.2.1.1) is set to FALSE, Outlook 2010 and Outlook 2013 change the value of the **PidLidReminderSet** property to TRUE, set the **PidLidReminderDelta** property ([MS-OXORMDR] section 2.2.1.3) to its default value (as defined by the client), and recalculate the **PidLidReminderSignalTime** property ([MS-OXORMDR] section 2.2.1.2) regardless of the value of the **PidLidAppointmentSubType** property (section 2.2.1.9).

<44> Section 3.1.4.7.2.2: Office Outlook 2003 does not set the **PidLidAppointmentReplyName** property (section 2.2.4.5).

<45> Section 3.1.4.7.2.2: Office Outlook 2003 does not copy the **RecipientRow** structures, as described in [MS-OXCDATA] section 2.8.3, of the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25) of the Meeting Request object to the **RecipientRow** structures of the Meeting object.

<46> Section 3.1.4.7.2.2: Office Outlook 2003 and Office Outlook 2007 do not copy the busy status for the exception.

<47> Section 3.1.4.7.2.2: Office Outlook 2003 and Office Outlook 2007 do not set the **PidLidServerProcessed** property (section 2.2.5.4).

<48> Section 3.1.4.7.2.3: Office Outlook 2003 automatically sends Meeting Response objects if the **PidLidServerProcessed** property (section 2.2.5.4) is set.

<49> Section 3.1.4.7.2.3: Office Outlook 2003 and Office Outlook 2007 do not set the **PidLidServerProcessingActions** property (section 2.2.5.5).

<50> Section 3.1.4.7.3: Office Outlook 2003 does not set the **PidLidOldWhenStartWhole** (section 2.2.6.8) and **PidLidOldWhenEndWhole** (section 2.2.6.9) properties.

<51> Section 3.1.4.7.3.1: Office Outlook 2003 sets the value of the **PidLidMeetingType** property (section 2.2.6.5) to **mtgFull** regardless of the change made.

<52> Section 3.1.4.7.3.2: Office Outlook 2003 always clears responses whenever any update is sent out.

<53> Section 3.1.4.7.3.4: Office Outlook 2003 does not treat an attendee that has been marked sendable as a new attendee.

<54> Section 3.1.4.7.3.4: Office Outlook 2003 does not set the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25) as specified.

<55> Section 3.1.4.7.3.5: Office Outlook 2003 does not send a Meeting Cancellation object for the exception to each attendee included in the recurring series that is not included in the exception.

<56> Section 3.1.4.7.3.5: Office Outlook 2003 does not send out cancelations to exceptions when the recurrence pattern has changed.

<57> Section 3.1.4.7.4: Office Outlook 2003 does not support the calendar options dictionary.

<58> Section 3.1.4.7.4.1: Office Outlook 2003 does not skip automatic updating of the Meeting object based on the values of the **PidLidServerProcessed** (section 2.2.5.4) and the **PidLidServerProcessingActions** (section 2.2.5.5) properties.

<59> Section 3.1.4.7.4.2: Office Outlook 2003 re-creates the exception regardless of whether these properties are set.

<60> Section 3.1.4.7.4.2: Office Outlook 2003 does not copy the **PidLidLocation** property (section 2.2.1.4), the **PidLidAppointmentStartWhole** property (section 2.2.1.5), or the **PidLidAppointmentEndWhole** property (section 2.2.1.6) onto the Meeting Update object.

<61> Section 3.1.4.7.4.2: Office Outlook 2003 performs these actions regardless of whether the **cpsCopiedOldProperties** bit is set on the **PidLidServerProcessingActions** property (section 2.2.5.5).

<62> Section 3.1.4.7.4.2: Office Outlook 2003 overwrites private values of the **PidTagSensitivity** property ([MS-OXCMSG] section 2.2.1.13).

<63> Section 3.1.4.7.4.2: Office Outlook 2003 resets the value of the **PidLidResponseStatus** property (section 2.2.1.11) to "respNotResponded" (0x00000005) regardless of whether the Meeting Update object contains a significant change.

<64> Section 3.1.4.7.5: Office Outlook 2003 and Office Outlook 2007 set the value of the **PidLidMeetingType** property (section 2.2.6.5) to 0x00000000.

<65> Section 3.1.4.7.5: Office Outlook 2003 does not copy the **RecipientRow** structures ([MS-OXCDATA] section 2.8.3) to the **PidLidAppointmentUnsendableRecipients** property (section 2.2.1.25) of the new object.

<66> Section 3.1.4.7.5.1: Office Outlook 2003 does not forward exceptions to a recurring series.

<67> Section 3.1.4.8.4: Office Outlook 2003 allows an organizer to send a response to their own meeting, but only if the **asfReceived** bit is not set on the **PidLidAppointmentStateFlags** property (section 2.2.1.10).

<68> Section 3.1.4.8.4: Office Outlook 2003 and Office Outlook 2007 also write the following properties, which are not used by Office Outlook 2003 or Office Outlook 2007: **PidLidInternetAccountName** ([MS-OXOMSG] section 2.2.1.62) and **PidLidInternetAccountStamp** ([MS-OXOMSG] section 2.2.1.63).

<69> Section 3.1.4.8.4: Office Outlook 2003 and Office Outlook 2007 also write the following properties when the Meeting Response object represents a recurring series. These properties are not used by Office Outlook 2003 or Office Outlook 2007: **PidLidRequiredAttendees** ([MS-OXPROPS] section 2.229), **PidLidOptionalAttendees** ([MS-OXPROPS] section 2.194), **PidLidResourceAttendees** ([MS-OXPROPS] section 2.230), **PidLidDelegateMail** ([MS-OXPROPS] section 2.92), **PidLidTimeZone** (section 2.2.5.6), **PidLidStartRecurrenceDate** ([MS-OXPROPS]

section 2.303), **PidLidStartRecurrenceTime** ([MS-OXPROPS] section 2.304), **PidLidEndRecurrenceDate** ([MS-OXPROPS] section 2.115), **PidLidEndRecurrenceTime** ([MS-OXPROPS] section 2.116), **PidLidDayInterval** ([MS-OXPROPS] section 2.90), **PidLidWeekInterval** ([MS-OXPROPS] section 2.352), **PidLidMonthInterval** ([MS-OXPROPS] section 2.172), **PidLidYearInterval** ([MS-OXPROPS] section 2.362), **PidLidMonthOfYearMask** ([MS-OXPROPS] section 2.174), **PidLidRecurrenceType** (section 2.2.1.45).

<70> Section 3.1.4.8.5: Office Outlook 2003 does not support the calendar options dictionary.

<71> Section 3.1.4.8.5.1: Office Outlook 2003 will process the response regardless of the value of the **PidLidServerProcessingActions** property (section 2.2.5.5).

<72> Section 3.1.4.8.5.1: When processing a Meeting Response object that represents an exception to a recurring appointment, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 ignore the **cpsUpdatedCalItem** bit and still record the response.

<73> Section 3.1.4.8.5.2: Office Outlook 2007 does not set the **PidLidPromptSendUpdate** (section 2.2.7.8) property.

<74> Section 3.1.4.8.5.2: Office Outlook 2003 and Office Outlook 2007 do not verify that the attendee exists on an out-of-date Meeting Response object.

<75> Section 3.1.4.8.5.2: Office Outlook 2003 and Office Outlook 2007 compare the two time values rounded down to the nearest minute so that if an attendee responds twice within the same minute, both responses will be seen as having been sent at the same time.

<76> Section 3.1.4.8.5.2: Office Outlook 2003 and Office Outlook 2007 round the time value from the **PidLidAttendeeCriticalChange** property (section 2.2.5.2) down to the nearest minute before setting the value in the **PidTagRecipientTrackStatusTime** property (section 2.2.4.10.3).

<77> Section 3.1.4.8.5.2: Office Outlook 2003 and Office Outlook 2007 allow the user to decide whether to delete empty responses.

<78> Section 3.1.4.9.1.1: Office Outlook 2003 does not send cancellations to unsendable attendees.

<79> Section 3.1.4.9.1.2: Office Outlook 2007 does not send out cancellations for deleted exceptions when sending out a cancellation for a recurring series.

<80> Section 3.1.4.9.1.2: Office Outlook 2007 sends Meeting Cancellation objects to exceptions when sending a Meeting Cancellation object to a recurring series to a Partial Attendee List, but Office Outlook 2003 does not.

<81> Section 3.1.4.9.2: Office Outlook 2003 does not support the calendar options dictionary.

<82> Section 3.1.4.9.2.1: Office Outlook 2003 does not skip automatic updating of the Meeting object based on the **PidLidServerProcessed** (section 2.2.5.4) and **PidLidServerProcessingActions** (section 2.2.5.5) properties.

<83> Section 3.1.4.9.2.1: Office Outlook 2003 and Office Outlook 2007 re-create the Exception object.

<84> Section 3.1.4.9.2.1: Office Outlook 2003 and Office Outlook 2007 re-create the Meeting object.

<85> Section 3.1.4.9.2.2: Office Outlook 2003 and Office Outlook 2007 both set the **PidTagProcessed** property (section 2.2.5.7).

<86> Section 3.1.4.10.1: Exchange 2003, Exchange 2007, Exchange 2010, the initial release version of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release version of Outlook 2013 do not support the **X-ServerApplication** header and the **Connect** request type. The **X-ServerApplication** header and the **Connect** request type were introduced in Exchange 2013 SP1 and Outlook 2013 SP1.

<87> Section 3.1.4.10.1: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 also write the following properties, which are not read by Office Outlook 2003, Office Outlook 2007, Outlook 2010, or Outlook 2013: **PidLidInternetAccountName** ([MS-OXOMSG] section 2.2.1.62) and **PidLidInternetAccountStamp** ([MS-OXOMSG] section 2.2.1.63).

<88> Section 3.1.4.10.1: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 also write the following properties when the Meeting Response object represents a recurring series. These are not read by Office Outlook 2003, Office Outlook 2007, Outlook 2010, or Outlook 2013: **PidLidRequiredAttendees** ([MS-OXPROPS] section 2.229), **PidLidOptionalAttendees** ([MS-OXPROPS] section 2.194), **PidLidResourceAttendees** ([MS-OXPROPS] section 2.230), **PidLidDelegateMail** ([MS-OXPROPS] section 2.92), **PidLidTimeZone** (section 2.2.5.6), **PidLidStartRecurrenceDate** ([MS-OXPROPS] section 2.303), **PidLidStartRecurrenceTime** ([MS-OXPROPS] section 2.304), **PidLidEndRecurrenceDate** ([MS-OXPROPS] section 2.115), **PidLidEndRecurrenceTime** ([MS-OXPROPS] section 2.116), **PidLidDayInterval** ([MS-OXPROPS] section 2.90), **PidLidWeekInterval** ([MS-OXPROPS] section 2.352), **PidLidMonthInterval** ([MS-OXPROPS] section 2.172), **PidLidYearInterval** ([MS-OXPROPS] section 2.362), **PidLidMonthOfYearMask** ([MS-OXPROPS] section 2.174), and **PidLidRecurrenceType** (section 2.2.1.45).

<89> Section 3.1.4.10.2: Office Outlook 2003 does not support the calendar options dictionary.

<90> Section 3.1.5.4: When sending a Meeting Update object for an exception to a recurring series, Office Outlook 2003 does not increment the sequence number (2) for the exception.

<91> Section 3.1.5.5.1: Office Outlook 2003 and Office Outlook 2007 do not interpret data in this manner.

<92> Section 3.1.5.5.2: Office Outlook 2003 and Office Outlook 2007 do not interpret data in this manner.

<93> Section 3.1.5.6: Office Outlook 2003 does not support the **PidTagScheduleInfoDelegatorWantsInfo** property ([MS-OXODLGT] section 2.2.2.2.2).

<94> Section 3.1.5.6: Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 2013 do not set the **cpsDelegatorWantsCopy** bit of the **PidLidServerProcessingActions** property (section 2.2.5.5).

7 Change Tracking

This section identifies changes that were made to the [MS-OXOCAL] protocol document between the July 2014 and October 2014 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

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

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

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

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

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

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

- Obsolete document removed.

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.1.44.1.4 PatternTypeSpecific Week	Changed the name of the first unused field from "unused" to "X".	N	Content updated.
2.2.1.44.1.6 PatternTypeSpecific MonthNth	Changed the name of the first unused field from "unused" to "X".	N	Content updated.
2.2.1.44.2 ExceptionInfo Structure	Updated descriptions of the ARO_APPTCOLOR flag and the AppointmentColor field. Removed the ReservedBlock1Size and ReservedBlock1 fields from the byte diagram.	N	Content updated.
3.1.1 Abstract Data Model	Deleted Calendar from the list of ADM types.	N	Content updated.

8 Index

N

O

P

R

S

T