

[MS-OXOPFFB]: Public Folder Based Free/Busy Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp/default.aspx>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

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

Preliminary Documentation. This documentation is preliminary documentation for these protocols. Since the documentation may change between this preliminary version and the final version, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and networking programming art, and assumes that the reader is either familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for a Licensee to develop an implementation. Licensees who have access to Microsoft programming tools and environments are free to take advantage of them.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability

Preliminary

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	6
1.2.1	Normative References	6
1.2.2	Informative References	7
1.3	Protocol Overview (Synopsis).....	7
1.4	Relationship to Other Protocols.....	8
1.5	Prerequisites/Preconditions.....	8
1.6	Applicability Statement.....	8
1.7	Versioning and Capability Negotiation.....	8
1.8	Vendor-Extensible Fields	8
1.9	Standards Assignments	8
2	Messages	8
2.1	Transport.....	8
2.2	Message Syntax.....	8
2.2.1	Free/Busy Message.....	9
2.2.2	Public Folder Free/Busy Related Properties	12
3	Protocol Details	13
3.1	Client Details	13
3.1.1	Abstract Data Model	13
3.1.2	Timers	14
3.1.3	Initialization	14
3.1.4	Higher-Layer Triggered Events.....	14
3.1.5	Message Processing Events and Sequencing Rules.....	16
3.1.6	Timer Events.....	16
3.1.7	Other Local Events.....	16
4	Protocol Examples	16
4.1	Free/Busy Message	16
4.2	Finding Free/Busy Messages Using E-mail Addresses	18
4.3	PidTagSchdinfoMonthsBusy Calculation	32
4.4	PidTagSchdinfoFreebusyBusy Calculation	33
4.4.1	PidTagSchdinfoFreebusyBusy Calculation with two non-consecutive events ...	34
4.4.2	PidTagSchdinfoFreebusyBusy Calculation with two consecutive events.....	34
4.4.3	PidTagSchdinfoFreebusyBusy Calculation with events in multiple months	34
4.4.4	PidTagSchdinfoFreebusyBusy calculation when an event is spread across multiple months	35
4.5	PidTagSchdinfoFreebusyMerged Calculation.....	35
5	Security	36
5.1	Security Considerations for Implementers.....	36
5.2	Index of Security Parameters.....	36
6	Appendix A: Office/Exchange Behavior	36

Preliminary

1 Introduction

Public Folder Based Free/Busy Protocol is a format used to publish information describing the availability of a user or resource. This information can be leveraged by a broad range of consumers to efficiently schedule meetings and/or provide presence information.

The Public Folder Based Free/Busy protocol specifies:

- A format in which Free/Busy data is represented.
- A method for publishing data in the prescribed format.
- A method for discovering and interpreting data in the prescribed format.

1.1 Glossary

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

Address Book Object
ambiguous name resolution (ANR)
Appointment
calendar object
delegate
entry ID
folder
folder ID
message
property
property tag
public folder
special folder
store
delegate information object
Resource

NON-IPM-SUBTREE

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

byte
BOOLEAN

The following terms are specific to this document:

Free/Busy Data: This data indicates whether a certain user or resource is available at a certain time.

Free/Busy Status: This indicates how an appointment on a user/resource's calendar affects their availability. More specifically, this is the value of the property PidLidBusyStatus on an appointment, defined in Appointment and Meeting Object Protocol Specification.

Out of Office: This is one of the possible values for Free/Busy Status on an appointment. It means that the user has indicated that they will be out of their office during the time of this appointment.

OOO: Same as Out of Office.

Free: This is one of the possible values for Free/Busy Status on an appointment. It means that the user is available during this appointment.

Busy: This is one of the possible values for Free/Busy Status on an appointment. It means that the user is not available for other appointments during this time.

Tentative: This is one of the possible values for Free/Busy Status on an appointment. It means that the user is tentatively booked during this appointment.

Conflict: When an appointment is scheduled at the same time as another, the appointments are said to be in conflict.

Publishing: In the context of this specification, this term is used to denote writing Free/Busy data to a shared location.

Publishing Range: The number of months of calendar data to be published.

Start of Range: Start date of **Publishing Range**.

End of Range: End date of **Publishing Range**.

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

1.2 References

1.2.1 Normative References

[MS-DTYP] Microsoft Corporation, "Windows Data Types", March 2007, <http://go.microsoft.com/fwlink/?LinkId=111558>.

[MS-OXCDATA] Microsoft Corporation, "Data Structures Protocol Specification", April 2008.

[MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol Specification", April 2008.

[MS-OXOPFFB] - v0.1

Public Folder Based Free/Busy Protocol Specification
Copyright © 2008 Microsoft Corporation.
Release: Friday, April 4, 2008

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol Specification", April 2008.

[MS-OXCSTOR] Microsoft Corporation, "Store Object Protocol Specification", April 2008.

[MS-OXCTABL] Microsoft Corporation, "Table Object Protocol Specification", April 2008.

[MS-OXGLOS] Microsoft Corporation, "Office Exchange Protocols Master Glossary", April 2008.

[MS-OXOCAL] Microsoft Corporation, "Appointment and Meeting Object Protocol Specification", April 2008.

[MS-OXODLGT] Microsoft Corporation, "Delegate Access Configuration Protocol Specification", April 2008.

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol Specification", April 2008.

[MS-OXPROPS] Microsoft Corporation, "Office Exchange Protocols Master Property List Specification", April 2008.

[MS-OXWAVLS] Microsoft Corporation, "Availability Web Service Protocol Specification", April 2008.

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

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

1.2.2 Informative References

None.

1.3 Protocol Overview (Synopsis)

Free/Busy data is derived from calendar data and falls into four categories: **Free**, **Busy**, **Tentative** and **Out of Office**.

The Public Folder Based Free/Busy Protocol specifies how Free/Busy data is read from and written to a shared location so that it can be used to effectively and efficiently schedule meetings.

Free/Busy information can also be obtained using the Availability Web service as specified in [MS-OXWAVLS], and by directly accessing a user's calendar and reading the PidLidBusyStatus property as specified in [MS-OXOCAL].

1.4 Relationship to Other Protocols

The Public Folder Based Free/Busy protocol extends the Message and Attachment Object protocol.

1.5 Prerequisites/Preconditions

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 that schedule meetings to avoid scheduling conflicts.

This specification is not an appropriate choice to accomplish the following related functionality. Below are the alternatives:

- a) Seeing complete Calendar details, is best accomplished by receiving permissions and directly browsing their calendar. See [MS-OXODLGT] for more details.
- b) Seeing Free/Busy information outside of the **Publishing Range** is best accomplished by using the Availability Web Service protocol. See [MS-OXWAVLS] for more details.

It is recommended that this specification be used only if Availability Service, [MS-OXWAVLS] is not supported by the Server <2>. To determine whether Availability Service is supported by the server, please see Autodiscover Publishing and Lookup Protocol specified in [MS-OXDSCLI].

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Public Folder Based Free/Busy protocol uses the protocols defined in [MS-OXCMSG] as its primary transport mechanism.

2.2 Message Syntax

None.

2.2.1 Free/Busy Message

Free/Busy data is represented as a set of properties set on a Message in the public folders store. The message is referred to as the Free/Busy Message. The location of this message is described in Section 3. Unless otherwise specified below, the Free/Busy Message adheres to the guidance in [MS-OXCMSG].

2.2.1.1 Common Properties

2.2.1.1.1 *PidTagNormalizedSubject*

This is a PtypString property and specifies the subject of the Free/Busy Message. Its value is a string derived from the e-mail address of the user. The e-mail address is the value of PidTagEmailAddress property of the address book object of the user. The subject is derived by taking the sub-string starting with “/cn” and prepending “USER-“ and converting all the characters to upper-case.

2.2.1.1.2 *PidTagMessageClass*

This is a PtypString property and its value MUST be set to “IPM.Post”.

2.2.1.2 Free/Busy Properties

2.2.1.2.1 *PidTagScheduleInfoMonthsTentative*

This is a PtypMultipleInteger32 property and it specifies the months for which Free/Busy data of type Tentative is present in the Free/Busy Message. The number of PtypInteger32 values in this property MUST be between zero and the number of months covered by the Publishing Range, which is the period between PidTagFreebusyPublishStart and PidTagFreebusyPublishEnd.

Each value in this PtypMultipleInteger32, interpreted as a signed PtypInteger32 has month and year encoded in it. This is calculated using the expression “year × 16 + month” where year and month are based on the Gregorian calendar. 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, then this property and PidTagSchdinfoFreebusyTentative MUST NOT be set or MUST be deleted if they already exist. Otherwise, this property MUST be set.

2.2.1.2.2 *PidTagScheduleInfoFreebusyTentative*

This is a PtypMultipleBinary property that specifies the blocks of times for which the Free/Busy status is Tentative. There are as many values as the number of values in

PidTagSchdinfoMonthsTentative. Each binary value represents a month and corresponds to the value at the same index, in PidTagSchdinfoMonthsTentative. The binary values are sorted in the same order as the values in PidTagSchdinfoMonthsTentative.

Each binary value has one or more 4-BYTE blocks and each of them contains the start time in the first two bytes and end time in the second two bytes in little-endian format. The start time is the number of minutes between 12 A.M. 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 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, then this property and PidTagSchdinfoMonthsTentative MUST NOT be set or MUST be deleted if they already exist. Otherwise, this property MUST be set.

2.2.1.2.3 PidTagScheduleInfoMonthsBusy

This is a PtypMultipleInteger32 property that specifies the months for which Free/Busy data of type Busy is present in the Free/Busy Message. The format, computation and constraints of this property are the same as those of PidTagSchdinfoMonthsTentative but refer to appointments that are marked Busy on the associated Calendar.

2.2.1.2.4 PidTagScheduleInfoFreebusyBusy

This is a PtypMultipleBinary property that specifies the blocks of time of Free/Busy type Busy. The format, computation and constraints of this property are the same as those of PidTagSchdinfoFreebusyTentative but refer to appointments that are marked Busy on the associated Calendar.

2.2.1.2.5 PidTagScheduleInfoMonthsAway

This is a PtypMultipleInteger32 property that specifies the months for which Free/Busy data of type OOF is present in the Free/Busy message. The format, computation and constraints of this property are the same as those of PidTagSchdinfoMonthsTentative but refer to appointments that are marked OOF on the associated Calendar.

2.2.1.2.6 PidTagScheduleInfoFreebusyAway

This is a PtypMultipleBinary property that specifies the times for which the Free/Busy status is set to OOF. The format, computation and constraints of this property are the same as those of PidTagSchdinfoFreebusyTentative but refer to appointments that are marked OOF on the associated Calendar.

2.2.1.2.7 PidTagScheduleInfoMonthsMerged

This is a PtypMultipleInteger32 property that specifies the months for which Free/Busy data of type Busy or OOF is present in the Free/Busy Message. Events of Free/Busy type Tentative are not included in this property. The syntax/format and constraints of this property are the same as those of PidTagSchdinfoMonthsTentative but refer to appointments that are marked OOF or Busy on the associated Calendar.

2.2.1.2.8 PidTagScheduleInfoFreebusyMerged

This is a PtypMultipleBinary property that specifies the times for which the Free/Busy status is set to Busy or OOF. Events of Free/Busy type Tentative are not included in this property. The format, computation and the restrictions of this property are the same as those of PidTagSchdinfoFreebusyTentative but refer to appointments that are marked OOF or Busy on the associated Calendar.

2.2.1.2.9 PidTagFreeBusyPublishStart

This is a signed PtypInteger32 property that specifies the start time of the Publishing Range. This value is expressed as the number of minutes since midnight, January 1, 1601 in UTC.

2.2.1.2.10 PidTagFreeBusyPublishEnd

This is a signed PtypInteger32 property that specifies the end time of the Publishing Range. It is computed by adding the value of PidTagFreebusyCountMonths to the start date of the Publishing Range. This value is expressed as the number of minutes since midnight, January 1, 1601 in UTC.

2.2.1.2.11 PidTagFreeBusyRangeTimestamp

This is a PtypTime property that specifies the time that the data was published, in UTC.

2.2.1.2.12 PidTagFreeBusyMessageEmailAddress

This is a PtypString property. It specifies the e-mail address of the user to whom this Free/Busy message applies. It is the PidTagEmailAddress property of the user's or resource's address book object.

2.2.1.3 Delegate Information properties

The following properties are used in Delegate Access Configuration Protocol and Appointment and Meeting Object Protocol as defined in [MS-OXODLGT] and [MS-

OXOCAL] respectively. These properties are optional. If they are set on the Free/Busy Message, the value of these properties MUST be equal to the value of the same property in the Delegate Information Message as defined in [MS-OXODLGT].

- *PidTagSchdinfoAutoAcceptAppts*
- *PidTagSchdinfoDisallowRecurringAppts*
- *PidTagSchdinfoDisallowOverlappingAppts*
- *PidTagSchdinfoDelegatorWantsCopy*
- *PidTagSchdinfoDontMailDelegates*
- *PidTagSchdinfoDelegatorWantsInfo*
- *PidTagFreebusyCountMonths*

2.2.1.4 Deprecated Properties

2.2.1.4.1 *PidTagGatewayNeedsToRefresh*

This is a PtypBoolean property. It is deprecated and SHOULD NOT be used. <3>

2.2.1.4.2 *PidTagScheduleInfoResourceType*

This is a signed PtypInteger32 property. It MUST be set to 0. This is a deprecated property and MUST not be read. <4>

2.2.1.4.3 *PidTagScheduleInfoFreeBusy*

This property is deprecated. It SHOULD NOT be set and MUST be ignored. <5>

2.2.2 Public Folder Free/Busy Related Properties

2.2.2.1 *PidTagFreebusyEntryIds*

This is a PtypMultipleBinary property. The value on the root folder of the local store MUST be equal to the value on the Inbox special folder. It is set on the root folder and on the Inbox special folder of the local store. It contains 4 binary values.

- i) The first value MUST be set to NULL.

- ii) The second value **MUST** be set to the entry ID of the Delegate Information Message. For more information about Delegate Information Message, see [MS-OXODLGT].
- iii) The third value **MAY** be set to the entry ID of the logged in user's Free/Busy Message. It **MUST** be set to NULL if public folder Free/Busy is unavailable. This value is set when a client or server creates the Free/Busy Message for the first time for this user.
- iv) The fourth value **MUST** be set to the entry ID of the folder whose PidTagDisplayName is equal to "FreeBusy Data" and is a child folder under the root folder of the store.

3 Protocol Details

3.1 Client Details

Free/Busy information is kept in a specific message in public folders store. There is one message for each user for whom Free/Busy data is published. This message is called Free/Busy Message. It is contained in a folder representing the Administrative Group to which the user belongs. There is one folder for each Administrative Group in the organization. All these folders are descendants of the special folder "SCHEDULE+ FREE BUSY". Each folder representing an Administrative Group is a sibling of the folders representing other Administrative Groups. An Administrative Group represents an "Organisational Unit", within an organization, as defined in RFC 1279. It is the *org-unit-rdn* component of the Distinguished Name of a user's Address Book Object as defined [MS-OXOABK].

A client **MAY** read or write Free/Busy information of another user. To find the Free/Busy Message for a specific user, see Section 3.1.4.1.2.

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.

3.1.1.1 Non-persisted Free/Busy Related Properties

Some properties are not persisted on any object but are temporarily cached in memory or computed on demand. Here are the Free/Busy related properties that are not persisted:

3.1.1.1.1 PidTagSplusFreeBusyEntryid

This is a PtypBinary property that contains the entry ID of the folder named “SCHEDULE+ FREE BUSY” under the NON_IPM_SUBTREE of the public folder store. This property is not stored on any object but is computed on demand. The folder ID of “SCHEDULE+ FREE BUSY” is returned in response to ROPLogon. The folder ID is preserved and used to compute entry ID when it’s needed.

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 should be published whenever appointments are created, deleted or when the following three properties of an appointment are modified – PidLidApptStartWhole, PidLidApptEndWhole, PidLidBusyStatus.

A client or server may keep a local copy of Free/Busy data and publish the changes periodically or immediately after the data changes.

The client may follow the steps below to publish the data:

- Determine the data to be published using the steps in section 3.1.4.1.1.
- Find the Free/Busy Message using the steps in Section 3.1.4.1.2.
- If the Message doesn’t exist, create a new Message. The sub-folder and the subject of the message should be as determined in the previous step.
- If the Message was created, set the third binary value of PidTagFreeBusyEntryids property to the entry ID of the Free/Busy Message. For more about how to create a message, see Message and Attachment Object Protocol Specification.

3.1.4.1.1 Determining Data Set for Publishing

Data to be published is determined by finding all the Calendar events that are in the Publishing Range. The Publishing Range is calculated as follows:

- **Start of Range** is 12 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.
- **End of Range** is calculated by adding the value of PidTagFreebusyCountMonths to Start of Range.
- All the appointments with end time greater than the Start of the Range and start time lesser than the End of the Range are considered to be in the Publishing Range.

3.1.4.1.2 Finding Free/Busy Message

The Free/Busy messages are stored in a descendant folder of the “SCHEDULE+ FREE BUSY” folder, under the NON_IPM_SUBTREE of the public folders store. There MUST be only one Free/Busy Message for each user. A client can read or write Free/Busy information for users other than the logged on users. The client MUST find the Free/Busy Message corresponding to the user before reading or writing information for that user. At the time of Publishing, the client MUST determine whether a Free/Busy Message already exists for the user in question before creating one.

The following steps MAY be used to locate the Free/Busy message corresponding to a given user.

To find the Free/Busy Message of the logged on user, get the entry Id of the Free/Busy Message. The Free/Busy Message’s entry ID property is set in the third binary value of PidTagFreebusyEntryids property of the Inbox. Use it to open the message. To determine how to open a message using the entry ID, see [MS-OXCSTOR].

To find the Free/Busy Message of another user, use the name (full name or part of their name) and perform an ambiguous name resolution to get their address book object, as described in “Address Book Object Protocol Specification”, [MS-OXOABK].

If PidTagFreebusyEntryids property does not exist or if the third binary value of this property is empty, or if Free/Busy Message is being requested for another user, then subject and the name of the sub-folder can be used to find the Free/Busy Message, by following the steps below:

3.1.4.1.2.1 Determining the e-mail address

- Get the address book object:
 - If the client is Publishing the data for the logged on user, find the address book object of the logged on user by matching its entry ID with PidTagMailboxOwnerEntryId of the store.
 - If the client is Publishing the data for another user, get the address book object by performing an ambiguous name resolution using their full name or partial name. See Address Book Object Protocol, specified in [MS-OXOABK] for instructions on how to resolve a name.
 - Get the e-mail address from the PidTagEmailAddress property of the address book object.

3.1.4.1.2.2 Determining the name of the appropriate sub-folder of the “SCHEDULE+ FREE BUSY” folder

- Derive the name of the sub-folder by truncating the e-mail address up to, but not including, the first occurrence of “/cn”.

- Prepend the string “EX:” to it.
- Find the folder using the name we derived in the step above. See Table Object Protocol specified in [MS-OXCTABL] for instructions on how to find a folder by name.

3.1.4.1.2.3 Determining the subject of the appropriate message within the above folder

- Derive the subject by taking the sub-string that starts with the first occurrence of “/cn” in the e-mail address.
- Prepend the string “USER-“ to it.
- Convert all the characters to upper-case.

3.1.4.1.2.4 Finding the message

- Derive the folder name and the subject of the message using the logic above.
- Open the “SCHEDULE + FREE BUSY” folder using PidTagSplusFreeBusyEntryid property.
- Open the specific folder for this user using the name derived using steps in section 3.1.4.1.2.2.
- Find the specific message for this user by looking for the message where PidTagSubject equals to the subject derived using steps in section 3.1.4.1.2.3. See [MS-OXCTABL] for instructions on how to find a message that matches a subject.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Free/Busy Message

Joe sets his Publishing interval to 3 months. Joe creates the following appointments on his calendar. All the events have a Free/Busy type of Busy.

Note All times are in Pacific Time.

Feb 2nd

12 P.M. – 1 P.M.

1 P.M. – 2 P.M.

April 2nd

12 P.M. – 1 P.M.

3 P.M. – 4 P.M.

The client publishes the availability information by using RopCreateMessage to create the message in the sub-folder for the user's Administrative Group under "SCHEDULE + FREE/BUSY" folder. The server returns a success code and a handle to a message object.

Then the client uses RopSetProperties to transmit Availability data to the server.

Property	Property ID	Property Type	Value
PidTagSchdinfoResourceType	0x6841	signed PtypInteger32	0x00000000
PidTagGatewayNeedsToRefresh	0x6846	PtypBoolean	0x00000001
PidTagNormalizedSubject	0x0E1D	PtypString	USER- /CN=RECIPIENTS/CN=JOE
PidTagSchdinfoMonthsMerged	0x684F	signed PtypMultipleInteger32	See examples below.
PidTagSchdinfoFreebusyMerged	0x6850	PtypMultipleBinary	See examples below.
PidTagSchdinfoMonthsBusy	0x6853	signed PtypMultipleInteger32	See examples below.
PidTagSchdinfoFreebusyBusy	0x6854	signed PtypMultipleBinary	See examples below.
PidTagFreebusyPublishStart	0x6847	signed PtypInteger32	214105440
PidTagFreebusyPublishEnd	0x6848	signed PtypInteger32	214234980
PidTagFreebusyRangeTimestamp	0x6868	PtypTime	2008/02/22 01:13:00.000

Because there are no events with Free/Busy type of Tentative or OOF, the client does not set PidTagSchdinfoFreebusyTentative, PidTagSchdinfoMonthsTentative, PidTagSchdinfoFreebusyOof, PidTagSchdinfoMonthsOof.

Then client uses RopSaveChanges to commit the properties on the server, and then RopRelease to release the message object.

The values of some properties will change during the execution of RopSaveChanges, but the properties specified in this protocol will not change.

4.2 Finding Free/Busy Messages Using E-mail Addresses

Determine the folder name in which the Free/Busy Message exists and its subject from the e-mail address of the user.

For example, a user who has the PidTagEmailAddress value of “/o=Adventure-Works/ou=New York/cn= Recipients/cn=David” would have his or her Free/Busy information stored in the folder named “EX:/o=Adventure-Works/ou=New York” and in a message with subject equal to “USER-/CN=RECIPIENTS/CN=David”.

For examples of how to get a user’s address book object and its PidTagEmailAddress, see [MS-OXOABK].

The folder name and subject have now been determined. The following is an example of the ROPs the client uses to find the Free/Busy Message. David has one event in his calendar which starts on December 25th, 2007 at 7:10 P.M. and ends on Dec 25th, 2008 at 7:10 P.M., Pacific Time.

ROPLogon

ROPid: 0xFE

LogonIndex: 0

HandleIndex: 0 (HSOT=0xffffffff)

LogonFlags: 0x04 Replicated

OpenFlags: 0x00000406 PUBLIC HOME_LOGON NO_MAIL

StoreState: 0x00000000 <none>

LegacyDNSize: 0x00

Public Logon <no mailbox>

Response to ROPLogon

ROPLogon

ROPid: 0xFE

HandleIndex: 0 (HSOT=0x00000017)

ReturnValue: ecNone (success) (0x00000000)

LogonFlags: 0x04 Replicated

Public Store

FolderArray:

FolderID 1: 0001-000000000006 Root Folder

FolderID 2: 0001-000000000001 IPM subtree

FolderID 3: 0001-000000000002 Non-IPM subtree

FolderID 4: 0001-000000000003 Eforms registry

FolderID 5: 0001-000000000004

FolderID 6: 0001-000000000005 Offline address book

FolderID 7: 0000-000000000000 Local Eforms registry

FolderID 8: 0003-000000000007 SCHEDULE+ FREEBUSY

FolderID 9: 0004-000000000008 Local address book

FolderID 10: 0000-000000000000 Article index

FolderID 11: 0000-000000000000 Schedule

FolderID 12: 0000-000000000000 <not used>

FolderID 13: 0000-000000000000 <not used>

ServerGUID: a608eae8-6603-4509-89b3-6dac886dca4d

PublicFolderPerUserGUID: fbdd61f1-4863-4d07-902a-64d07f4ca88d

Open the folder named SCHEDULE+ FREEBUSY which has an ID of 0001-000000000004 as determined from above.

ROOpenFolder

ROPid: 0x02

LogonIndex: 5

HandleIndex: 0 (HSOT=0x0000008c)

FID: 0001-000000000004

OpenModeFlags: 0x00 ReadOnly

Response to ROPOpenFolder is omitted for readability and clarity.

To find the sub-folder that matches the name we determined earlier, the client makes the following ROPs:

ROPGetHierarchyTable

ROPid: 0x04

LogonIndex: 5

HandleIndex: 0 (HSOT=0x0000008b)

HierarchyHandleIndex: 1 (HSOT=0xffffffff)

TableFlags: 0x00 Standard

Response to ROPGetHierarchyTable has been omitted for readability and clarity.

ROPSetColumns

ROPid: 0x12

LogonIndex: 5

HandleIndex: 1 (HSOT=0xffffffff)

WantAsync: 0x00 Wait

PropCount: 3 (0x03)

0x67480014 PidTagFID

0x3001001F PidTagDisplayName

Response to ROPSetColumns has been omitted for readability and clarity.

ROPFindRow

ROPid: 0x4F

LogonIndex: 5

HandleIndex: 1 (HSOT=0xffffffff)

WantsBackwardSearch: 0x00 Direction: forward

RestrictionSize: 0x0078 (120)

Restriction:

ConditionType: 0x04 RES_PROPERTY:

RelationalOperator: 0x04 RELOP_EQ

0x3001001F PidTagDisplayName EX:/o=Adventure-
Works/ou=New York

BookmarkFlag: 0x00 BOOKMARK_BEGINNING

BookmarkSize: 0x0000 (0)

Response to ROPFindRow

ROPid: 0x4F

HandleIndex: 1 (HSOT=0x00000039)

ReturnValue: ecNone (success) (0x00000000)

HasChangedPosition: 0x00 (FALSE)

HasRowData: 0x01 (TRUE)

HasError: 0

PropertyArray:

PropCount: 3

0x67480014 PidTagFID 0x0700000000000003

0x3001001F PidTagDisplayName EX:/o=Adventure-
Works/ou=New York

Now that the folder is found, and we got the folder ID from ROPFindRow above, open the folder.

ROPOpenFolder

ROPid: 0x02

LogonIndex: 5

HandleIndex: 0 (HSOT=0x0000008c)

FID: 0003-000000000007

OpenModeFlags: 0x00 ReadOnly

Response to ROPOpenFolder has been omitted for readability and clarity.

ROPGetContentsTable

ROPid: 0x05

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000023)

ContentHandleIndex: 1 (HSOT=0xffffffff)

TableFlags: 0x00 Standard

ROPSetColumns

ROPid: 0x12

LogonIndex: 0

HandleIndex: 1 (HSOT=0xffffffff)

WantAsync: 0x00 Wait

PropCount: 5 (0x05)

0x67480014 PidTagFID

0x674A0014 PidTagMID

0x674D0014 PidTagInstID
0x674E0003 PidTagInstanceNum
0x0E1D001F PidTagNormalizedSubject

ROPSortTable

ROPid: 0x13
LogonIndex: 0
HandleIndex: 1 (HSOT=0xffffffff)
WantAsync: 0x00 Wait
SortCount: 0x0001 (1)
CategoryCount: 0x0000 (0)
ExpandedCount: 0x0000 (0)
PropertyTag:
 0x0E1D001F PidTagNormalizedSubject
Order: 0x00 Flag: TABLE_SORT_ASCEND

ROPFindRow

ROPid: 0x4F
LogonIndex: 0
HandleIndex: 1 (HSOT=0xffffffff)
WantsBackwardSearch: 0x00 Direction: forward
RestrictionSize: 0x0054 (84)
Restriction:
ConditionType: 0x04 RES_PROPERTY:

RelationalOperator: 0x04 RELOP_EQ

0x0E1D001F PidTagNormalizedSubject USER-
/CN=RECIPIENTS/CN=DAVID

BookmarkFlag: 0x00 BOOKMARK_BEGINNING

BookmarkSize: 0x0000 (0)

Response to ROPGetContentsTable

ROPid: 0x05

HandleIndex: 1 (HSOT=0x00000022)

ReturnValue: ecNone (success) (0x00000000)

RowCount: 113

Response to ROPSetColumns

ROPid: 0x12

HandleIndex: 1 (HSOT=0x00000022)

ReturnValue: ecNone (success) (0x00000000)

CompletionStatus: TBLSTAT_COMPLETE (0x00)

Response to ROPSortTable

ROPid: 0x13

HandleIndex: 1 (HSOT=0x00000022)

ReturnValue: ecNone (success) (0x00000000)

TableStatus: TBLSTAT_COMPLETE (0x00)

Response to ROPFindRow

ROPid: 0x4F

HandleIndex: 1 (HSOT=0x00000022)

ReturnValue: NotFound (0x8004010f)

The message is not found, so create it:

ROPCreateMessage

ROPid: 0x06

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000023)

MessageHandleIndex: 1 (HSOT=0xffffffff)

CodePageId: 0x0FFF (4095)

FolderId: 0003-000000000007

AssociatedFlag: 0x00

Response to ROPCreateMessage

ROPid: 0x06

HandleIndex: 1 (HSOT=0x00000020)

ReturnValue: ecNone (success) (0x00000000)

HasMessageId: 0

Now set properties:

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x000F (15)

PropCount: 2 (0x02)

0x68410003 PidTagSchdinfoResourceType 0x00000000 (0)

0x6846000B PidTagGatewayNeedsToRefresh 0x0001 (TRUE)

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x0156 (342)

PropCount: 7 (0x07)

0x68410003 PidTagSchdinfoResourceType

0x6842000B PidTagSchdinfoBossWantsCopy 0x0001 (TRUE)

0x6843000B PidTagSchdinfoDontMailDelegates 0x0001 (TRUE)

0x686D000B PidTagSchdinfoAutoAcceptAppts 0x0000 (FALSE)

0x686E000B PidTagSchdinfoDisallowRecurringAppts 0x0000 (FALSE)

(FALSE) 0x686F000B PidTagSchdinfoDisallowOverlappingAppts 0x0000

0x684B000B PidTagSchdinfoBossWantsInfo 0x0001 (TRUE)

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x0056 (86)

PropCount: 2 (0x02)

0x003D001F PidTagSubjectPrefix (null)

0x0E1D001F PidTagNormalizedSubject USER-
/CN=RECIPIENTS/CN=DAVID

ROPSaveChangesMessage

ROPid: 0x0C

LogonIndex: 0

HandleIndex: 1 (HSOT=0x00000023)

MessageHandleIndex: 0 (HSOT=0x00000020)

SaveOptions: 0x0A KeepOpenReadWrite DelayedCall

Response to ROPSaveChangesMessage:

ROPSaveChangesMessage

ROPid: 0x0C

HandleIndex: 1 (HSOT=0x00000023)

ReturnValue: ecNone (success) (0x00000000)

MessageHandleIndex: 0 (HSOT=0x00000020)

MessageId: 0001-0000000051e3

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 2 (HSOT=0x00000020)

PropertySize: 0x00AC (172)

PropCount: 1 (0x01)

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x0026 (38)

PropCount: 2 (0x02)

0x684F1003 PidTagSchdinfoMonthsMerged

PT_MV_LONG[0]: 32130

PT_MV_LONG[1]: 32131

0x68501102 PidTagSchdinfoFreebusyMerged

PT_MV_BINARY[0] (4 bytes):

0000: E0 01 20 A3 ...

PT_MV_BINARY[1] (4 bytes):

0000: 00 00 E0 01

ROPDeletePropertiesNoReplicate

ROPid: 0x7A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropCount: 2 (0x02)

0x68511003 PidTagSchdinfoMonthsTentative

0x68521102 PidTagSchdinfoFreebusyTentative

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x0026 (38)

PropCount: 2 (0x02)

0x68531003 PidTagSchdinfoMonthsBusy

PT_MV_LONG[0]: 32130

PT_MV_LONG[1]: 32131

0x68541102 PidTagSchdinfoFreebusyBusy

PT_MV_BINARY[0] (4 bytes):

0000: E0 01 20 A3 ...

PT_MV_BINARY[1] (4 bytes):

0000: 00 00 E0 01

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x0007 (7)

PropCount: 1 (0x01)

0x6846000B PidTagGatewayNeedsToRefresh 0x0001 (TRUE)

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x000A (10)

PropCount: 1 (0x01)

0x68470003 PidTagFreebusyPublishStart PT_LONG
0x0CC2FD60 (214105440)

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x000A (10)

PropCount: 1 (0x01)

0x68480003 PidTagFreebusyPublishEnd 0x0CC3A080 (214147200)

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x000E (14)

PropCount: 1 (0x01)

0x68680040 PidTagFreebusyRangeTimestamp High: 0x01C87A68 Low:
0x430A6000 (2008/02/29 00:16:00.000)

ROPSetProperties

ROPid: 0x0A

LogonIndex: 0

HandleIndex: 0 (HSOT=0x00000020)

PropertySize: 0x000A (10)

PropCount: 1 (0x01)

0x68410003 PidTagSchdinfoResourceType 0x00000000 (0)

ROPSaveChangesMessage

ROPid: 0x0C

LogonIndex: 0

HandleIndex: 1 (HSOT=0x00000023)

MessageHandleIndex: 0 (HSOT=0x00000020)

SaveOptions: 0x08 DelayedCall

Response to ROPSaveChangesMessage:

ROPSaveChangesMessage

ROPid: 0x0C

HandleIndex: 1 (HSOT=0x00000023)

ReturnValue: ecNone (success) (0x00000000)

MessageHandleIndex: 0 (HSOT=0x00000020)

MessageId: 0001-0000000051e3

The client then calls ROPRelease on all open folders and the newly created message.

4.3 PidTagSchdinfoMonthsBusy Calculation

This property is calculated by using the following equations:

1. $(Year \times 16) + month$
2. *Converted result of the above equation to hexadecimal*

Consider the following example:

Free/Busy range is 3 months from the time of Publishing. The time of Publishing is 12 A.M. on Feb 25th UTC. There is at least one calendar item with Free/Busy type of Busy in the months of February, March, and April.

So, we will publish data in the months of February, March, April and May. More specifically,

Feb 25th 12 A.M. to Feb 29th 11:59 P.M.

March 1st 12 A.M. to March 31st 11:59 P.M.

April 1st 12 A.M. to April 30th 11:59 P.M.

May 1st 12 A.M. to May 25th 12 A.M.

All the times above are in UTC. The property value is determined by using the following calculations:

For February
Year = 2008
Month = February = 2
$(2008 * 16) + 2 = 32130$
32130 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

So the PtypMultipleInteger32 will have 3 values with 0x00007D82, 0x00007D83, 0x00007D84 in each of the LONG values. Because the month of May did not have any events, there will not be a value representing May. Otherwise, there would have been 4 values. If the month of March, for example, did not have any calendar items with Free/Busy type of Busy, and there would be only two values.

4.4 *PidTagSchdinfoFreebusyBusy Calculation*

This property is calculated by using the following equations:

Start Time is the minutes between midnight on the first day of the month and the start time of the event in hexadecimal.

*Number of days before the scheduled date in that particular month × 24 (hours/day) * 60 (minutes/hour) + (hour of start time on a 24-hour clock adjusted for Coordinated Universal Time [UTC]) * 60 (minutes/hour) = Text value of start time in minutes*

Note: Each time zone requires a different adjustment for UTC.

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

For example, if there is only one event with Free/Busy type of Busy that occurs during the months of February, March, April and May. It occurs between noon and 1 P.M Pacific Time on February 2nd. The Publishing interval is 3 months. The time of Publishing is 12 A.M. on Feb 25th UTC.

The value is determined by using the following calculations:

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

End time
<i>Number of days before the scheduled date in that particular month = 1</i>
<i>End hour = 1 P.M. Pacific Time = 13 on a 24-hour clock + 8 for UTC conversion = 21</i>
Note:
<i>+8 is the difference between UTC and Pacific Time Zone. The adjustment is</i>

End time

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 may change the date of the start time or end time.

The Multi-Value Binary will have one binary value. The binary value will have the following 4 bytes, encoded in little-endian; the first two bytes for Start Time and the second two bytes for End Time: 0x50, 0x0A, 0x8C, 0x50.

4.4.1 PidTagSchdinfoFreebusyBusy Calculation with two non-consecutive events

Let's say there are two events with Free/Busy type of Busy, in the month of February.

February 2nd (times are in Pacific Time)
12 PM to 1 PM
3PM to 4PM

The PtypMultipleBinary will have one binary value. The binary value will have the following 8 bytes. 0x50, 0x0A, 0x8C, 0x0A, 0x04, 0x0B, 0x40, 0x0B.

4.4.2 PidTagSchdinfoFreebusyBusy Calculation with two consecutive events

If there are two consecutive events with the same Free/Busy type, the two times are merged together. For example, if there are two events of type Busy.

February 2nd (times are in Pacific Time)
12 PM to 1 PM
1PM to 2 PM

There will be only one block with Start Time 12 PM and End Time 2 PM. The PtypMultipleBinary will have one binary value. The binary value will have the following 4 bytes: 0x50, 0x0A, 0xC8, 0x0A.

4.4.3 PidTagSchdinfoFreebusyBusy Calculation with events in multiple months

Let's say there are two events of type Busy in the month of February and April each. There are no events in March.

February 2nd (times are in Pacific Time)
12 P.M. to 1 P.M.
1 P.M. to 2 P.M.

April 2nd

12 P.M. to 1 P.M.
3 P.M. to 4 P.M.

The `PtypMultipleBinary` will have two binary values.

The first binary value will have the following 4 bytes: 0x50, 0x0A, 0xC8, 0x0A

The second binary value will have the following 8 bytes: 0x14, 0x0A, 0x50, 0x0A, 0xC8, 0x0A, 0x04, 0x0B.

Note that the time difference between UTC and PST in April is +7 hours and hence the values are calculated by adding 7 hours to PST time.

4.4.4 `PidTagSchdinfoFreebusyBusy` calculation when an event is spread across multiple months

Consider an example where an event starts on Dec 25th 2007, 9 A.M. and ends on Dec 25th 2008, 10 A.M. This event must be considered as a series of events that start and end as follows:

Dec 25th 2007, 9 A.M. - Dec 31st 2007, 12:00 A.M.
Jan 1st 2008, 12:00 A.M. - Jan 31st 2008, 12:00 A.M.

....
....

Dec 1st 2008, 12:00 A.M. - Dec 25th, 10:00 A.M.

The values are calculated as described in the examples above.

If the Publishing interval is 12 months and the time of Publishing is 12 A.M. Dec 25th, 2007 UTC then there would be 13 binary values in both `PidTagSchdinfoMonthsFreebusy` and `PidTagSchdinfoFreebusyBusy` properties; one for Dec 2007 and one for each month in 2008.

If the Publishing interval is 12 months and the time of Publishing is 12 A.M. Feb 25th, 2007, UTC, then there would be 11 binary values.

If the Publishing interval is 1 month and the time of Publishing is 12 A.M. Feb 25th, 2007, UTC, then there would be 2 binary values, for the months of Feb and March.

4.5 `PidTagSchdinfoFreebusyMerged` Calculation

The `PidTagSchdinfoFreeBusyMerged` property includes events of type Busy and Out of Office. For example, if a user has an event from 1 P.M. to 2 P.M. of type Busy and another from 4 P.M. to 5 P.M. of type Out of Office, then `PidTagSchdinfoFreeBusyMergedProperty` will contain two binary values.

If there are consecutive or overlapping scheduled events of type Busy and Out Of Office, they will be merged into one block. For example, if a user has an event from 1 P.M. to 2 P.M., of type Busy and another from 1:30 P.M. to 3 P.M., of type Out Of Office, the `PidTagSchdinfoFreeBusyMerged` property will contain one binary value with start time of 1 P.M. and an end time of 3 P.M.

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to this protocol. General security considerations pertaining to the underlying [MS-OXOCMSG] protocol apply.

5.2 Index of Security Parameters

None.

6 Appendix A: Office/Exchange Behavior

The information in this specification is applicable to the following versions of Microsoft Office and Microsoft Exchange:

- Office 2003 with Service Pack 3 applied
- Exchange 2003 with Service Pack 2 applied
- Office 2007 with Service Pack 1 applied
- Exchange 2007 with Service Pack 1 applied

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Microsoft Office and Microsoft Exchange behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies Office and Exchange do not follow the prescription.

<1> Public folders are supported by default on Exchange 2003. But they may have to be configured explicitly on an Exchange 2007 server.

<2 > Exchange 2003 does not support Availability Service.

<3> Outlook 2003 and 2007, will set this PidTagGatewayNeedsToRefresh to 1. This property is not read by Outlook 2003 or 2007 and Exchange 2003 or 2007.

<4> PidTagSchdInfoResourceType from Free/Busy message is read from and written to by Outlook 2003 and 2007, but the value is not used.

<5> PidTagSchdinfoFreebusy: This property might be set on Delegate Info message by lower-level versions of Outlook 2003. This property is deprecated and MUST not be read from or written to.

Index

Appendix A

Office/Exchange behavior, 36

Introduction, 5

Applicability statement, 8

Glossary, 5

Prerequisites/Preconditions, 8

Protocol overview (synopsis), 7

References, 6

Relationship to other protocols, 8

Standards assignments, 8

Vendor-extensible fields, 8

Versioning and capability statement, 8

Messages, 8

Message syntax, 8

Transport, 8

Protocol details, 13

Client details, 13

Protocol examples, 16

Finding free/busy messages using e-mail addresses, 18

Free/Busy message, 16

PidTagSchdinfoFreebusyBusy calculation, 33

PidTagSchdinfoFreebusyMerged calculation, 35

PidTagSchinfoMonthsBusy calculation, 32

References

Informative references, 7

Normative references, 6

Security, 36

Index of security parameters, 36

Security considerations for implementers, 36