

[MS-OXOPFFB]:

Public Folder-Based Free/Busy Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability.
4/25/2008	0.2	Minor	Revised and updated property names and other technical content.
6/27/2008	1.0	Major	Initial Release.
8/6/2008	1.01	Minor	Updated references to reflect date of initial release.
9/3/2008	1.02	Minor	Updated references.
12/3/2008	1.03	Minor	Revised and edited technical content.
4/10/2009	2.0	Major	Updated technical content and applicable product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	3.1.0	Minor	Updated the technical content.
2/10/2010	4.0.0	Major	Updated and revised the technical content.
5/5/2010	5.0.0	Major	Updated and revised the technical content.
8/4/2010	5.1	Minor	Clarified the meaning of the technical content.
11/3/2010	5.2	Minor	Clarified the meaning of the technical content.
3/18/2011	5.2	None	No changes to the meaning, language, and formatting of the technical content.
8/5/2011	5.2	None	No changes to the meaning, language, or formatting of the technical content.
10/7/2011	5.3	Minor	Clarified the meaning of the technical content.
1/20/2012	6.0	Major	Significantly changed the technical content.
4/27/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	6.1	Minor	Clarified the meaning of the technical content.
10/8/2012	6.2	Minor	Clarified the meaning of the technical content.
2/11/2013	6.2	None	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	6.3	Minor	Clarified the meaning of the technical content.
11/18/2013	6.3	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	6.3	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	6.3	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	6.3	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
10/30/2014	6.4	Minor	Clarified the meaning of the technical content.
5/26/2015	6.4	None	No changes to the meaning, language, or formatting of the technical content.
6/3/2016	6.4	None	No changes to the meaning, language, or formatting of the technical content.
6/13/2016	6.4	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	6.4	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	8
1.3	Overview	8
1.4	Relationship to Other Protocols	8
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	9
1.7	Versioning and Capability Negotiation	9
1.8	Vendor-Extensible Fields	9
1.9	Standards Assignments.....	9
2	Messages.....	10
2.1	Transport	10
2.2	Message Syntax	10
2.2.1	Free/Busy Message	10
2.2.1.1	Common Properties.....	10
2.2.1.1.1	PidTagNormalizedSubject Property	10
2.2.1.1.2	PidTagMessageClass Property	10
2.2.1.2	Free/Busy Properties	10
2.2.1.2.1	PidTagScheduleInfoMonthsTentative Property	10
2.2.1.2.2	PidTagScheduleInfoFreeBusyTentative Property	11
2.2.1.2.3	PidTagScheduleInfoMonthsBusy Property.....	11
2.2.1.2.4	PidTagScheduleInfoFreeBusyBusy Property.....	11
2.2.1.2.5	PidTagScheduleInfoMonthsAway Property	11
2.2.1.2.6	PidTagScheduleInfoFreeBusyAway Property	11
2.2.1.2.7	PidTagScheduleInfoMonthsMerged Property	12
2.2.1.2.8	PidTagScheduleInfoFreeBusyMerged Property	12
2.2.1.2.9	PidTagFreeBusyPublishStart Property	12
2.2.1.2.10	PidTagFreeBusyPublishEnd Property.....	12
2.2.1.2.11	PidTagFreeBusyRangeTimestamp Property.....	12
2.2.1.2.12	PidTagFreeBusyMessageEmailAddress Property	12
2.2.1.3	Delegate Information Properties	13
2.2.1.4	Deprecated Properties	13
2.2.1.4.1	PidTagGatewayNeedsToRefresh Property	13
2.2.1.4.2	PidTagScheduleInfoResourceType Property	13
2.2.1.4.3	PidTagScheduleInfoFreeBusy Property.....	13
2.2.2	Public Folder Free/Busy Related Properties.....	13
2.2.2.1	PidTagFreeBusyEntryIds Property.....	13
2.2.2.2	PidTagSchedulePlusFreeBusyEntryId Property	14
3	Protocol Details	15
3.1	Client Details	15
3.1.1	Abstract Data Model	15
3.1.2	Timers	15
3.1.3	Initialization	15
3.1.4	Higher-Layer Triggered Events	15
3.1.4.1	Publishing Free/Busy Data.....	15
3.1.4.1.1	Determining the Data Set for Publishing	16
3.1.4.1.2	Finding the Free/Busy Message	16
3.1.4.1.2.1	Determining the Email Address.....	16
3.1.4.1.2.2	Determining the Name of the Subfolder	17
3.1.4.1.2.3	Determining the Subject of the Message	17
3.1.4.1.2.4	Finding the Message.....	17

3.1.5	Message Processing Events and Sequencing Rules	17
3.1.6	Timer Events.....	17
3.1.7	Other Local Events.....	17
3.2	Server Details.....	17
3.2.1	Abstract Data Model.....	18
3.2.2	Timers	18
3.2.3	Initialization	18
3.2.4	Higher-Layer Triggered Events	18
3.2.5	Message Processing Events and Sequencing Rules	18
3.2.6	Timer Events.....	18
3.2.7	Other Local Events.....	18
4	Protocol Examples	19
4.1	Updating the Free/Busy Message	19
4.2	Finding Free/Busy Messages by Using Email Addresses	20
4.3	Calculating the PidTagScheduleInfoMonthsBusy Property	27
4.4	Calculating the PidTagScheduleInfoFreeBusyBusy Property.....	28
4.4.1	Calculating the PidTagScheduleInfoFreeBusyBusy Property for Two Nonconsecutive Events.....	29
4.4.2	Calculating the PidTagScheduleInfoFreeBusyBusy Property for Two Consecutive Events.....	29
4.4.3	Calculating the PidTagScheduleInfoFreeBusyBusy Property for Events in Multiple Months	29
4.4.4	Calculating the PidTagScheduleInfoFreeBusyBusy Property for an Event Spread Across Multiple Months	30
4.5	Calculating the PidTagScheduleInfoFreeBusyMerged Property	30
5	Security	31
5.1	Security Considerations for Implementers	31
5.2	Index of Security Parameters	31
6	Appendix A: Product Behavior	32
7	Change Tracking.....	33
8	Index.....	34

1 Introduction

The Public Folder-Based Free/Busy Protocol is used to publish information that describes the availability of a user or resource. This protocol extends the Message and Attachment Object Protocol, which is described in [\[MS-OXCMSG\]](#).

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

1.1 Glossary

This document uses the following terms:

Address Book object: An entity in an address book that contains a set of attributes, each attribute with a set of associated values.

ambiguous name resolution (ANR): A search algorithm that permits a client to search multiple naming-related attributes on objects by way of a single clause of the form "(anr=value)" in a Lightweight Directory Access Protocol (LDAP) search filter. This permits a client to query for an object when the client possesses some identifying material related to the object but does not know which attribute of the object contains that identifying material.

Availability service: A web service that provides free/busy information.

Calendar folder: A Folder object that contains **Calendar objects**.

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

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

Delegate Information object: A **Message object** that contains properties specifying delegate access settings for resources in a delegator's mailbox.

distinguished name (DN): A name that uniquely identifies an object by using the relative distinguished name (RDN) for the object, and the names of container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in a tree.

email address: A string that identifies a user and enables the user to receive Internet messages.

EntryID: A sequence of bytes that is used to identify and access an object.

free/busy message: A message that is stored in a public folder and contains free/busy data.

free/busy status: A property of an appointment that indicates how an appointment on the calendar of an attendee or resource affects their availability.

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

Inbox folder: A **special folder** that is the default location for **Message objects** received by a user or resource.

little-endian: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

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

message store: A unit of containment for a single hierarchy of Folder objects, such as a mailbox or public folders.

non-interpersonal messaging subtree: A hierarchy of folders at the root level not commonly visible in a client. This includes server and client-created folders used principally for containing operational metadata.

Out of Office (OOF): One of the possible values for the **free/busy status** on an appointment. It indicates that the user will not be in the office during the appointment.

public folder: A Folder object that is stored in a location that is publicly available.

publishing range: The number of months of free/busy calendar data to be published, beginning at the start date of the publishing range, which is defined by the PidTagFreeBusyPublishStart property, and continuing for the number of months defined by the PidTagFreeBusyCountMonths property.

remote operation (ROP): An operation that is invoked against a server. Each ROP represents an action, such as delete, send, or query. A ROP is contained in a ROP buffer for transmission over the wire.

root folder: The folder at the top of a hierarchy of folders in a list.

ROP request: See ROP request buffer.

ROP response: See ROP response buffer.

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

tentative: One of the possible values for the **free/busy status** on an appointment. A tentative status indicates that the user is tentatively booked during the appointment.

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

1.2 References

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

1.2.1 Normative References

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

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

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

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

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

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

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

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

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

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

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

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

[RFC1279] Hardcastle-Kille, S., "X.500 and Domains", RFC 1279, November 1991, <http://www.rfc-editor.org/rfc/rfc1279.txt>

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

1.2.2 Informative References

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

[MS-OXWAVLS] Microsoft Corporation, "[Availability Web Service Protocol](#)".

1.3 Overview

The Public Folder–Based Free/Busy protocol allows information about the availability of a user or resource to be written to and read from a shared location so that it can be used to effectively and efficiently schedule meetings. This information, referred to as free/busy data, is specified in the properties of a **free/busy message** that is created in the **public folders message store**. The properties of this message are set to reflect the **free/busy status** and related details of the user's or resource's **Calendar object**. The free/busy status falls into four categories: free, busy, **tentative**, and **Out of Office (OOO)**.

Free/busy data can also be obtained by using the **Availability service**, as described in [\[MS-OXWAVLS\]](#), and by directly accessing the calendar of another user or a resource and reading the free/busy status property, as described in [\[MS-OXOCAL\]](#).

This protocol extends the Message and Attachment Object Protocol, described in [\[MS-OXCMSG\]](#), in that it defines new properties on a **Message object** and adds constraints to the existing properties of a Message object.

1.4 Relationship to Other Protocols

The Public Folder–Based Free/Busy Protocol has the same dependencies as the Message and Attachment Object Protocol, as described in [\[MS-OXCMSG\]](#).

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

1.5 Prerequisites/Preconditions

This protocol has the same prerequisites and preconditions as the Message and Attachment Object Protocol, as specified in [\[MS-OXCMSG\]](#). In addition, the Public Folder–Based Free/Busy Protocol assumes that the server is configured to support **public folders**.[<1>](#)

1.6 Applicability Statement

The Public Folder–Based Free/Busy Protocol is appropriate for use by higher layers of a server or client implementation that schedule meetings to avoid scheduling conflicts.

The following related functionality is best accomplished by using other protocols.

- Displaying complete calendar details. This is best accomplished by receiving permissions and directly browsing the calendar of the meeting attendee or resource. For information about how a client opens the **Calendar folder** of another user, see [\[MS-OXODLGT\]](#) section 3.2.4.1.
- Displaying free/busy data outside the **publishing range**. This is best accomplished by using the **Availability service**, as described in [\[MS-OXWAVLS\]](#).

It is recommended that this protocol be used only if the Availability service is not supported by the server.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

This protocol provides no vendor extensibility beyond that specified in [\[MS-OXCMSG\]](#).

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Public Folder–Based Free/Busy Protocol uses the same underlying transport as that used by the Message and Attachment Object Protocol, as specified in [\[MS-OXCMSG\]](#).

2.2 Message Syntax

2.2.1 Free/Busy Message

Free/busy data is represented as a set of properties that are set on a message in the **public folders message store**. The message is referred to as the **free/busy message**. The location of this message is specified in section [3.1.4.1.2](#). Unless otherwise specified, the free/busy message conforms to a **Message object**, as specified in [\[MS-OXCMSG\]](#).

2.2.1.1 Common Properties

2.2.1.1.1 PidTagNormalizedSubject Property

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

The **PidTagNormalizedSubject** property ([\[MS-OXCMSG\]](#) section 2.2.1.10) specifies the subject of the **free/busy message**. Its value is a string that is derived from the **email address** of the user. The email address is the value of the **PidTagEmailAddress** property ([\[MS-EXOABK\]](#) section 2.2.3.14) of the **Address Book object** of the user. The subject is derived by taking the substring, starting with "/CN", prepending "USER-", and converting all the characters to uppercase.

2.2.1.1.2 PidTagMessageClass Property

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

The **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) MUST be set to "IPM.Post".

2.2.1.2 Free/Busy Properties

2.2.1.2.1 PidTagScheduleInfoMonthsTentative Property

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

The **PidTagScheduleInfoMonthsTentative** property ([\[MS-OXPROPS\]](#) section 2.984) specifies the months for which the **free/busy status** is **tentative**. The number of **PtypInteger32** values in this property MUST be between 0 (zero) and the number of months covered by the **publishing range**, which is the interval between the values of the **PidTagFreeBusyPublishStart** property (section [2.2.1.2.9](#)) and the **PidTagFreeBusyPublishEnd** property (section [2.2.1.2.10](#)).

Each value in this **PtypMultipleInteger32** property data type, interpreted as a signed **PtypInteger32**, has a month and year encoded in it. This is calculated by using the following expression, in which year and month are based on the Gregorian calendar:

$\text{year} \times 16 + \text{month}$

The values are sorted in ascending order and are encoded in **little-endian** format.

If an event is spread across multiple months or multiple years, there MUST be one value for each of the months that fall in the publishing range. If there are no tentative events in the publishing range, this property and the **PidTagScheduleInfoFreeBusyTentative** property (section [2.2.1.2.2](#)) are not set or, if they already exist, are deleted.

2.2.1.2.2 PidTagScheduleInfoFreeBusyTentative Property

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

The **PidTagScheduleInfoFreeBusyTentative** property ([\[MS-OXPROPS\]](#) section 2.980) specifies the blocks of times for which the **free/busy status** is **tentative**. This property has as many values as the number of values in the **PidTagScheduleInfoMonthsTentative** property (section [2.2.1.2.1](#)). Each binary value represents a month and corresponds to the value at the same index in the **PidTagScheduleInfoMonthsTentative** property. The binary values are sorted in the same order as the values in the **PidTagScheduleInfoMonthsTentative** property.

Each binary value has one or more 4-byte blocks, and each block contains the start time in the first two bytes and the end time in the second two bytes in **little-endian** format. The start time is the number of minutes between 12:00 A.M. **Coordinated Universal Time (UTC)** of the first day of the month and the start time of the event in UTC. The end time is the number of minutes between 12:00 A.M. UTC of the first day of the month and the end time of the event in UTC. The 4-byte blocks are sorted in ascending order.

Consecutive or overlapping blocks of time are merged into one block with start time as the start time of the first block and end time as the end time of the last block. If an event is spread across multiple months or years, the event is split into multiple blocks, one for each month. If there are no tentative events in the **publishing range**, this property and the **PidTagScheduleInfoMonthsTentative** property are not set or, if they already exist, are deleted. Otherwise, this property MUST be set.

2.2.1.2.3 PidTagScheduleInfoMonthsBusy Property

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

The **PidTagScheduleInfoMonthsBusy** property ([\[MS-OXPROPS\]](#) section 2.982) specifies the months for which the **free/busy status** is busy. The format, computation, and constraints of this property are the same as those for the **PidTagScheduleInfoMonthsTentative** property (section [2.2.1.2.1](#)) but refer to appointments that are marked as busy on the associated **Calendar object**.

2.2.1.2.4 PidTagScheduleInfoFreeBusyBusy Property

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

The **PidTagScheduleInfoFreeBusyBusy** property ([\[MS-OXPROPS\]](#) section 2.978) specifies the blocks of time for which the **free/busy status** is busy. The format, computation, and constraints of this property are the same as those for the **PidTagScheduleInfoFreeBusyTentative** property (section [2.2.1.2.2](#)) but refer to appointments that are marked as busy on the associated **Calendar object**.

2.2.1.2.5 PidTagScheduleInfoMonthsAway Property

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

The **PidTagScheduleInfoMonthsAway** property ([\[MS-OXPROPS\]](#) section 2.981) specifies the months for which **free/busy status** is **OOF**. The format, computation, and constraints of this property are the same as those for the **PidTagScheduleInfoMonthsTentative** property (section [2.2.1.2.1](#)) but refer to appointments that are marked as OOF on the associated **Calendar object**.

2.2.1.2.6 PidTagScheduleInfoFreeBusyAway Property

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

The **PidTagScheduleInfoFreeBusyAway** property ([\[MS-OXPROPS\]](#) section 2.977) specifies the blocks of time for which the **free/busy status** is **OOF**. The format, computation, and constraints of this property are the same as those for the **PidTagScheduleInfoFreeBusyTentative** property (section [2.2.1.2.2](#)) but refer to appointments that are marked as OOF on the associated **Calendar object**.

2.2.1.2.7 PidTagScheduleInfoMonthsMerged Property

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

The **PidTagScheduleInfoMonthsMerged** property ([\[MS-OXPROPS\]](#) section 2.983) specifies the months for which **free/busy status** is either busy or **OOF**. The syntax/format and constraints of this property are the same as those for the **PidTagScheduleInfoMonthsTentative** property (section [2.2.1.2.1](#)) but refer to appointments that are marked as either busy or OOF on the associated **Calendar object**.

2.2.1.2.8 PidTagScheduleInfoFreeBusyMerged Property

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

The **PidTagScheduleInfoFreeBusyMerged** property ([\[MS-OXPROPS\]](#) section 2.979) specifies the blocks of time for which the **free/busy status** is either busy or **OOF**. The format, computation, and constraints of this property are the same as those for the **PidTagScheduleInfoFreeBusyTentative** property (section [2.2.1.2.2](#)) but refer to appointments that are marked as either busy or OOF on the associated **Calendar object**.

2.2.1.2.9 PidTagFreeBusyPublishStart Property

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

The **PidTagFreeBusyPublishStart** property ([\[MS-OXPROPS\]](#) section 2.708) specifies the start time of the **publishing range**. This property is set to 12:00 A.M. **UTC** on the first day of the month or the first day of the week, whichever occurs earlier at the time of publishing. This value is expressed as the number of minutes since midnight, January 1, 1601, in UTC.

2.2.1.2.10 PidTagFreeBusyPublishEnd Property

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

The **PidTagFreeBusyPublishEnd** property ([\[MS-OXPROPS\]](#) section 2.707) specifies the end time of the **publishing range**. The value of this property is computed by adding the value of the **PidTagFreeBusyCountMonths** property ([\[MS-OXOCAL\]](#) section 2.2.12.1) to the value of the **PidTagFreeBusyPublishStart** property (section [2.2.1.2.9](#)). This value is expressed as the number of minutes since midnight, January 1, 1601 in **UTC**.

2.2.1.2.11 PidTagFreeBusyRangeTimestamp Property

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

The **PidTagFreeBusyRangeTimestamp** property ([\[MS-OXPROPS\]](#) section 2.709) specifies the time, in **UTC**, that the data was published.

2.2.1.2.12 PidTagFreeBusyMessageEmailAddress Property

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

The **PidTagFreeBusyMessageEmailAddress** property ([MS-OXPROPS] section 2.706) specifies the **email address** of the user or resource to whom this **free/busy message** applies. The value of this property is the same as the value of the **PidTagEmailAddress** property ([MS-EXOABK] section 2.2.3.14) of the **Address Book object** for the attendee or resource.

2.2.1.3 Delegate Information Properties

The following properties are optional on a **free/busy message**:

- **PidTagScheduleInfoAutoAcceptAppointments** ([MS-OXOCAL] section 2.2.12.2)
- **PidTagScheduleInfoDisallowRecurringAppts** ([MS-OXOCAL] section 2.2.12.3)
- **PidTagScheduleInfoDisallowOverlappingAppts** ([MS-OXOCAL] section 2.2.12.4)
- **PidTagScheduleInfoDelegatorWantsCopy** ([MS-OXODLGT] section 2.2.2.2.1)
- **PidTagScheduleInfoDontMailDelegates** ([MS-OXPROPS] section 2.975)
- **PidTagScheduleInfoDelegatorWantsInfo** ([MS-OXODLGT] section 2.2.2.2.2)
- **PidTagFreeBusyCountMonths** ([MS-OXOCAL] section 2.2.12.1)

If any of these properties is set on the free/busy message, the value MUST be equal to the value of the same property on the **Delegate Information object**. For details about the Delegate Information object, see [MS-OXODLGT].

2.2.1.4 Deprecated Properties

2.2.1.4.1 PidTagGatewayNeedsToRefresh Property

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

The **PidTagGatewayNeedsToRefresh** property ([MS-OXPROPS] section 2.711) SHOULD NOT [<2>](#) be set and MUST be ignored upon receipt.

2.2.1.4.2 PidTagScheduleInfoResourceType Property

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

The **PidTagScheduleInfoResourceType** property ([MS-OXPROPS] section 2.985) is set to 0 when sending and ignored on receipt. [<3>](#)

2.2.1.4.3 PidTagScheduleInfoFreeBusy Property

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

The **PidTagScheduleInfoFreeBusy** property ([MS-OXPROPS] section 2.976) SHOULD NOT be set and MUST be ignored upon receipt.

2.2.2 Public Folder Free/Busy Related Properties

2.2.2.1 PidTagFreeBusyEntryIds Property

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

The **PidTagFreeBusyEntryIds** property ([MS-OXOSFLD] section 2.2.6) contains the **EntryID** of the **free/busy message** of the logged-on user and the EntryID of the Freebusy Data folder. For details

about the Freebusy Data folder, see [MS-OXOSFLD] section 2.2.1. This property is set on the **Root folder** and on the **Inbox folder** of the local **message store**. For complete details about setting this property, see [MS-OXOSFLD].

2.2.2.2 PidTagSchedulePlusFreeBusyEntryId Property

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

The **PidTagSchedulePlusFreeBusyEntryId** property ([\[MS-OXPROPS\]](#) section 2.986) specifies the **EntryID** of the folder named SCHEDULE+ FREE BUSY, which is under the **non-interpersonal messaging subtree** of the **public folders**.

This property is not stored on any object but is computed on demand from the folder ID (FID) of the SCHEDULE+ FREE BUSY folder. The FID, which is returned in a **RopLogon ROP response** ([\[MS-OXCROPS\]](#) section 2.2.3.1), is preserved and used to compute the EntryID when it is needed.

3 Protocol Details

3.1 Client Details

The client creates and manipulates a **free/busy message** and in all other ways operates within the client role as specified in [\[MS-OXCMSG\]](#).

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.

Free/busy data is kept in a specific message in the **public folders message store**. There is one message for each user for whom free/busy data is published. This message is called the **free/busy message**. It is contained in a folder that represents the administrative group to which the user belongs. There is one folder for each administrative group in the organization. All of these folders are descendants of the **special folder** SCHEDULE+ FREE BUSY. Each folder that represents an administrative group is a sibling of the folders that represent other administrative groups. An administrative group represents an organizational unit within an organization, as specified in [\[RFC1279\]](#). The organizational unit is specified by the *org-unit-rdn* component of the **distinguished name (DN)** of the **Address Book object** for the user, as defined in [\[MS-OXOABK\]](#) section 2.2.1.1.

The **PidTagSchedulePlusFreeBusyEntryId** property (section [2.2.2.2](#)) is not persisted on any object but is temporarily cached in memory or computed on demand.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Publishing Free/Busy Data

The free/busy data is published whenever appointments are created or deleted or when any of the following three properties of an appointment are modified: **PidLidAppointmentStartWhole** ([\[MS-OXOCAL\]](#) section 2.2.1.5), **PidLidAppointmentEndWhole** ([\[MS-OXOCAL\]](#) section 2.2.1.6), or **PidLidBusyStatus** ([\[MS-OXOCAL\]](#) section 2.2.1.2).

A client keeps a local copy of free/busy data and publishes the changes periodically or immediately after the data changes. A client can read or write free/busy data for another user.

The client performs the following steps to publish the data:

1. The client determines the data to be published by using the process specified in section [3.1.4.1.1](#).
2. The client finds the **free/busy message** by using the process specified in section [3.1.4.1.2](#).

3. If the message does not exist, the client creates a new **Message object** and performs step 2 to determine the subfolder and the subject of the message.
4. If the free/busy message was created, the client sets the third binary value of the **PidTagFreeBusyEntryIds** property (section [2.2.2.1](#)) to the **EntryID** of the free/busy message. For details about how to create a Message object, see [\[MS-OXCMSG\]](#).

3.1.4.1.1 Determining the Data Set for Publishing

Data to be published is determined by finding all of the calendar events that are within the **publishing range**, which is established by the **PidTagFreeBusyPublishStart** (section [2.2.1.2.9](#)) and **PidTagFreeBusyPublishEnd** (section [2.2.1.2.10](#)) properties. A calendar event is within the publishing range if it meets the following criteria:

- The end time of the calendar event is greater than the start time of the publishing range.
- The start time of the calendar event is less than the end time of the publishing range.

3.1.4.1.2 Finding the Free/Busy Message

The **free/busy messages** are stored in a descendant folder of the SCHEDULE+ FREE BUSY folder, under the **non-interpersonal messaging subtree** of the **public folders message store**. There MUST be only one free/busy message for each user. A client can read or write free/busy data for users other than the logged-on users. The client MUST find the free/busy message that corresponds to the user before reading or writing information for that user. At the time of publishing, the client determines whether a free/busy message already exists for the user in question before creating one.

To locate the free/busy message that corresponds to a given user, the client does one of the following:

- To find the free/busy message of the logged-on user, the client gets the **EntryID** of the free/busy message. The EntryID of the free/busy message is set in the third binary value of the **PidTagFreeBusyEntryIds** property (section [2.2.2.1](#)) of the **Inbox folder**. The EntryID of the free/busy message is used to generate the message ID of the message, as specified in [\[MS-OXCDATA\]](#) section 2.2.1.2. The client uses the message ID of the message to open the message, as specified in [\[MS-OXCMSG\]](#).
- To find the free/busy message of another user, the client uses the user's name (full name or part of the name) and performs an **ambiguous name resolution (ANR)** to get the **Address Book object** for that user, as specified in [\[MS-OXOABK\]](#) section 3.1.4.4.
- If the **PidTagFreeBusyEntryIds** property does not exist, or if the third binary value of this property is empty, or if the free/busy message is being requested for another user, the subject of the message and the name of the subfolder can be used to find the free/busy message, as specified in sections [3.1.4.1.2.1](#), [3.1.4.1.2.2](#), [3.1.4.1.2.3](#), and [3.1.4.1.2.4](#).

3.1.4.1.2.1 Determining the Email Address

To determine the **email address**, the client gets the **Address Book object** by using one of the following methods:

- If the client is publishing the data for the logged-on user, the client finds the Address Book object of the logged on user by matching its **EntryID** with the value of the **PidTagMailboxOwnerEntryId** property ([\[MS-OXCSTOR\]](#) section 2.2.2.1.1.7) of the **message store**.
- If the client is publishing the data for another user, the client gets the Address Book object by performing an **ANR** with the full name or partial name of the delegator. For details about how to resolve a name, see [\[MS-OXOABK\]](#) section 3.1.4.4.

The client gets the email address from the **PidTagEmailAddress** property ([MS-OXOABK] section 2.2.3.14) of the Address Book object.

3.1.4.1.2.2 Determining the Name of the Subfolder

To determine the name of the subfolder of the SCHEDULE+ FREE BUSY folder, the client performs the following steps:

1. The client derives the name of the subfolder by truncating the **email address** up to, but not including, the first occurrence of "/cn".
2. The client prepends the string "EX:" to the value computed in step 1.

3.1.4.1.2.3 Determining the Subject of the Message

To determine the subject of the **free/busy message**, the client performs the following steps:

1. The client derives the subject by taking the substring that starts with the first occurrence of "/cn" in the **email address**.
2. The client prepends the string "USER-" to the value computed in step 1.
3. The client converts all characters to uppercase.

3.1.4.1.2.4 Finding the Message

To find the message, the client performs the following steps:

1. The client derives the name of the subfolder and the subject of the **free/busy message** by using the logic specified in sections [3.1.4.1.2.2](#) and [3.1.4.1.2.3](#).
2. The client opens the SCHEDULE+ FREE BUSY folder that is specified by the **PidTagSchedulePlusFreeBusyEntryId** property (section [2.2.2.2](#)).
3. The client opens the specific folder for this user by using the name that is derived as specified in section 3.1.4.1.2.2. For details about opening a folder, see [\[MS-OXCFOLD\]](#).
4. The client finds the specific free/busy message for this user by finding the message in which the value of the **PidTagSubject** property ([\[MS-OXPROPS\]](#) section 2.1033) equals the subject that is derived as specified in section 3.1.4.1.2.3. For details about how to find a message that matches a subject, see [\[MS-OXCTABL\]](#).

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

The server processes a client's requests regarding a **free/busy message** and in all other ways operates within the server role as specified in [\[MS-OXCMSG\]](#).

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.

Free/busy data is kept in a specific message in the **public folders message store**. There is one message for each user for whom free/busy data is published. This message is called the **free/busy message**. It is contained in a folder that represents the administrative group to which the user belongs. There is one folder for each administrative group in the organization. All of these folders are descendants of the **special folder** SCHEDULE+ FREE BUSY. Each folder that represents an administrative group is a sibling of the folders that represent other administrative groups. An administrative group represents an organizational unit within an organization, as specified in [\[RFC1279\]](#). The organizational unit is specified by the *org-unit-rdn* component of the **DN** of the **Address Book object** for the user, as defined in [\[MS-OXOABK\]](#) section 2.2.1.1.

The **PidTagSchedulePlusFreeBusyEntryId** property (section [2.2.2.2](#)) is not persisted on any object but is temporarily cached in memory or computed on demand.

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

The server responds to client requests as specified in [\[MS-OXCMSG\]](#) section 3.2.5.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Updating the Free/Busy Message

The following example describes how the **free/busy message** of a user is updated by the client after the user adds new appointments to his calendar.

Joe sets his **publishing range** to three months.

Joe creates the following appointments on his calendar. All of the events have a **free/busy status** of busy.

Note All times in this example are in Pacific Time.

```
Feb 2nd
  appointment time: 12:00 P.M. - 1:00 P.M.
  appointment time: 1:00 P.M. - 2:00 P.M.

April 2nd
  appointment time: 12:00 P.M. - 1:00 P.M.
  appointment time: 3:00 P.M. - 4:00 P.M.
```

The client publishes the availability information by using the **RopCreateMessage ROP** ([MS-OXCROPS] section 2.2.6.2) to create the message in the subfolder for the administrative group of the user under the SCHEDULE+ FREE BUSY folder. The server returns a success code and a **handle** to a **Message object**.

Then the client transmits availability data to the server by using the **RopSetProperties ROP** ([MS-OXCROPS] section 2.2.8.6). The properties listed in the following table are sent.

Property	Property ID	Property type	Value
PidTagScheduleInfoResourceType (section 2.2.1.4.2)	0x6841	Signed PtypInteger32 ([MS-OXCROPS] section 2.11.1)	0x00000000
PidTagGatewayNeedsToRefresh (section 2.2.1.4.1)	0x6846	PtypBoolean ([MS-OXCROPS] section 2.11.1)	0x00000001
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	PtypString ([MS-OXCROPS] section 2.11.1)	USER-/CN=recipients/CN=JOE
PidTagScheduleInfoMonthsMerged (section 2.2.1.2.7)	0x684F	Signed PtypMultipleInteger32 ([MS-OXCROPS] section 2.11.1)	See section 4.3. The PidTagScheduleInfoMonthsMerged property is calculated similarly to the PidTagScheduleInfoMonthsBusy property (section 2.2.1.2.3).
PidTagScheduleInfoFreeBusyMerged (section 2.2.1.2.8)	0x6850	PtypMultipleBinary ([MS-OXCROPS] section 2.11.1)	See section 4.5.
PidTagScheduleInfoMonthsBusy	0x6853	Signed PtypMultipleInteger	See section 4.3.

Property	Property ID	Property type	Value
		32	
PidTagScheduleInfoFreeBusyBusy (section 2.2.1.2.4)	0x6854	Signed PtypMultipleBinary	See section 4.4 .
PidTagFreeBusyPublishStart (section 2.2.1.2.9)	0x6847	Signed PtypInteger32	214105440
PidTagFreeBusyPublishEnd (section 2.2.1.2.10)	0x6848	Signed PtypInteger32	214234980
PidTagFreeBusyRangeTimestamp (section 2.2.1.2.11)	0x6868	PtypTime ([MS-OXCDATA] section 2.11.1)	2008/02/22 01:13:00.000

Because no events have a free/busy status of **tentative** or **OOF**, the client does not set the **PidTagScheduleInfoFreeBusyTentative** property (section [2.2.1.2.2](#)), the **PidTagScheduleInfoMonthsTentative** property (section [2.2.1.2.1](#)), the **PidTagScheduleInfoFreeBusyAway** property (section [2.2.1.2.6](#)), or the **PidTagScheduleInfoMonthsAway** property (section [2.2.1.2.5](#)).

The client then commits the properties on the server by using the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) and then releases the Message object by using the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3).

The values of some properties change during the processing of the **RopSaveChangesMessage** ROP, but the properties specified in this protocol do not change.

4.2 Finding Free/Busy Messages by Using Email Addresses

The following example shows how the client finds the **free/busy message** of a user by using the user's **email address**.

First, the client obtains the user's email address from the **PidTagEmailAddress** property ([\[MS-OXOABK\]](#) section 2.2.3.14). The client then uses the email address to determine the name of the folder in which the free/busy message exists and to determine the subject of the free/busy message.

For example, a user who has an email address of "o=Adventure-Works/ou=New York/CN=recipients/CN=David" would have his or her free/busy data stored in a free/busy message that has a subject of "USER-/CN=recipients/CN=David" and the free/busy message would be stored in the folder named "EX:/o=Adventure-Works/ou=New York".

The following is an example of the **ROPs** the client uses to find the free/busy message. David has one event in his calendar that starts on December 25, 2007, at 7:10 P.M. and ends on December 25, 2008, at 7:10 P.M., Pacific Standard Time.

RopLogon ROP Request ([\[MS-OXCROPS\]](#) section 2.2.3.1)

```

RopId : 0xFE
LogonID : 0
OutputHandleIndex : 0 (HSOT=0xffffffff)
LogonFlags : 0x04 Replicated
OpenFlags : 0x00000406 PUBLIC HOME LOGON NO MAIL
StoreState : 0x00000000 <none>
EssdnSize : 0x00
ESSDN <no Mailbox>

```

RopLogon ROP Response

```
RopId : 0xFE
OutputHandleIndex : 0 (HSOT=0x00000017)
ReturnValue : ecNone (success) (0x00000000)
LogonFlags : 0x04 Replicated
Public store
FolderArray :
  FolderID 1: 0001-0000000000006 Root folder
  FolderID 2: 0001-0000000000001 IPM subtree
  FolderID 3: 0001-0000000000002 non-IPM subtree
  FolderID 4: 0001-0000000000003 Eforms registry
  FolderID 5: 0001-0000000000004
  FolderID 6: 0001-0000000000005 offline address book
  FolderID 7: 0000-0000000000000 Local Eforms registry
  FolderID 8: 0003-0000000000007 SCHEDULE+ FREEBUSY
  FolderID 9: 0004-0000000000008 Local address book
  FolderID 10: 0000-0000000000000 Article index
  FolderID 11: 0000-0000000000000 Schedule
  FolderID 12: 0000-0000000000000 <not used>
  FolderID 13: 0000-0000000000000 <not used>
ServerGUID : a608eae8-6603-4509-89b3-6dac886dca4d
PublicFolderPerUserGUID : fbdd61f1-4863-4d07-902a-64d07f4ca88d
```

The client opens the folder named SCHEDULE+ FREE BUSY, which has an ID of 0001-0000000000004 as determined from the **RopLogon** response.

RopOpenFolder ROP Request ([MS-OXCROPS] section 2.2.4.1)

```
RopId : 0x02
LogonID : 5
InputHandleIndex : 0 (HSOT=0x0000008c)
FolderId : 0001-0000000000004
OpenModeFlags : 0x00 ReadOnly
```

The response to the **RopOpenFolder ROP request** is omitted for readability and clarity.

To find the subfolder that matches the name that was determined earlier, the client sends the following ROP requests.

RopGetHierarchyTable ROP Request ([MS-OXCROPS] section 2.2.4.13)

```
RopId : 0x04
LogonID : 5
InputHandleIndex : 0 (HSOT=0x0000008b)
OutputHandleIndex : 1 (HSOT=0xffffffff)
TableFlags : 0x00 Standard
```

The response to the **RopGetHierarchyTable** ROP request has been omitted for readability and clarity.

RopSetColumns ROP Request ([MS-OXCROPS] section 2.2.5.1)

```
RopId : 0x12
LogonID : 5
InputHandleIndex : 1 (HSOT=0xffffffff)
SetColumnsFlag : 0x00 Wait
PropertyTagCount : 3 (0x03)
PropertyTags : 0x67480014 PidTagFolderId
               0x3001001F PidTagDisplayName
```

The response to the **RopSetColumns** ROP request has been omitted for readability and clarity.

RopFindRow ROP Request ([MS-OXCROPS] section 2.2.5.13)

```
RopId : 0x4F
LogonID : 5
InputHandleIndex : 1 (HSOT=0xffffffff)
FindRowFlags : 0x00 Direction: forward
RestrictionDataSize : 0x0078 (120)
RestrictionData :
ConditionType : 0x04 RES_PROPERTY:
RelationalOperator : 0x04 RELOP_EQ
0x3001001F PidTagDisplayName EX:/o=Adventure-Works/ou=New York
Origin: 0x00 BOOKMARK_BEGINNING
BookmarkSize : 0x0000 (0)
```

RopFindRow ROP Response

```
RopId : 0x4F
InputHandleIndex : 1 (HSOT=0x00000039)
ReturnValue : ecNone (success) (0x00000000)
RowNoLongerVisible: 0x00 (FALSE)
HasRowData: 0x01 (TRUE)
RowData:
  HasError: 0
  PropertyArray:
  PropCount: 3
  0x67480014 PidTagFolderId 0x0700000000000003
  0x3001001F PidTagDisplayName EX:/o=Adventure-Works/ou=New York
```

Now that the folder has been found, and the FID was retrieved by using the **RopFindRow** ROP, the client opens the folder.

RopOpenFolder ROP Request

```
RopId : 0x02
LogonID : 5
InputHandleIndex : 0 (HSOT=0x0000008C)
FolderId : 0003-00000000000007
OpenModeFlags : 0x00 ReadOnly
```

The response to the **RopOpenFolder** ROP request has been omitted for readability and clarity.

RopGetContentsTable ROP Request ([MS-OXCROPS] section 2.2.4.14)

```
RopId : 0x05
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000023)
OutputHandleTable : 1 (HSOT=0xFFFFFFFF)
TableFlags : 0x00 Standard
```

RopSetColumns ROP Request

```
RopId : 0x12
LogonID : 0
InputHandleIndex : 1 (HSOT=0xFFFFFFFF)
```

SetColumnFlags : 0x00 Wait
PropertyTagCount : 5 (0x05)
PropertyTags : 0x67480014 PidTagFolderId
 0x674A0014 PidTagMid
 0x674D0014 PidTagInstID
 0x674E0003 PidTagInstanceNum
 0x0E1D001F PidTagNormalizedSubject

RopSortTable ROP Request ([MS-OXCROPS] section 2.2.5.2)

RopId : 0x13
LogonID : 0
InputHandleIndex : 1 (HSOT=0xFFFFFFFF)
SortTableFlags : 0x00 Wait
SortOrderCount : 0x0001 (1)
CategoryCount : 0x0000 (0)
ExpandedCount : 0x0000 (0)
SortOrders :
 PropertyTag : 0x0E1D001F PidTagNormalizedSubject
 0x00 Flag: TABLE_SORT_ASCEND

RopFindRow ROP Request

RopId : 0x4F
LogonID : 0
InputHandleIndex : 1 (HSOT=0xFFFFFFFF)
FindRowFlags : 0x00 Direction: forward
RestrictionDataSize : 0x0054 (84)
RestrictionData :
 ConditionType : 0x04 RES_PROPERTY:
 RelationalOperator : 0x04 RELOP_EQ
 0x0E1D001F PidTagNormalizedSubject USER-/CN=recipients/CN=DAVID
 Origin : 0x00 BOOKMARK BEGINNING
 BookmarkSize : 0x0000 (0)

RopGetContentsTable ROP Response

RopId : 0x05
OutputHandleIndex : 1 (HSOT=0x00000022)
ReturnValue : ecNone (success) (0x00000000)
RowCount : 113

RopSetColumns ROP Response

RopId : 0x12
InputHandleIndex : 1 (HSOT=0x00000022)
ReturnValue : ecNone (success) (0x00000000)
TableStatus : TBLSTAT_COMPLETE (0x00)

RopSortTable ROP Response

RopId : 0x13
InputHandleIndex : 1 (HSOT=0x00000022)
ReturnValue : ecNone (success) (0x00000000)
TableStatus : TBLSTAT_COMPLETE (0x00)

RopFindRow ROP Response

```
RopId : 0x4F
InputHandleIndex : 1 (HSOT=0x00000022)
ReturnValue : NotFound (0x8004010F)
```

The message is not found, so the client creates it.

RopCreateMessage ROP Request ([MS-OXCROPS] section 2.2.6.2)

```
RopId : 0x06
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000023)
OutputHandleIndex : 1 (HSOT=0xFFFFFFFF)
CodePageId : 0x0FFF (4095)
FolderId : 0003-0000000000007
AssociatedFlag : 0x00
```

RopCreateMessage ROP Response

```
RopId : 0x06
OutputHandleIndex : 1 (HSOT=0x00000020)
ReturnValue : ecNone (success) (0x00000000)
HasMessageId : 0
```

Now the client sets the properties.

RopSetProperties ROP Request ([MS-OXCROPS] section 2.2.8.6)

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x000F (15)
PropertyValueCount : 2 (0x02)
PropertyValues : 0x68410003 PidTagScheduleInfoResourceType
                  0x00000000 (0) 0x6846000B PidTagGatewayNeedsToRefresh 0x0001 (TRUE)
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x0156 (342)
PropertyValueCount : 7 (0x07)
PropertyValues : 0x68410003 PidTagScheduleInfoResourceType
                  0x6842000B PidTagScheduleInfoDelegatorWantsCopy 0x0001 (TRUE)
                  0x6843000B PidTagScheduleInfoDontMailDelegates 0x0001 (TRUE)
                  0x686D000B PidTagScheduleInfoAutoAcceptAppointments 0x0000 (FALSE)
                  0x686E000B PidTagScheduleInfoDisallowRecurringAppts 0x0000 (FALSE)
                  0x686F000B PidTagScheduleInfoDisallowOverlappingAppts 0x0000 (FALSE)
                  0x684B000B PidTagScheduleInfoDelegatorWantsInfo 0x0001 (TRUE)
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x0056 (86)
```



```
PropertyValueCount : 2 (0x02)
PropertyValues : 0x003D001F PidTagSubjectPrefix (null)
                 0x0E1D001F PidTagNormalizedSubject USER-/CN=recipients/CN=DAVID
```

RopSaveChangesMessage ROP Request ([MS-OXCROPS] section 2.2.6.3)

```
RopId : 0x0C
LogonID : 0
ResponseHandleIndex : 1 (HSOT=0x00000023)
InputHandleIndex : 0 (HSOT=0x00000020)
SaveFlags : 0x0A KeepOpenReadWrite DelayedCall
```

RopSaveChangesMessage ROP Response

```
RopId : 0x0C
ResponseHandleIndex : 1 (HSOT=0x00000023)
ReturnValue : ecNone (success) (0x00000000)
InputHandleIndex : 0 (HSOT=0x00000020)
MessageID : 0001-0000000051E3
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 2 (HSOT=0x00000020)
PropertyValueSize : 0x00AC (172)
PropertyValueCount : 1 (0x01)
PropertyValues : 0x6849001F PidTagFreeBusyMessageEmailAddress /o=Adventure-Works/ou=New
York/CN=recipients/CN=David
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x0026 (38)
PropertyValueCount : 2 (0x02)
PropertyValues : 0x684F1003 PidTagScheduleInfoMonthsMerged
                  PtypMultipleInteger32 [0]: 32130
                  PtypMultipleInteger32 [1]: 32131
                  0x68501102 PidTagScheduleInfoFreeBusyMerged
                  PtypMultipleBinary [0] (4 bytes): 0000: E0 01 20 A3
                  PtypMultipleBinary [1] (4 bytes): 0000: 00 00 E0 01
```

RopDeletePropertiesNoReplicate ROP Request ([MS-OXCROPS] section 2.2.8.9)

```
RopId : 0x7A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyTagCount : 2 (0x02)
PropertyTags : 0x68511003 PidTagScheduleInfoMonthsTentative
               0x68521102 PidTagScheduleInfoFreeBusyTentative
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x0026 (38)
PropertyValueCount : 2 (0x02)
PropertyValues : 0x68531003 PidTagScheduleInfoMonthsBusy
                  PtypMultipleInteger32 [0]: 32130
                  PtypMultipleInteger32 [1]: 32131
0x68541102 PidTagScheduleInfoFreeBusyBusy
                  PtypMultipleBinary [0] (4 bytes):0000: E0 01 20 A3
                  PtypMultipleBinary [1] (4 bytes):0000: 00 00 E0 01
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x0007 (7)
PropertyValueCount : 1 (0x01)
PropertyValues : 0x6846000B PidTagGatewayNeedsToRefresh 0x0001 (TRUE)
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x000A (10)
PropertyValueCount : 1 (0x01)
PropertyValues : 0x68470003 PidTagFreeBusyPublishStart PtypInteger32 0x0CC2FD60 (214105440)
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x000A (10)
PropertyValueCount : 1 (0x01)
PropertyValues : 0x68480003 PidTagFreeBusyPublishEnd 0x0CC3A080 (214147200)
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x000E (14)
PropertyValueCount : 1 (0x01)
PropertyValues : 0x68680040 PidTagFreeBusyRangeTimestamp
High: 0x01C87A68
Low: 0x430A6000 (2008/02/29 00:16:00.000)
```

RopSetProperties ROP Request

```
RopId : 0x0A
LogonID : 0
InputHandleIndex : 0 (HSOT=0x00000020)
PropertyValueSize : 0x000A (10)
PropertyValueCount : 1 (0x01)
```

PropertyValues : 0x68410003 PidTagScheduleInfoResourceType 0x00000000 (0)

RopSaveChangesMessage ROP Request

RopId : 0x0C
LogonID : 0
ResponseHandleIndex : 1 (HSOT=0x00000023)
InputHandleIndex : 0 (HSOT=0x00000020)
SaveFlags : 0x08 DelayedCall

RopSaveChangesMessage ROP Response

RopId : 0x0C
ResponseHandleIndex : 1 (HSOT=0x00000023)
ReturnValue : ecNone (success) (0x00000000)
InputHandleIndex : 0 (HSOT=0x00000020)
MessageID : 0001-0000000051e3

The client then calls the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3) on all open folders and the newly created message.

4.3 Calculating the PidTagScheduleInfoMonthsBusy Property

The **PidTagScheduleInfoMonthsBusy** property (section [2.2.1.2.3](#)) is calculated by using the following equations:

1. $(\text{Year} \times 16) + \text{month}$
2. Convert result of equation 1 to hexadecimal

Consider the following example:

The free/busy range is three months from the time of publishing. The time of publishing is 12:00 A.M. on February 25, **UTC**. There is at least one calendar item with a **free/busy status** of busy in the months of February, March, and April.

Therefore, data in the months of February, March, April, and May will be published. More specifically:

- February 25, 12:00 A.M., to February 29, 11:59 P.M.
- March 1, 12:00 A.M., to March 31, 11:59 P.M.
- April 1, 12:00 A.M., to April 30, 11:59 P.M.
- May 1, 12:00 A.M., to May 25, 12:00 A.M.

All of the times in this example are in UTC. The property value is determined by using the following calculations.

For February
Year = 2008 Month = February = 2 $(2008 * 16) + 2 = 32130$ 32120 converted to hexadecimal = 7D82

For March
Year = 2008 Month = March = 3 $(2008 * 16) + 3 = 32131$ 32131 converted to hexadecimal = 7D83

For April
Year = 2008 Month = April = 4 $(2008 * 16) + 4 = 32132$ 32132 converted to hexadecimal = 7D84

The **PtypMultipleInteger32** will have the following three 4-byte values: 0x00007D82, 0x00007D83, and 0x00007D84. Because the month of May had no events, there will not be a value that represents May. Otherwise, there would have been four values.

If the month of March, for example, did not have any calendar items with a free/busy status of busy, there would be only two values.

4.4 Calculating the PidTagScheduleInfoFreeBusyBusy Property

The **PidTagScheduleInfoFreeBusyBusy** property (section [2.2.1.2.4](#)) is calculated by using the following equations:

- Start time is the number of minutes between midnight on the first day of the month and the start time of the event in hexadecimal. The number of days before the scheduled date in that particular month $\times 24$ (hours/day) $\times 60$ (minutes/hour) + (hour of start time on a 24-hour clock adjusted for **UTC**) $\times 60$ (minutes/hour) = Text value of start time in minutes

Note Each time zone requires a different adjustment for UTC.

- End time is the number of minutes between midnight on the first day of the month and the end time of the event in hexadecimal.

For example, assume there is only one event with a **free/busy status** of busy that occurs during the months of February, March, April, and May. It occurs between noon and 1:00 P.M. Pacific Standard Time on February 2. The publishing interval is three months. The time of publishing is 12:00 A.M. on February 25, UTC.

The value is determined by using the following calculations.

Start time
Number of days before the scheduled date in that particular month = 1 Start hour = noon = 12 on a 24-hour clock + 8 for UTC conversion = 20 Note: +8 is the difference between UTC and Pacific Standard Time. The adjustment is different for each time zone. Start time in minutes = $(1 * 60 * 24) + (20 * 60) = 1440 + 1200 = 2640$ Hexadecimal start time = 0A50

End time
Number of days before the scheduled date in that particular month = 1 End hour = 1:00 P.M. Pacific Standard Time = 13 on a 24-hour clock + 8 for UTC conversion = 21 Note: +8 is the difference between UTC and Pacific Standard Time. The adjustment is different for each time zone. End time in minutes = (1 * 60 * 24) + (21 * 60) = 1440 + 1260 = 2700 Hexadecimal end time = 0A8C

Note In some cases, the UTC conversion changes the date of the start time or end time.

The multivalue binary will have one binary value. The binary value will have the following four bytes, encoded in **little-endian** — the first two bytes for the start time and the second two bytes for the end time: 0x50, 0x0A, 0x8C, 0x50.

4.4.1 Calculating the PidTagScheduleInfoFreeBusyBusy Property for Two Nonconsecutive Events

Assume that there are two events with a **free/busy status** of busy in the month of February.

February 2 (times are in Pacific Standard Time)

Appointment time: 12:00 P.M. to 1:00 P.M.

Appointment time: 3:00 P.M. to 4:00 P.M.

The **PtypMultipleBinary** will have one binary value. The binary value will have the following eight bytes: 0x50, 0x0A, 0x8C, 0x0A, 0x04, 0x0B, 0x40, 0x0B.

4.4.2 Calculating the PidTagScheduleInfoFreeBusyBusy Property for Two Consecutive Events

If there are two consecutive events with the same **free/busy status**, the two times are merged together. For example, assume the following two events are marked as busy.

February 2 (times are in Pacific Standard Time)

Appointment time: 12:00 P.M. to 1:00 P.M.

Appointment time: 1:00 P.M. to 2:00 P.M.

There will be only one block with a start time of 12:00 P.M. and an end time of 2:00 P.M. The **PtypMultipleBinary** will have one binary value. The binary value will have the following four bytes: 0x50, 0x0A, 0xC8, 0x0A.

4.4.3 Calculating the PidTagScheduleInfoFreeBusyBusy Property for Events in Multiple Months

Assume that there are two events with a **free/busy status** of busy in the month of February and two events in the month of April. There are no events in March.

February 2 (times are in Pacific Time)

Appointment time: 12:00 P.M. to 1:00 P.M.

Appointment time: 1:00 P.M. to 2:00 P.M.

April 2

Appointment time: 12:00 P.M. to 1:00 P.M.

Appointment time: 3:00 P.M. to 4:00 P.M.

The **PtypMultipleBinary** will have two binary values.

The first binary value will have the following four bytes: 0x50, 0x0A, 0xC8, 0x0A

The second binary value will have the following eight bytes: 0x14, 0x0A, 0x50, 0x0A, 0xC8, 0x0A, 0x04, 0x0B.

Note that the time difference between **UTC** and Pacific Daylight Time in April is 7 hours and therefore the values are calculated by adding 7 hours to the time.

4.4.4 Calculating the PidTagScheduleInfoFreeBusyBusy Property for an Event Spread Across Multiple Months

Consider an example where an event starts on December 25, 2007, at 9:00 A.M. and ends on December 25, 2008, at 10:00 A.M. This event is considered as a series of events that start and end as follows.

December 25 2007, 9:00 A.M. to December 31, 2007, 12:00 A.M.

January 1, 2008, 12:00 A.M. to January 31, 2008, 12:00 A.M.

....

....

December 1, 2008, 12:00 A.M. to December 25, 2008, 10:00 A.M.

The values are calculated as specified in the examples in sections [4.4.1](#), [4.4.2](#), and [4.4.3](#).

If the publishing interval is 12 months and the time of publishing is 12:00 A.M. December 25, 2007, **UTC**, there would be 13 binary values in both the **PidTagScheduleInfoMonthsBusy** (section [2.2.1.2.3](#)) and the **PidTagScheduleInfoFreeBusyBusy** (section [2.2.1.2.4](#)) properties; one for December 2007 and one for each month in 2008.

If the publishing interval is 12 months and the time of publishing is 12:00 A.M. February 25, 2007, **UTC**, there would be 11 binary values.

If the publishing interval is one month and the time of publishing is 12:00 A.M. February 25, 2007, **UTC**, there would be two binary values, for the months of February and March.

4.5 Calculating the PidTagScheduleInfoFreeBusyMerged Property

The **PidTagScheduleInfoFreeBusyMerged** property (section [2.2.1.2.8](#)) includes events with a **free/busy status** of either busy or **OOF**. For example, if a user has an event from 1:00 P.M. to 2:00 P.M. with a free/busy status of busy and another from 4:00 P.M. to 5:00 P.M. with a free/busy status of OOF, the **PidTagScheduleInfoFreeBusyMerged** property will contain two binary values.

If there are consecutive or overlapping scheduled events with free/busy status of busy and OOF, they will be merged into one block. For example, if a user has an event from 1:00 P.M. to 2:00 P.M. with a free/busy status of busy, and another from 1:30 P.M. to 3:00 P.M. with a free/busy status of OOF, the **PidTagScheduleInfoFreeBusyMerged** property will contain one binary value with a start time of 1:00 P.M. and an end time of 3:00 P.M.

5 Security

5.1 Security Considerations for Implementers

There are no security considerations specific to this protocol. General security considerations pertaining to the underlying transport apply, as described in [\[MS-OXCMSG\]](#).

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

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

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

[<1> Section 1.5](#): **Public folders** are supported by default by Exchange 2003. Public folders might have to be configured explicitly on a computer running Exchange 2007 or Exchange 2010.

[<2> Section 2.2.1.4.1](#): Office Outlook 2003 and Office Outlook 2007 will set the **PidTagGatewayNeedsToRefresh** property (section [2.2.1.4.1](#)) to 1.

[<3> Section 2.2.1.4.2](#): The **PidTagScheduleInfoResourceType** property (section [2.2.1.4.2](#)) of the **free/busy message** is read and written by Office Outlook 2003 and Office Outlook 2007, but the value is not used.

7 Change Tracking

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

8 Index

A

Abstract data model

[client](#) 15
[server](#) 18

[Applicability](#) 9

C

[Capability negotiation](#) 9

[Change tracking](#) 33

Client

[abstract data model](#) 15
[initialization](#) 15
[message processing](#) 17
[other local events](#) 17
[overview](#) 15
[sequencing rules](#) 17
[timer events](#) 17
[timers](#) 15

Client – higher-layer triggered events

[publishing free/busy data](#) 15

D

Data model - abstract

[client](#) 15
[server](#) 18

[Delegate information properties on free/busy message](#) 13

E

Examples

[calculating the PidTagScheduleInfoFreeBusyBusy property for an event spread across multiple months](#) 30
[calculating the PidTagScheduleInfoFreeBusyBusy property for events in multiple months](#) 29
[calculating the PidTagScheduleInfoFreeBusyBusy property for two consecutive events](#) 29
[calculating the PidTagScheduleInfoFreeBusyBusy property for two nonconsecutive events](#) 29
[finding free/busy messages by using email addresses](#) 20
[PidTagScheduleInfoFreeBusyBusy calculation overview](#) 28
[PidTagScheduleInfoFreeBusyMerged calculation](#) 30
[PidTagScheduleInfoMonthsBusy calculation](#) 27
[updating the free/busy message](#) 19

F

[Fields - vendor-extensible](#) 9

[Finding free/busy messages by using email addresses example](#) 20

Free/busy message

[delegate information properties](#) 13

[Free/Busy Message message](#) 10

G

[Glossary](#) 6

H

Higher-layer triggered events

[server](#) 18

Higher-layer triggered events - client

[publishing free/busy data](#) 15

I

[Implementer - security considerations](#) 31

[Index of security parameters](#) 31

[Informative references](#) 8

Initialization

[client](#) 15
[server](#) 18

[Introduction](#) 6

M

Message processing

[client](#) 17
[server](#) 18

Messages

[Free/Busy Message](#) 10
[transport](#) 10

N

[Normative references](#) 7

O

Other local events

[client](#) 17
[server](#) 18

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

P

[Parameters - security index](#) 31

[PidTagFreeBusyEntryIds public folder free/busy related property](#) 13

PidTagScheduleInfoFreeBusyBusy calculation example

[calculating the PidTagScheduleInfoFreeBusyBusy property for an event spread across multiple months](#) 30

[calculating the PidTagScheduleInfoFreeBusyBusy property for events in multiple months](#) 29

[calculating the PidTagScheduleInfoFreeBusyBusy property for two consecutive events](#) 29

[calculating the PidTagScheduleInfoFreeBusyBusy property for two nonconsecutive events](#) 29

[overview](#) 28

[PidTagScheduleInfoFreeBusyMerged calculation example](#) 30

[PidTagScheduleInfoMonthsBusy calculation example](#) 27

[PidTagSchedulePlusFreeBusyEntryId public folder free/busy related property](#) 14

[Preconditions](#) 9
[Prerequisites](#) 9
[Product behavior](#) 32
Public folder free/busy related properties
 [PidTagFreeBusyEntryIds property](#) 13
 [PidTagSchedulePlusFreeBusyEntryId property](#) 14

R

[References](#) 7
 [informative](#) 8
 [normative](#) 7
[Relationship to other protocols](#) 8

S

Security
 [implementer considerations](#) 31
 [parameter index](#) 31
Sequencing rules
 [client](#) 17
 [server](#) 18
Server
 [abstract data model](#) 18
 [higher-layer triggered events](#) 18
 [initialization](#) 18
 [message processing](#) 18
 [other local events](#) 18
 [overview](#) 17
 [sequencing rules](#) 18
 [timer events](#) 18
 [timers](#) 18
[Standards assignments](#) 9

T

Timer events
 [client](#) 17
 [server](#) 18
Timers
 [client](#) 15
 [server](#) 18
[Tracking changes](#) 33
[Transport](#) 10
Triggered events - client
 [publishing free/busy data](#) 15
Triggered events - higher-layer
 [server](#) 18

U

[Updating the free/busy message example](#) 19

V

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