

[MS-OXOABK]: Address Book Object 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.
Microsoft Corporation	April 25, 2008	0.2	Revised and updated property names and other technical content.

Preliminary

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	8
1.3	Protocol Overview (Synopsis)	8
1.4	Relationship to Other Protocols	9
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	10
1.7	Versioning and Capability Negotiation	10
1.8	Vendor-Extensible Fields	10
1.9	Standards Assignments	10
2	Messages	10
2.1	Transport	10
2.2	Message Syntax	10
2.2.1	Definitions	11
2.2.2	Properties That Apply to Containers in the Address Book Hierarchy Table	13
2.2.3	Properties That Apply to All Address Book Objects	16
2.2.4	Properties That Apply to Mail User Objects	24
2.2.5	Properties That Reference Other Address Book Objects	35
2.2.6	Properties That Apply to Distribution Lists	36
2.2.7	Properties That Apply to Organization Objects	37
2.2.8	Properties That Apply to Department Objects	37
2.2.9	Properties That Apply to Resources	38
2.2.10	Properties That Have Special Purposes	38
2.2.11	Named Properties	39
3	Protocol Details	39
3.1	Client Details	39
3.1.1	Abstract Data Model	39
3.1.2	Timers	40
3.1.3	Initialization	40
3.1.4	Higher-Layer Triggered Events	40
3.1.5	Message Processing Events and Sequencing Rules	42
3.1.6	Timer Events	42
3.1.7	Other Local Events	42
3.2	Client and Server Details	43
4	Protocol Examples	43
5	Security	48
5.1	Security Considerations for Implementers	48
5.2	Index of Security Parameters	48
6	Appendix A: Office/Exchange Behavior	48

Preliminary

1 Introduction

An address book is a repository of information about address book objects, many of which have information that can be retrieved by messaging clients to display to messaging users, or to enable users to address email to other email-enabled users and entities represented by these objects. The objects in the address book are organized in a fashion that makes it possible for users to locate and look up information about these objects.

The client retrieves data about a number of different address book objects from one of two possible data sources. The client retrieves data from the Name Service Provider Interface (NSPI) Server using the protocol as specified in [MS-NSPI]. Alternatively, the client retrieves data from an Offline Address Book as specified in [MS-OXOAB]. These include properties on mail users, distribution lists, resources, address book containers and the address book hierarchy. The client chooses the data source based on whatever source is available, or based on user settings.

The Address Book Object Protocol specifies:

- Properties of various address book objects
- How these properties interrelate

1.1 Glossary

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

alias

address book

address book distinguished name (ABDN)

address book object

address book container

address book hierarchy table

address list

ambiguous name resolution (ANR)

ASCII

binary large object (BLOB)

code page

display template

distinguished name (DN)

distribution list

Globally Unique Identifier (GUID)
Mail User
mailbox
message database
Multipurpose Internet Mail Extensions (MIME)
Name Service Provider Interface (NSPI)
Offline Address Book (OAB)
property
property ID
property type
recipient
Remote Procedure Call (RPC)
Rich Text Format (RTF)
Simple Mail Transfer Protocol (SMTP)
Tagged property
Transport Neutral Encapsulation Format (TNEF)
Unicode
Unicode Transformation Format, 16-bits, Little-Endian(UTF-16LE)

The following terms are specific to this document:

department: An **address book object** that describes a department within an organization.

Global Address List (GAL): The address list that conceptually represents the default address list for an address book obtained from an Offline Address Book or NSPI server.

organization: An **address book object** that describes an entire organization.

resource: An **address book object** that represents a resource that can be reserved, such as a room or equipment.

template: A template that describes how to display or allow the user to modify information about another address book object.

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-NSPI] Microsoft Corporation, "Name Service Provider Interface (NSPI) Protocol Specification", April 2008.
- [MS-OXABREF] Microsoft Corporation, "Address Book Name Service Provider Interface (NSPI) Referral Protocol Specification", April 2008.
- [MS-OXCDATA] Microsoft Corporation, "Data Structures Protocol Specification", April 2008.
- [MS-OXCMAIL] Microsoft Corporation, "RFC2822 and MIME to E-mail Object Conversion Protocol Specification", April 2008.
- [MS-OXGLOS] Microsoft Corporation, "Office Exchange Protocols Master Glossary", April 2008.
- [MS-OXOAB] Microsoft Corporation, "Offline Address Book (OAB) Format and Schema Protocol Specification", April 2008.
- [MS-OXOABKT] Microsoft Corporation, "Address Book User Interface Templates Protocol Specification", April 2008.
- [MS-OXPFOAB] Microsoft Corporation, "Offline Address Book (OAB) Public Folder Retrieval Protocol Specification", April 2008.
- [MS-OXPROPS] Microsoft Corporation, "Office Exchange Protocols Master Property List Specification", April 2008.
- [MS-OXTNEF] Microsoft Corporation, "Transport Neutral Encapsulation Format (TNEF) Protocol Specification", April 2008.
- [MS-OXWOAB] Microsoft Corporation, "Offline Address Book (OAB) Retrieval Protocol Specification", April 2008.
- [MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions", July 2006, <http://go.microsoft.com/fwlink/?LinkId=112246>.
- [RFC1034] Mockapetris, P., "Domain Names—Concepts and Facilities", RFC 1034, November 1987, <http://www.ietf.org/rfc/rfc1034.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>.

[RFC2876] Pawling, J., "Use of the KEA and SKIPJACK Algorithms in CMS", RFC2876, July 2000, <http://www.ietf.org/rfc/rfc2876.txt>.

[RFC3174] Eastlake III, D. and Jones, P., "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001, <http://www.ietf.org/rfc/rfc3174.txt>.

[RFC3280] Housley, R., Polk, W., Ford, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002, <http://www.ietf.org/rfc/rfc3280.txt>.

[RFC3851] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004, <http://www.ietf.org/rfc/rfc3851.txt>.

[RFC3852] Housley, R. "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004, <http://www.ietf.org/rfc/rfc3852.txt>.

[RFC4234] Crocker, D., Ed. and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, <http://www.ietf.org/rfc/rfc4234.txt>.

[RFC959] Postel, J., Reynolds, J., "File Transfer Protocol (FTP)", RFC 959, October 1985, <http://www.ietf.org/rfc/rfc959.txt>.

1.2.2 Informative References

[ISO/IEC 8825-1] "ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ISO/IEC 8825-1:1998, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32306.

1.3 Protocol Overview (Synopsis)

An **address book** is a collection of **address book objects**. There are many types of objects in an address book, many of which can be added as an addressee to an outbound message to be the destination. Every object in the address book has various properties, and the values of these properties contain information used by messaging clients to route the message correctly or to display information about the address book object to a messaging user.

The address book also contains one or more address book containers, each describing an address list. An address list is a collection of address book objects that is possible to be rendered in a table to be browsed by a messaging user. A collection of **address book containers**, each representing an address list, is arranged in an **address book hierarchy table**.

The address book can be stored remotely on a **Name Service Provider Interface (NSPI)** server and accessed through NSPI calls using the NSPI protocol, as specified in [MS-NSPI], or can be stored locally on the client machine as an **Offline Address Book (OAB)** using the OAB Format and Schema structure, as specified in [MS-OXOAB].

Some of the address book object types include:

- mail users, each of which describes a person or entity that can receive e-mail
- distribution lists, each of which is a collection of other mail users, distribution lists, or other address book objects that can receive e-mail
- resources, which can be reserved such as a room or equipment
- organizations, each of which describes an organization
- departments, each of which describes the departmental structure of an organization
- address book containers, each of which represents an address list containing address book objects that can be viewed as a table
- templates, each of which describes a physical view that can be used to show details on other address book objects to a messaging user, and is described in more detail in [MS-OXOABKT].

1.4 Relationship to Other Protocols

This Address Book Object Protocol specification relies on understanding how NSPI calls are transmitted to the server using the underlying RPC transport using the Name Service Provider Interface (NSPI) protocol as specified in [MS-NSPI], and an understanding of the Offline Address Book (OAB) Format and Schema structure, as specified in [MS-OXOAB].

The specification also relies on an understanding of how to locate and establish a connection with an NSPI server using the Name Service Provider Interface (NSPI) Referral protocol (see [MS-OXABREF]), and an understanding of how to obtain an Offline Address Book using either the Offline Address Book Web Retrieval protocol (see [MS-OXWOAB]) or the Offline Address Book Public Folder Retrieval protocol (see [MS-OXPFOAB]). The Address Book User Interface Templates Protocol, as specified in [MS-OXOABKT], is needed in order to render information about an address book object to a messaging user, which relies on an understanding of this Address Book Object Protocol specification.

1.5 Prerequisites/Preconditions

This Address Book Object Protocol specification assumes that either the messaging client has been referred to a NSPI server using the Name Service Provider Interface (NSPI) Referral protocol, as specified in [MS-OXABREF], and established a connection to a server that supports the Name Service Provider Interface (NSPI) protocol, as specified in [MS-NSPI], or that the messaging client has access to an Offline Address Book, as specified in [MS-OXOAB].

1.6 Applicability Statement

This Address Book Object protocol is used to access information about address book objects in an organization. Messaging clients use this protocol to determine the destination for outbound messages addressed to these objects, and to display information about these objects to a messaging user.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

The following sections specify the properties of address book objects and their formats.

Unless otherwise specified, all numeric values in this protocol are in little-endian format.

Unless otherwise specified, all Unicode string representations are in UTF-16LE format.

2.1 Transport

This protocol uses the Name Service Provider Interface (NSPI) protocol, as specified in [MS-NSPI], as the underlying transport protocol when handling address books online using an NSPI server, or the Offline Address Book (OAB) Format and Schema structure, as specified in [MS-OXOAB], as the underlying structure when handling address books using a local Offline Address Book.

2.2 Message Syntax

When messaging clients are using a NSPI server, the properties described in this specification are returned and given to various NSPI functions of the Name Service Provider Interface (NSPI) protocol as specified in [MS-NSPI]. The usage of NSPI calls depends on the object type and its associated property list. As a collection, the entire set of NSPI calls enable messaging clients to access and browse address lists, and manipulate or obtain data on address book objects in that list.

Alternatively, these structures are maintained in an Offline Address Book using the Offline Address Book (OAB) Format and Schema structure as specified in [MS-OXOAB]. The Offline Address Book is a collection of address lists, each containing address book objects and their properties. Messaging clients use the Offline Address Book (OAB) Format and Schema

structure specified in [MS-OXOAB] to access and browse the address lists and to obtain data of the address book objects in those lists.

There are many types of address book objects, including mail users, distribution lists, address book containers, resources, department objects, organization objects, templates, and other object types. This protocol specification does not require values for all properties that are defined for any object type, nor limit the properties to those of their own type or those listed in this specification. When a data source, such as an Offline Address Book or NSPI server, includes properties that are not in this set, the format and meaning of the property is defined by the implementation of that data source.

In all of the properties of type PtypString listed below, a request to an NSPI server of that property with type PtypString8 or PtypString is permitted. If a conversion is required, the string will be converted to the type requested by the client, as specified in [MS-NSPI].

In all of the properties of type PtypString listed below, an Offline Address Book that includes a value for that string property MUST contain exactly one string representation. The internal representation of strings in an Offline Address Book, using the Offline Address Book (OAB) Format and Schema structure, is described in [MS-OXOAB]. A client that looks up the value for a string property MUST convert the value to the string type native to the client before interpreting its value.

2.2.1 Definitions

The definitions below describe structures used by various address book object properties.

2.2.1.1 Distinguished Names for Objects

Address Book Distinguished Names (ABDNs) are used to uniquely identify objects in the address book. Throughout the rest of this specification, the term Distinguished Name, or DN, will refer to an Address Book Distinguished Name. Each address book object MUST have a unique DN value, expressed as a NULL terminated ASCII string. The DN is stored in the PidTagEmailAddress property. The DN is also embedded in the “Distinguished Name” field of the Permanent Entry ID, as specified in [MS-NSPI]. DNs are structured as the following ABNF definition illustrates. For more information on ABNF, see [RFC4234].

```
dn           = organization-dn /  
              addresslist-dn /  
              x500-dn  
organization-dn = org-rdn  
addresslist-dn  = "/guid=" container-guid /  
                 gal-addresslist-dn  
container-guid  = 32 (HEXDIG)  
gal-addrlist-dn = "/"
```

```

x500-dn          = x500-container-dn object-rdn
                  ; x500-dns are limited to 16 levels

x500-container-dn = org-rdn org-unit-rdn 0*13(container-rdn)

org-rdn          = "/o=" rdn

org-unit-rdn     = "/ou=" rdn

container-rdn    = "/cn=" rdn

object-rdn       = "/cn=" rdn

rdn              = ( non-space-teletex ) /
                  ( non-space-teletex *62(teletex-char)
                    non-space-teletex )
                  ; rdn values are limited to 64 characters
                  ; the number of rdns is limited to 16 but the
                  ; total cumulative length of rdn characters in
                  ; An x500-dn is limited to 256.

teletex-char     = SP / non-space-teletex

non-space-teletex = "!" / DQUOTE / "%" / "&" / "\" / "(" / ")" /
                  "*" / "+" / "," / "-" / "." / "0" / "1" /
                  "2" / "3" / "4" / "5" / "6" / "7" / "8" /
                  "9" / ":" / "<" / "=" / ">" / "?" / "@" /
                  "A" / "B" / "C" / "D" / "E" / "F" / "G" /
                  "H" / "I" / "J" / "K" / "L" / "M" / "N" /
                  "O" / "P" / "Q" / "R" / "S" / "T" / "U" /
                  "V" / "W" / "X" / "Y" / "Z" / "[" / "]" /
                  "_" / "a" / "b" / "c" / "d" / "e" / "f" /
                  "g" / "h" / "i" / "j" / "k" / "l" / "m" /
                  "n" / "o" / "p" / "q" / "r" / "s" / "t" /
                  "u" / "v" / "w" / "x" / "y" / "z" / "|"

```

In general, the Distinguished Names for all address book objects MUST follow the *dn* format. The DNs for different types of objects MUST follow more strict *dn* formats, depending on their type, according to the following table:

Object type:	DN format:	Notes:
Address book container	<i>addresslist-dn</i>	
Global Address List container	<i>gal-addrlist-dn</i>	

Object type:	DN format:	Notes:
Mail user	<i>x500-dn</i>	<i>org-rdn</i> string is the mail user's organization.
Organization	<i>organization-dn</i>	
Message database	<i>x500-dn</i>	The <i>x500-container-dn</i> is the mailbox server.
Mailbox server	<i>x500-dn</i>	The <i>rdn</i> in the object- <i>rdn</i> is the name of the mailbox server.
Room container reference	<i>x500-dn</i> with no container- <i>rdn</i>	The <i>rdn</i> of the object- <i>rdn</i> matches the <i>container-guid</i> of the address book container.
All other address book objects	<i>dn</i>	

When the DN of an address book object obtained from an NSPI server matches the DN of one obtained from an Offline Address Book, the objects represent the same entity. For such an object visible on both data sources, the properties available in the Offline Address Book SHOULD<1> be a subset of the properties in the NSPI server, MAY<2> include additional properties, and SHOULD<3><4> have the same value when present on both data sources.

2.2.2 Properties That Apply to Containers in the Address Book Hierarchy Table

An address book is a collection of address book objects, each of which are contained in any number of address lists. The address book is arranged as a hierarchy of address book containers. Each address book container, in turn, describes an address list that contains many address book objects.

When using an Offline Address Book, messaging clients obtain information about the address book hierarchy and its address book containers using the Offline Address Book Web and Public Folder Retrieval protocols as specified in [MS-OXWOAB] and [MS-OXPFOAB]. Since an Offline Address Book maintains its own structure for the hierarchy table, none of the properties of address book containers described below apply to the Offline Address Book.

When using an NSPI server, messaging clients call `NspiGetSpecialTable`, as described in [MS-NSPI] to obtain the address book hierarchy table. The hierarchy table is a set of rows, each row describing one address book container. The NSPI server MUST return the following properties for each container in the hierarchy, in this order:

- PidTagEntryId
- PidTagContainerFlags
- PidTagDepth
- PidTagAddressBookContainerId
- PidTagDisplayName
- PidTagAddressBookIsMaster
- PidTagAddressBookParentEntryId (optional, and MUST be the seventh column if it is included)

For every row returned, all of the above properties except PidTagAddressBookParentEntryId MUST be present and have a value prescribed under its definition.

In addition, the PidTagEntryId MUST be in the form of a Permanent Entry ID, specified in [MS-NSPI], with its DisplayType having the value DT_CONTAINER, as specified in [MS-NSPI], and its DN following the *addresslist-dn* format specification, as specified in section 2.2.1.1. When the object is the Global Address List container, its DN MUST follow the *gal-addrlist-dn* format specification.

2.2.2.1 PidTagContainerFlags

The PidTagContainerFlags property of type PtypInteger32 contains a bitmask of flags describing capabilities of an address book container.

The following flags are defined for the PidTagContainerFlags bitmask. The PidTagContainerFlags MUST NOT contain any other flags.

Name	Value	Description
AB_RECIPIENTS	0x00000001	The container holds address book objects. This flag does not indicate whether any objects are actually present in the container. This flag MUST be set for all containers and distribution lists returned by the NSPI server.

Name	Value	Description
AB_SUBCONTAINERS	0x00000002	The container holds child containers. This flag does not indicate whether any sub-containers are actually present in the container.
AB_UNMODIFIABLE	0x00000008	It is not possible to add or remove address book objects from the container. This flag MUST be set for containers returned by the NSPI server.

For distribution lists obtained from a NSPI server, this property **MUST** be present, and its value **MUST** be AB_RECIPIENTS.

Since this property applies to a container in the hierarchy table, it is not present on objects in the Offline Address Book. An Offline Address Book has its own structure for maintaining the hierarchy, using the Offline Address Book Web and Public Folder Retrieval protocols, as specified in [MS-OXWOAB] and [MS-XPFOAB].

2.2.2.2 PidTagDepth

The PidTagDepth property of type PtypInteger32 represents the relative level of depth of a container in a hierarchy table. Objects in the hierarchy table that share the same PidTagDepth value **SHOULD** be considered peer containers by clients if they are adjacent or if they are separated only by containers that have equivalent or larger depth values.

Since this property applies to a container in the hierarchy table, it is not present on objects in an Offline Address Book. An Offline Address Book has its own structure for maintaining the hierarchy, using the Offline Address Book Web and Public Folder Retrieval protocols, as specified in [MS-OXWOAB] and [MS-XPFOAB].

2.2.2.3 PidTagAddressBookContainerId

The PidTagAddressBookContainerId property of type PtypInteger32 is another way, other than PidTagEntryId, of identifying a container on an NSPI server. It is a Minimal Entry ID. A value of zero represents the Global Address List. This value is used in other NSPI calls (such as NspiResolveNamesW) to identify which container the NSPI call applies. If the value is non-zero, it is only a valid representation of the specific container while the connection to the NSPI server lasts or, after disconnection and reconnection to the same or other NSPI server using NspiBind, as long as the new server identifies itself as having the same GUID in its return value for pServerGuid, as specified in [MS-NSPI].

Since this property applies to a container in the hierarchy table, it is not present on objects in an Offline Address Book. An Offline Address Book has its own structure for maintaining the hierarchy, using the Offline Address Book Web and Public Folder Retrieval protocols, as specified in [MS-OXWOAB] and [MS-XPFOAB].

2.2.2.4 PidTagAddressBookIsMaster

The PidTagAddressBookIsMaster property of type PtypBoolean is TRUE if it is possible to create address book objects in that container, and FALSE otherwise. The value does not pertain to parent containers or sub-containers of this container.

Since this property applies to a container in the hierarchy table, it is not present on objects in an Offline Address Book. An Address Book has its own structure for maintaining the hierarchy, using the Offline Address Book Web and Public Folder Retrieval protocols, as specified in [MS-OXWOAB] and [MS-XPFOAB].

2.2.2.5 PidTagAddressBookParentEntryId

The PidTagAddressBookParentEntryId property of type PtypBinary is the Entry ID of the parent container in a hierarchy of address book containers. This property is not present if there is no parent container. Messaging clients use this to expand and collapse a hierarchy of address book containers in an address book hierarchy table.

Since this property applies to a container in the hierarchy table, it is not present on objects in an Offline Address Book. An Offline Address Book has its own structure for maintaining the hierarchy, using the Offline Address Book Web and Public Folder Retrieval protocols, as specified in [MS-OXWOAB] and [MS-XPFOAB].

2.2.3 Properties That Apply to All Address Book Objects

Address book objects are listed in an Offline Address Book using the Offline Address Book (OAB) Format and Schema structure (see [MS-OXOAB]) or accessed on a NSPI server using various NSPI calls of the Name Service Provider Interface (NSPI) protocol (see [MS-NSPI]). Each object represents any addressable entity, such as a mail user, distribution list, department, organization, address book container, resource, or other object.

It is possible to set any address book object as an addressee of an outbound message to be sent by a messaging client. A mail user is an example, and generally contains an e-mail address to be used for messaging. A distribution list is a special type of address book object that represents a collection of other address book objects, can be an addressee of an outbound message, and MAY<5> have additional properties and operations, such as the ability to determine the distribution list membership. Mail user objects are further broken down into other subtypes, such as rooms, equipment, messaging forums, or other types.

The department object and the organization object are types of address book objects that are needed to support a hierarchical address book view needed by messaging clients.

All address book objects obtained from an Offline Address Book or an NSPI server MUST define values for the following properties:

PidTagDisplayName
PidTagEmailAddress
PidTagObjectType
PidTagDisplayType

In addition to the above properties, all address book objects obtained from an NSPI server MUST define values for the following properties:

PidTag7BitDisplayName

In addition to the above properties, all address book objects obtained from a NSPI server MUST define values for the following properties, which MUST NOT be defined for any objects in an Offline Address Book:

PidTagEntryId
PidTagTemplateid
PidTagRecordKey
PidTagSearchKey
PidTagInstanceKey
PidTagAddressBookContainerId
PidTagInitialDetailsPane
PidTagTransmittableDisplayName
PidTagAddressType
PidTagAddressBookObjectDistinguishedName

In addition to the preceding list, an Offline Address Book MUST NOT define values for these properties:

PidTagDepth
PidTagContainerFlags

A NSPI server MUST define values for these properties for distribution lists and define no values for other object types:

PidTagContainerContents
PidTagContainerFlags

If either a NSPI server or Offline Address Book does not conform to these rules, and to the descriptions of the properties below, client behavior is undefined.

2.2.3.1 PidTagDisplayName

The PidTagDisplayName property of type PtypString represents a displayable form of the address book object. When the address book object is a mail user, the PidTagDisplayName

string property is the name of the mail user, usually the mail user's full name. When the object is a distribution list, PidTagDisplayName is the name of the distribution list that would be displayed when addressing messages to that distribution list. When the object is a department object, PidTagDisplayName is the name of that department. When the object is any other address book object, PidTagDisplayName is the displayable name of that object.

PidTagDisplayName is also one of the columns returned for the set of address book containers in the address book hierarchy table. The PidTagDisplayName value for each row in the hierarchy table represents the name of the address book container of that row.

2.2.3.2 PidTagEntryId

The PidTagEntryId property of type PtypBinary is used to identify many different types of messaging objects, including objects that are not in an address book. The formats of the PidTagEntryId property for non-addressing messaging objects are specified in [MS-OXCADATA]. These other formats do not apply to any objects in an Offline Address Book or NSPI server.

The PidTagEntryId property of type PtypBinary identifies an address book object on an NSPI server, and represents either the Permanent Entry ID or the Ephemeral Entry ID, as specified in [MS-NSPI]. Messaging clients use it to open the object and to perform operations on it, such as obtaining other properties. A description of the types of operations that can be performed is specified in [MS-NSPI]. When the object is in Permanent Entry ID format, its DN MUST match the value for PidTagEmailAddress and MUST follow the *dn* format specification particular to the type of object, as described in section 2.2.1.1.

The Offline Address Book (OAB) Format and Schema structure specification [MS-OXOAB] does not include values for PidTagEntryId for address book objects in its data structure. Instead, the PidTagEmailAddress property identifies objects in an Offline Address Book.

2.2.3.3 PidTagTemplateid

The PidTagTemplateid property of type PtypBinary contains the PidTagEntryId, expressed as a Permanent Entry ID format. This value MUST be present for all address book objects on a NSPI server, its DN MUST match the value for PidTagEmailAddress, and its DN MUST follow the *dn* format specification particular to the type of object, as described in section 2.2.1.1.

The PidTagTemplateid property is not present on objects in an Offline Address Book.

2.2.3.4 PidTagRecordKey

The PidTagRecordKey property of type PtypBinary contains a unique binary-comparable identifier for a specific address book object. It MUST be present on all objects on a NSPI server and MUST match PidTagTemplateid.

The PidTagRecordKey property is not present on objects in an Offline Address Book.

2.2.3.5 PidTagSearchKey

The **PidTagSearchKey** property of type PtypBinary is a binary value formed by concatenating the ASCII string "EX:" followed by the DN for the object converted to all upper case, followed by a zero byte value. This value **MUST** be present for all address book objects on a NSPI server and **MUST** follow this form.

The PidTagSearchKey property is not present on objects in an Offline Address Book.

2.2.3.6 PidTagInstanceKey

The PidTagInstanceKey property of type PtypBinary value is another way, other than PidTagEntryId, of identifying an object on an NSPI server. It is a Minimal Entry ID, represented as a 4 byte binary value, in little-endian byte order.

The PidTagInstanceKey property is not present on objects in an Offline Address Book.

2.2.3.7 PidTag7BitDisplayName

The PidTag7BitDisplayName property of type PtypString contains a displayable form of the address book object that can be rendered in the client user's own code page.

2.2.3.8 PidTagTransmittableDisplayName

The PidTagTransmittableDisplayName property of type PtypString contains an address book object's display name that is transmitted with the message. It **MUST** be present on all objects on a NSPI server, and its value **MUST** match the value for PidTagDisplayName.

The PidTagTransmittableDisplayName property is not present on objects in an Offline Address Book.

2.2.3.9 PidTagAddressBookPhoneticDisplayName

The PidTagAddressBookPhoneticDisplayName property of type PtypString is the phonetic representation of the PidTagDisplayName property.

2.2.3.10 PidTagObjectType

The **PidTagObjectType** property of type PtypInteger32 contains a value that specifies the type of an object.

The **PidTagObjectType** property **MUST** be present for all address book objects and **MUST** have one of the following values:

Name	Value	Description
MAILUSER	0x00000006	A mail user object, or any address book object that is not a distribution list or forum.

Name	Value	Description
DISTLIST	0x00000008	A distribution list object.
FOLDER	0x00000003	A messaging forum, such as a bulletin board service or a public or shared folder.

2.2.3.11 PidTagDisplayType

The PidTagDisplayType property of type PtypInteger32 contains a value indicating how to display an address book object in a table or as an addressee on a message. Messaging clients use this to display an icon, bold the item, or some other display element to make it easy for a user viewing the object to distinguish its type. In addition, the property PidTagDisplayTypeEx provides a mean to further refine the display.

PidTagDisplayType MUST have one of the following values of Display Type, as specified in [MS-NSPI], according to the object's type: DT_MAILUSER, DT_DISTLIST, DT_FORUM, DT_AGENT, DT_ORGANIZATION, DT_PRIVATE_DISTLIST, DT_REMOTE_MAILUSER. If the object is not one of these types, its PidTagDisplayType has the value DT_MAILUSER.

2.2.3.12 PidTagDisplayTypeEx

The PidTagDisplayTypeEx property of type PtypInteger32 contains a value indicating how to display an address book object in a table or as a recipient on a message. Messaging clients use this to display an icon, bold the item, or some other display element to make it easy for a user viewing the object to distinguish its type. This property contains more detailed information about the object's display information than PidTagDisplayType. In addition to the display information contained in PidTagDisplayType, it distinguishes between additional object types. When the object comes from a remote server, it also includes information about the type of object on that remote server, as well as the type on the local server.

The PidTagDisplayTypeEx property of type PtypInteger32 also includes information indicating whether it is possible to share information from the user's own mailbox to the entity represented by this address book object.

The PidTagDisplayTypeEx is a bitmask of flags and values, and has the following structure:

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
r	s	reserved														dtRemote					dtLocal										

r (1 bit): 1 indicates the value in dtRemote is the remote display type. 0 means dtRemote is undefined.

s (1 bit): 1 indicates the mailbox server supports sharing to the entity represented by this address book object. 0 means it does not.

reserved (14 bits): Undefined. It MUST contain all zeroes and MUST be ignored by clients.

dtRemote (1 byte): Display type of the address book object in the remote forest. This is undefined if r is 0. If r is 1, it contains one of the values listed below.

dtLocal (1 byte): Display type of the address book object in the messaging user's local forest. It contains one of the values listed below.

dtLocal and/or dtRemote MUST have one of the following values of Display Type, as defined in [MS-NSPI], according to the object's type: DT_MAILUSER, DT_DISTLIST, DT_FORUM, DT_AGENT, DT_ORGANIZATION, DT_PRIVATE_DISTLIST, DT_REMOTE_MAILUSER, or one of the values in the following table:

Name	Value	Description
DT_ROOM	0x00000007	A conference room. It is possible for messaging clients to send meeting requests to this address book object to book the room.
DT_EQUIPMENT	0x00000008	Equipment. It is possible for messaging clients to send meeting requests to this address book object to reserve the equipment.
DT_SEC_DISTLIST	0x00000009	A distribution list.

2.2.3.13 PidTagAddressType

The PidTagAddressType property of type PtypString contains the address book object's e-mail address type. It MUST have the value "EX" for all objects on a NSPI server.

The PidTagAddressType property is not present on objects in an Offline Address Book.

2.2.3.14 PidTagEmailAddress

The PidTagEmailAddress property of type PtypString contains the address book object's e-mail address, expressed in X500 format, using the *dn* format specification particular to the

type of object, defined in section 2.2.1.1. This property MUST be present for every address book object. Its value MUST match the DN of the Permanent Entry ID for the object if the object is present on an NSPI server. Its DN MUST follow the *dn* format specification particular to the type of object, as described in section 2.2.1.1.

2.2.3.15 PidTagAddressBookObjectDistinguishedName

The **PidTagAddressBookObjectDistinguishedName** property of type **PtypString** contains the Distinguished Name (DN) of the address book object in *dn* format, as described in section 2.2.1.1. If present, its DN MUST follow the *dn* format specification particular to the type of object, as described in section 2.2.1.1. Its value, if present, MUST match the value for **PidTagEmailAddress**. This value MUST be present on all address book objects on a NSPI server.

2.2.3.16 PidTagCreationTime

The **PidTagCreationTime** property of type **PtypTime** contains the creation date and time for the address book object in UTC.

2.2.3.17 PidTagLastModificationTime

The **PidTagLastModificationTime** property of type **PtypTime** contains the date and time that the address book object was last modified in UTC.

2.2.3.18 PidTagSendRichInfo

The **PidTagSendRichInfo** property of type **PtypBoolean** contains TRUE if the e-mail-enabled entity represented by the address book object can receive all message content, including Rich Text Format and other embedded objects. When sending mail using the RFC2822 and MIME to E-mail Object Conversion protocol, as specified in [MS-OXCMAIL], the **PidTagSendRichInfo** property specifies whether to encode the message in MIME or in TNEF, as specified in [MS-OXCMAIL].

2.2.3.19 PidTagSendInternetEncoding

The **PidTagSendInternetEncoding** property of type **PtypInteger32** contains a bitmask of message encoding preferences for mail sent to e-mail-enabled entity represented by this address book object. When sending mail using the RFC2822 and MIME to E-mail Object Conversion protocol, as specified in [MS-OXCMAIL], the **PidTagSendInternetEncoding** property specifies the format of the MIME body, as specified in [MS-OXCMAIL].

2.2.3.20 PidTagAccount

The **PidTagAccount** property of type **PtypString** contains the address book object's alias, which is an alternative name by which the object can be identified.

2.2.3.21 PidTagSmtpAddress

The **PidTagSmtpAddress** property of type **PtypString** is the address book object's SMTP address.

2.2.3.22 PidTagAddressBookTargetAddress

The PidTagAddressBookTargetAddress property of type PtypString contains the foreign system e-mail address of an address book object. If present, it MUST be prefixed with the address type of the foreign address, followed by a colon (":"), followed by the foreign email address expressed in that address type. For SMTP addresses, it is "SMTP:" followed by the foreign SMTP address.

2.2.3.23 PidTagAddressBookProxyAddresses

The PidTagAddressBookProxyAddresses property of type PtypMultipleString contains alternate e-mail addresses for the address book object. Each string MUST be prefixed by an e-mail type, followed by a colon (":"), followed by the e-mail address in a format for that e-mail type. For an SMTP address, the string MUST start with the string "smtp:" or "SMTP:" followed by an SMTP address. The first string starting with the upper case string "SMTP:" MUST match the PidTagSmtpAddress property for the address book object.

2.2.3.24 PidTagAddressBookSeniorityIndex

The PidTagAddressBookSeniorityIndex property of type PtypInteger32 contains a signed integer that specifies the seniority order of address book objects in a department, with larger values specifying members that are more senior, and specifies the sort order of department objects, in descending order.

2.2.3.25 PidTagAddressBookObjectGuid

The PidTagAddressBookObjectGuid property of type PtypBinary is a GUID that uniquely identifies an address book object.

2.2.3.26 PidTagComment

The PidTagComment property of type PtypString contains a comment about the purpose or content of the address book object.

2.2.3.27 PidTagMappingSignature

The PidTagMappingSignature property of type PtypBinary MUST be present on all address book objects on a NSPI server and MUST contain the following 16 bytes:

0xDC, 0xA7, 0x40, 0xC8, 0xC0, 0x42, 0x10, 0x1A, 0xB4, 0xB9, 0x08, 0x00,
0x2B, 0x2F, 0xE1, 0x82

The PidTagMappingSignature property is not present on objects in an Offline Address Book.

2.2.3.28 PidTagOriginalEntryId

The PidTagOriginalEntryId property of type PtypBinary MUST NOT be present on objects on a NSPI server or an Offline Address Book.

2.2.3.29 PidTagOriginalDisplayName

The PidTagOriginalDisplayName property of type PtypString MUST NOT be present on objects on a NSPI server or an Offline Address Book.

2.2.3.30 PidTagOriginalSearchKey

The PidTagOriginalSearchKey property of type PtypBinary MUST NOT be present on objects on a NSPI server or an Offline Address Book.

2.2.3.31 PidTagInitialDetailsPane

The PidTagInitialDetailsPane property of type PtypInteger32 indicates the page of a display template to display first. It MUST be present on all address book objects on an NSPI server, and MUST have the value 0.

2.2.3.32 PidTagAddressBookExtensionAttribute1... PidTagAddressBookExtensionAttribute15

These properties of type PtypString are defined and populated by the organization to be added to display templates.

2.2.4 Properties That Apply to Mail User Objects

2.2.4.1 PidTagSurname

The PidTagSurname property of type PtypString contains the mail user's family name.

2.2.4.2 PidTagGivenName

The PidTagGivenName property of type PtypString contains the mail user's given name.

2.2.4.3 PidTagNickname

The PidTagNickname property of type PtypString contains the mail user's nickname.

2.2.4.4 PidTagDisplayNamePrefix

The PidTagDisplayNamePrefix property of type PtypString contains the mail user's honorific title.

2.2.4.5 PidTagInitials

The PidTagInitials property of type PtypString contains the initials for parts of the full name of the mail user.

2.2.4.6 PidTagGeneration

The PidTagGeneration property of type PtypString contains a generational abbreviation that follows the full name of the mail user.

2.2.4.7 PidTagTitle

The PidTagTitle property of type PtypString contains the mail user's job title.

2.2.4.8 PidTagOfficeLocation

The PidTagOfficeLocation property of type PtypString contains the mail user's office location.

2.2.4.9 PidTagDepartmentName

The PidTagDepartmentName property of type PtypString contains a name for the department in which the mail user works.

2.2.4.10 PidTagCompanyName

The PidTagCompanyName property of type PtypString contains the mail user's company name.

2.2.4.11 PidTagAssistant

The PidTagAssistant property of type PtypString contains the name of the mail user's administrative assistant.

2.2.4.12 PidTagManagerName

The PidTagManagerName property of type PtypString contains the name of the mail user's manager.

2.2.4.13 PidTagAddressBookManagerDistinguishedName

The PidTagAddressBookManagerDistinguishedName property of type PtypString contains the distinguished name (DN) of the mail user's manager.

2.2.4.14 PidTagAddressBookPhoneticGivenName

The PidTagAddressBookPhoneticGivenName property of type PtypString is the phonetic representation of the PidTagGivenName property.

2.2.4.15 PidTagAddressBookPhoneticSurname

The PidTagAddressBookPhoneticSurname property of type PtypString is the phonetic representation of the PidTagSurname property.

2.2.4.16 PidTagAddressBookPhoneticCompanyName

The PidTagAddressBookPhoneticCompanyName property of type PtypString is the phonetic representation of the PidTagCompanyName property.

2.2.4.17 PidTagAddressBookPhoneticDepartmentName

The PidTagAddressBookPhoneticDepartmentName property of type PtypString is the phonetic representation of the PidTagDepartmentName property.

2.2.4.18 PidTagPostalAddress

The PidTagPostalAddress property of type PtypString contains the mail user's postal address.

2.2.4.19 PidTagStreetAddress

The PidTagStreetAddress property of type PtypString contains the mail user's street address.

2.2.4.20 PidTagPostOfficeBox

The PidTagPostOfficeBox property of type PtypString contains the number or identifier of the mail user's post office box.

2.2.4.21 PidTagLocality

The PidTagLocality property of type PtypString contains the name of the mail user's locality, such as the town or city.

2.2.4.22 PidTagStateOrProvince

The PidTagStateOrProvince property of type PtypString contains the name of the mail user's state or province.

2.2.4.23 PidTagPostalCode

The PidTagPostalCode property of type PtypString contains the postal code for the mail user's postal address.

2.2.4.24 PidTagCountry

The PidTagCountry property of type PtypString contains the name of the mail user's country/region.

2.2.4.25 PidTagHomeAddressStreet

The PidTagHomeAddressStreet property of type PtypString contains the mail user's home street address.

2.2.4.26 PidTagHomeAddressPostOfficeBox

The PidTagHomeAddressPostOfficeBox property of type PtypString contains the number or identifier of the mail user's home post office box.

2.2.4.27 PidTagHomeAddressCity

The PidTagHomeAddressCity property of type PtypString contains the name of the mail user's home locality, such as the town or city.

2.2.4.28 PidTagHomeAddressStateOrProvince

The PidTagHomeAddressStateOrProvince property of type PtypString contains the name of the mail user's home state or province.

2.2.4.29 PidTagHomeAddressPostalCode

The PidTagHomeAddressPostalCode property of type PtypString contains the postal code for the mail user's home postal address.

2.2.4.30 PidTagHomeAddressCountry

The PidTagHomeAddressCountry property of type PtypString contains the name of the mail user's home country/region.

2.2.4.31 PidTagOtherAddressStreet

The PidTagOtherAddressStreet property of type PtypString contains the mail user's other street address.

2.2.4.32 PidTagOtherAddressPostOfficeBox

The PidTagOtherAddressPostOfficeBox property of type PtypString contains the number or identifier of the mail user's other post office box.

2.2.4.33 PidTagOtherAddressCity

The PidTagOtherAddressCity property of type PtypString contains the name of the mail user's other locality, such as the town or city.

2.2.4.34 PidTagOtherAddressStateOrProvince

The PidTagOtherAddressStateOrProvince property of type PtypString contains the name of the mail user's other state or province.

2.2.4.35 PidTagOtherAddressPostalCode

The PidTagOtherAddressPostalCode property of type PtypString contains the postal code for the mail user's other postal address.

2.2.4.36 PidTagOtherAddressCountry

The PidTagOtherAddressCountry property of type PtypString contains the name of the mail user's other country/region.

2.2.4.37 PidTagPrimaryTelephoneNumber

The PidTagPrimaryTelephoneNumber property of type PtypString contains the mail user's primary telephone number.

2.2.4.38 PidTagBusinessTelephoneNumber

The PidTagBusinessTelephoneNumber property of type PtypString contains the primary telephone number of the mail user's place of business.

2.2.4.39 PidTagHomeTelephoneNumber

The PidTagHomeTelephoneNumber property of type PtypString contains the primary telephone number of the mail user's home.

2.2.4.40 PidTagBusiness2TelephoneNumber

The PidTagBusiness2TelephoneNumber property of type PtypString contains a secondary telephone number at the mail user's place of business.

2.2.4.41 PidTagBusiness2TelephoneNumbers

The PidTagBusiness2TelephoneNumbers property of type PtypMultipleString contains secondary telephone numbers at the mail user's place of business.

2.2.4.42 PidTagHome2TelephoneNumber

The PR PidTagHome2TelephoneNumber property of type PtypString contains a secondary telephone number at the mail user's home.

2.2.4.43 PidTagHome2TelephoneNumbers

The PR PidTagHome2TelephoneNumbers property of type PtypMultipleString contains secondary telephone numbers at the mail user's home.

2.2.4.44 PidTagCallbackTelephoneNumber

The PidTagCallbackTelephoneNumber property of type PtypString contains a telephone number to reach the mail user.

2.2.4.45 PidTagMobileTelephoneNumber

The PidTagMobileTelephoneNumber property of type PtypString contains the mail user's cellular telephone number.

2.2.4.46 PidTagRadioTelephoneNumber

The PidTagRadioTelephoneNumber property of type PtypString contains the mail user's radio telephone number.

2.2.4.47 PidTagCarTelephoneNumber

The PidTagCarTelephoneNumber property of type PtypString contains the mail user's car telephone number.

2.2.4.48 PidTagOtherTelephoneNumber

The PidTagOtherTelephoneNumber property of type PtypString contains an alternate telephone number for the mail user.

2.2.4.49 PidTagPagerTelephoneNumber

The PidTagPagerTelephoneNumber property of type PtypString contains the mail user's pager telephone number.

2.2.4.50 PidTagPrimaryFaxNumber

The PidTagPrimaryFaxNumber property of type PtypString contains the telephone number of the mail user's primary fax machine.

2.2.4.51 PidTagBusinessFaxNumber

The PidTagBusinessFaxNumber property of type PtypString contains the telephone number of the mail user's business fax machine.

2.2.4.52 PidTagHomeFaxNumber

The PidTagHomeFaxNumber property of type PtypString contains the telephone number of the mail user's home fax machine.

2.2.4.53 PidTagCompanyMainTelephoneNumber

The PidTagCompanyMainTelephoneNumber property of type PtypString contains the main telephone number of the mail user's company.

2.2.4.54 PidTagTelecommunicationsDeviceForDeafTelephoneNumber

The PidTagTelecommunicationsDeviceForDeafTelephoneNumber property of type PtypString contains the mail user's Telecommunications Device For The Deaf (TTYTDD) telephone number.

2.2.4.55 PidTagTelexNumber

The PidTagTelexNumber property of type PtypString contains the mail user's telex number.

2.2.4.56 PidTagIsdnNumber

The PidTagIsdnNumber property of type PtypString contains the mail user's ISDN-capable telephone number.

2.2.4.57 PidTagAssistantTelephoneNumber

The PidTagAssistantTelephoneNumber property of type PtypString contains the telephone number of the mail user's administrative assistant.

2.2.4.58 PidTagKeyword

The PidTagKeyword property of type PtypString contains a keyword identifying the mail user to the mail user's system administrator.

2.2.4.59 PidTagGovernmentIdNumber

The PidTagGovernmentIdNumber property of type PtypString contains a government identifier for the mail user.

2.2.4.60 PidTagMessageHandlingSystemCommonName

The PidTagMessageHandlingSystemCommonName property of type PtypString contains the common name of a messaging user for use in a message header.

2.2.4.61 PidTagLanguage

The PidTagLanguage property of type PtypString contains a value indicating the language in which the messaging user is writing messages.

2.2.4.62 PidTagLocation

The PidTagLocation property of type PtypString contains the location of the mail user in a format that is useful to the mail user's organization.

2.2.4.63 PidTagOrganizationalIdNumber

The PidTagOrganizationalIdNumber property of type PtypString contains an identifier for the mail user used within the mail user's organization.

2.2.4.64 PidTagUserCertificate

The PidTagUserCertificate property of type PtypBinary has been deprecated. This property MUST be ignored by clients.

2.2.4.65 PidTagAddressBookX509Certificate

The PidTagAddressBookX509Certificate property of type PtypMultipleBinary specifies ASN.1 DER encoded X.509 certificates for the mail user. Each binary value MUST be an ASN.1 DER encoded X.509 certificate, as specified in [RFC3280].

2.2.4.66 PidTagUserX509Certificate

The PidTagUserX509Certificate property of type PtypMultipleBinary specifies a list certificates for the mail user. Each binary value MUST be either an ASN.1 DER encoded SignedData Type BLOB which contains the user's certificates and is signed with the user's certificate, as specified in [RFC3852], or a binary property as specified in the following paragraph. To determine which of these choices each binary value is, the application MUST examine the first BYTE of each binary value. If the first byte has the value 0x30, it is an ASN.1 DER encoded SignedData Type BLOB. Otherwise, the binary value MUST be interpreted according to the format specified in this section. Whenever possible, the client and

the server SHOULD use the PidTagUserX509Certificate whenever looking for certificates instead of PidTagAddressBookX509Certificate.

Non-ASN.1 Binary Value Format

If the binary value is not an ASN.1 DER encoded SignedData Type BLOB, then it MUST be a BLOB containing a set of security settings as specified in sections 2.2.4.66.1.1 through 2.2.4.66.1.12, one after another, in a continuous block of data. All settings in these sections MUST appear at most once in the binary value unless stated otherwise. Each security setting has the following format:

0	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	2	0	1	2	3	4	5	6	7	8	9	3	0	1
TAG										LENGTH																								
DATA (variable)																																		

TAG (PtypInteger16): An unsigned value that identifies this particular security setting. If the TAG found in a security setting is not among the ones listed in the sections 2.2.4.66.1.1 through 2.2.4.66.1.12, the client and the server MUST ignore these settings.

LENGTH (PtypInteger16): The total length of the security setting, including the TAG field, the LENGTH field and the DATA field

DATA (variable structure): Contains the data associated with this security setting. Its length, in bytes, can be computed from the value of the LENGTH field.

The following sections specify the security settings that appear in a non-ASN.1 certificate.

2.2.4.66.1.1 Property Version

The TAG for this setting is 0x0001. The LENGTH for this setting MUST be 0x0008. The DATA specifies the version of the certificate. The unsigned PtypInteger32 DATA MUST be 0x00000001. This setting MUST be part of each PtypBinary value.

2.2.4.66.1.2 Encryption Type

The TAG for this setting is 0x0006. The LENGTH for this setting MUST be 0x0008. The unsigned PtypInteger32 DATA specifies the type of encryption to be used with this certificate. A value of 0x00000001 specifies that the encryption type is S/MIME, as specified in [RFC3852]. A DATA value of 0x00000006 specifies that the encryption type is Fortezza, as specified in [RFC2876]. All other values have no defined meaning and MUST be ignored by both the server and the client. This setting MUST be part of each PtypBinary value.

2.2.4.66.1.3 Defaults

The TAG for this setting is 0x0020. The LENGTH for this setting MUST be 0x0008. The DATA is a 4-byte bit field which can contain any combination of the following bit values:

Bit	Meaning
0x00000001	This is the default certificate for S/MIME.
0x00000002	This is the default certificate for all formats. If this bit is set, then 0x00000001 MUST be set.

Any other bit flags set on this setting MUST be ignored. This setting MUST be part of each PtypBinary value.

2.2.4.66.1.4 ASCII Display Name

The TAG for this setting is 0x000B. The LENGTH for this setting MUST be larger than 4. The DATA field contains the user-readable name, as a NULL terminated ASCII string. Either the ASCII display name or the Unicode display name (as specified in section 2.2.4.66.1.5) SHOULD be part of the binary value, but not both <6>.

2.2.4.66.1.5 Unicode Display Name

The TAG for this setting is 0x0051. The LENGTH for this setting MUST be larger than 4. The DATA field contains the user-readable name, as a NULL-terminated Unicode string. Either the Unicode display name or the ASCII display name SHOULD be part of the certificate, but not both <6>.

2.2.4.66.1.6 KeyExSHA1Hash

The TAG for this setting is 0x0022. The LENGTH for this setting MUST be larger than 4. The DATA field contains the SHA1 Hash as specified in [RFC3174] to look up the encryption certificate in the certificate store on the user's computer. The certificate that matches this hash SHOULD be used when sending encrypted mail to the contact.

For a given PtypBinary value in PidTagUserX509Certificate, if the KeyExSHA1Hash setting is present, then KeyExchangeCertificate MUST NOT be present.

Whenever possible, the client and the server SHOULD use the KeyExchangeCertificate setting instead of the KeyExSHA1Hash setting.

2.2.4.66.1.7 SignSHA1Hash

The TAG for this optional setting is 0x0009. The LENGTH for this setting MUST be larger than 4. The DATA field contains the SHA1 Hash as specified in [RFC3174] to look up the signing certificate in the certificate store on the user's computer<6>.

For a given PtypBinary value in PidTagUserX509Certificate, if the SignSHA1Hash setting is present, then SignCertificate MUST NOT be present.

Whenever possible, the client and the server SHOULD use the SignCertificate setting instead of the SignSHA1Hash setting.

2.2.4.66.1.8 KeyExchangeCertificate

The TAG for this setting is 0x0003. The LENGTH for this setting MUST be larger than 4. The DATA field contains an ASN.1 DER encoded X.509 certificate, as specified in [RFC3280]. This certificate SHOULD be used when sending encrypted S/MIME mail to the contact.

For a given PtypBinary value in PidTagUserX509Certificate, if the KeyExchangeCertificate setting is present, then KeyExSHA1Hash MUST NOT be present. If KeyExSHA1Hash is not present, then KeyExchangeCertificate MUST be present.

Whenever possible, the client and the server SHOULD use the KeyExchangeCertificate setting instead of the KeyExSHA1Hash setting.

2.2.4.66.1.9 SignCertificate

The TAG for this optional setting is 0x0008. The LENGTH for this setting MUST be larger than 4. The DATA field contains an ASN.1 DER encoded X.509 certificate, as specified in [RFC3280] <6>.

For a given PtypBinary value in PidTagUserX509Certificate, if the SignCertificate setting is present, then SignSHA1Hash MUST NOT be present.

Whenever possible, the client and the server SHOULD use the SignCertificate setting instead of the SignSHA1Hash setting.

2.2.4.66.1.10 ChainCertificate

The TAG for this optional setting is 0x0004. The LENGTH for this setting MUST be larger than 4. This setting can appear multiple times in the PtypBinary value to provide multiple certificates to use in the building of chains. The DATA field contains an ASN.1 DER encoded X.509 certificate, as specified in [RFC3280]. If present, these optional certificates SHOULD be used to build the chain of trust for the certificate specified by KeyExchangeCertificate.

2.2.4.66.1.11 AsymmetricCapabilities

The TAG for this setting is 0x0002. The LENGTH for this setting MUST be larger than 4. The DATA field contains the ASN.1 DER encoded sMIMECapabilities type, as specified in [RFC3851]. This setting MUST be part of the PtypBinary value.

2.2.4.66.1.12 SavedTime

The TAG for this optional setting is 0x000D. The LENGTH for this setting MUST be 0x000C. The DATA MUST be a PtypFloatingTime that specifies the time (in UTC) when the certificate was added to the contact.

2.2.4.67 PidTagAddressBookHomeMessageDatabase

The PidTagAddressBookHomeMessageDatabase property of type PtypString of a mail user is the DN, expressed in the x500-dn format specification described in section 2.2.1.1, of an address book object that represents the mail user's message database. The x500-container-dn

portion of the DN is an x500-dn string that references an address book object that represents the mail user's mailbox server. The rdn in the object-rdn of the x500-dn for the mailbox server's address book object is the host-name, as described in [RFC1034], of the server containing the mail user's message database.

2.2.4.68 PidTagAddressBookNetworkAddress

The PidTagAddressBookNetworkAddress property of type PtypMultipleString of an address book object of a mailbox server contains a list of names by which a server is known to the various transports in use by the network. Each PtypString is a RPC protocol sequence, as specified in [MS-RPCE], followed by a colon (":"), followed by the host-name of the server under that RPC protocol sequence.

2.2.4.69 PidTagHobbies

The PidTagHobbies property of type PtypString contains the names of the mail user's hobbies.

2.2.4.70 PidTagProfession

The PidTagProfession property of type PtypString contains the name of the mail user's line of business.

2.2.4.71 PidTagReferredByName

The PidTagReferredByName property of type PtypString contains the name of the mail user's referral.

2.2.4.72 PidTagSpouseName

The PidTagSpouseName property of type PtypString contains the name of the mail user's spouse/partner.

2.2.4.73 PidTagGender

The PidTagGender short property contains a value representing the mail user's gender. If present, it **MUST** contain one of the following values:

Value	Description
0x00000000	No gender is specified.
0x00000001	Specifies a gender of female.
0x00000002	Specifies a gender of male.

2.2.4.74 PidTagComputerNetworkName

The PidTagComputerNetworkName property of type PtypString contains the name of the mail user's computer network.

2.2.4.75 PidTagCustomerId

The PidTagCustomerId property of type PtypString contains the mail user's customer identification number.

2.2.4.76 PidTagFtpSite

The PidTagFtpSite property of type PtypString contains the mail user's File Transfer Protocol (FTP) site address.

2.2.4.77 PidTagPersonalHomePage

The PidTagPersonalHomePage property of type PtypString contains web address (URL) of the mail user's personal home page.

2.2.4.78 PidTagBusinessHomePage

The PidTagBusinessHomePage property of type PtypString contains web address (URL) of the mail user's business home page.

2.2.4.79 PidTagBirthday

The PidTagBirthday property of type PtypTime contains the date of the mail user's birthday at 12:00 AM UTC.

2.2.4.80 PidTagWeddingAnniversary

The PidTagWeddingAnniversary property of type PtypTime contains the date of the mail user's wedding anniversary at 12:00 AM UTC.

2.2.5 Properties That Reference Other Address Book Objects

Some address book objects contain references to other address book objects, in the form of a table, through various properties. Please see "Property Value based Explicit Tables" in [MS-NSPI] for a description of these tables. Each of these properties is of type PtypEmbeddedTable, as specified in [MS-NSPI]. For example, a mail user in an organization reports to a manager which is also listed in that address book as another mail user object. In such a case, there is a property of type PtypEmbeddedTable, PidTagAddressBookManager, which references their manager. In general, a property of type PtypEmbeddedTable references any number of other address book objects. Properties of type PtypEmbeddedTable are made available on an NSPI server using a property value based explicit table, through the call NspiGetMatches, and modified through NspiModLinkAtt, as specified in [MS-NSPI]. The structure for an Offline Address Book does not specify any way to store properties of type PtypEmbeddedTable.

2.2.5.1 PidTagAddressBookManager

The PidTagAddressBookManager property of type PtypEmbeddedTable of a mail user contains one row referencing the mail user's manager.

2.2.5.2 PidTagAddressBookReports

The PidTagAddressBookReports property of type PtypEmbeddedTable of a mail user is a list of all the mail user's direct reports.

2.2.5.3 PidTagAddressBookIsMemberOfDistributionList

The PidTagAddressBookIsMemberOfDistributionList property of type PtypEmbeddedTable of an address book object that lists all the distribution lists for which this object is a member.

2.2.5.4 PidTagAddressBookOwnerBackLink

The PidTagAddressBookOwnerBackLink property of type PtypEmbeddedTable of a mail user that lists the distribution lists that this mail user owns.

2.2.5.5 PidTagAddressBookPublicDelegates

The PidTagAddressBookPublicDelegates property of type PtypEmbeddedTable contains a list of mail users allowed to send mail on behalf of the mailbox owner.

2.2.5.6 PidTagAddressBookHierarchicalShowInDepartments

The PidTagAddressBookHierarchicalShowInDepartments property of type PtypEmbeddedTable of a mail user lists all the department objects that this mail user is a member.

2.2.6 Properties That Apply to Distribution Lists

2.2.6.1 PidTagAddressBookMember

The PidTagAddressBookMember property of type PtypEmbeddedTable of a distribution list that lists the members of the distribution list.

2.2.6.2 PidTagAddressBookOwner

The PidTagAddressBookOwner property of type PtypEmbeddedTable of a distribution list contains one row referencing the distribution list's owner.

2.2.6.3 PidTagContainerContents

The PidTagContainerContents property of type PtypEmbeddedTable of a distribution list is always empty. An NSPI server MUST define this value for distribution lists, and not be present for all other objects.

2.2.6.4 PidTagAddressBookFolderPathname

The PidTagAddressBookFolderPathname property of type PtypString has been deprecated and MUST be ignored by clients.

2.2.7 Properties That Apply to Organization Objects

An organization object is an address book object that represents an organization, and contains properties specific to organization objects. Messaging clients access the organization object of a mail user by extracting the *org-rdn* string from the mail user's DN, which is in the format of an X500 address and the *x500-dn* format specification described in section 2.2.1.1. Messaging clients use the resulting *org-rdn* string as a DN of the mail user's organization, which is in the *organization-dn* format specification. Once the organization object is obtained, if present, messaging clients are able to perform operations as they would any other address book object.

2.2.7.1 PidTagAddressBookRoomContainers

The PidTagAddressBookRoomContainers property of type PtypMultipleString contains a list of DNs that represent the address book containers that hold resources, such as conference rooms and equipment. Messaging clients use this list to determine which containers contain mainly resource objects in order to do special handling on these containers, such as displaying a different column set when browsing address lists represented by these containers, or for features that require choosing among a set of rooms or equipment.

The DNs in the PidTagAddressBookRoomContainers property each use the *x500-dn* format specification described in section 2.2.1.1, with the additional requirement that there is no *container-rdn* in its *x500-dn* and that the *rdn* of the object-*rdn* follows the *container-guid* format specification. When the *rdn* is extracted from the DN, it represents the GUID of the address book container this is referencing. Messaging clients compare this GUID to the GUID of another address book container by extracting that GUID from the *container-guid* of the other container's DN, which follows the *addresslist-dn* format specification. If the GUIDs are the same, then the other address book container is a room container.

2.2.7.2 PidTagAddressBookHierarchicalRootDepartment

The PidTagAddressBookHierarchicalRootDepartment property of type PtypEmbeddedTable is a reference to the root department object in the department hierarchy for the organization. The table has either zero or one row, referencing a department object. If either the organization object is missing, or this property is missing or the property value is empty, then the NSPI server does not have a department hierarchy for that organization.

2.2.8 Properties That Apply to Department Objects

A department object is an address book object that represents a department within an organization, and contains properties specific to a department object.

Messaging clients obtain the root of the department hierarchy using the property PidTagAddressBookHierarchicalRootDepartment of the organization object, or by obtaining the root department that is not specific to any organization, which has a DN specified by using

the organization-dn format specification described in section 2.2.1.1 with a value of “/o=FF46312B-D8AE-406C-B8E6-BC1A22A4C69E”.

2.2.8.1 PidTagAddressBookHierarchicalChildDepartments

The PidTagAddressBookHierarchicalChildDepartments property of type PtypEmbeddedTable on a department object references the child departments in a hierarchy of departments.

2.2.8.2 PidTagAddressBookHierarchicalParentDepartment

The PidTagAddressBookHierarchicalParentDepartment property of type PtypEmbeddedTable on a department object references all the departments to which this department is a child.

2.2.8.3 PidTagAddressBookHierarchicalDepartmentMembers

The PidTagAddressBookHierarchicalDepartmentMembers property of type PtypEmbeddedTable of a department object lists all the mail users that belong to this department.

2.2.9 Properties That Apply to Resources

2.2.9.1 PidTagAddressBookRoomCapacity

The PidTagAddressBookRoomCapacity property of type PtypInteger32 is the maximum occupancy of the room.

2.2.9.2 PidTagAddressBookRoomDescription

The PidTagAddressBookRoomDescription property of type PtypString is a description of the resource.

2.2.10 Properties That Have Special Purposes

2.2.10.1 PidTagAnr

The PidTagAnr property of type PtypString is a special property that is not actually a property on the address book objects themselves. Rather, this property is used by messaging clients as a property value to a Filter to the NSPI call NspiGetMatches (see [MS-NSPI]<7>. Messaging clients pass this property as a target string to NspiGetMatches to identify objects in an address list that are a possible match for the target string. This operation is known as Ambiguous Name Resolution (ANR). NSPI servers respond by returning the Minimal Entry IDs of all address book objects that are possible matches against the target string. This protocol does not prescribe the choice of Ambiguous Name Resolution results of an NSPI server<8>. Please see [MS-NSPI] for more information about Ambiguous Name Resolution.

An Offline Address Book does not contain this property for any address book objects. It is up to messaging clients to determine how to perform name matching among the objects in an Offline Address Book.

2.2.10.2 PidTagAddressBookManageDistributionList

The PidTagAddressBookManageDistributionList object property is a property tag for use in display templates for distribution lists. When PidTagAddressBookManageDistributionList is the dwType property tag in a Button control to a CNTRL structure of a display template, it tells messaging clients to include a button to enable UI to edit the members of a distribution list. See [MS-OXOABKT] for information on Button control for a template. This is not a property of objects in an address book.

2.2.11 Named Properties

This protocol specification specifies no named properties. Offline Address Books and NSPI servers are free to expose any named properties in their implementation.

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

The address book contains one or more address book containers. The collection of address book containers is arranged in an address book hierarchy table. An address book container represents an address list, which is a collection of address book objects rendered in a table and browsed by a messaging user. Among other things, messaging users are able to scroll through this table of objects, set positioning based on a name typed by a messaging user, perform searches, and perform Ambiguous Name Resolution (ANR) against the names in that address list. Ambiguous Name Resolution means the user types part of a name, and the address book identifies potential matches for that name.

Messaging users are able to display information about an address book object from the table, an object obtained through Ambiguous Name Resolution, or a recipient on a message. The information displayed is obtained from a display template supplied in the address book, and depends on the type of address book object being displayed.

Among the many types of objects, an address book typically includes:

- mail users, each of which describes a person or entity that can receive e-mail;
- distribution lists, each of which is a collection of other mail users, distribution lists, or other address book objects that can receive e-mail;
- resources, which can be reserved such as a room or equipment;
- organizations, each of which describes an organization; departments, which describes the departmental structure of an organization;
- departments, each of which describes the departmental structure of an organization
- address book containers, each of which represents an address list containing address book objects that can be viewed as a table; and

- templates, each of which describes a physical view that can be used to show details on other address book objects to a messaging user, and is specified in more detail in [MS-OXOABKT].

3.1.2 Timers

There are no timers specific to this protocol.

3.1.3 Initialization

Initialization is accomplished in two possible ways: either through a NSPI connection to a server via NSPI, through the contents of an Offline Address Book, or both.

3.1.3.1 Initialization Through NSPI connection

Initialization is accomplished via NspiBind, as specified in [MS-NSPI], and MUST occur before any address book objects are able to be accessed through NSPI calls.

3.1.3.2 Initialization Through an Offline Address Book

Messaging clients obtain address lists in an Offline Address Book using the Offline Address Book Web and Public Folder Retrieval protocols, specified in [MS-OXWOAB] and [MS-OXPFOAB]. Each address list, in turn, contains information about objects in that address list, using the Offline Address Book (OAB) Format and Schema structure, as specified in [MS-OXOAB]. Once one or more address lists from an Offline Address Book are obtained, messaging clients navigate any of the address lists to perform the lookups needed to perform these operations, as specified in [MS-OXOAB].

3.1.4 Higher-Layer Triggered Events

It is possible for messaging clients to obtain a hierarchy of address book containers, browse the address book objects in an address list represented by an address book container, obtain information about an address book object, and perform Ambiguous Name Resolution to find address book objects that match a target string.

3.1.4.1 Obtaining a Hierarchy of Address Book Containers

When using an NSPI server, messaging clients obtain the hierarchy using NspiGetSpecialTable, as specified in [MS-NSPI]. The table returns several rows of containers, and for each row, returns the properties PidTagDisplayName, PidTagEntryId, PidTagContainerFlags, PidTagDepth, PidTagAddressBookContainerId, PidTagAddressBookIsMaster, and PidTagAddressBookParentEntryId which are needed to determine the hierarchy of containers.

The hierarchy table in an Offline Address Book is obtained using the Offline Address Book Web and Public Folder Retrieval protocols, specified in [MS-OXWOAB] and [MS-OXPFOAB].

3.1.4.2 Browsing an Address Book

Once the hierarchy of containers is obtained, and a single container is chosen by a messaging user, it is possible for the messaging user to browse the address list represented by the container.

When browsing an address list for an Offline Address Book, it is up to the messaging client to perform the lookups necessary to present a browse-able view on the address list, using the Offline Address Book (OAB) Format and Schema structure, as specified in [MS-OXOAB].

When browsing an address list from an NSPI server, messaging clients use the call `NspiQueryRows` to obtain a set of rows to display to the messaging user. Messaging clients choose the properties they wish to render, but would normally request `PidTagEntryId`, `PidTagDisplayName`, `PidTagSmtpAddress`, `PidTagTitle`, among other properties that the messaging client deems useful for displaying to the user.

It is possible to scroll to certain approximate positions in the address book view. When using NSPI, this is accomplished by modifying the STAT structure, which describes a table position as specified in [MS-NSPI], and calling `NspiUpdateStat`. Messaging clients usually follow up by calling `NspiQueryRows` to display rows starting at the new position.

It is possible for the user to type a certain name, and have the address list view scroll to the first display name typed, like a rolodex. When using NSPI, this is accomplished through the call `NspiSeekEntries`, as specified in [MS-NSPI]. The result updates positioning information in the STAT structure, specified in [MS-NSPI], and returns a screen full of address book objects.

3.1.4.3 Obtaining Properties on an Address Book Object

In order to obtain properties for an address book object from an Offline Address Book, the messaging client needs to have a Distinguished Name (DN) for the object, or the object's SMTP address. This means the messaging client will need to keep track of the DN after any lookup in the Offline Address Book, and will need to make sure that the DN or SMTP address is maintained for the addressee of a message when messages are sent or received through a messaging server. Generally, incoming messages contain either the SMTP address or the DN of message addressees and is provided by the messaging server. The Offline Address Book (OAB) Format and Schema structure document [MS-OXOAB] describes how this information is organized and structured. Once the information for the address book object is located, it is possible for the client to retrieve the value of any property in that record. For example, to obtain properties such as the business phone number and SMTP address, the client looks up `PidTagBusinessTelephoneNumber` and `PidTagSmtpAddress` in the record.

In order to obtain properties for an address book object on an NSPI server, the messaging client needs to either have a Distinguished Name (DN), a Minimal Entry ID, or the object's SMTP address. Certain NSPI calls that return address book object information, such as `NspiQueryRows`, `NspiSeekEntries`, or `NspiGetMatches`, etc, includes the DN or Minimal Entry ID in its `PidTagEntryId`, or as a return value of the call. Clients call `NspiDNToMid` to obtain the Minimal Entry ID from a DN. To obtain a DN from an SMTP address, messaging clients construct a string starting with "=`SMTP:`" followed by the SMTP address as the `paStr`

or paWStr parameter to the call NspiResolveNames or NspiResolveNamesW, as specified in [MS-NSPI], and extract it from the PidTagEntryId property returned, as specified in the Permanent Entry ID structure of [MS-NSPI].

Once the Minimal Entry ID is known, clients use it as the CurrentRec member of the STAT structure passed to NspiGetProps, as specified in [MS-NSPI]. It is possible for clients to specify any number of properties in the pPropTags parameter to NspiGetProps that the client wants to request. For example, to request properties such as the business phone number and SMTP address, the client includes the property IDs PidTagBusinessTelephoneNumber and PidTagSmtpAddress in the pPropTags parameter.

3.1.4.4 Performing Ambiguous Name Resolution (ANR)

Messaging clients perform Ambiguous Name Resolution when a user is addressing a message and types only part of a name. The role of the address book in this case is to identify the best possible matches for the name entered, and if there is more than one good match, present the list of possible address book objects to the user.

When performing Ambiguous Name Resolution using an Offline Address Book, it is up to the messaging client to decide the best method for name matching among address book objects in the Offline Address Book. The client is free to use whatever means to decide on good matches on any choice of properties using the Offline Address Book (OAB) Format and Schema structure, specified in [MS-EXOAB].

When performing Ambiguous Name Resolution using an NSPI server, the client calls NspiResolveNames or NspiResolveNamesW passing the target string in the parameter paStr or paWStr, as specified in [MS-NSPI]. If there is exactly one match, then the row corresponding to the target string will contain the properties requested for that match. If there is more than one match, then it is possible to obtain the set of possible matches by calling NspiGetMatches, as specified in [MS-NSPI], using a Filter with the property PidTagAnr, and the value for PidTagAnr as the target string for Ambiguous Name Resolution<7>. The set of matches, along with the requested properties for the matches, will be returned. Messaging clients usually display the returned results in a dialog to the user, so the user is able to choose the best match from the list of results.

3.1.5 Message Processing Events and Sequencing Rules

There are no messages specific to this protocol.

3.1.6 Timer Events

There are no timers specific to this protocol.

3.1.7 Other Local Events

There are no local events specific to this protocol.

3.2 Client and Server Details

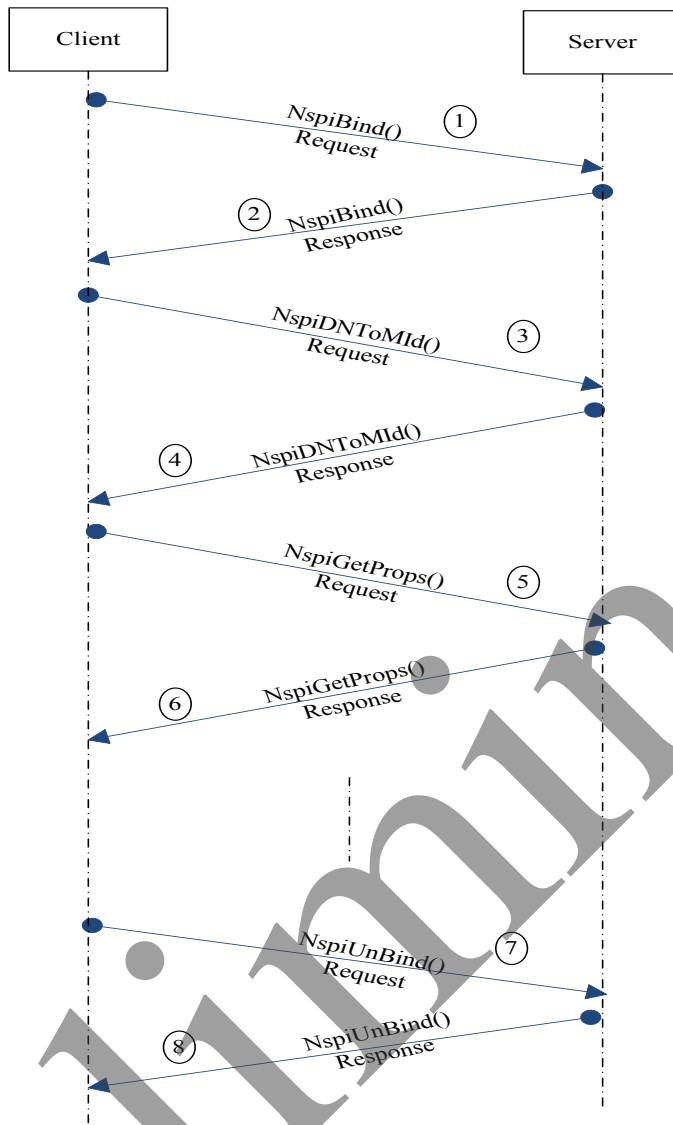
A messaging client uses either an Offline Address Book or a NSPI server, or both, to access an address book. A client obtains one or more address lists from an Offline Address Book using the Offline Address Book Web and Public Folder Retrieval protocols as specified in [MS-OXWOAB] and in [MS-OXPFOAB]. Usually, the client downloads the contents of address lists from the Offline Address Book onto the local machine into one or more files. Messaging clients access information about the address book objects contained in an address list from an Offline Address Book using the Offline Address Book (OAB) Format and Schema structure specified in [MS-OXOAB]. Alternatively, the client accesses information about objects in an address book using an NSPI server, using the Name Service Provider Interface (NSPI) protocol specified in [MS-NSPI]. Because both data sources represent the same address book, information about an address book object that is contained on both an NSPI server and in an Offline Address Book SHOULD be identical, with the Offline Address Book containing a subset of information available on the NSPI server. As the PidTagEmailAddress, which matches the DN for an address book object, is used to identify that object, the value for PidTagEmailAddress MUST match for an object contained in both data sources.

The role of the server that contains an Offline Address Book is to supply all the necessary information about the address book objects using the Offline Address Book (OAB) Format and Schema structure specified in [MS-OXOAB] and deliver them to the client using the Offline Address Book Web and Public Folder Retrieval protocols specified in [MS-OXWOAB] and [MS-OXPFOAB]. The server provides periodic updates to the Offline Address Book, as necessary. The updating mechanism is specified in [MS-OXOAB]. Once the information is delivered to the client, the server plays no role.

The role of the NSPI server is to service the requests of the client, using the Name Service Provider Interface (NSPI) protocol as specified in [MS-NSPI]. The NSPI server does not initiate any requests, but MAY drop a client connection that has remained idle for too long, has become disconnected due to a transient network error, or when the server needs to drop the connection to make other resources available. If a connection is dropped, the server MUST return an error when the messaging client requests information using an NSPI call using the dropped connection. Clients respond to the error by reestablishing that connection using NspiBind, as specified in [MS-NSPI].

4 Protocol Examples

This section illustrates the call sequences of obtaining two string properties namely PidTagDisplayName and PidTagGivenName of a mail user whose Distinguished Name is *"/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=user1"*.



Note: In this illustration of calls, not all parameters are shown, only relevant information is shown. For more details about the parameter information, see [MS-NSPI].

1. Client initiates a session to the NSPI server by calling NspiBind(). The messaging client sends in the following values to the server that looks like the following:

dwFlags	0x00000000	DWORD	
pStat			
hIndex	0x00000000	unsigned long	
ContainerID	0x00000000	unsigned long	
CurrentRec	0x00000000	unsigned long	

Delta	0x00000000	long
NumPos	0x00000000	unsigned long
TotalRecs	0x00000000	unsigned long
CodePage	0x000004e4	unsigned long
TemplateLocale	0x00000409	unsigned long
SortLocale	0x00000409	unsigned long

pServerGuid

pointer to an array of 16 unsigned char to be returned by the server

2. The server responds to NspiBind call with return code Success and a valid Server Guid.

Typical parameters will look like the following:

pServerGuid

[0x0] 0xab 0xbc 0x8b 0x86 0x79 0x33 0xc4 0x48 0xa1 0xef

[0xa] 0x1b 0x53 0xe6 0x3b 0xdc 0x46

contextHandle

< a token which will be used by the NspiUnbind call >

3. The client requests Minimal EntryId for the distinguished name “/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=user1” by calling NspiDNToMid() with parameters that typically look like the following:

m_pNames

Count 0x00000001 DWORD

Strings char **

[0x0] char *

“/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=user1”

ppMids

< a pointer a valid memory location of type PropertyTagArray_r ** for server to return Mid >

4. The server responds with return code Success with a value of 0x00001927 for Mid that typically looks like the following:

m_pNames

Count 0x00000001 DWORD

Strings char **

[0x0] char *

“/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=user1”

ppMids

```

cValues      0x00000001    DWORD
aulPropTag   DWORD[]
              [0x0]
              0x00001927    DWORD

```

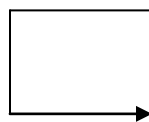
5. The client requests two string properties PidTagDisplayName and PidTagGivenName by calling NspiGetProps with parameters that typically looks like the following:

[Note: MId 0x00001927 value obtained in step 4 is used as CurrentRec field of pStat]

```

hRpc    <a valid RPC handle>    void *
dwFlags 0x00000000    DWORD
pStat
  hIndex      0x00000000    unsigned long
  ContainerID 0x00000000    unsigned long
  CurrentRec  0x00001927    unsigned long
  Delta       0x00000000    long
  NumPos      0x00000000    unsigned long
  TotalRecs   0x00000000    unsigned long
  CodePage    0x000004b0    unsigned long
  TemplateLocale 0x00000409    unsigned long
  SortLocale  0x00000409    unsigned long
pPropTags  _SPropTagArray_r *
  {
    cValues    0x00000002    DWORD
    aulPropTag=<a pointer to an array of proptags>
  }
  aulPropTag  unsigned long []
    [0x0]     PidTagDisplayName    unsigned long
    [0x1]     PidTagGivenName      unsigned long
ppRow       _SRow_r * *
  <memory location for server return values>

```



6. The server responds to NspiGetProps with return code Success. In this example, the server has returned us the string value "user1" for both the property requested and the return values typically look like the following:

```

dwFlags 0x00000000    DWORD
pStat
  {
    hIndex      0x00000000    unsigned long

```

```

ContainerID      0x00000000      unsigned long
CurrentRec      0x00001927      unsigned long
Delta           0x00000000      long
NumPos         0x00000000      unsigned long
TotalRecs      0x00000000      unsigned long
CodePage       0x000004b0      unsigned long
TemplateLocale 0x00000409      unsigned long
SortLocale     0x00000409      unsigned long

```

```

}
pPropTags      _SPropTagArray_r *
{

```

```

    cValues      0x00000002      DWORD
    aulPropTag=<a pointer to an array of proptags>
}
    aulPropTag      unsigned long []
                    [0x0]  PidTagDisplayName      unsigned long
                    [0x1]  PidTagGivenName       unsigned long

```

```

ppRows      _SRowSet_r **
{
    cRows      0x00000001      DWORD
    aRow=<a pointer to an array of rows>
}

```

In this example, the server has returned a total of 0x1 row denoted as [0x0] that typically looks like the following:

```

aRow [0x0] _SRow_r *
{
    cValues      0x00000002      DWORD
    lpProps=<a pointer to an array of columns>
}

```

In this example, server has returned a column set of 2 properties and each column will look like –the following:

```

[0x0] _SPropValue_r
{
    [0x0]
        ulPropTag      PidTagDisplayName      unsigned long
        Value
        lpszA = "user1"      char *

```

```

[0x1]
    ulPropTag          PidTagGivenName          unsigned long
    Value
        lpszA = "user1"          char *
    }

```

[Note: The client MAY invoke additional NSPI calls to access other information from the server before calling NSPIUnbind().]

7. The client terminates the connection by calling NspiUnbind with a token that the server returned in response to NspiBind call.

```

contextHandle          NSPI_HANDLE *
    <a token which was sent by the server in the NspiBind call >
dwFlags
    0x00000000          unsigned long

```

8. The server responds with return code 0x00000001 and destroys the token that the client passed in.

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to the Address Book Object protocol. General security considerations pertaining to the underlying NSPI RPC-based transport apply (see [MS-NSPI]).

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 that Office/Exchange does not follow the prescription.

<1> The Offline Address Book implemented in Exchange 2003 SP2 and Exchange 2007 SP1 contains all the properties listed in the appendices of [MS-OXOAB]. This MAY include some properties not present on some implementations of an NSPI server.

<2> The Offline Address Book Version 4 implemented in Exchange 2003 SP2 and Exchange 2007 SP1 includes a property PidTagOabTruncatedProps that is not included in objects in a NSPI server. This property is further described in [MS-OXOAB].

<3> The values for string and binary properties may be truncated in an Offline Address Book according to the limitations specified in [MS-OXOAB].

<4> The values for properties in an Offline Address Book may have a different value if the value on a NSPI server has changed since the Offline Address Book was created, or if the NSPI server was restored from a backup after the Offline Address Book was created. In such a case, the NSPI server and the Offline Address Book are said to be “out of sync.” That is, the data in each source reflects a different time period.

<5> The Exchange 2003 SP2 and Exchange 2007 SP1 NSPI server includes the additional properties described in section 2.2.6. Certain properties MAY be restricted from access by clients, such as a distribution lists where the members are hidden.

<6> Sections 2.2.4.66.1.4, 2.2.4.66.1.5, 2.2.4.66.1.7, and 2.2.4.66.1.9: These settings are not used by either Outlook or Exchange

<7> Outlook 2003 SP3 and Outlook 2007 SP1 specify an Ambiguous Name Resolution search by setting up a Filter to NspiGetMatches, as specified in [MS-NSPI]. The Filter is a Restriction_r structure, whose members are set up according to the following table:

Member	Value
Filter.rt	0x00000004 (same as RES_PROPERTY as specified in [MS-OXCDATA])
Filter.res.resProperty.relop	0x00000004 (same as RELOP_EQ as specified in [MS-OXCDATA])
Filter.res.resProperty.ulPropTag	PidTagAnr (PtypString8 or PtypString)
Filter.res.resProperty.lpProp.ulPropTag	Same as Filter.res.resProperty.ulPropTag
Filter.res.resProperty.lpProp.Value.lpszA	Target string in code page of pStat.CodePage, if PtypString8 specified.
Filter.res.resProperty.lpProp.Value.lpszW	Target string in Unicode, if PtypString specified.

<8> Outlook 2003 SP3 and Outlook 2007 SP1 performs a lookup of SMTP addresses by constructing a specific target string for Ambiguous Name Resolution that is understood by Exchange 2003 SP2 and Exchange 2007 SP1 NSPI servers. When the target string starts with “=SMTP:” and is followed by a valid SMTP address, the NSPI server returns exactly one match, if any are found. The match will be an address book entry that has the target string as a valid SMTP address. Such a target string can be specified as a PidTagAnr property restriction to NspiGetMatches, described in <7>. Such a target string is also understood by the paStr parameter to NspiResolveNames or the paWStr parameter to NspiResolveNamesW.

7 Index

Applicability statement, 10
Client and server details, 43
Client details, 39
Glossary, 5
Index of security parameters, 48
Informative references, 8
Introduction, 5
Message syntax, 10
Messages, 10
 Message syntax, 10
 Transport, 10
Normative references, 7
Office/Exchange behavior, 48
Prerequisites/preconditions, 9
Protocol details, 39
 Client and server details, 43
 Client details, 39
Protocol examples, 43
Protocol overview (synopsis), 8
References, 7
 Informative references, 8
 Normative references, 7
Relationship to other protocols, 9
Security, 48
 Index of security parameters, 48
 Security considerations for implementers, 48
Security considerations for implementers, 48
Standards assignments, 10
Transport, 10
Vendor-extensible fields, 10
Versioning and capability negotiation, 10