

[MS-OXOSRCH]: Search Folder List Configuration 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>). 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.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it. A protocol specification 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.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability.
Microsoft Corporation	April 25, 2008	0.2	Revised and updated property names and other technical content.
Microsoft Corporation	June 27, 2008	1.0	Initial Release.
Microsoft Corporation	August 6, 2008	1.01	Revised and edited technical content.

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References	5
1.2.1	Normative References	5
1.2.2	Informative References	6
1.3	Protocol Overview	6
1.4	Relationship to Other Protocols	6
1.5	Prerequisites/Preconditions	7
1.6	Applicability Statement	7
1.7	Versioning and Capability Negotiation	7
1.8	Vendor-Extensible Fields	7
1.9	Standards Assignments	7
2	Messages	7
2.1	Transport	7
2.2	Message Syntax	7
2.2.1	Definition Message	7
2.2.1.1	Common Properties	8
2.2.1.1.1	PidTagMessageClass	8
2.2.1.1.2	PidTagDisplayName	8
2.2.1.2	Additional Properties	8
2.2.1.2.1	PidTagSearchFolderId	8
2.2.1.2.2	PidTagSearchFolderTemplateId	8
2.2.1.2.3	PidTagSearchFolderTag	8
2.2.1.2.4	PidTagSearchFolderLastUsed	8
2.2.1.2.5	PidTagSearchFolderExpiration	8
2.2.1.2.6	PidTagSearchFolderStorageType	8
2.2.1.2.7	PidTagSearchFolderEfpFlags	9
2.2.1.2.8	PidTagSearchFolderDefinition	9
2.2.1.2.9	PidTagSearchFolderRecreateInfo	15
2.2.2	Search Folder Container	15
2.2.2.1	Common Properties	15
2.2.2.1.1	PidTagContainerClass	15
2.2.2.1.2	PidTagExtendedFolderFlags	15
2.2.3	Search Templates	16
2.2.3.1	Unread Messages	16
2.2.3.2	Marked for Followup	17
2.2.3.3	Unread or Marked for Followup	17
2.2.3.4	Important Mail	17
2.2.3.5	Conversations	18
2.2.3.6	From a Specific Person	18
2.2.3.7	Sent Directly to Me	18
2.2.3.8	Sent to a Specific Distribution List	19

2.2.3.9	Large Messages	19
2.2.3.10	Old Mail	19
2.2.3.11	With Attachments.....	20
2.2.3.12	With Specific Words	20
2.2.3.13	Categorized	21
2.2.3.14	Custom	21
2.2.4	Search Folder Definition Messages and Search Folder Containers.....	21
3	<i>Protocol Details</i>	22
3.1	Client Details	22
3.1.1	Abstract Data Model	22
3.1.2	Timers	22
3.1.3	Initialization	22
3.1.4	Higher-Layer Triggered Events.....	23
3.1.4.1	Creating a Search Folder	23
3.1.4.1.1	Obtaining Data.....	23
3.1.4.1.2	Creating a New Search Folder Container.....	23
3.1.4.1.3	Creating a New Definition Message.....	24
3.1.4.2	Opening a Search Folder.....	26
3.1.4.3	Modifying a Search Folder	26
3.1.4.4	Deleting a Search Folder	26
3.1.4.5	Current Time Exceeds PidTagSearchFolderExpiration	26
3.1.5	Message Processing Events and Sequencing Rules	26
3.1.6	Timer Events.....	26
3.1.7	Other Local Events.....	26
4	<i>Protocol Examples</i>	27
4.1	Search Folder Message Object	27
5	<i>Security</i>	28
5.1	Security Considerations for Implementers.....	28
5.2	Index of Security Parameters.....	28
6	<i>Appendix A: Office/Exchange Behavior</i>	28
	<i>Index</i>	30

1 Introduction

This document specifies the Search Folder List Configuration protocol used by the client to create, read, and execute search folders. A search folder is used to query for items that match specified criteria.

1.1 Glossary

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

AddressList
associated message
big-endian
binary large object (BLOB)
Coordinated Universal Time (UTC)
EntryList
folder
Folder objects
GUID
mailbox
message
message database (MDB)
Message object
property
search folder
search folder container
search folder definition message
search folder criteria
Unicode

The following data types are defined in [MS-DTYP]:

Boolean
byte
FILETIME
INT32
ULONG

The following terms are specific to this document:

active search folder: A search folder where the **search folder container** exists and is up to date with the correct search criteria.

definition message: See **search folder definition message**.

inactive search folder: A search folder that does not have a search folder container .

skip block: The block in a **binary large object (BLOB)** that acts as padding, reserving space that can be used by future versions to insert data. The block consists of a **ULONG** describing how many additional ULONGs to skip ahead.

TickCount: The number of milliseconds since a system was started.

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", June 2008.

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

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

[MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol Specification", June 2008.

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

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

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

[MS-OXOABK] Microsoft Corporation, "Address Book Object Protocol Specification", June 2008.

[MS-OXOCFG] Microsoft Corporation, "Configuration Information Protocol Specification", June 2008.

[MS-OXOMSG] Microsoft Corporation, "E-mail Object Protocol Specification", June 2008.

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

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

[MS-OXPROTO] Microsoft Corporation, "Office Exchange Protocols Overview", June 2008.

[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

[MSDN-FAIT] Microsoft Corporation, "Folder Associated Information Tables", <http://go.microsoft.com/fwlink/?LinkID=92924>.

1.3 Protocol Overview

A **search folder** provides a means of querying for items that match certain criteria. To the user, a search folder appears in the client as a normal **folder** that populates itself when opened. A search folder uses one of the standard templates or a custom search created by the user to include specific search criteria.

This protocol enables the client to create, read, and execute search folders. To create a search folder, the client collects the data used to define the search criteria, creates a **search folder container** to contain the search results, and creates a **search folder definition message** to persist the search folder. Search folder criteria is persisted on the server, although it is not necessary for the server to understand the criteria. **Search folder criteria** is saved as an **associated message** in a hidden folder outside the root **mailbox** and is not directly visible to the end user.

1.4 Relationship to Other Protocols

The Search Folder List Configuration protocol specification relies on the following:

- An understanding of the Office Exchange Protocols Overview as specified in [MS-OXPROTO].
- An understanding of messaging as specified in [MS-OXCMSG] and of **Message objects** as specified in [MS-OXOMSG].
- An understanding of using **folders** as specified in [MS-OXCFOLD] and of **Folder objects** as specified in [MS-OXOSFLD].
- An understanding of properties as specified in [MS-OXPROPS] and of setting properties on **message** and Folder objects as specified in [MS-OXCPRPT].

- An understanding of the **message database (MDB)** as specified in [MS-OXCSTOR] and of manipulating tables in the **message database** as specified in [MS-OXCTABL].

1.5 Prerequisites/Preconditions

This protocol specification assumes that the messaging client has logged on to the **message database**, with the ability to open tables and read/write **Message objects**, **Folder objects**, and their properties.

1.6 Applicability Statement

A client can use this protocol to save user-created search queries that can be invoked again at a later time.

1.7 Versioning and Capability Negotiation

The **binary large object (BLOB)** stored in **PidTagSearchFolderDefinition** contains a version for that BLOB format.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The properties specified in this protocol are set by and returned from a server message, as specified in [MS-OXCMSG], or from a **folder** [MS-OXCFOLD] that uses the underlying Property and Stream Object protocol, as specified in [MS-OXCPRPT].

2.2 Message Syntax

The remaining subsections within section 2.2 specify the format of **properties** that are specific to this protocol.

2.2.1 Definition Message

Search folder definition messages are stored as **associated messages** (as specified in [MS-OXCMSG]) in the associated information table [as specified in MSDN-FAIT] of the Common Views **folder** (as specified in [MS-OXOSFLD]) within a **message database**. The **definition message** is how a **search folder** is persisted; a search folder ceases to exist if its definition message is deleted. For more details about how definition messages relate to search folders and **search folder containers**, see section 2.2.4.

Search folder definition messages possess additional properties that describe the search folder criteria. These properties are described in the remaining subsections of section 2.2.1.

2.2.1.1 Common Properties

These properties are common to most **messages**, and their values do not specifically relate to **search folder** functionality. For more details about these properties, see [MS-OXPROPS].

2.2.1.1.1 PidTagMessageClass

The client uses **PidTagMessageClass** as a means of identifying a **search folder definition message**. A search folder definition message **MUST** set this **property** value to: IPM.Microsoft.WunderBar.SFInfo.

2.2.1.1.2 PidTagDisplayName

PidTagDisplayName is the name of the **search folder**. The client **SHOULD** use this **property** value as the display name of the **Folder object**.

2.2.1.2 Additional Properties

The following properties are specifically indicated for **search folders**, and they contain the information that defines a search folder.

2.2.1.2.1 PidTagSearchFolderId

PidTagSearchFolderId is a **GUID** that identifies the **search folder**. This GUID is used to tie the **definition message** to the corresponding **search folder container**. For more details, see section 2.2.4.

2.2.1.2.2 PidTagSearchFolderTemplateId

PidTagSearchFolderTemplateId is the ID of the template that is being used for the search. For more details about search templates, see section 2.2.3.

2.2.1.2.3 PidTagSearchFolderTag

PidTagSearchFolderTag is used to synchronize this **definition message** with the matching **search folder container**. It is changed when this definition message is changed. It **MUST** change each iteration, but it **MAY NOT** be unique. For more details, see section 2.2.4.

2.2.1.2.4 PidTagSearchFolderLastUsed

PidTagSearchFolderLastUsed is the last time the **folder** was accessed. It **MUST** be formatted as the number of minutes since midnight (**UTC**) January 1, 1601.

2.2.1.2.5 PidTagSearchFolderExpiration

PidTagSearchFolderExpiration is the time at which the **search folder container** will be stale and **SHOULD** be updated or recreated. It **MUST** be formatted as the number of minutes since midnight (**UTC**) January 1, 1601.

2.2.1.2.6 PidTagSearchFolderStorageType

PidTagSearchFolderStorageType contains flags that specify the **BLOB** data that appears in the **PidTagSearchFolderDefinition** property. These flags are duplicated inside the BLOB. For more details about flag fields, see the definition of A through L in section 2.2.1.2.8. The specific flags to use depends on the template; section 2.2.3 specifies the correct flags for each template definition.

The definitions of the flags are specified in network order in section 2.2.1.2.8. This property is stored as a 4-byte integer. The following table shows the flags in **big-endian** order.

Flag	Big-endian bit
B	0x00000040
C	0x00000020
D	0x00000010
E	0x00000008
F	0x00000004
G	0x00000002
H	0x00000001
I	0x00008000
J	0x00004000
K	0x00002000
L	0x00001000

2.2.1.2.7 PidTagSearchFolderEfpFlags

PidTagSearchFolderEfpFlags SHOULD contain extended **folder** flags that apply to the **search folder container** for this **search folder**. Specifically, it SHOULD contain the flags in the **PidTagExtendedFolderFlags** property, **ExtendedFlags** sub-property, field **b** for that folder, as specified in [MS-OXOCFG]. <1>

2.2.1.2.8 PidTagSearchFolderDefinition

PidTagSearchFolderDefinition contains data that specifies the search criteria.

The structure of the **BLOB** contained in **PidTagSearchFolderDefinition** is specified as follows. The specific content of each field is dependent upon the template ID that is specified in **PidTagSearchFolderTemplateId**. For details, see section 2.2.3.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version field																															
A	B	C	D	E	F	G	H	I	J	K	L	(ignored)																			
Numerical search field																															

Text search field length	Text search field length extended (optional)	Text search field (optional, variable)
Text search field (optional, variable)		
Skip block (variable)		
Deep search field		
Folder list field 1 length	Folder list field 1 length extended (optional)	Folder list field 1 (optional, variable)
Folder list field 1 (optional, variable)		
Folder list field 2 byte count		
Folder list field 2 (optional, variable)		
Address list field (optional, variable)		
Skip block (variable)		
SRestriction field (optional, variable)		
Advanced search field (optional, variable)		
Skip block (variable)		

Version field: This field SHOULD specify the version of this BLOB definition <2>.

Flag fields:

- A. Ignored. This bit SHOULD be zero (0).
- B. MAY indicate whether **Folder list field 2** contains data that defines the search criteria.
- C. SHOULD indicate whether **Folder list field 1** contains data that defines the search criteria.
- D. MUST indicate whether **Advanced search field** exists and contains the data that defines the search criteria.
- E. MUST indicate whether **SRestriction field** exists and contains the data that defines the search criteria.

- F. MUST indicate whether **Address list field** exists and contains addresses that define the search criteria.
- G. SHOULD indicate that there is data in the **Text search field** that defines the search criteria.
- H. SHOULD indicate that there is data in the **Numerical search field** that defines the search criteria.
- I. MAY indicate that this **search folder** is not an **active search folder**.
- J. Indicates that this search folder SHOULD be refreshed daily. That is, when updating the **PidTagSearchFolderExpiration** value, it SHOULD be set one day in the future.
- K. Indicates that this search folder SHOULD be refreshed weekly. That is, when updating the **PidTagSearchFolderExpiration** value, it SHOULD be set one week in the future.
- L. Indicates that this search folder SHOULD be refreshed monthly. That is, when updating the **PidTagSearchFolderExpiration** value, it SHOULD be set one month in the future.

Numerical search field: This field MUST contain a 4-byte integer that is used by some templates to define the search criteria. If the template specified in **PidTagSearchFolderTemplateId** does not require it (and therefore does not set the **H** flag in this BLOB), this field is ignored.

Text search field length: This field MUST contain a single byte integer defining the length of the **Text search field**. The length is the number of characters. If **Text search field** is an empty string, this field MUST be zero (0). If **Text search field** is longer than 254 characters, this field MUST be 255.

Text search field length extended: This field MUST NOT exist if the value of **Text search field length** is less than 255. If **Text search field length** is 255, this field MUST contain a 2-byte integer defining the length of the **Text search field**. The length is the number of characters.

Text search field: If **Text search field length** is zero (0), this field MUST NOT exist. If **Text search field length** is non-zero, this field MUST contain a string used by some templates to define the search criteria. If the template specified in **PidTagSearchFolderTemplateId** does not require it (and therefore does not set the **G** flag in this BLOB), this field is ignored. It MUST NOT be longer than 65,536 characters.

Skip block: This MUST contain at least a 4-byte integer that defines how many bytes to skip ahead. This is for future versions to insert new data. If there is no data to skip, the value of this

field MUST be zero (0), and its size will be 4 bytes. If there is data to skip, the size of this field MUST be 4 bytes plus the size of the data to be skipped.

Deep search field: This field MUST contain a 4-byte integer that represents a **Boolean** value. It defines whether the search criteria SHOULD include sub-folders. This MUST be set to zero (0) for false, and it MUST be set to a non-zero value for true.

Folder list field 1 length: This field MUST contain a single byte integer that defines the length of **Folder list field 1**. The length is the number of characters. If **Folder list field 1** is an empty string, this field MUST be zero (0). If **Folder list field 1** is longer than 254 characters, this field MUST be 255.

Folder list field 1 length extended: This field MUST NOT exist if the value of **Folder list field 1 length** is less than 255. If **Folder list field 1 length** is 255, this field MUST contain a 2-byte integer that defines the length of the **Folder list field 1**. The field length is the number of characters.

Folder list field 1: If **Folder list field 1 length** is zero (0), this field MUST NOT exist. If **Folder list field 1 length** is non-zero, this field MUST contain a string that is used by some templates to define the **folder** names to search, delimited by semicolons. If the template specified in **PidTagSearchFolderTemplateId** does not require it (and therefore does not set the **C** flag in this BLOB), this field is ignored. It MUST NOT be longer than 65,536 characters.

*Folder list field 1 = [folder_name] [“;” folder_name]**

Folder list field 2-byte count: This field MUST contain a 4-byte integer that defines the number of bytes in the next field, **Folder list field 2**. If the template specified in **PidTagSearchFolderTemplateId** does not require **Folder list field 2** (and therefore does not set the **B** flag in this BLOB), this field MUST be set to zero (0).

Folder list field 2: If **Folder list field 2 byte count** is greater than zero (0), this field MUST exist and MUST contain an **EntryList** [MS-OXCDATA] that is used by some templates to define the search criteria. If **Folder list field 2 byte count** is equal to zero (0), this field MUST NOT exist.

Address list field: If the template specified in **PidTagSearchFolderTemplateId** requires this field (and sets the **F** flag in this BLOB), this field MUST exist and MUST contain an **AddressList** of addresses used by some templates to define the search criteria. If the template does not require it (and therefore does not set the **F** flag in this BLOB), this field MUST NOT exist. This field is formatted as follows:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
AddressCount																															
Addresses (Variable)																															
...																															

AddressCount: This field MUST contain a 4-byte unsigned integer that gives the number of addressees to follow.

Addresses: This field MUST contain **AddressCount** and **AddressEntry** structures. An AddressEntry is a set of properties that represent one addressee, and is specified as follows:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
PropertyCount																															
Pad																															
Values (Variable)																															
...																															

PropertyCount: This field MUST contain a 4-byte unsigned integer that gives the number of values to follow.

Pad: Ignored, this 4-byte field MAY be any value.

Values: This field MUST contain **PropertyCount** properties that specify the properties of an addressee. These properties are each formatted as follows:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
PropertyType																PropertyId															
PropertyValue (variable)																															

PropertyType: This field MUST contain a 2-byte integer that specifies the type of **PropertyValue**.

PropertyId: This field MUST contain a 2-byte integer that identifies the data in **PropertyValue**.

PropertyType and **PropertyId** together specify a property. Each **AddressEntry** MUST include the following nine properties: **PidTagDisplayName**, **PidTagPrimarySmtpAddress**, **PidTagAddressType**, **PidTagEntryId**, **PidTagObjectType**, **PidTagDisplayType**, **PidTagDisplayTypeEx**, **PidTagEmailAddress**, and **PidTagRecipientType**. Other **PropertyIds** MAY be included. See [MS-OXPROPS] for more details and a complete list of properties.

PropertyValue: The contents of this field depend on **PropertyType**. It MUST contain appropriately formatted data as specified in the following table.

PropertyType name	Property Type value	PropertyValue specification	Alternate names
PtypInteger32	0x0003	4 bytes, a 32-bit integer. [MS-DTYP]: INT32	PT_LONG, PT_I4
PtypErrorCode	0x000A	4 bytes, a 32-bit integer encoding error information	PT_ERROR
PtypBoolean	0x000B	2 bytes, a 16-bit integer, zero (0) is false, non-zero is true.	PT_BOOLEAN
PtypString	0x001F	Variable size, a 16-bit byte count followed by a string of Unicode characters in UTF-16LE encoding with terminating null character (2 bytes of zero).	PT_UNICODE
PtypString8	0x001E	Variable size, a 16-bit byte count followed by a string of multi-byte characters in externally specified encoding with terminating null character (single 0 byte).	PT_STRING8
PtypTime	0x0040	8 bytes, a 64-bit integer representing the number of 100-nanosecond intervals since January 1, 1601. [MS-DTYP]: FILETIME	PT_SYSTIME
PtypBinary	0x0102	Variable size, a 16-bit byte count followed by that many bytes.	PT_BINARY
PtypMultipleString8	0x101E	Variable size, a 16-bit byte count followed by that many PtypString8 values (as specified above)	PT_MV_STRING8

PropertyType name	Property Type value	PropertyValue specification	Alternate names
PtypMultipleBinary	0x1102	Variable size, a 16-bit byte count followed by that many PtypBinary values (as specified above).	PT_MV_BINAR Y

Skip block: See **Skip block**, previously in this section.

SRestriction field: If the template specified by **PidTagSearchFolderTemplateId** requires this field (and sets the **E** flag in this BLOB), this field **MUST** exist and **MUST** contain an **SRestriction** that explicitly defines the search criteria. If the template does not require it (and therefore does not set the **E** flag in this BLOB), this field **MUST NOT** exist.

Advanced search field: If the template specified by **PidTagSearchFolderTemplateId** requires this field (and sets the **D** flag in this BLOB), this field **MUST** exist and **MUST** contain a BLOB representing the advanced search dialog. If the template does not require it (and therefore does not set the **D** flag in this BLOB), this field **MUST NOT** exist.

Skip block: See **Skip block**, previously in this section.

2.2.1.2.9 PidTagSearchFolderRecreateInfo

This **property** **MAY NOT** be used.

2.2.2 Search Folder Container

While the **definition message** persists, the definition of a **search folder**, a **search folder container** only exists if the search folder is an **active search folder**. If a search folder is inactive, the search folder container will not exist. For more details about active and **inactive search folders**, and the relationship between definition messages and search folder containers, see section 2.2.4.

A search folder container is created and modified as described in sections 3.1.4.1 and 3.1.4.3. A search folder container uses search criteria, a feature that is not unique to this protocol. These folders also require the use of several common properties.

2.2.2.1 Common Properties

These properties are not unique to **search folder** containers, but their values **MUST** be correct to function in the Search Folder protocol.

2.2.2.1.1 PidTagContainerClass

PidTagContainerClass **MUST** be set to "IPF.Note" for the folder to be recognized as a **search folder container**.

2.2.2.1.2 PidTagExtendedFolderFlags

PidTagExtendedFolderFlags is a **BLOB** specified in [MS-OXOCFG]. It **MUST** contain equivalent data to the **PidTagSearchFolderId** of the **definition message** and **PidTagSearchFolderTag** properties of the definition message. For more details about how these properties work together, see section 2.2.4.

The **Folder flags** BLOB specified in [MS-OXOCFG] defines a number of sub-properties, each consisting of an Id, Cb, and Data block. These sub-properties include **SearchFolderID**, which corresponds to the **PidTagSearchFolderId property** of the definition message.

The Search Folder List Configuration protocol extends that BLOB by defining another sub-property, **SearchFolderTag**. This sub-property corresponds with the **PidTagSearchFolderTag** property of the definition message and is specified as follows:

Sub-property field	Value
Id	0x03
Cb	0x04
Data	A 4-byte number that matches the PidTagSearchFolderTag property of the definition message.

2.2.3 Search Templates

Search folder criteria is specified by a template <3>. The **PidTagSearchFolderTemplateId property** on the **message** that defines the search folder identifies its corresponding template. In addition to defining search criteria, a template also defines **folders** to exclude from the search, defines items to exclude from the search, and specifies the value of **PidTagSearchFolderStorageType**. For more details about the folders that are excluded from the search, see [MS-OXOSFLD]. For more details about the item types that are excluded from the search, see the following:

- Appointments [MS-OXOCAL]
- Contacts [MS-OXOCNTC]
- Distribution lists [MS-OXOABK]
- Journal items [MS-OXOJRNL]
- Sticky notes [MS-OXONOTE]
- Tasks [MS-OXOTASK]

The flags set in **PidTagSearchFolderStorageType** (as specified in section 2.2.1.2.6) and in the second field of the binary data in **PidTagSearchFolderDefinition** (as specified in section 2.2.1.2.8) vary according to template. The fields required in the binary data of **PidTagSearchFolderDefinition** also vary according to template. The remaining subsections within section 2.2.3 define the templates <4>, including their requirements for **PidTagSearchFolderStorageType** flags and **PidTagSearchFolderDefinition** fields.

2.2.3.1 Unread Messages

ID	2
Folders excluded	Failed sync items, deleted items, junk mail, outbox,

	drafts
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x0000048 (B and E)
PidTagSearchFolderDefinition fields	Folder list field 2: The EntryList of folders to search. This will include all mailbox folders, except the ones specifically excluded for this template. SRestriction field: An SRestriction describing Unread Messages.

2.2.3.2 Marked for Followup

ID	3
Folders excluded	Failed sync items, deleted items, junk mail, outbox
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x0000048 (B and E)
PidTagSearchFolderDefinition fields	Folder list field 2: The EntryList of folders to search. This will include all mailbox folders, except the ones specifically excluded for this template. SRestriction field: An SRestriction describing messages that are marked for follow-up.

2.2.3.3 Unread or Marked for Followup

ID	4
Folders excluded	Failed sync items, deleted items, junk mail, outbox
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x0000048 (B and E)
PidTagSearchFolderDefinition fields	Folder list field 2: The EntryList of folders to search. This includes all mailbox folders, except the ones specifically excluded for this template. SRestriction field: An SRestriction describing the unread message AND messages marked for follow-up.

2.2.3.4 Important Mail

ID	5
Folders excluded	Failed sync items, deleted items, junk mail, outbox, drafts
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x0000048 (B and E)
PidTagSearchFolderDefinition fields	Folder list field 2: The EntryList of folders to search. This includes all mailbox folders, except the ones

	specifically excluded for this template. SRestriction field: SRestriction messages that have been marked as important.
--	---

2.2.3.5 Conversations

ID	6
Folders excluded	Failed sync items, deleted items, junk mail, outbox, drafts
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x0000004e (B, E, F, and G)
PidTagSearchFolderDefinition fields	Folder list field 2: The EntryList of folders to search. This includes all mailbox folders, except the ones specifically excluded for this template. SRestriction field: An SRestriction describing messages to and from people in the Address list field . Address list field: An AddressList of the people by which to filter conversations. Text search field: A text list of the people by which to filter conversations.

2.2.3.6 From a Specific Person

ID	7
Folders excluded	Failed sync items, deleted items, junk mail, outbox, drafts, sent items
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x0000004e (B, E, F, and G)
PidTagSearchFolderDefinition fields	Folder list field 2: The EntryList of folders to search. This will include all mailbox folders, except the ones specifically excluded for this template. SRestriction field: An SRestriction describing messages from people in the Address list field . Address list field: An AddressList of the people by which to filter received messages. Text search field: A text list of the people by which to filter received messages.

2.2.3.7 Sent Directly to Me

ID	8
Folders excluded	Failed sync items, deleted items, junk mail, outbox, drafts, sent items
Item Types excluded	Appointments, contacts, distribution lists, journal items,

	sticky notes, tasks
PidTagSearchFolderStorageType	0x00000048 (B and E)
PidTagSearchFolderDefinition fields	Folder list field 2: The EntryList of folders to search. This will include all mailbox folders, except the ones specifically excluded for this template. SRestriction field: An SRestriction describing messages that include the user as a recipient.

2.2.3.8 Sent to a Specific Distribution List

ID	9
Folders excluded	Failed sync items, deleted items, junk mail
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks,
PidTagSearchFolderStorageType	0x0000004e (B , E , F , and G)
PidTagSearchFolderDefinition fields	Folder list field 2: The EntryList of folders to search. This will include all mailbox folders, except the ones specifically excluded for this template. SRestriction field: An SRestriction describing messages to distribution lists in the Address list field . Address list field: An AddressList of the distribution lists by which to filter messages. Text search field: A text list of the distribution lists by which to filter messages.

2.2.3.9 Large Messages

ID	10
Folders excluded	Failed sync items, deleted items
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x0000004b (B , E , G , and H)
PidTagSearchFolderDefinition fields	Folder list field 2: The EntryList of folders to search. This includes all mailbox folders, except the ones specifically excluded for this template. SRestriction field: An SRestriction describing messages that are larger than <i>n</i> kilobytes, where <i>n</i> is the number specified in Number search field . Text search field: The size as a string, including units. Example: 99 KB Number search field: The size to filter by, in kilobytes.

2.2.3.10 Old Mail

ID	11
Folders excluded	Failed sync items, deleted items, junk mail

Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x00004049 (B, E, H, and J) This SHOULD include J (daily), to indicate that these folders SHOULD update daily. It MAY include K or L instead of J .
PidTagSearchFolderDefinition fields	<p>Folder list field 2: The EntryList of folders to search. This will include all mailbox folders, except the ones specifically excluded for this template.</p> <p>SRestriction field: An SRestriction describing messages that are older than the age specified in Number search field.</p> <p>Number search field: The age. This is formatted as follows (in big-endian byte order):</p> <p>HIWORD: These 2 bytes describe the units.</p> <p>0x0000: Days</p> <p>0x0001: Weeks</p> <p>0x0002: Months</p> <p>LOWORD: These 2 bytes describe the amount.</p> <p>For example, 0x0001002a would mean an age of 42 weeks.</p>

2.2.3.11 With Attachments

ID	12
Folders excluded	Failed sync items, deleted items, junk mail, outbox, drafts
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x00000048 (B and E)
PidTagSearchFolderDefinition fields	<p>Folder list field 2: The EntryList of folders to search. This will include all mailbox folders, except the ones specifically excluded for this template.</p> <p>SRestriction field: An SRestriction describing messages that have file attachments.</p>

2.2.3.12 With Specific Words

ID	14
Folders excluded	Failed sync items, deleted items, junk mail, outbox, drafts
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x0000004a (B, E, and G)
PidTagSearchFolderDefinition fields	<p>Folder list field 2: The EntryList of folders to search. This will include all mailbox folders, except the ones specifically excluded for this template.</p>

	<p>SRestriction field: An SRestriction describing messages that contain words specified in the Text search field.</p> <p>Text search field: The words for which to search.</p>
--	--

2.2.3.13 Categorized

ID	15
Folders excluded	failed sync items, deleted items, junk mail
Item Types excluded	appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x00000048 (B and E) (any category) 0x0000004a (B , E , and G) (specific categories)
PidTagSearchFolderDefinition fields	<p>Folder list field 2: The EntryList of folders to search. This will include all mailbox folders, except the ones specifically excluded for this template.</p> <p>SRestriction field: SRestriction describing messages that have a category. If G is specified in PidTagSearchFolderStorageType, the SRestriction specifies messages that have a category matching the string in Text search field.</p> <p>Text search field: The category text to match, if G is specified.</p>

2.2.3.14 Custom

ID	1
Folders excluded	Failed sync items
Item Types excluded	Appointments, contacts, distribution lists, journal items, sticky notes, tasks
PidTagSearchFolderStorageType	0x00000010 (D) This template MAY contain other flags, but MUST include this flag.
PidTagSearchFolderDefinition fields	Advanced search field: A client specific binary serialization of the advanced find dialog.

2.2.4 Search Folder Definition Messages and Search Folder Containers

A **search folder** exists only if it has a **definition message**. Each definition message MUST have a **GUID**, stored in the **PidTagSearchFolderId** property. This GUID is fixed and MUST NOT change. **Search folder containers** also have a GUID, stored in the **SearchFolderId** sub-property of the **PidTagExtendedFolderFlags** property. A search folder container MUST have the same **SearchFolderId** as the **PidTagSearchFolderId** of its corresponding definition message. If it does not, that search folder container SHOULD be deleted. This is how a definition message is connected to a corresponding search folder container. If both the search folder container and the definition message exist (with search

folder container **SearchFolderId** matching definition message **PidTagSearchFolderId**), the search folder is active. If the search folder container does not exist, the search folder is inactive. The client can make an **inactive search folder** active by creating the search folder container as specified in section 3.1.4.1.2.

To keep the search folder container updated, the client uses the **PidTagSearchFolderTag** property on the **search folder definition message** and the corresponding **SearchFolderTag** property stored in the **PidTagExtendedFolderFlags** property of the search folder container. When a search folder container is created, this **SearchFolderTag** property MUST have the value as the **PidTagSearchFolderTag** property on its corresponding definition message. If the definition message is changed, the value of **PidTagSearchFolderTag** MUST change. If this scenario occurs, the corresponding search folder container would have an unmatched **SearchFolderTag** value, so the client MUST update the physical search folder to set the **SearchFolderTag** value to match the value of **PidTagSearchFolderTag** on the definition message. For more details, see section 3.1.4.3.

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

This section specifies 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.

All data necessary for the search folder protocol is persisted in the **search folder definition message**, the format of which is specified in section 2.2.1. This data is used to create and maintain search folder containers, as specified in section 2.2.2.

3.1.2 Timers

None.

3.1.3 Initialization

For this protocol, there is no specific initialization, as all **messages** use the existing connection assumed in section 1.5.

To display the list of existing **search folders** to the user at start-up, the client:

- SHOULD return the **associated messages** (as specified in [MS-OXCMSG]) in the Common Views **folder** (as specified in [MS-OXOSFLD]) from the message database (as specified in [MS-OXCSTOR]). Each message with **PidTagMessageClass** set to "IPM.Microsoft.Wunderbar.SFInfo" is a **search folder definition message**.

- SHOULD load the **Folder objects** (as specified in [MS-OXCFOLD]) in the Finder folder (as specified in [MS-OXOSFLD]). Each folder with **property PidTagFolderType** set to FOLDER_SEARCH (0x00000002) and **PidTagContainerClass** set to "IPF.Note" is a **search folder container**.
- SHOULD examine the **PidTagExtendedFolderFlags** property, **SearchFolderTag** sub-property (as specified in [MS-OXOCFG]) of each **search folder container**. If that **GUID** matches the value of a **search folder definition message** **PidTagSearchFolderId** property, that Folder object is associated with that search message.
- MUST delete any search folder container that does not have a corresponding search folder definition message.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Creating a Search Folder

To create a **search folder**, the client MUST complete three steps:

- Obtain data to define the search criteria.
- Create the **search folder container** to contain the search results.
- Create the **definition message** to persist the **search folder**.

The remaining subsections of section 3.1.4.1 specify details for each of these steps.

3.1.4.1.1 Obtaining Data

The client SHOULD do the following:

- Obtain a name for the **search folder**.
- Identify which template to use. For more details about search templates, see section 2.2.3.
- Obtain specific data needed by the chosen template.

3.1.4.1.2 Creating a New Search Folder Container

In the "Finder" **folder** of the **message database**, the client MUST create a new **search folder container** that has the following properties:

Property	Value
PidTagFolderType	MUST be FOLDER_SEARCH (0x00000002)
PidTagDisplayName	SHOULD be name for the search folder , as specified in section 3.1.4.1.1.
PidTagContainerClass	MUST be IPF.Note.
PidTagExtendedFolderFlags	MUST contain a BLOB , as specified in [MS-OXOCFG]. Among others, that BLOB MUST contain: <ul style="list-style-type: none"> • SearchFolderID: a generated GUID identifying the message object. This MUST be the same as PidTagSearchFolderId of the Message object

	<p>created (as specified in section 3.1.4.1.3).</p> <ul style="list-style-type: none"> • SearchFolderTag: another ID, often a TickCount, used to indicate whether this Folder object is synchronized with the associated message object. • ExtendedFlags: Search folders support the total count and unread count flags.
--	---

The client MUST also set the search criteria. For more details, see [MS-OXCFOLD] and [MS-OXOCFG].

3.1.4.1.3 Creating a New Definition Message

In the associated message table of the Common Views folder in the message database, the client MUST create a new definition message and populate each property specified in section 2.2.1.2.

Property	Value
PidTagMessageClass	MUST be "IPM.Microsoft.WunderBar.SFInfo".
PidTagDisplayName	SHOULD match the name of the search folder .
PidTagSearchFolderId	MUST be the same GUID as stored in the extended folder properties BLOB on the Folder object .
PidTagSearchFolderTemplateId	MUST be the ID of the template chosen, as specified in section 2.2.3.
PidTagSearchFolderTag	SHOULD be the same value as the SearchFolderTag stored in the PidTagExtendedFolderFlags property BLOB on the Folder object .
PidTagSearchFolderLastUsed	SHOULD be set to the current time.
PidTagSearchFolderExpiration	SHOULD be set to the date and time at which the search folder container object will be deleted.
PidTagSearchFolderStorageType	SHOULD clear the 0x00008000 flag, because the search folder container object exists. The value of this flag SHOULD be the value specified by the template definition in section 2.2.3. The 0x00000004, 0x00000008, and 0x00000010 (big-endian) flags MUST be set as specified by the template being used.
PidTagSearchFolderEfpFlags	SHOULD be the same as the flags stored in the extended folder properties BLOB on the folder object.
PidTagSearchFolderDefinition	<p>MUST contain a BLOB as specified in section 2.2.1.2.8.</p> <ul style="list-style-type: none"> • ULONG version of the search folder implementation. <5> • ULONG value in PidTagSearchFolderStorageType.

	<ul style="list-style-type: none"> • ULONG number used by the search template. This number MUST be present, regardless of whether the template specifies it or not. • Unicode string with preceding length used by the search template. This MUST be present, regardless of whether the template specifies it or not. See section 2.2.1.2.8 for the exact format. • A Skip block, a ULONG defining how many more ULONGs to read and skip. If no skipping is needed, this value MUST be 0x00000000. • Boolean: whether the search SHOULD search sub-folders. This MUST be present. • Unicode string with preceding length of semi-colon delimited folder names indicating the folders to search. See section 2.2.1.2.8 for the exact format. This field is only used by the Custom template (as specified in section 2.2.3.14). • ULONG indicating the byte size of the following field. • EntryList containing the folders to search. This MUST be present if the previous field is greater than zero (0). • AddressList containing the addresses related to the search criteria. This MUST be present if PidTagSearchFolderStorageType contains the 0x00000004 flag. If PidTagSearchFolderStorageType does not contain that flag, this field MUST NOT be present. • Another Skip block. • The SRestriction criteria for the search folder. This MUST be present if PidTagSearchFolderStorageType contains 0x00000008. If PidTagSearchFolderStorageType does not contain that flag, this field MUST NOT be present. • BLOB detailing an advanced custom search folder. This BLOB MUST be present if PidTagSearchFolderStorageType contains 0x00000010. If PidTagSearchFolderStorageType does not contain that flag, this field MUST NOT be
--	---

	present.
--	----------

3.1.4.2 Opening a Search Folder

If the **search folder** is not active, the client MUST create the folder in the finder of the **message database**, as specified in section 3.1.4.1.

If the current date/time is later than the **PidTagSearchFolderExpiration** value of the search folder message, the client SHOULD recreate the criteria and update the **message** and the **folder**.

After the **Folder object** is updated, or if the search folder was active already, the client MUST open the folder. When the search folder is opened, the client SHOULD set the value of **PidTagSearchFolderLastUsed** to the current time.

3.1.4.3 Modifying a Search Folder

Any changes to the **search folder** MUST be made to the **definition message** in the Common Views associated table. The **search folder container** (if one exists) MUST be updated or deleted. If it is deleted, it MUST be updated or recreated when the search folder is accessed. In addition to any change:

- The **PidTagSearchFolderTag** of the **Message object** and the, **SearchFolderTag** sub-property in the search folder container **PidTagExtendedFolderFlags** property SHOULD be updated. These new values MUST be equal.
- The **PidTagSearchFolderLastUsed** SHOULD be set to the current time.

3.1.4.4 Deleting a Search Folder

To delete a **search folder**, the client MUST delete the **Message object** from the Common Views **associated message** table and delete the **Folder object** in the Finder folder. For more details about deleting **messages** and **folders**, see [MS-OXCMSG] and [MS-OXCFOLD].

3.1.4.5 Current Time Exceeds PidTagSearchFolderExpiration

When the current time passes the time in **PidTagSearchFolderExpiration**, the client SHOULD delete (mark inactive) the actual **folder** items in the **message database** finder.

3.1.5 Message Processing Events and Sequencing Rules

In the event that **messages** with potentially conflicting changes arrive close to one another, this protocol follows the standard messaging behavior as specified in [MS-OXCMSG].

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

	000050000000201090E0201090E2E000000000000A19D6BCC8B44A4CBF5DF63A922E170C010014E20014EE879243A1A829B0620DBD890000020DEFAC000004000000050000000201090E0201090E2E0000000000A19D6BCC8B44A4CBF5DF63A922E170C010014E20014EE879243A1A829B0620DBD8900000210B4D2000004000000050000000201090E0201090E2E0000000000A19D6BCC8B44A4CBF5DF63A922E170C010014E20014EE879243A1A829B0620DBD8900000210B4D300000400000050000000201090E0201090E2E000000000000A19D6BCC8B44A4CBF5DF63A922E170C010014E20014EE879243A1A829B0620DBD8900000210B4D4000004000000050000000201090E0201090E2E0000000000A19D6BCC8B44A4CBF5DF63A922E170C010014E20014EE879243A1A829B0620DBD8900000210B4D1000001000000020000000600000000000000300070E010000000600000001000000030097100100000000000000
PidTagSearchFolderStorageType	72
PidTagSearchFolderTag	1045439171
PidTagSearchFolderEfpFlags	0

5 Security

5.1 Security Considerations for Implementers

There are no security considerations beyond those specified in [MS-OXCMSG] and [MS-OXCFOLD].

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 Office/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 Office/Exchange behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies Office/Exchange does not follow the prescription.

<1> Outlook 2007 SP1 and Outlook 2003 SP3 set the **PidTagExtendedFolderFlags** property of the container, sub-property **ExtendedFlags**, field b to 0x1 even if the **PidTagSearchFolderEfpFlags** property of the **definition message** is 0x00000000. If the value of **PidTagSearchFolderTemplateId** is 3 or 4, Outlook 2007 SP1 and Outlook 2003 SP3 set field b to 0x2, even if **PidTagSearchFolderEfpFlags** is 0x00000000.

<2> Section 2.2.1.2.8: Outlook 2007 SP1 and Outlook 2003 SP3 use 0x04100000 (network order).

<3> Section 2.2.3: Exchange (all versions) ignores the definition messages. Exchange does expose **active search folders** through its Outlook Web Access Web client by identifying search folder containers in the Finder folder [MS-OXOSFLD]. These search folder containers are identified by examining the **PidTagExtendedFolderFlags** property of the folder, as specified in [MS-OXOCFG]. If the sub-property **SearchFolderID** is defined, Exchange treats as a search folder. Exchange (and Outlook Web Access) do not support **inactive search folders**.

<4> Section 2.2.3.13: The Categorized template is not supported in Outlook 2003 SP3.

<5> Section 2.2.1.2.8: Outlook 2007 SP1 and Outlook 2003 SP3 use 0x04100000 (network order).

Index

- Applicability statement, 7
- Client details, 22
- Glossary, 4
- Index of security parameters, 28
- Informative references, 6
- Introduction, 4
- Message syntax, 7
- Messages, 7
 - Message syntax, 7
 - Transport, 7
- Normative references, 5
- Office/Exchange behavior, 28
- Prerequisites/preconditions, 7
- Protocol details, 22
 - Client details, 22
- Protocol examples, 27
 - Search folder message object, 27
- Protocol Overview, 6
- References, 5
 - Informative references, 5
 - Normative references, 5
- Relationship to other protocols, 6
- Search folder message object, 27
- Security, 28
 - Index of security parameters, 28
 - Security considerations for implementers, 28
- Security considerations for implementers, 28
- Standards assignments, 7
- Transport, 7
- Vendor-extensible fields, 7
- Versioning and capability negotiation, 7