[MS-OXOABK]:

Address Book Object Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **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 technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@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. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	Major	Initial Availability.
4/25/2008	0.2	Minor	Revised and updated property names and other technical content.
6/27/2008	1.0	Major	Initial Release.
8/6/2008	1.0.1	Editorial	Updated references to reflect date of initial release.
9/3/2008	1.0.2	Editorial	Revised and edited technical content.
12/3/2008	1.0.3	Editorial	Revised and edited technical content.
3/4/2009	1.0.4	Editorial	Revised and edited technical content.
4/10/2009	2.0	Major	Updated technical content and applicable product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	3.1.0	Minor	Updated the technical content.
2/10/2010	3.2.0	Minor	Updated the technical content.
5/5/2010	4.0.0	Major	Updated and revised the technical content.
8/4/2010	4.1	Minor	Clarified the meaning of the technical content.
11/3/2010	4.2	Minor	Clarified the meaning of the technical content.
3/18/2011	5.0	Major	Significantly changed the technical content.
8/5/2011	6.0	Major	Significantly changed the technical content.
10/7/2011	6.1	Minor	Clarified the meaning of the technical content.
1/20/2012	7.0	Major	Significantly changed the technical content.
4/27/2012	8.0	Major	Significantly changed the technical content.
7/16/2012	8.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	8.1	Minor	Clarified the meaning of the technical content.
2/11/2013	9.0	Major	Significantly changed the technical content.
7/26/2013	10.0	Major	Significantly changed the technical content.
11/18/2013	11.0	Major	Significantly changed the technical content.
2/10/2014	11.0	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	12.0	Major	Significantly changed the technical content.
7/31/2014	12.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	12.1	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
3/16/2015	13.0	Major	Significantly changed the technical content.
5/26/2015	14.0	Major	Significantly changed the technical content.
9/14/2015	14.0	No Change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Address Book Object Protocol provides implementers with information about all the properties defined on various Address Book objects and how the properties on **Address Book objects** interrelate with one another.

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

1.1 Glossary

The following terms are specific to this document:

address book: A collection of **Address Book objects**, each of which are contained in any number of **address lists**.

address book container: An **Address Book object** that describes an **address list**.

address book hierarchy table: A collection of **address book containers** arranged in a hierarchy.

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

address list: A collection of distinct **Address Book objects**.

address type: An identifier for the type of email address, such as **SMTP** and EX.

alias: An alternate name that can be used to reference an object or element.

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

ASCII: The American Standard Code for Information Interchange (ASCII) is an 8-bit character-encoding scheme based on the English alphabet. ASCII codes represent text in computers, communications equipment, and other devices that work with text. ASCII refers to a single 8-bit ASCII character or an array of 8-bit ASCII characters with the high bit of each character set to zero.

attribute: A characteristic of some object or entity, typically encoded as a name-value pair.

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [RFC5234].

binary large object (BLOB): A discrete packet of data that is stored in a database and is treated as a sequence of uninterpreted bytes.

code page: An ordered set of characters of a specific script in which a numerical index (code-point value) is associated with each character. Code pages are a means of providing support for character sets (1) and keyboard layouts used in different countries. Devices such as the display

and keyboard can be configured to use a specific code page and to switch from one code page (such as the United States) to another (such as Portugal) at the user's request.

common name (CN): A string attribute of a certificate (1) that is one component of a **distinguished name (DN)**. In Microsoft Enterprise uses, a CN must be unique within the forest where it is defined and any forests that share trust with the defining forest. The website or email address of the certificate owner is often used as a common name. Client applications often refer to a certification authority (CA) by the CN of its signing certificate.

contact: A person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and **attributes**, such as an Internet message address or login name.

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

Department object: An **Address Book object** that describes a department within an organization.

departmental group: A **distribution list** that describes a department within an organization.

display template: A template that describes how to display or allow a user to modify information about an **Address Book object**.

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

distribution list: A collection of users, computers, contacts, or other groups that is used only for email distribution, and addressed as a single recipient.

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

entry ID: See EntryID.

flags: A set of values used to configure or report options or settings.

Global Address List (GAL): An **address list** that conceptually represents the default address list for an **address book**.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

hierarchy table: A Table object whose rows represent the Folder objects that are contained in another Folder object.

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure

Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

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

locale: A collection of rules and data that are specific to a language and a geographical area. A locale can include information about sorting rules, date and time formatting, numeric and monetary conventions, and character classification.

mail tip: A note that is presented to the author of a message when the author is composing the message. A mail tip provides information about the recipients of a message and issues that might impact delivery of the message, such as moderation or delivery restrictions.

mail user: An **Address Book object** that represents a person or entity that can receive deliverable messages.

mailbox: A **message store** that contains email, calendar items, and other Message objects for a single recipient.

meeting request: An instance of a Meeting Request object.

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

Minimal Entry ID: A property of an **Address Book object** that can be used to uniquely identify the object.

Multipurpose Internet Mail Extensions (MIME): A set of extensions that redefines and expands support for various types of content in email messages, as described in [RFC2045], [RFC2046], and [RFC2047].

name service provider interface (NSPI): A method of performing address-book-related operations on Active Directory.

named property: A property that is identified by both a GUID and either a string name or a 32-bit identifier.

offline address book (OAB): A collection of **address lists** that are stored in a format that a client can save and use locally.

Organization object: An **Address Book object** that describes an entire organization.

Permanent Entry ID: A property of an **Address Book object** that can be used to uniquely identify the object.

property ID: A 16-bit numeric identifier of a specific **attribute**. A property ID does not include any **property type** information.

property tag: A 32-bit value that contains a property type and a property ID. The low-order 16 bits represent the property type. The high-order 16 bits represent the property ID.

property type: A 16-bit quantity that specifies the data type of a property value.

recipient: An entity that is in an **address list**, can receive email messages, and contains a set of **attributes**. Each attribute has a set of associated values.

Recipient object: A set of properties that represent the recipient of a Message object.

relative distinguished name (RDN): In the Active Directory directory service, the unique name of a child element relative to its parent in Active Directory. The RDN of a child element combined with the fully qualified domain name (FQDN) (2) of the parent forms the FQDN of the child.

remote procedure call (RPC): A context-dependent term commonly overloaded with three meanings. Note that much of the industry literature concerning RPC technologies uses this term interchangeably for any of the three meanings. Following are the three definitions: (*) The runtime environment providing remote procedure call facilities. The preferred usage for this meaning is "RPC runtime". (*) The pattern of request and response message exchange between two parties (typically, a client and a server). The preferred usage for this meaning is "RPC exchange". (*) A single message from an exchange as defined in the previous definition. The preferred usage for this term is "RPC message". For more information about RPC, see [C706].

Resource object: An **Address Book object** that represents an asset that can be reserved, such as a room or equipment.

Rich Text Format (RTF): Text with formatting as described in [MSFT-RTF].

S/MIME (Secure/Multipurpose Internet Mail Extensions): A set of cryptographic security services, as described in [RFC5751].

Simple Mail Transfer Protocol (SMTP): A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [RFC5321].

Transport Neutral Encapsulation Format (TNEF): A binary type-length-value encoding that is used to encode properties for transport, as described in [MS-OXTNEF].

Unicode: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The **Unicode** standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

UTF-16LE: The Unicode Transformation Format - 16-bit, Little Endian encoding scheme. It is used to encode **Unicode** characters as a sequence of 16-bit codes, each encoded as two 8-bit bytes with the least-significant byte first.

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

1.2 References

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

1.2.1 Normative References

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

[ASF] Microsoft Corporation, "Advanced Systems Format Specification", December 2004, http://download.microsoft.com/download/7/9/0/790fecaa-f64a-4a5e-a430-0bccdab3f1b4/ASF_Specification.doc

[MS-NSPI] Microsoft Corporation, "Name Service Provider Interface (NSPI) Protocol".

[MS-OXABREF] Microsoft Corporation, "Address Book Name Service Provider Interface (NSPI) Referral Protocol".

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

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

[MS-OXCMAIL] Microsoft Corporation, "RFC 2822 and MIME to Email Object Conversion Algorithm".

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

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

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

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

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

[MS-OXNSPI] Microsoft Corporation, "Exchange Server Name Service Provider Interface (NSPI) Protocol".

[MS-OXOABKT] Microsoft Corporation, "Address Book User Interface Templates Protocol".

[MS-OXOAB] Microsoft Corporation, "Offline Address Book (OAB) File Format and Schema".

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

[MS-OXPFOAB] Microsoft Corporation, "Offline Address Book (OAB) Public Folder Retrieval Protocol".

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

[MS-OXWOAB] Microsoft Corporation, "Offline Address Book (OAB) Retrieval File Format".

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions".

[RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, 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.rfc-editor.org/rfc/rfc2119.txt

[RFC2822] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001, http://www.ietf.org/rfc/rfc2822.txt

[RFC2876] Pawling, J., "Use of the KEA and SKIPJACK Algorithms in CMS", RFC 2876, July 2000, http://www.rfc-editor.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

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

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, http://www.rfc-editor.org/rfc/rfc5234.txt

[RFC5751] Ramsdell, B., and Turner, S., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010, http://www.rfc-editor.org/rfc/rfc5751.txt

1.2.2 Informative References

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

1.3 Overview

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. Every object in the address book has various properties, and the values of these properties contain information that is 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 of which describes an **address list**. An address list is a collection of Address Book objects that can 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 an address book server and accessed as follows:

- Through **NSPI** calls by using either of the NSPI protocols, as described in [MS-NSPI] and [MS-OXNSPI].
- Through the Messaging Application Programming Interface (MAPI) Extensions for **HTTP**, as described in [MS-OXCMAPIHTTP].

Alternatively, the address book can be stored locally on the client computer as an **offline address book (OAB)** by using the OAB Format and Schema Protocol, as described in [MS-OXOAB]. An OAB is obtained by using either the OAB Web Retrieval protocol, as described in [MS-OXWOAB], or the OAB Public Folder Retrieval Protocol, as described in [MS-OXPFOAB]. The Address Book User Interface Templates Protocol, as described in [MS-OXOABKT], is used to render information about an Address Book object to a messaging user.

This protocol defines the properties on each of the Address Book objects and defines how they interrelate with one another.

1.4 Relationship to Other Protocols

This protocol relies on the following:

- The Exchange Server Name Service Provider Interface (NSPI) Protocol, as described in [MS-OXNSPI], or the Name Service Provider Interface (NSPI) Protocol, as described in [MS-NSPI].
- The Messaging Application Programming Interface (MAPI) Extensions for HTTP, as described in [MS-OXCMAPIHTTP].
- The Offline Address Book (OAB) File Format and Schema, as described in [MS-OXOAB].
- The Address Book User Interface Templates Protocol, as described in [MS-OXOABKT].
- The Offline Address Book (OAB) Retrieval File Format, as described in [MS-OXWOAB].
- The Offline Address Book (OAB) Public Folder Retrieval Protocol, as described in [MS-OXPFOAB].

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

1.5 Prerequisites/Preconditions

This specification assumes one of the following:

- That the messaging client has been referred to an address book server either by using the NSPI Referral protocol, as specified in [MS-OXABREF], or by using the Messaging Application Programming Interface (MAPI) Extensions for HTTP, as specified in [MS-OXCMAPIHTTP], and has established a connection to an address book server, as specified in [MS-NSPI], [MS-OXNSPI], and [MS-OXCMAPIHTTP].
- That the messaging client has access to an OAB, as specified in [MS-OXOAB].

1.6 Applicability Statement

This 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 that are 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 one of the following as the underlying transport protocol when handling address books online on an address book server:

- The Name Service Provider Interface (NSPI) Protocol, as specified in [MS-NSPI], or the Exchange Server NSPI Protocol, as specified in [MS-OXNSPI].<1>
- The Messaging Application Programming Interface (MAPI) Extensions for HTTP, as specified in [MS-OXCMAPIHTTP].<2>

This protocol uses the OAB File Format and Schema, as specified in [MS-OXOAB], as the underlying structure when handling address books via a local offline address book (OAB).

2.2 Message Syntax

The properties used by this protocol are maintained on an address book server. The properties are returned by an address book server in one of the following:

- Various NSPI functions of either the Name Service Provider Interface (NSPI) Protocol, as specified in [MS-NSPI], or the Exchange Server NSPI Protocol, as specified in [MS-OXNSPI].
- Various response bodies of an HTTP **POST** method, as specified in [MS-OXCMAPIHTTP].<3>

Alternatively, many of these properties are maintained in an offline address book (OAB) by using the OAB Format and Schema Protocol, as specified in [MS-OXOAB]. The OAB is a collection of address lists, each of which contains Address Book objects and their properties. Refer to [MS-OXOAB] for a list of the properties maintained in OABs.

There are many types of Address Book objects, including but not limited to **mail users**, **distribution lists**, address book containers, **Resource objects**, **Department objects**, **Organization objects**, and templates. This protocol does not require values for all properties that are defined for any object type, nor does it limit the properties to those of their own type or those listed in this specification. When a data source, such as an OAB or an address book 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 the properties of type **PtypString** ([MS-OXCDATA] section 2.11.1) that are listed in this document, a request to an address book server of that property with type **PtypString8** ([MS-OXCDATA] section 2.11.1) or **PtypString** is permitted. If a conversion is required, the string will be converted to the type requested by the client, as specified in either [MS-NSPI] or [MS-OXNSPI].

For all of the properties of type **PtypString** that are listed in this document, an OAB that includes a value for that string property MUST contain exactly one string representation. The internal representation of strings in an OAB that use the OAB Format and Schema Protocol is specified in [MS-OXOAB]. A client that looks up the value for a string property MUST convert the value to the string type that is native to the client before it interprets the value of that property.

2.2.1 Definitions

The following sections describe structures that are 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 (DN)** is used to refer to an address book DN. 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 (section 2.2.3.14). The DN is also embedded in the **Distinguished Name** field of the PermanentEntryID structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.9.3.<4> DNs are structured as shown in the following **Augmented Backus-Naur Form (ABNF)** definition, as specified in [RFC5234].

```
dn = organization-dn /
  addresslist-dn /
  x500-dn
organization-dn= org-rdn
addresslist-dn    = "/guid=" container-guid /
  gal-addrlist-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" / "|"
```

DNs for specific objects have a strict format, as shown in the following table.

Object type	DN format	Notes
Address book container	*addresslist-dn*	
Global Address List container	*gal-addrlist-dn*	
Mail user	*x500-dn*	The *org-rdn* string is the mail user's organization.
Organization	*organization-dn*	
Store	*x500-dn*	The *x500-container-dn* is the **mailbox** server.
Mailbox server	*x500-dn*	The **relative distinguished name (RDN)** in the *object-rdn* is the name of the mailbox server.

Object type	DN format	Notes
Room container reference	*x500-dn* with no *container-rdn*	The RDN of the *object-rdn* matches the *container-guid* of the address book container.
All other Address Book objects	*dn*	

When the DN of an Address Book object that is obtained from an NSPI server matches the DN of an Address Book object that is obtained from an OAB, the objects represent the same entity. The OAB SHOULD<5> include additional properties not available on the NSPI server. Properties SHOULD have the same value when present on both data sources. One exception to having the same value on both data sources is if the properties are truncated in the OAB, according to the limitations specified in [MS-OXOAB] section 2.9.2.2.1. Another exception is if the value on an NSPI server has changed since the OAB was created, or if the NSPI server was restored from a backup after the OAB was created. In such a case, the NSPI server and the OAB are said to be "out of sync". That is, the data in the two sources reflects two different time periods.

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 is 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 (OAB), messaging clients SHOULD obtain information about the address book hierarchy and its address book containers by using the Offline Address Book Retrieval File Format, as specified in [MS-OXWOAB], or MAY<6> obtain information by using the Offline Address Book Public Folder Retrieval Protocol, as specified in [MS-OXPFOAB]. Because an OAB maintains its own structure for the **hierarchy table**, none of the properties of address book containers described in this document apply to the OAB.

The hierarchy table is a set of rows, each of which describes one address book container. For details about retrieving the address book hierarchy table, see section 3.1.4.1.

2.2.2.1 PidTagContainerFlags

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

The **PidTagContainerFlags** property ([MS-OXPROPS] section 2.635) contains a bitmask of **flags** that describe capabilities of an address book container.

The flags listed in the following table are defined for the bitmask of the **PidTagContainerFlags** property. The **PidTagContainerFlags** property MUST NOT contain any other flags.

Flag 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 that are returned by the NSPI server.
AB_SUBCONTAINERS	0x00000002	The container holds child containers. This flag does not indicate whether any subcontainers 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 that are returned by the NSPI server.

Because this property applies to a container in the hierarchy table, it is not present on objects in the offline address book (OAB). An OAB has its own structure for maintaining the hierarchy, using the OAB Retrieval File Format, as specified in [MS-OXWOAB], and the OAB Public Folder Retrieval Protocol, as specified in [MS-OXPFOAB].

2.2.2.2 PidTagDepth

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

The **PidTagDepth** property ([MS-OXCTABL] section 2.2.1.4) represents the relative level of depth of a container in a hierarchy table. Objects in the hierarchy table that share the same value for the **PidTagDepth** property 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.

Because this property applies to a container in the hierarchy table, it is not present on objects in an offline address book (OAB). An OAB has its own structure for maintaining the hierarchy, using the OAB Public Folder Retrieval Protocol, as specified in [MS-OXPFOAB].

2.2.2.3 PidTagAddressBookContainerId

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

The **PidTagAddressBookContainerId** property ([MS-OXPROPS] section 2.503) identifies an address book container on an address book server. Note that this property is a **Minimal Entry ID**. A value of 0 (zero) represents the Global Address List (GAL). If the value is nonzero, it is only a valid representation of the specific container. This representation lasts for the time that the connection to the address book server lasts, or, after disconnection from and reconnection to the same or another address book server, for as long as the new server identifies itself as having the same **GUID**. The server returns the GUID in the *pServerGuid* parameter of the **NspiBind** method,<7> as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.1, or in the **ServerGuid** field of the **Bind** request type response,<8> as specified in [MS-OXCMAPIHTTP] section 2.2.5.1.2.

Because this property applies to a container in the hierarchy table, it is not present on objects in an offline address book (OAB). An OAB has its own structure for maintaining the hierarchy, using the OAB Public Folder Retrieval Protocol, as specified in [MS-OXPFOAB].

2.2.2.4 PidTagAddressBookIsMaster

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

The **PidTagAddressBookIsMaster** property ([MS-OXPROPS] section 2.536) is **TRUE** (0x01) if it is possible to create Address Book objects in that container, and **FALSE** (0x00) otherwise. The value does not pertain to parent containers or subcontainers of this container.

Because this property applies to a container in the hierarchy table, it is not present on objects in an offline address book (OAB). An OAB has its own structure for maintaining the hierarchy, using the OAB Public Folder Retrieval Protocol, as specified in [MS-OXPFOAB].

2.2.2.5 PidTagAddressBookParentEntryId

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

The **PidTagAddressBookParentEntryId** property ([MS-OXPROPS] section 2.550) is the **entry ID** of the parent container in a hierarchy of address book containers. This property is not present if no parent container exists. Messaging clients use this property to expand and collapse a hierarchy of address book containers in an address book hierarchy table.

Because this property applies to a container in the hierarchy table, it is not present on objects in an offline address book (OAB). An OAB has its own structure for maintaining the hierarchy, using the OAB Public Folder Retrieval Protocol, as specified in [MS-OXPFOAB].

2.2.3 Properties that Apply to All Address Book Objects

Address Book objects are listed in an offline address book (OAB) by the OAB Format and Schema Protocol, as specified in [MS-OXOAB], or accessed on an address book server by using various NSPI method calls,<9> as specified in [MS-NSPI] and [MS-OXNSPI], or by using various HTTP request types for an address book server **endpoint**,<10> as specified in [MS-OXCMAPIHTTP]. Each object represents any addressable entity, including but not limited to mail user, distribution list, Department object, Organization object, address book container, or Resource object. Not all properties defined in this section are stored in the OAB. For a list of the properties stored in the OAB, see [MS-OXOAB].

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 it can contain 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 SHOULD have additional properties, as specified in section 2.2.6. Mail user objects are further broken down into subtypes, such as rooms, equipment, messaging forums, or other types. Note that servers can restrict client access to specific properties as required by the implementation. One example is distribution list membership; the server can restrict access to distribution lists in which the members are hidden.

The Department object and the Organization object are types of Address Book objects that are required to support a hierarchical address book view that messaging clients need.

2.2.3.1 PidTagDisplayName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagDisplayName** property ([MS-OXCFOLD] section 2.2.2.2.2.5) represents a displayable form of an Address Book object. When the Address Book object is a mail user, the **PidTagDisplayName** property is the full name of the mail user. When the object is a distribution list, the value of the **PidTagDisplayName** property 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, the **PidTagDisplayName** property value contains the name of that department. When the object is any other Address Book object, the value of the **PidTagDisplayName** property is the displayable name of that object.

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

2.2.3.2 PidTagEntryId

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

The **PidTagEntryId** property ([MS-OXCPERM] section 2.2.4) contains an entry ID that identifies an Address Book object on an NSPI server. The entry ID MUST be formatted as either a **PermanentEntryID** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.9.3, or an **EphemeralEntryID** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.9.2.<11> Messaging clients use this property to open an Address Book object. The client can then perform operations on the Address Book object, such as obtaining other properties. The types of operations that can be performed on an Address Book object are specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4 and in [MS-OXCMAPIHTTP] section 2.2.5. When the entry ID is in **Permanent Entry ID** format, its DN MUST match the value of the **PidTagEmailAddress** property (section 2.2.3.14) and MUST follow the format that is specified in section 2.2.1.1.

The OAB Format and Schema Protocol specification, as specified in [MS-OXOAB], does not include value for the **PidTagEntryId** property for Address Book objects in its data structure. Instead, the **PidTagEmailAddress** property (section 2.2.3.14) identifies objects in an OAB.

2.2.3.3 PidTagTemplateid

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

The **PidTagTemplateid** property ([MS-OXPROPS] section 2.1033) contains the **PidTagEntryId** property (section 2.2.3.2), expressed in Permanent Entry ID format. This value MUST be present for all Address Book objects on an NSPI server, its DN MUST match the value for the **PidTagEmailAddress** property (section 2.2.3.14), and its DN MUST follow the format 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 (OAB).

2.2.3.4 PidTagRecordKey

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

The **PidTagRecordKey** property ([MS-OXCPRPT] section 2.2.1.8) contains a unique binary-comparable identifier for an Address Book object. This value MUST be present on all objects on an NSPI server and MUST match the value for the **PidTagTemplateid** property (section 2.2.3.3).

The **PidTagRecordKey** property is not present on objects in an offline address book (OAB).

2.2.3.5 PidTagSearchKey

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

The **PidTagSearchKey** property ([MS-OXCPRPT] section 2.2.1.9) is formed by concatenating the ASCII string "EX: " followed by the DN for the object converted to all uppercase, followed by a zero-byte value. This value MUST be present for all Address Book objects on an NSPI server and MUST be in the aforementioned format.

The **PidTagSearchKey** property is not present on objects in an offline address book (OAB).

2.2.3.6 PidTagInstanceKey

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

The **PidTagInstanceKey** property ([MS-OXPROPS] section 2.734) identifies 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 (OAB).

2.2.3.7 PidTagAddressBookDisplayNamePrintable

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookDisplayNamePrintable** property ([MS-OXPROPS] section 2.505) contains a displayable form of an Address Book object that can be rendered in the client user's own **code page**.

2.2.3.8 PidTagTransmittableDisplayName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagTransmittableDisplayName** property ([MS-OXPROPS] section 2.1039) contains an Address Book object's display name that is transmitted with the message. This value MUST be present on all objects on an NSPI server except the organization object. When present the value MUST match the value for the **PidTagDisplayName** property (section 2.2.3.1).

The **PidTagTransmittableDisplayName** property is not present on objects in an offline address book (OAB).

2.2.3.9 PidTagAddressBookPhoneticDisplayName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookPhoneticDisplayName** property ([MS-OXPROPS] section 2.553) contains the phonetic representation of the **PidTagDisplayName** property (section 2.2.3.1).<12>

2.2.3.10 PidTagObjectType

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

The **PidTagObjectType** property ([MS-OXCPRPT] section 2.2.1.7) 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 values listed in the following table.

Value name	Value	Description
MAILUSER	0x00000006	A mail user, or any Address Book object that is not a distribution list or forum.
DISTLIST	0x00000008	A distribution list.
folder	0x00000003	A messaging forum, such as a bulletin board service or a public or shared folder.

2.2.3.11 PidTagDisplayType

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

The **PidTagDisplayType** property ([MS-OXPROPS] section 2.670) contains a value that indicates how to display an Address Book object in a table or as an addressee on a message. Messaging clients use this property to display an icon, make the object bold, or apply some other display element to make it easy for a user who is viewing the object to distinguish its type. In addition, the **PidTagDisplayTypeEx** property (section 2.2.3.12) provides a means to refine the display further.

The **PidTagDisplayType** property MUST have one of the following display type values, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.1.3, according to the object's type: **DT_MAILUSER**, **DT_DISTLIST**, **DT_FORUM**, **DT_AGENT**, **DT_ORGANIZATION**, **DT_PRIVATE_DISTLIST**, or **DT_REMOTE_MAILUSER**.<13>

2.2.3.12 PidTagDisplayTypeEx

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

The **PidTagDisplayTypeEx** property ([MS-OXPROPS] section 2.671) contains a value that indicates 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, make the object bold, or apply some other display element to make it easy for a user who is viewing the object to distinguish its type. This property contains more detailed

information about the object's display information than does the **PidTagDisplayType** property (section 2.2.3.11). In addition to the display information contained in the **PidTagDisplayType** property, this property distinguishes among additional object types. When the object comes from a remote server, the **PidTagDisplayType** property also includes information about the type of object on that remote server, as well as the type on the local server.

The **PidTagDisplayTypeEx** property also includes information that indicates whether it is possible to share information from the user's own mailbox to the entity represented by an Address Book object.

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

r (1 bit): 1 indicates that the value in **dtRemote** is the remote display type. The numeral 0 (zero) means that **dtRemote** is undefined. This represents the high order bit.

s (1 bit): 1 indicates that the mailbox server supports sharing to the entity that an Address Book object represents. The numeral 0 (zero) means that it does not support such sharing.

reserved (14 bits): Undefined. This value MUST contain all zeros and MUST be ignored by clients.

dtRemote (1 byte): The display type of an Address Book object in the remote forest. This is undefined if **r** is 0 (zero). If **r** is 1, it contains one of the values listed in the following table.

dtLocal (1 byte): The display type of an Address Book object in the messaging user's local forest. It contains one of the values listed in the following table.

Note that **dtLocal** and/or **dtRemote** MUST have one of the following display type values, as defined in [MS-NSPI] and [MS-OXNSPI] section 2.2.1.3,<14> 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 listed in the following table.

Value name	Value	Description
DT_ROOM	0x07	A conference room. It is possible for messaging clients to send **meeting requests** to the specified Address Book object to book the room.
DT_EQUIPMENT	0x08	Equipment. It is possible for messaging clients to send meeting requests to the specified Address Book object to reserve the equipment.
DT_SEC_DISTLIST	0x09	A distribution list used as a security group on the server.

2.2.3.13 PidTagAddressType

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressType** property ([MS-OXPROPS] section 2.567) contains an Address Book object's e-mail **address type**. It MUST have the value "EX" for all objects on an NSPI server.

The **PidTagAddressType** property is not present on objects in an offline address book (OAB).

2.2.3.14 PidTagEmailAddress

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagEmailAddress** property ([MS-OXPROPS] section 2.672) contains an Address Book object's e-mail address, expressed in X500 format, using the format that is particular to the type of object, as specified 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 format particular to the type of object, as specified in section 2.2.1.1.

2.2.3.15 PidTagAddressBookObjectDistinguishedName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookObjectDistinguishedName** property ([MS-OXPROPS] section 2.545) contains the DN of an Address Book object in DN format, as specified in section 2.2.1.1. If present, its DN MUST follow the format that is particular to the type of object, as specified in section 2.2.1.1. Its value, if present, MUST match the value for the **PidTagEmailAddress** property. This value MUST be present on all Address Book objects on an NSPI server.

2.2.3.16 PidTagCreationTime

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

The **PidTagCreationTime** property ([MS-OXCMSG] section 2.2.2.3) contains the creation date and time for an Address Book object in **Coordinated Universal Time (UTC)**.

2.2.3.17 PidTagLastModificationTime

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

The **PidTagLastModificationTime** property ([MS-OXCMSG] section 2.2.2.2) contains the date and time that an Address Book object was last modified in Coordinated Universal Time (UTC).

2.2.3.18 PidTagSendRichInfo

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

The **PidTagSendRichInfo** property ([MS-OXPROPS] section 2.1000) contains **TRUE** (0x01) if the e-mail-enabled entity represented by an Address Book object can receive all message content, including **Rich Text Format (RTF)** and other embedded objects. When sending mail by using the [RFC2822] Internet message format and **Multipurpose Internet Mail Extensions (MIME)** to the E-mail Object Conversion Protocol, as specified in [MS-OXCMAIL], the **PidTagSendRichInfo** property specifies whether to encode the message in MIME or in **Transport Neutral Encapsulation Format (TNEF)**, as specified in [MS-OXCMAIL]. The **PidTagSendRichInfo** property contains **FALSE** (0x00) if the e-mail-enabled entity cannot receive formatted message content.

2.2.3.19 PidTagSendInternetEncoding

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

The **PidTagSendInternetEncoding** property ([MS-OXPROPS] section 2.999) contains a bitmask of message-encoding preferences for mail sent to an e-mail-enabled entity that is represented by an Address Book object. When sending mail by using the [RFC2822] Internet message format and Multipurpose Internet Mail Extensions (MIME) to the 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

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAccount** property ([MS-OXOCNTC] section 2.2.1.10.11) contains an Address Book object's **alias**, which is an alternative name by which the object can be identified.

2.2.3.21 PidTagSmtpAddress

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagSmtpAddress** property ([MS-OXPROPS] section 2.1010) contains an Address Book object's **Simple Mail Transfer Protocol (SMTP)** address.

2.2.3.22 PidTagAddressBookTargetAddress

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookTargetAddress** property ([MS-OXPROPS] section 2.564) contains the foreign system e-mail address of an Address Book object. If this value is present, it MUST be prefixed with the address type of the foreign address, followed by a colon (":"), followed by the foreign e-mail address expressed in that address type. For Simple Mail Transfer Protocol (SMTP) addresses, this value is "SMTP:" followed by the foreign SMTP address.

2.2.3.23 PidTagAddressBookProxyAddresses

Data type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookProxyAddresses** property ([MS-OXPROPS] section 2.556) contains alternate e-mail addresses for an 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 a Simple Mail Transfer Protocol (SMTP) address, the string MUST start with the string "SMTP:" or "SMTP:" followed by an SMTP address. The first string starting with the uppercase string "SMTP:" MUST match the **PidTagSmtpAddress** property (section 2.2.3.21) for the Address Book object.

2.2.3.24 PidTagAddressBookSeniorityIndex

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

The **PidTagAddressBookSeniorityIndex** property<15> ([MS-OXPROPS] section 2.563) contains a signed integer that specifies the seniority order of Address Book objects that represent members of a department and are referenced by a Department object or **departmental group**, with larger values specifying members that are more senior. It also specifies the sort order of Department objects or departmental groups that represent child departments of another parent department in a hierarchy of departments and that are referenced by the parent's Department object or departmental group, in descending order.

2.2.3.25 PidTagAddressBookObjectGuid

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

The **PidTagAddressBookObjectGuid** property<16> ([MS-OXPROPS] section 2.546) contains a GUID that uniquely identifies an Address Book object.

2.2.3.26 PidTagAddressBookSenderHintTranslations

Data type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookSenderHintTranslations** property<17> ([MS-OXPROPS] section 2.562) contains the **locale** ID and translations of the default **mail tip**. An example of this value is "en-US:Hello" or "es:Hola".

2.2.3.27 PidTagAddressBookDeliveryContentLength

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

The **PidTagAddressBookDeliveryContentLength** property<18> ([MS-OXPROPS] section 2.504) specifies the maximum size of a message that a recipient can receive.

2.2.3.28 PidTagAddressBookModerationEnabled

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

The **PidTagAddressBookModerationEnabled** property<19> ([MS-OXPROPS] section 2.543) indicates whether moderation is enabled for the mail user of the distribution list. This property is set to **TRUE** (0x01) if moderation is enabled for the mail user or distribution list; otherwise, this property is set to **FALSE** (0x00).

2.2.3.29 PidTagAddressBookDistributionListMemberCount

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

The **PidTagAddressBookDistributionListMemberCount** property<20> ([MS-OXPROPS] section 2.508) contains the total number of recipients in a specified distribution list. This value includes the members of all the distribution lists that are members of the specified distribution list.

This property is calculated during OAB generation and is not available on the NSPI server.

2.2.3.30 PidTagAddressBookDistributionListExternalMemberCount

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

The **PidTagAddressBookDistributionListExternalMemberCount** property<21> ([MS-OXPROPS] section 2.507) contains the number of external recipients in a specified distribution list.

This property is calculated during OAB generation and is not available on the NSPI server.

2.2.3.31 PidTagComment

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagComment** property ([MS-OXCFOLD] section 2.2.2.2.2.2) contains a comment about the purpose or content of an Address Book object.

2.2.3.32 PidTagMappingSignature

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

The **PidTagMappingSignature** property ([MS-OXPROPS] section 2.771) MUST be present on all Address Book objects on an 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 (OAB).

2.2.3.33 PidTagInitialDetailsPane

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

The **PidTagInitialDetailsPane** property ([MS-OXPROPS] section 2.731) indicates which page of a **display template** to display first. This value MUST be present on all Address Book objects on an NSPI server, and it MUST have the value 0 (zero).

2.2.3.34 PidTagAddressBookExtensionAttribute1 through PidTagAddressBookExtensionAttribute15

These 15 properties are used to store custom values. These properties are **PtypString** values ([MS-OXCDATA] section 2.11.1) that are defined and populated by the organization that modified the display templates.

2.2.3.35 PidTagAddressBookDisplayTypeExtended

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

The **PidTagAddressBookDisplayTypeExtended** property ([MS-OXPROPS] section 2.506) is limited to **Recipient objects**. The **PidTagAddressBookDisplayTypeExtended** property SHOULD be present on objects on an NSPI server or an offline address book (OAB).<22>

2.2.4 Properties that Apply to Mail User Objects

2.2.4.1 PidTagSurname

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagSurname** property ([MS-OXPROPS] section 2.1026) contains a mail user's family name.

2.2.4.2 PidTagGivenName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagGivenName** property ([MS-OXPROPS] section 2.705) contains a mail user's given name.

2.2.4.3 PidTagInitials

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagInitials** property ([MS-OXPROPS] section 2.732) contains the initials for parts of a mail user's full name.

2.2.4.4 PidTagTitle

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagTitle** property ([MS-OXPROPS] section 2.1036) contains a mail user's job title.

2.2.4.5 PidTagOfficeLocation

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagOfficeLocation** property ([MS-OXPROPS] section 2.805) contains a mail user's office location.

2.2.4.6 PidTagDepartmentName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagDepartmentName** property ([MS-OXPROPS] section 2.663) contains a name for the department in which a mail user works.

2.2.4.7 PidTagCompanyName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagCompanyName** property ([MS-OXPROPS] section 2.630) contains a mail user's company name.

2.2.4.8 PidTagAssistant

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAssistant** property ([MS-OXPROPS] section 2.573) contains the name of a mail user's administrative assistant.

2.2.4.9 PidTagAddressBookManagerDistinguishedName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookManagerDistinguishedName** property ([MS-OXPROPS] section 2.540) contains the DN of a mail user's manager.

2.2.4.10 PidTagAddressBookPhoneticGivenName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookPhoneticGivenName** property ([MS-OXPROPS] section 2.554) contains the phonetic representation of the **PidTagGivenName** property (section 2.2.4.2).<23>

2.2.4.11 PidTagAddressBookPhoneticSurname

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookPhoneticSurname** property ([MS-OXPROPS] section 2.555) contains the phonetic representation of the **PidTagSurname** property (section 2.2.4.1).<24>

2.2.4.12 PidTagAddressBookPhoneticCompanyName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookPhoneticCompanyName** property ([MS-OXPROPS] section 2.551) contains the phonetic representation of the **PidTagCompanyName** property (section 2.2.4.7).<25>

2.2.4.13 PidTagAddressBookPhoneticDepartmentName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookPhoneticDepartmentName** property ([MS-OXPROPS] section 2.552) contains the phonetic representation of the **PidTagDepartmentName** property (section 2.2.4.6).<26>

2.2.4.14 PidTagStreetAddress

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagStreetAddress** property ([MS-OXPROPS] section 2.1021) contains a mail user's street address.

2.2.4.15 PidTagPostOfficeBox

Data type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1)

The **PidTagPostOfficeBox** property ([MS-OXPROPS] section 2.857) contains the number or identifier of a mail user's post office box.

2.2.4.16 PidTagLocality

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagLocality** property ([MS-OXPROPS] section 2.766) contains the name of a mail user's locality, such as the town or city.

2.2.4.17 PidTagStateOrProvince

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagStateOrProvince** property ([MS-OXPROPS] section 2.1017) contains the name of a mail user's state or province.

2.2.4.18 PidTagPostalCode

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagPostalCode** property ([MS-OXPROPS] section 2.856) contains the postal code for a mail user's postal address.

2.2.4.19 PidTagCountry

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagCountry** property ([MS-OXPROPS] section 2.644) contains the name of a mail user's country/region.

2.2.4.20 PidTagHomeAddressStreet

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagHomeAddressStreet** property ([MS-OXPROPS] section 2.721) contains a mail user's home street address.

2.2.4.21 PidTagBusinessTelephoneNumber

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagBusinessTelephoneNumber** property ([MS-OXPROPS] section 2.617) contains the primary telephone number of a mail user's place of business.

2.2.4.22 PidTagHomeTelephoneNumber

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagHomeTelephoneNumber** property ([MS-OXPROPS] section 2.723) contains the primary telephone number of a mail user's home.

2.2.4.23 PidTagBusiness2TelephoneNumber

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagBusiness2TelephoneNumber** property ([MS-OXPROPS] section 2.613) contains a secondary telephone number at a mail user's place of business.

2.2.4.24 PidTagBusiness2TelephoneNumbers

Data type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1)

The **PidTagBusiness2TelephoneNumbers** property ([MS-OXPROPS] section 2.614) contains secondary telephone numbers at a mail user's place of business.

2.2.4.25 PidTagHome2TelephoneNumber

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagHome2TelephoneNumber** property ([MS-OXPROPS] section 2.714) contains a secondary telephone number at a mail user's home.

2.2.4.26 PidTagHome2TelephoneNumbers

Data type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1)

The **PidTagHome2TelephoneNumbers** property ([MS-OXPROPS] section 2.715) contains secondary telephone numbers at a mail user's home.

2.2.4.27 PidTagMobileTelephoneNumber

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagMobileTelephoneNumber** property ([MS-OXPROPS] section 2.795) contains a mail user's mobile telephone number.

2.2.4.28 PidTagPagerTelephoneNumber

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagPagerTelephoneNumber** property ([MS-OXPROPS] section 2.848) contains a mail user's pager telephone number.

2.2.4.29 PidTagPrimaryFaxNumber

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagPrimaryFaxNumber** property ([MS-OXPROPS] section 2.859) contains the telephone number of a mail user's primary fax machine.

2.2.4.30 PidTagTelexNumber

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1) in an OAB file, **PtypMultipleBinary** ([MS-OXCDATA] section 2.11.1) when returned from an NSPI server

The **PidTagTelexNumber** property ([MS-OXPROPS] section 2.1031) contains a mail user's telex number.

2.2.4.31 PidTagAssistantTelephoneNumber

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAssistantTelephoneNumber** property ([MS-OXPROPS] section 2.574) contains the telephone number of a mail user's administrative assistant.

2.2.4.32 PidTagKeyword

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagKeyword** property ([MS-OXPROPS] section 2.753) contains a keyword that identifies a mail user to his or her system administrator.

2.2.4.33 PidTagMessageHandlingSystemCommonName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagMessageHandlingSystemCommonName** property ([MS-OXPROPS] section 2.783) contains the **common name** of a messaging user for use in a message header. This property is read-only.

2.2.4.34 PidTagUserCertificate

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

The **PidTagUserCertificate** property ([MS-OXPROPS] section 2.1042) has been deprecated. This property MUST be ignored by clients.

2.2.4.35 PidTagAddressBookX509Certificate

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

The **PidTagAddressBookX509Certificate** property ([MS-OXPROPS] section 2.566) contains ASN.1 DER encoded X.509 certificates for a mail user. Each binary value MUST be an ASN.1 DER encoded X.509 certificate, as specified in [RFC3280].

2.2.4.36 PidTagUserX509Certificate

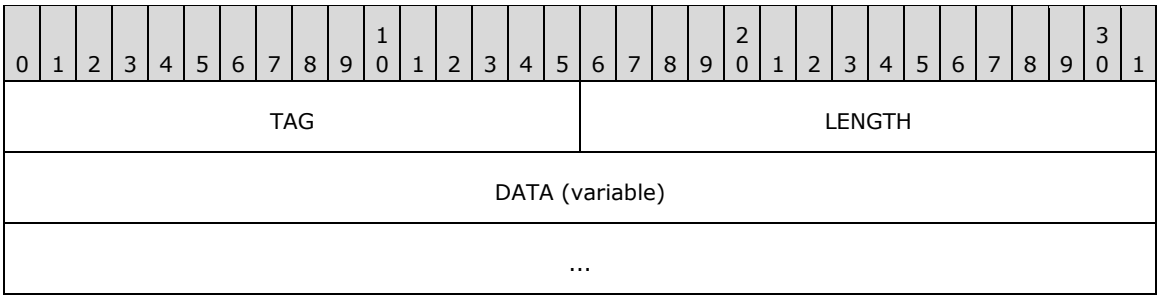
Data type: **PtypMultipleBinary** ([MS-OXCDATA] section 2.11.1)

The **PidTagUserX509Certificate** property ([MS-OXPROPS] section 2.1044) contains a list certificate for a mail user. Each binary value MUST be either an ASN.1 DER encoded SignedData Type **binary large object (BLOB)** that 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 the preceding types 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. The client and the server SHOULD use the **PidTagUserX509Certificate** property instead of the **PidTagAddressBookX509Certificate** property (section 2.2.4.35) when looking for certificates.

Non-ASN.1 Binary Value Format

If the binary value is not an ASN.1 DER encoded SignedData Type BLOB, it MUST be a BLOB containing a set of security settings as specified in sections 2.2.4.36.1 through 2.2.4.36.12, one after another, in a continuous block of data. All settings in these sections MUST appear no more than once in the binary value unless stated otherwise. Each security setting has the format shown in the following packet diagram.



TAG (2 bytes): An unsigned value that identifies this particular security setting. If the **TAG** value found in a security setting is not among the ones listed in sections 2.2.4.36.1 through 2.2.4.36.12, the client and the server MUST ignore these settings.

LENGTH (2 bytes): The total length of the security setting, including the **TAG** field, the **LENGTH** field, and the **DATA** field.

DATA (variable): 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.

2.2.4.36.1 Property Version

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

2.2.4.36.2 Encryption Type

The **TAG** value for this setting is 0x0006. The **LENGTH** value for this setting MUST be 0x0008. The unsigned **PtypInteger32** ([MS-OXCDATA] section 2.11.1) **DATA** value specifies the type of encryption to be used with this certificate. A value of 0x00000001 specifies that the encryption type is **S/MIME (Secure/Multipurpose Internet Mail Extensions)**, as specified in [RFC3852]. A **DATA** value of 0x00000006 specifies that the encryption type is Fortezza, as specified in [RFC2876]. All other values do not have any defined meanings and MUST be ignored by both the server and the client. This setting MUST be part of each **PtypBinary** ([MS-OXCDATA] section 2.11.1) value.

2.2.4.36.3 Defaults

The **TAG** value for this setting is 0x0020. The **LENGTH** value for this setting MUST be 0x0008. The **DATA** field is a 4-byte bit field, which can contain either or both of the bit values listed in the following table.

Bit	Meaning
0x00000001	This is the default certificate for S/MIME (Secure/Multipurpose Internet Mail Extensions).
0x00000002	This is the default certificate for all formats. If this bit is set, 0x00000001 MUST also be set.

Any other bit flags set on this setting MUST be ignored. This setting MUST be part of each **PtypBinary** ([MS-OXCDATA] section 2.11.1) value.

2.2.4.36.4 ASCII Display Name

The **TAG** value for this setting is 0x000B. The **LENGTH** value 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 2.2.4.36.5) — but not both — SHOULD be part of the binary value.<27>

2.2.4.36.5 Unicode Display Name

The **TAG** value for this setting is 0x0051. The **LENGTH** value 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 — but not both — SHOULD be part of the certificate.<28>

2.2.4.36.6 KeyExSHA1Hash

The **TAG** value for this setting is 0x0022. The **LENGTH** value for this setting MUST be larger than 4. The **DATA** field contains the SHA1 Hash setting, as specified in [RFC3174], which is used 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 specified **PtypBinary** ([MS-OXCDATA] section 2.11.1) value in the **PidTagUserX509Certificate** property (section 2.2.4.36), if the **KeyExSHA1Hash** setting is present, the **KeyExchangeCertificate** setting MUST NOT be present.

The client and the server SHOULD use the **KeyExchangeCertificate** setting instead of the **KeyExSHA1Hash** setting.

2.2.4.36.7 SignSHA1Hash

The **TAG** value for this optional setting is 0x0009. The **LENGTH** value for this setting MUST be larger than 4. The **DATA** field contains the SHA1 Hash setting, as specified in [RFC3174], which is used to look up the signing certificate in the certificate store on the user's computer.<29>

For a specified **PtypBinary** ([MS-OXCDATA] section 2.11.1) value in the **PidTagUserX509Certificate** property (section 2.2.4.36), if the **SignSHA1Hash** setting is present, the **SignCertificate** setting MUST NOT be present.

The client and the server SHOULD use the **SignCertificate** setting instead of the **SignSHA1Hash** setting.

2.2.4.36.8 KeyExchangeCertificate

The **TAG** value for this setting is 0x0003. The **LENGTH** value 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 (Secure/Multipurpose Internet Mail Extensions) mail to the contact.

For a specified **PtypBinary** ([MS-OXCDATA] section 2.11.1) value in the **PidTagUserX509Certificate** property (section 2.2.4.36), if the **KeyExchangeCertificate** setting is present, the **KeyExSHA1Hash** setting MUST NOT be present. Conversely, if **KeyExSHA1Hash** is not present, the KeyExchangeCertificate setting MUST be present.

The client and the server SHOULD use the **KeyExchangeCertificate** setting instead of the **KeyExSHA1Hash** setting.

2.2.4.36.9 SignCertificate

The **TAG** value for this optional setting is 0x0008. The **LENGTH** value for this setting MUST be larger than 4. The **DATA** field contains an ASN.1 DER encoded X.509 certificate, as specified in [RFC3280].<30>

For a specified **PtypBinary** ([MS-OXCDATA] section 2.11.1) value in the **PidTagUserX509Certificate** property (section 2.2.4.36), if the **SignCertificate** setting is present, the **SignSHA1Hash** setting MUST NOT be present.

The client and the server SHOULD use the **SignCertificate** setting instead of the **SignSHA1Hash** setting.

2.2.4.36.10 ChainCertificate

The **TAG** value for this optional setting is 0x0004. The **LENGTH** value for this setting MUST be larger than 4. This setting can appear multiple times in the **PtypBinary** ([MS-OXCDATA] section 2.11.1) 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 the **KeyExchangeCertificate** setting.

2.2.4.36.11 AsymetricCapabilities

The **TAG** value for this setting is 0x0002. The **LENGTH** value for this setting MUST be equal to or larger than 4. The **DATA** field contains the ASN.1 DER encoded **sMIMECapabilities** type, as specified in [RFC5751]. This setting MUST be part of the **PtypBinary** ([MS-OXCDATA] section 2.11.1) value.

2.2.4.36.12 SavedTime

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

2.2.4.37 PidTagAddressBookHomeMessageDatabase

Data type: **PtypString8** ([MS-OXCDATA] section 2.11.1) in an OAB file, **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1.5) when returned from an NSPI server

The **PidTagAddressBookHomeMessageDatabase** property ([MS-OXPROPS] section 2.535) contains the DN of a mail user, 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 store** 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 specified in [RFC1034], of the server that contains the mail user's message store database.

2.2.4.38 PidTagAddressBookNetworkAddress

Data type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookNetworkAddress** property ([MS-OXPROPS] section 2.544) contains the Address Book object of a mailbox server that contains a list of names by which a server is known to the various transports in use by the network. Each **PtypString** is a **remote procedure call (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.

2.2.4.39 PidTagAddressBookOrganizationalUnitRootDistinguishedName

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookOrganizationalUnitRootDistinguishedName** property ([MS-OXPROPS] section 2.547) contains the DN of the Organization object of the mail user's organization. When this property is not present, clients obtain the mail user's organization as described in section 2.2.7.<31>

2.2.4.40 PidTagThumbnailPhoto

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

The **PidTagThumbnailPhoto**<32> property ([MS-OXPROPS] section 2.1035), if present, MUST contain an image of a mail user's photo in Joint Photographic Experts Group (JPEG) format.

2.2.4.41 PidTagSpokenName

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

The **PidTagSpokenName** property<33> ([MS-OXPROPS] section 2.1013), if present, MUST contain an Advanced Systems Format binary media object, as specified in [ASF] section 9.5, that contains a recording of the mail user's name pronunciation. The major media type MUST be set to 73647561-0000-0010-8000-00AA00389B71, the media subtype MUST be set to 00000001-0000-0010-8000-00AA00389B71, and the format type MUST be set to 05589f81-c356-11ce-bf01-00aa0055595a.

2.2.4.42 PidTagAddressBookAuthorizedSenders

Data type: **PtypObject** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookAuthorizedSenders** property<34> ([MS-OXPROPS] section 2.502) indicates whether delivery restrictions exist for the specified recipient. A value other than NULL indicates that delivery restrictions exist for this recipient. The address book does not contain the lists of senders who are allowed for this recipient; it only indicates whether such restrictions exist.

2.2.4.43 PidTagAddressBookUnauthorizedSenders

Data type: **PtypObject** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookUnauthorizedSenders** property<35> ([MS-OXPROPS] section 2.565) indicates whether delivery restrictions exist for the specified recipient. A value other than NULL indicates that delivery restrictions exist for this recipient. The address book does not contain the lists of senders who are prohibited for this recipient; it only indicates whether such restrictions exist.

2.2.4.44 PidTagAddressBookDistributionListMemberSubmitAccepted

Data type: **PtypObject** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookDistributionListMemberSubmitAccepted** property<36> ([MS-OXPROPS] section 2.509) indicates whether delivery restrictions exist for the specified recipient. A value other than NULL indicates that delivery restrictions exist for this recipient. The address book does not

contain the lists of the group of senders who are allowed for this recipient; it only indicates whether such restrictions exist.

2.2.4.45 PidTagAddressBookDistributionListMemberSubmitRejected

Data type: **PtypObject** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookDistributionListMemberSubmitRejected** property<37> ([MS-OXPROPS] section 2.510) indicates whether delivery restrictions exist for this recipient. A value other than NULL indicates that delivery restrictions exist for this recipient. The address book does not contain the lists of the group of senders who are prohibited for this recipient; it only indicates whether such restrictions exist.

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.

For details about accessing these tables from an address book server, see [MS-NSPI] and [MS-OXNSPI] section 3.1.4.4.2.2.<38> Each of these properties is of type **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1). For example, a mail user in an organization reports to a manager, who is also listed in that address book as another mail user. In this case, a property of type **PtypEmbeddedTable** or **PidTagAddressBookManager** (section 2.2.5.1) references the 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 address book server by using a property value–based explicit table through either the **NspiGetMatches** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.10, or the **GetMatches** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.5, and are modified through either the **NspiModLinkAtt** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.15, or the **ModLinkAtt** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.10.<39><40>

These tables are represented in an offline address book (OAB) as a property either of type **PtypString** ([MS-OXCDATA] section 2.11.1) (for tables that consist of no more than one reference) or of type **PtypMultipleString** ([MS-OXCDATA] section 2.11.1) (for tables that can consist of multiple references), as specified in [MS-OXOAB]. Each string value represents the DN of another Address Book object. For example, a mail user in an organization reports to a manager who is also listed in the address book as another mail user. In this case, the **PidTagAddressBookManager** property (section 2.2.5.1), with a **property type** of **PtypString**, contains the DN of the manager.

2.2.5.1 PidTagAddressBookManager

Data type: **PtypObject** ([MS-OXCDATA] section 2.11.1.5)

The **PidTagAddressBookManager** property ([MS-OXPROPS] section 2.539) contains one row that references a mail user's manager.

2.2.5.2 PidTagAddressBookReports

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookReports** property ([MS-OXPROPS] section 2.558) lists all of the mail user's direct reports.

2.2.5.3 PidTagAddressBookIsMemberOfDistributionList

Data type: **PtypString8** ([MS-OXCDATA] section 2.11.1) in an OAB file, **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1.5) when returned from an NSPI server.

The **PidTagAddressBookIsMemberOfDistributionList** property<41> ([MS-OXPROPS] section 2.537) is an Address Book object that lists all the distribution lists for which a specified object is a member.

2.2.5.4 PidTagAddressBookOwnerBackLink

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookOwnerBackLink** property ([MS-OXPROPS] section 2.549) is a mail user list that contains the distribution lists that the mail user owns.

2.2.5.5 PidTagAddressBookPublicDelegates

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookPublicDelegates** property ([MS-OXPROPS] section 2.557) contains a list of mail users who are allowed to send mail on behalf of the mailbox owner.

2.2.5.6 PidTagAddressBookHierarchicalShowInDepartments

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookHierarchicalShowInDepartments** property ([MS-OXPROPS] section 2.534) is a mail user list that contains all of the Department objects of which the mail user is a member.

2.2.6 Properties That Apply to Distribution Lists

2.2.6.1 PidTagAddressBookMember

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookMember** property<42> ([MS-OXPROPS] section 2.541) contains the members of a distribution list. If the distribution list is also a departmental group (as specified by the **PidTagAddressBookHierarchicalIsHierarchicalGroup** property (section 2.2.6.5)), the **PidTagAddressBookMember** property contains the members of the department and the child departmental groups in the hierarchy of departments.

2.2.6.2 PidTagAddressBookOwner

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookOwner** property ([MS-OXPROPS] section 2.548) contains one row that references a distribution list's owner.

2.2.6.3 PidTagContainerContents

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagContainerContents** property ([MS-OXPROPS] section 2.634) contains a distribution list that is always empty. An NSPI server MUST define this value for distribution lists. This value is not present for any other objects.

2.2.6.4 PidTagAddressBookFolderPathname

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookFolderPathname** property ([MS-OXPROPS] section 2.528) has been deprecated and MUST be ignored by clients.

2.2.6.5 PidTagAddressBookHierarchicalIsHierarchicalGroup

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

The **PidTagAddressBookHierarchicalIsHierarchicalGroup** property<43> ([MS-OXPROPS] section 2.531) contains **TRUE** (0x01) if the distribution list represents a departmental group. Members of this departmental group that are also distribution lists with the **PidTagAddressBookHierarchicalIsHierarchicalGroup** property set to **TRUE** represent the child department of this departmental group in a hierarchy of departments. Other members represent members of this departmental group. If the distribution list is not a departmental group, the **PidTagAddressBookHierarchicalIsHierarchicalGroup** property can be either **FALSE** (0x00) or absent.

2.2.7 Properties That Apply to Organization Objects

An Organization object is an Address Book object that represents an organization and contains properties that are specific to that organization. A mail user's Organization object is obtained by using the **PidTagAddressBookOrganizationalUnitRootDistinguishedName** property (section 2.2.4.39) of the mail user.<44> When this property is not present on a mail user, 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, as specified 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. After the Organization object is obtained, messaging clients are able to perform operations the same way as they would on any other Address Book object.

2.2.7.1 PidTagAddressBookRoomContainers

Data type: **PtypMultipleString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookRoomContainers** property ([MS-OXPROPS] section 2.560) contains a list of DNs representing the address book containers that hold Resource objects, such as conference rooms and equipment. Messaging clients use this list to determine which containers have mainly Resource objects so that they can perform special actions on these containers, such as displaying a different column set when browsing address lists that are represented by these containers, or using features that require selecting from a set of rooms or equipment.

The DNs in the **PidTagAddressBookRoomContainers** property each use the *x500-dn* format specification, as specified 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 that 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, the other address book container is a room container.

2.2.7.2 PidTagAddressBookHierarchicalRootDepartment

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookHierarchicalRootDepartment** property ([MS-OXPROPS] section 2.533) contains a reference to either the root Department object or the root departmental group in the department hierarchy for an organization.<45> The table has either zero rows or one row, which references a Department object or departmental group. If either the Organization object is missing, this property is missing, or the property value is empty, 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 that are specific to a Department object. Department objects are distinct from departmental groups, which are a specific type of distribution list with all its properties. A server MAY<46> represent a department hierarchy by using either Department objects or departmental groups, and it MUST NOT mix the representations.

Messaging clients obtain the root of the department hierarchy by using the **PidTagAddressBookHierarchicalRootDepartment** property (section 2.2.7.2) of the Organization object (which MAY be either a Department object or a departmental group), or by obtaining the root department that is not specific to any organization, which MUST be a Department object, and which has a DN that is specified by using the *organization-dn* format specification, as specified in 2.2.1.1, with a value of "/o=FF46312B-D8AE-406C-B8E6-BC1A22A4C69E".

2.2.8.1 PidTagAddressBookHierarchicalChildDepartments

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookHierarchicalChildDepartments** property ([MS-OXPROPS] section 2.529) contains a Department object that references the child departments in a hierarchy of departments.

2.2.8.2 PidTagAddressBookHierarchicalParentDepartment

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookHierarchicalParentDepartment** property ([MS-OXPROPS] section 2.532) contains a Department object that references all of the departments to which this specified department is a child department.

2.2.8.3 PidTagAddressBookHierarchicalDepartmentMembers

Data type: **PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookHierarchicalDepartmentMembers** property ([MS-OXPROPS] section 2.530) contains a Department object that lists all of the mail users that belong to a specified department.

2.2.9 Properties That Apply to Resource Objects

2.2.9.1 PidTagAddressBookRoomCapacity

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

The **PidTagAddressBookRoomCapacity** property ([MS-OXPROPS] section 2.559) contains the maximum occupancy of a specified room.

2.2.9.2 PidTagAddressBookRoomDescription

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookRoomDescription** property ([MS-OXPROPS] section 2.561) contains a description of a Resource object.

2.2.10 Properties That Have Special Purposes

2.2.10.1 PidTagAnr

Data type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagAnr** property ([MS-OXPROPS] section 2.569) 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 in the *Filter* parameter of the **NspiGetMatches** method,<47> as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.10, or in the **Filter** field of the **GetMatches** request body,<48> as specified in [MS-OXCMAPIHTTP] section 2.2.5.5.1. Messaging clients pass this property as a target string in the **NspiGetMatches** method or **GetMatches** request body to identify objects in an address list that are possible matches for the target string. This operation is known as **ambiguous name resolution (ANR)**, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.7.<49> An address book server responds 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 ANR results of an address book server.<50>

An offline address book (OAB) 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 OAB.

2.2.10.2 PidTagAddressBookManageDistributionList

Data type: **PtypObject** ([MS-OXCDATA] section 2.11.1)

The **PidTagAddressBookManageDistributionList** property ([MS-OXPROPS] section 2.538) is a **property tag** for use in display templates for distribution lists. When the value for the **PidTagAddressBookManageDistributionList** property 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 in the user interface to enable them to edit the members of a distribution list. For details about the button control for a template, see [MS-OXOABKT] section 2.2.2.1.3.6. This is not a property of objects in an address book.

2.2.11 Named Properties

This document does not specify any **named properties**. OABs and NSPI servers can expose any named properties in their implementations.

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

All abstract data model (ADM) types maintained by the client are prefixed with "Client".

The following ADM types are defined in this section:

Client.AddressBook: A collection of **Client.AddressBookObject** ADM types, each of which are contained in any number of **AddressList** ADM types. Among the many types of objects, a **Client.AddressBook** ADM type includes the following object types:

- Mailbox user, each of which is a mailbox owner.
- Mail users, each of which describes a person or entity that can receive e-mail messages.
- Distribution lists, each of which is a collection of other mail users, distribution lists, or other **Client.AddressBookObject** ADM types that can receive e-mail messages.
- Resource objects that can be reserved, such as rooms or equipment.
- Organization objects, each of which describes an organization.
- Department objects, which describe the departmental structure of an organization.
- Templates, each of which describes a physical view that can be used to show details on other Address Book objects to a messaging user, as specified in [MS-OXOABKT].
- **Client.AddressBookContainers** ADM types.

Client.AddressBookContainer: A **Client.AddressBookObject** ADM type that describes a **Client.AddressList** ADM type.

Client.AddressBookHierarchyTable: A collection of **Client.AddressBookContainer** ADM types arranged in a hierarchy. 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 **Client.AddressList** ADM type. Messaging users are able to display information about a **Client.AddressBookObject** ADM type from the table, an object obtained through ANR, or a recipient on a message. The information displayed can be formatted according to a display template that is supplied in the **Client.AddressBookObject** ADM type, and the type of information depends on the type of **Client.AddressBookObject** ADM type being displayed.

Client.AddressBookObject: An entity in a **Client.AddressBook** ADM type that contains a set of **attributes**, each attribute with a set of associated values

Client.AddressList: A collection of distinct **Client.AddressBookObject** ADM types.

Client.OfflineAddressBook: A collection of **Client.AddressList** ADM types that are stored in a format that a client can save and use locally.

3.1.2 Timers

None.

3.1.3 Initialization

Initialization is accomplished in one of four ways: through a name service provider interface (NSPI) connection to a server, through the contents of an offline address book (OAB), or both, or through a Messaging Application Programming Interface (MAPI) Extensions for HTTP connection.<51>

3.1.3.1 Initialization Through an NSPI Connection

Initialization is accomplished by means of the **NspiBind** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.1,<52> and MUST occur before any Address Book objects can be accessed through NSPI calls.

3.1.3.2 Initialization Through an HTTP Connection

Initialization is accomplished by means of the **Bind** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.1, and MUST occur before any Address Book objects can be accessed through Messaging Application Programming Interface (MAPI) Extensions for HTTP request types.<53>

3.1.3.3 Initialization Through an Offline Address Book

Messaging clients MAY<54> obtain address lists in an OAB by using the OAB Public Folder Retrieval Protocol, as specified in [MS-OXPFOAB]. Each address list, in turn, contains information about objects in that address list, using the OAB Format and Schema Protocol, as specified in [MS-OXOAB]. After one or more address lists from an OAB are obtained, messaging clients navigate any of the address lists to perform the lookups that are required to perform these operations.

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 that is represented by an address book container, obtain information about an Address Book object, and perform ambiguous name resolution (ANR) to find Address Book objects that match a specified target string.

3.1.4.1 Obtaining a Hierarchy of Address Book Containers

When using an address book server, messaging clients obtain the address book hierarchy table by using either the **NspiGetSpecialTable** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.3, or the **GetSpecialTable** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.8.<55><56> The hierarchy table is a set of rows, each of which describes one address book container.

The hierarchy table in an OAB MAY<57> be obtained by using the OAB Public Folder Retrieval Protocol, as specified in [MS-OXPFOAB].

3.1.4.2 Browsing an Address Book

After 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 that is represented by the container.

When browsing an address list for an OAB, it is up to the messaging client to perform the lookups that are necessary to present a browsable view of the address list, by using the OAB Format and Schema Protocol, as specified in [MS-OXOAB].

When browsing an address list from an address book server, messaging clients use either the **NspiQueryRows** method, as specified in [MS-OXNSPI] section 3.1.4.1.8, or the **QueryRows** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.12, to obtain a set of rows to display to the messaging user.<58> Messaging clients choose the properties they want to render, minimally they SHOULD request the **PidTagEntryId** property (section 2.2.3.2), the **PidTagDisplayName** property (section 2.2.3.1), the **PidTagSmtpAddress** property (section 2.2.3.21), and the **PidTagTitle** property (section 2.2.4.4), 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 [MS-OXNSPI] section 2.2.8,<59> and then calling the **NspiUpdateStat** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.4.<60> When using the Messaging Application Programming Interface (MAPI) Extensions for HTTP protocol, clients update the **STAT** structure using the **UpdateStat** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.17.<61> Messaging clients follow up by using either the **NspiQueryRows** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.8, or the QueryRows request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.12, to display rows starting at the new position.<62>

It is possible for the user to type a particular 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 **NspiSeekEntries** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.9. When using the Messaging Application Programming Interface (MAPI) Extensions for HTTP protocol, clients use the **SeekEntries** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.16.<63> The result updates positioning information in the **STAT** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.8, and returns a screen full of Address Book objects.

3.1.4.3 Obtaining Properties on an Address Book Object

To obtain properties for an Address Book object from an OAB, the messaging client needs to have either a DN for the object or the object's Simple Mail Transfer Protocol (SMTP) address. This means that the messaging client will need to keep track of the DN after any lookup in the OAB, and it 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. Incoming messages can contain either the SMTP address or the DN of message addressees. The messaging server provides the SMTP address or the DN of the addressee. The OAB Format and Schema Protocol, as specified in [MS-OXOAB], describes how this information is organized and structured. After 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 telephone number and SMTP address, the client looks up the **PidTagBusinessTelephoneNumber** property (section 2.2.4.21) and the **PidTagSmtpAddress** property (section 2.2.3.21) in the record.

To obtain properties for an Address Book object on an address book server, the messaging client needs to either have a DN, a Minimal Entry ID, or the object's SMTP address. Specific NSPI calls and corresponding Messaging Application Programming Interface (MAPI) Extensions for HTTP request types that return Address Book object information — including but not limited to:

- The **NspiQueryRows** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.8<64>
- The **QueryRows** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.12<65>
- The **NspiSeekEntries** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.9
- The **SeekEntries** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.16<66>

- The **NspiGetMatches** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.10
- The **GetMatches** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.5<67>

— include the DN or Minimal Entry ID in their calls for the **PidTagEntryId** property, or as a return value of the calls. Clients use the **NspiDNToMId** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.13, or the **DnToMinId** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.4, to obtain the Minimal Entry ID from a DN.<68> To obtain a DN from an SMTP address, messaging clients construct a string that starts with "=SMTP: " followed by the SMTP address as the *paStr* or *paWStr* parameter to either the **NspiResolveNames** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.16, or the **NspiResolveNamesW** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.17, or the **ResolveNames** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.14, and extract the DN from the **PidTagEntryId** property (section 2.2.3.2) that is returned, as specified in the **PermanentEntryID** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.9.3.<69><70>

After the Minimal Entry ID is known, clients use it as the **CurrentRec** member of the **STAT** structure passed to the **NspiGetProps** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.7, or to the **GetProps** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.7.<71> It is possible for clients to specify any number of properties (either in the *pPropTags* parameter of the **NspiGetProps** method or in the **PropertyTags** structure of the **GetProps** request type) 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** for the **PidTagBusinessTelephoneNumber** (section 2.2.4.21) and **PidTagSmtpAddress** (section 2.2.3.21) properties in the *pPropTags* parameter.

To retrieve a property value from the NSPI server for the following properties, use the **NspiGetMatches** method or the **GetMatches** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.5:<72>

- **PidTagAddressBookHomeMessageDatabase** (section 2.2.4.37)
- **PidTagAddressBookMember** property (section 2.2.6.1)
- **PidTagAddressBookManager** property (section 2.2.5.1)
- **PidTagAddressBookReports** property (section 2.2.5.2)
- **PidTagAddressBookIsMemberOfDistributionList** property (section 2.2.5.3)
- **PidTagAddressBookOwnerBackLink** property (section 2.2.5.4)
- **PidTagAddressBookOwner** property (section 2.2.6.2)

To retrieve these properties, set the **CurrentRec** field of the *pStat* input parameter to the **EphemeralEntryID** of the object and the **ContainerId** field to the property tag of the property to retrieve. The **EphemeralEntryId** structure is specified in [MS-OXNSPI] section 2.2.9.2.

3.1.4.4 Performing Ambiguous Name Resolution

Messaging clients perform ambiguous name resolution (ANR) 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, to present the list of possible Address Book objects to the user.

When performing ANR by using an offline address book (OAB), it is up to the messaging client to decide the best method for name matching among Address Book objects. The client is free to use any available means to determine good matches on any choice of properties by using the OAB Format and Schema Protocol, as specified in [MS-OXOAB].

When performing ANR by using an address book server, the client uses the **NspiResolveNames** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.16,<73> or the **NspiResolveNamesW** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.17, or the **ResolveNames** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.14. The client passes the target string in the *paStr* parameter of the **NspiResolveNames** method, the *paWStr* parameter of the **NspiResolveNamesW** method, or the **NameValues** field of the **ResolveNames** request type.<74> If there is exactly one match, the row that corresponds to the target string will contain the properties that were requested for that match. If there is more than one match, it is possible to obtain the set of possible matches by using the **NspiGetMatches** method, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.10, or the **GetMatches** request type, as specified in [MS-OXCMAPIHTTP] section 2.2.5.5, by using a *Filter* parameter with the **PidTagAnr** property and using the value for the **PidTagAnr** property as the target string for ANR.<75> The set of matches, along with the requested properties for the matches, will be returned. Messaging clients display the returned results in a dialog box, so the user is able to select the best match from the list of results.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

All ADM types maintained by the server are prefixed with "Server".

The following ADM types are defined in this section:

Server.AddressBook: A collection of **Server.AddressBookObject** ADM types, each of which are contained in any number of **Server.AddressList** ADM types. Among the many types of objects, a **Server.AddressBook** ADM type includes the following object types:

- Mailbox user, each of is a mailbox owner.
- Mail users, each of which describes a person or entity that can receive e-mail messages.
- Distribution lists, each of which is a collection of other mail users, distribution lists, or other **Server.AddressBookObject** ADM types that can receive e-mail messages.
- Resource objects that can be reserved, such as rooms or equipment.
- Organization objects, each of which describes an organization.
- Department objects, which describe the departmental structure of an organization.

- Templates, each of which describe a physical view that can be used to show details on other Server.AddressBookObjects to a messaging user, as specified in [MS-OXOABKT].
- **Server.AddressBookContainer** ADM types.

Server.AddressBookContainer: A **Server.AddressBookObject** ADM type that describes a **Server.AddressList** ADM type.

Server.AddressBookHierarchyTable: A collection of **Server.AddressBookContainer** ADM types arranged in a hierarchy. 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 **Server.AddressList** ADM type. Messaging users are able to display information about a **Server.AddressBookObject** ADM type from the table, an object obtained through ANR, or a recipient on a message. The information displayed can be formatted according to a display template that is supplied in the **Server.AddressBookObject** ADM type, and the type of information depends on the type of **Server.AddressBookObject** ADM type being displayed.

Server.AddressBookObject: An entity in a **Server.AddressBook** ADM type that contains a set of attributes, each attribute with a set of associated values

Server.AddressList: A collection of distinct **Server.AddressBookObject** ADM types.

Server.OfflineAddressBook: A collection of **Server.AddressList** ADM types that are stored in a format that a client can save and use locally.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

An OAB does not contain the following properties, but the server MUST generate values for these properties before sending the OAB to the client:

- **PidTagDepth** (section 2.2.2.2)
- **PidTagContainerFlags** (section 2.2.2.1)

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

Note For examples of using the Messaging Application Programming Interface (MAPI) Extensions for HTTP, see [MS-OXCMAPIHTTP] section 4.<76>

4.1 Call Sequence To Retrieve Address Book Properties

This section describes the call sequence for obtaining two string properties — **PidTagDisplayName** (section 2.2.3.1) and **PidTagGivenName** (section 2.2.4.2) — for a mail user whose DN is "/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/CN=Recipients/CN=user1." The following figure shows the call sequence between the client and the server.

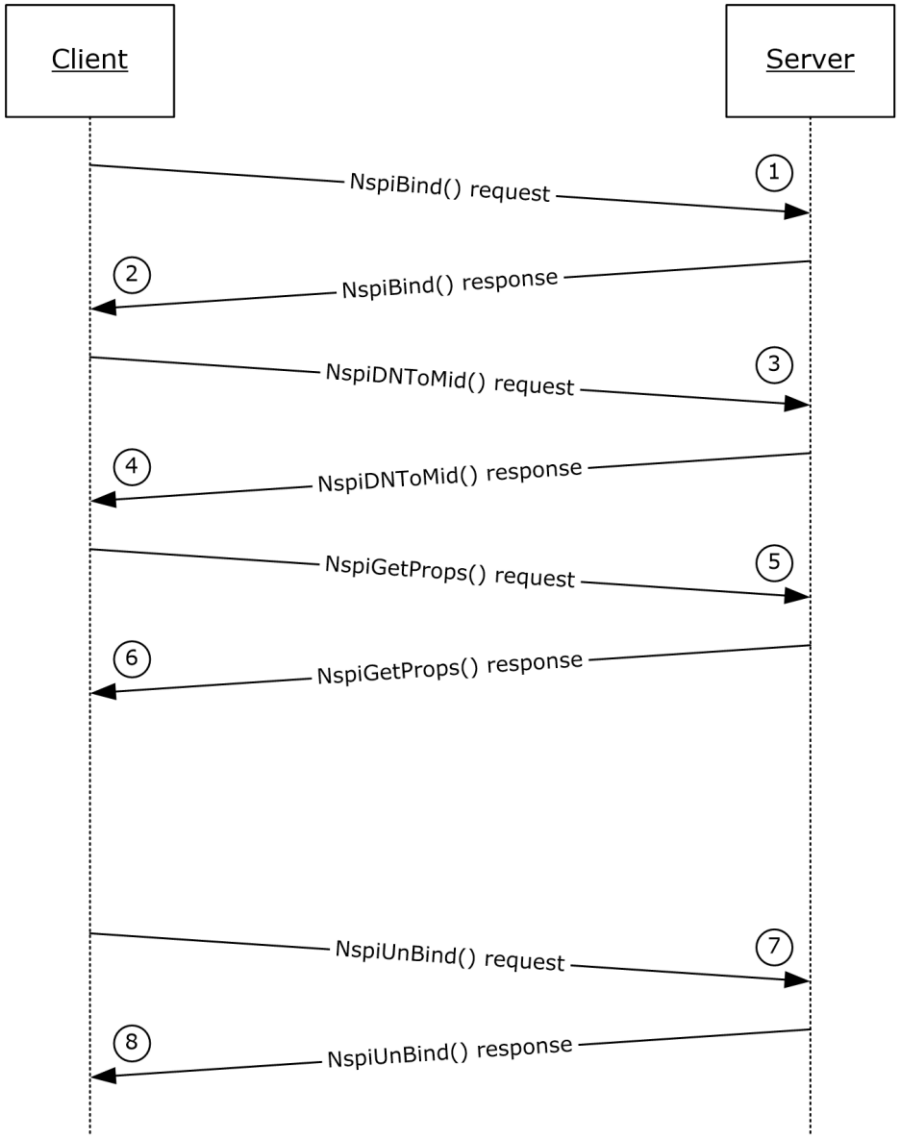


Figure 1: Call sequence between client and server

Note This figure shows relevant information only, not all of the possible parameters. For more details about the parameters, see [MS-NSPI] and [MS-OXNSPI].

1. The client initiates a session to the NSPI server by calling the **NspiBind** method. The messaging client sends in the following values to the server:

```
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
      < a pointer to an array of 16 unsigned chars to be returned by the server>
```

2. The server responds to the **NspiBind** method call with the return code "Success" and a valid server GUID. Typical parameters are as follows:

```
pServerGuid
               [0x0]     0xab 0xbc 0x8b 0x86 0x79 0x33 0xc4         0x48 0xa1 0xef
               [0xa]     0x1b 0x53 0xe6 0x3b 0xdc 0x46
contextHandle
               < a token that will be used by the NspiUnbind call>
```

3. The client requests the Minimal Entry ID for the DN "/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/CN=Recipients/CN=user1" by calling the **NspiDNToMId** method with the following parameters:

```
_pNames
               Count        0x00000001               DWORD
               Strings                               char **
                            [0x0]                    char *
               "/o=First Organization/ou=Exchange Administrative Group
               (FYDIBOHF23SPDLT)/cn=Recipients/cn=user1"
ppMIds
       <a pointer to a valid memory location of type PropertyTagArray r ** for server to
return Mid>
```

4. The server responds with return code of **Success** with a value of 0x00001927 for a Minimal Entry ID as follows:

```
              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** (section 2.2.3.1) and **PidTagGivenName** (section 2.2.4.2) — by calling the **NspiGetProps** method with the following parameters:

Note The Minimal Entry ID value of 0x00001927 that was obtained in step 4 is used as the **CurrentRec** field of the *pStat* parameter.

```
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 the **NspiGetProps** method call with the return code of **Success**. In this example, the server has returned the string value "user1" for both of the requested properties, and the return values are represented as follows:

```
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 is represented as follows:

```
aRow       [0x0]      SRow r  *
           {
                     cValues         0x00000002           DWORD
```

```
                    lpProps=<a pointer to an array of columns>
            }
```

In this example, the server has returned a column set of two properties, and each column will be represented as follows:

```
[0x0]     _SPropValue_r
          {
          [0x0]
                    ulPropTag         PidTagDisplayName      unsigned long
                    Value
                              lpszA =  "user1"char *
          [0x1]
                    ulPropTag         PidTagGivenName        unsigned long
                    Value
                              lpszA =  "user1" char *
          }
```

Note The client can invoke additional NSPI calls to access other information from the server before calling the **NSPIUnbind** method.

7. The client terminates the connection by calling the **NspiUnbind** method with a token that the server returned in response to an **NspiBind** method 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 the return code 0x00000001 and destroys the token that the client passed in.

4.2 PidTagUserX509Certificate Property in ASN.1 DER Encoded Format

The following example shows the parsed text output of a sample **PidTagUserX509Certificate** property (section 2.2.4.36) value in ASN.1 DER encoded **SignedData** type BLOB format.

The first column contains the offset to the node in bytes, the second column contains the length of the node in bytes, the third column contains the byte count of the hex value of the second column (for example, 2228 is DB 6D BC in hex, which is 3 bytes), and the fourth column contains the text output of the node contents.

```
  0|  2228|      3| SEQUENCE :
  4|     9|      1|    OBJECT IDENTIFIER : signedData [1.2.840.113549.1.7.2]
 15|  2213|      3|    CONTEXT SPECIFIC (0) :
 19|  2209|      3|       SEQUENCE :
 23|     1|      1|          INTEGER : 1
 26|    11|      1|          SET :
 28|     9|      1|             SEQUENCE :
 30|     5|      1|                OBJECT IDENTIFIER : sha1 [1.3.14.3.2.26]
 37|     0|      1|                NULL : ''
 39|    25|      1|          SEQUENCE :
 41|     9|      1|             OBJECT IDENTIFIER : data [1.2.840.113549.1.7.1]
 52|    12|      1|             CONTEXT SPECIFIC (0) :
 54|    10|      1|                OCTET STRING :
   |      |       |                   'Empty Body'
 66|  1509|      3|          CONTEXT SPECIFIC (0) :
 70|   638|      3|             SEQUENCE :
 74|   487|      3|                SEQUENCE :
```

```
    78|     3|      1|                  CONTEXT SPECIFIC (0) :
    80|     1|      1|                     INTEGER : 2
    83|    16|      1|                  INTEGER :
      |      |       |                     0274DE397E583BF0DB53CE0EECB024A0
   101|    13|      1|                  SEQUENCE :
   103|     9|      1|                     OBJECT IDENTIFIER : sha1withRSAEncryption
[1.2.840.113549.1.1.5]
   114|     0|      1|                     NULL : ''
   116|    86|      1|                  SEQUENCE :
   118|    60|      1|                     SET :
   120|    58|      1|                        SEQUENCE :
   122|     9|      1|                           OBJECT IDENTIFIER : emailAddress
[1.2.840.113549.1.9.1]
   133|    45|      1|                           IA5 STRING :
      |      |       |                              'Administrator@ssvsim-dom.extes'
      |      |       |                              't.microsoft.com'
   180|    22|      1|                     SET :
   182|    20|      1|                        SEQUENCE :
   184|     3|      1|                           OBJECT IDENTIFIER : commonName [2.5.4.3]
   189|    13|      1|                           UTF8 STRING :
      |      |       |                              'Administrator'
   204|    30|      1|                  SEQUENCE :
   206|    13|      1|                     UTC TIME :
      |      |       |                        '111108025113Z'
   221|    13|      1|                     UTC TIME :
      |      |       |                        '111208025113Z'
   236|    86|      1|                  SEQUENCE :
   238|    60|      1|                     SET :
   240|    58|      1|                        SEQUENCE :
   242|     9|      1|                           OBJECT IDENTIFIER : emailAddress
[1.2.840.113549.1.9.1]
   253|    45|      1|                           IA5 STRING :
      |      |       |                              'Administrator@ssvsim-dom.extes'
      |      |       |                              't.microsoft.com'
   300|    22|      1|                     SET :
   302|    20|      1|                        SEQUENCE :
   304|     3|      1|                           OBJECT IDENTIFIER : commonName [2.5.4.3]
   309|    13|      1|                           UTF8 STRING :
      |      |       |                              'Administrator'
   324|   159|      2|                  SEQUENCE :
   327|    13|      1|                     SEQUENCE :
   329|     9|      1|                        OBJECT IDENTIFIER : rsaEncryption
[1.2.840.113549.1.1.1]
   340|     0|      1|                        NULL : ''
   342|   141|      2|                     BIT STRING UnusedBits:0 :
      |      |       |
   346|   137|      2|                        SEQUENCE :
   349|   129|      2|                           INTEGER :
      |      |       |                              00921EA7244DD0D37C9107C2B46D7C
      |      |       |                              2B8BE61B22E971488E559B47E122D8
      |      |       |                              D255F248062BCB35FE3FEC0AE07EC3
      |      |       |                              2FB59196C987B953EC28D68741BA26
      |      |       |                              F794902FDC1AB9F7C8F33A9ED9398D
      |      |       |                              4C363EC56D05E21EDC01DF3F01EB18
      |      |       |                              C4B80087039B3D3C77E3792028EEF3
      |      |       |                              635DBC70C811F327D3493DAA431E23
      |      |       |                              69879056081F722827
   481|     3|      1|                           INTEGER : 65537
   486|    77|      1|                  CONTEXT SPECIFIC (3) :
   488|    75|      1|                     SEQUENCE :
   490|    29|      1|                        SEQUENCE :
   492|     3|      1|                           OBJECT IDENTIFIER : subjectKeyIdentifier
[2.5.29.14]
   497|    22|      1|                           OCTET STRING :
   499|    20|      1|                              OCTET STRING :
      |      |       |                                 C38DC1AB5265B974F7A5DFBDE8E
      |      |       |                                 B75F60E2AEF8F
   521|    11|      1|                        SEQUENCE :
   523|     3|      1|                           OBJECT IDENTIFIER : keyUsage [2.5.29.15]
```

```
   528|     4|      1|                           OCTET STRING :
   530|     2|      1|                              BIT STRING UnusedBits:7 :
      |      |       |                                 80
   534|    29|      1|                        SEQUENCE :
   536|     3|      1|                           OBJECT IDENTIFIER : extKeyUsage [2.5.29.37]
   541|    22|      1|                           OCTET STRING :
   543|    20|      1|                              SEQUENCE :
   545|     8|      1|                                 OBJECT IDENTIFIER : clientAuth
[1.3.6.1.5.5.7.3.2]
   555|     8|      1|                                 OBJECT IDENTIFIER : emailProtection
[1.3.6.1.5.5.7.3.4]
   565|    13|      1|                  SEQUENCE :
   567|     9|      1|                     OBJECT IDENTIFIER : sha1withRSAEncryption
[1.2.840.113549.1.1.5]
   578|     0|      1|                     NULL : ''
   580|   129|      2|                  BIT STRING UnusedBits:0 :
      |      |       |                     2DEC723B68A73E04779D17854E8F9068E6E0ACD36F
      |      |       |                     ABDCB5DA3A8FCB12772547AED11A03B13F6358DED2
      |      |       |                     2CE1B568D70AED842A9B099C4B77B21ABC0E48F751
      |      |       |                     CC4251A8804472DD87A4EE04709923AD711A1C6267
      |      |       |                     5460EA27A380773542A1C248BCD30EA5BDEEA923D8
      |      |       |                     385523176DF5987D9C668E847B3A2D5C57CD22F693
      |      |       |                     2CF4
   712|   863|      3|               SEQUENCE :
   716|   712|      3|                  SEQUENCE :
   720|     3|      1|                     CONTEXT SPECIFIC (0) :
   722|     1|      1|                        INTEGER : 2
   725|    10|      1|                     INTEGER :
      |      |       |                        7EC4FFE5000000001F5F
   737|    13|      1|                     SEQUENCE :
   739|     9|      1|                        OBJECT IDENTIFIER : sha1withRSAEncryption
[1.2.840.113549.1.1.5]
   750|     0|      1|                        NULL : ''
   752|    21|      1|                     SEQUENCE :
   754|    19|      1|                        SET :
   756|    17|      1|                           SEQUENCE :
   758|     3|      1|                              OBJECT IDENTIFIER : commonName [2.5.4.3]
   763|    10|      1|                              PRINTABLE STRING :
      |      |       |                                 'RSACERTSRV'
   775|    30|      1|                     SEQUENCE :
   777|    13|      1|                        UTC TIME :
      |      |       |                           '111108010318Z'
   792|    13|      1|                        UTC TIME :
      |      |       |                           '121108011318Z'
   807|    86|      1|                     SEQUENCE :
   809|    22|      1|                        SET :
   811|    20|      1|                           SEQUENCE :
   813|     3|      1|                              OBJECT IDENTIFIER : commonName [2.5.4.3]
   818|    13|      1|                              PRINTABLE STRING :
      |      |       |                                 'Administrator'
   833|    60|      1|                        SET :
   835|    58|      1|                           SEQUENCE :
   837|     9|      1|                              OBJECT IDENTIFIER : emailAddress
[1.2.840.113549.1.9.1]
   848|    45|      1|                              IA5 STRING :
      |      |       |                                 'Administrator@ssvsim-dom.extes'
      |      |       |                                 't.microsoft.com'
   895|   159|      2|                     SEQUENCE :
   898|    13|      1|                        SEQUENCE :
   900|     9|      1|                           OBJECT IDENTIFIER : rsaEncryption
[1.2.840.113549.1.1.1]
   911|     0|      1|                           NULL : ''
   913|   141|      2|                        BIT STRING UnusedBits:0 :
      |      |       |
   917|   137|      2|                           SEQUENCE :
   920|   129|      2|                              INTEGER :
      |      |       |                                 009B3B26A3CCF7D8F375B253D2AA95
      |      |       |                                 F2AE82AEEC1337732A455018689388
      |      |       |                                 75080011AD2F600B0B6FA11ABC91F1
```

```
      |      |       |                          4020235E743D3FB250D071C606A338
      |      |       |                          92F4657A30F55C4422ED2165C40E59
      |      |       |                          884CF92EE84848D840DE6E75741955
      |      |       |                          63C019546AFC59F81EC74F6D5543CE
      |      |       |                          AFA5C972507AE1D773363B6406DD1E
      |      |       |                          3AD11929E3F3839453
  1052|     3|      1|                       INTEGER : 65537
  1057|   371|      3|                 CONTEXT SPECIFIC (3) :
  1061|   367|      3|                    SEQUENCE :
  1065|    14|      1|                       SEQUENCE :
  1067|     3|      1|                          OBJECT IDENTIFIER : keyUsage [2.5.29.15]
  1072|     1|      1|                          BOOLEAN :
      |      |       |                             'ÿ'
  1075|     4|      1|                          OCTET STRING :
  1077|     2|      1|                             BIT STRING UnusedBits:4 :
      |      |       |                                30
  1081|    19|      1|                       SEQUENCE :
  1083|     3|      1|                          OBJECT IDENTIFIER : extKeyUsage [2.5.29.37]
  1088|    12|      1|                          OCTET STRING :
  1090|    10|      1|                             SEQUENCE :
  1092|     8|      1|                                OBJECT IDENTIFIER : emailProtection
[1.3.6.1.5.5.7.3.4]
  1102|    29|      1|                       SEQUENCE :
  1104|     3|      1|                          OBJECT IDENTIFIER : subjectKeyIdentifier
[2.5.29.14]
  1109|    22|      1|                          OCTET STRING :
  1111|    20|      1|                             OCTET STRING :
      |      |       |                                CAD24F57FA070174A0BB5C02594
      |      |       |                                2E0C72AB912B3
  1133|    31|      1|                       SEQUENCE :
  1135|     3|      1|                          OBJECT IDENTIFIER : authorityKeyIdentifier
[2.5.29.35]
  1140|    24|      1|                          OCTET STRING :
  1142|    22|      1|                             SEQUENCE :
  1144|    20|      1|                                CONTEXT SPECIFIC (0) :
      |      |       |                                   71DDD1A9BB0D7EE05F82FC9D
      |      |       |                                   C59D7A5E098E41EF
  1166|   107|      1|                       SEQUENCE :
  1168|     3|      1|                          OBJECT IDENTIFIER : cRLDistributionPoints
[2.5.29.31]
  1173|   100|      1|                          OCTET STRING :
  1175|    98|      1|                             SEQUENCE :
  1177|    96|      1|                                SEQUENCE :
  1179|    94|      1|                                   CONTEXT SPECIFIC (0) :
  1181|    92|      1|                                      CONTEXT SPECIFIC (0) :
  1183|    43|      1|                                         CONTEXT SPECIFIC (6) :
      |      |       |                                            'http://rsacerts'
      |      |       |                                            'rv/CertEnroll/R'
      |      |       |                                            'SACERTSRV.crl'
  1228|    45|      1|                                         CONTEXT SPECIFIC (6) :
      |      |       |                                            'file://\\rsacer'
      |      |       |                                            'tsrv\CertEnroll'
      |      |       |                                            '\RSACERTSRV.crl'
  1275|   154|      2|                       SEQUENCE :
  1278|     8|      1|                          OBJECT IDENTIFIER : authorityInfoAccess
[1.3.6.1.5.5.7.1.1]
  1288|   141|      2|                          OCTET STRING :
  1291|   138|      2|                             SEQUENCE :
  1294|    66|      1|                                SEQUENCE :
  1296|     8|      1|                                   OBJECT IDENTIFIER : caIssuers
[1.3.6.1.5.5.7.48.2]
  1306|    54|      1|                                   CONTEXT SPECIFIC (6) :
      |      |       |                                      'http://rsacertsrv/Cer'
      |      |       |                                      'tEnroll/rsacertsrv_RS'
      |      |       |                                      'ACERTSRV.crt'
  1362|    68|      1|                                SEQUENCE :
  1364|     8|      1|                                   OBJECT IDENTIFIER : caIssuers
[1.3.6.1.5.5.7.48.2]
  1374|    56|      1|                                   CONTEXT SPECIFIC (6) :
```

```
      |      |       |                                    'file://\\rsacertsrv\C'
      |      |       |                                    'ertEnroll\rsacertsrv_'
      |      |       |                                    'RSACERTSRV.crt'
  1432|    13|      1|               SEQUENCE :
  1434|     9|      1|                  OBJECT IDENTIFIER : sha1withRSAEncryption
[1.2.840.113549.1.1.5]
  1445|     0|      1|                  NULL : ''
  1447|   129|      2|               BIT STRING UnusedBits:0 :
      |      |       |                  8F409689A9F3125F18A1D52DC2C702533F993E3342
      |      |       |                  27967C07F682C27634D5DC65EA0A9BE66DF5E1BDF9
      |      |       |                  A2A780C4FC870BDF4B9D97D07F12B738B9B6636EBD
      |      |       |                  6B64E130BD85B100FA6AF676D5A318FBBFEEB03E2D
      |      |       |                  F86939B86F107753C89E1FAF76DDDA6B83F15AE950
      |      |       |                  5D754D3869B30DCA42F7C688AA935593E61ABC6F6A
      |      |       |                  F879
  1579|   649|      3|          SET :
  1583|   645|      3|             SEQUENCE :
  1587|     1|      1|                INTEGER : 1
  1590|   106|      1|                SEQUENCE :
  1592|    86|      1|                   SEQUENCE :
  1594|    60|      1|                      SET :
  1596|    58|      1|                         SEQUENCE :
  1598|     9|      1|                            OBJECT IDENTIFIER : emailAddress
[1.2.840.113549.1.9.1]
  1609|    45|      1|                            IA5 STRING :
      |      |       |                               'Administrator@ssvsim-dom.extes'
      |      |       |                               't.microsoft.com'
  1656|    22|      1|                      SET :
  1658|    20|      1|                         SEQUENCE :
  1660|     3|      1|                            OBJECT IDENTIFIER : commonName [2.5.4.3]
  1665|    13|      1|                            UTF8 STRING :
      |      |       |                               'Administrator'
  1680|    16|      1|                   INTEGER :
      |      |       |                      0274DE397E583BF0DB53CE0EECB024A0
  1698|     9|      1|                SEQUENCE :
  1700|     5|      1|                   OBJECT IDENTIFIER : sha1 [1.3.14.3.2.26]
  1707|     0|      1|                   NULL : ''
  1709|   373|      3|                CONTEXT SPECIFIC (0) :
  1713|    16|      3|                   SEQUENCE :
  1715|     9|      1|                      OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.16.3]
  1726|     3|      1|                      SET :
  1728|     1|      1|                         INTEGER : 7
  1731|    24|      1|                   SEQUENCE :
  1733|     9|      1|                      OBJECT IDENTIFIER : contentType
[1.2.840.113549.1.9.3]
  1744|    11|      1|                      SET :
  1746|     9|      1|                         OBJECT IDENTIFIER : data
[1.2.840.113549.1.7.1]
  1757|    35|      1|                   SEQUENCE :
  1759|     9|      1|                      OBJECT IDENTIFIER : messageDigest
[1.2.840.113549.1.9.4]
  1770|    22|      1|                      SET :
  1772|    20|      1|                         OCTET STRING :
      |      |       |                            68DD2DD63498036679AD9081C69A564A4
      |      |       |                            A6A0E1A
  1794|    50|      1|                   SEQUENCE :
  1796|     9|      1|                      OBJECT IDENTIFIER : Outlook Express
[1.3.6.1.4.1.311.16.4]
  1807|    37|      1|                      SET :
  1809|    35|      1|                         SEQUENCE :
  1811|    21|      1|                            SEQUENCE :
  1813|    19|      1|                               SET :
  1815|    17|      1|                                  SEQUENCE :
  1817|     3|      1|                                     OBJECT IDENTIFIER : commonName
[2.5.4.3]
  1822|    10|      1|                                     PRINTABLE STRING :
      |      |       |                                        'RSACERTSRV'
  1834|    10|      1|                            INTEGER :
      |      |       |                               7EC4FFE5000000001F5F
```

```
  1846|    52|      1|                  SEQUENCE :
  1848|    11|      1|                     OBJECT IDENTIFIER :  [1.2.840.113549.1.9.16.2.11]
  1861|    37|      1|                     SET :
  1863|    35|      1|                        CONTEXT SPECIFIC (0) :
  1865|    21|      1|                           SEQUENCE :
  1867|    19|      1|                              SET :
  1869|    17|      1|                                 SEQUENCE :
  1871|     3|      1|                                    OBJECT IDENTIFIER : commonName
[2.5.4.3]
  1876|    10|      1|                                    PRINTABLE STRING :
      |      |       |                                       'RSACERTSRV'
  1888|    10|      1|                           INTEGER :
      |      |       |                              7EC4FFE5000000001F5F
  1900|   183|      2|                  SEQUENCE :
  1903|     9|      1|                     OBJECT IDENTIFIER : sMIMECapabilities
[1.2.840.113549.1.9.15]
  1914|   169|      2|                     SET :
  1917|   166|      2|                        SEQUENCE :
  1920|    11|      1|                           SEQUENCE :
  1922|     9|      1|                              OBJECT IDENTIFIER :
[2.16.840.1.101.3.4.1.42]
  1933|    11|      1|                           SEQUENCE :
  1935|     9|      1|                              OBJECT IDENTIFIER :
[2.16.840.1.101.3.4.1.22]
  1946|    10|      1|                           SEQUENCE :
  1948|     8|      1|                              OBJECT IDENTIFIER : DES-EDE3-CBC
[1.2.840.113549.3.7]
  1958|    11|      1|                           SEQUENCE :
  1960|     9|      1|                              OBJECT IDENTIFIER :
[2.16.840.1.101.3.4.1.2]
  1971|    14|      1|                           SEQUENCE :
  1973|     8|      1|                              OBJECT IDENTIFIER : rc2CBC
[1.2.840.113549.3.2]
  1983|     2|      1|                              INTEGER : 128
  1987|    13|      1|                           SEQUENCE :
  1989|     8|      1|                              OBJECT IDENTIFIER : rc2CBC
[1.2.840.113549.3.2]
  1999|     1|      1|                              INTEGER : 64
  2002|     7|      1|                           SEQUENCE :
  2004|     5|      1|                              OBJECT IDENTIFIER : desCBC
[1.3.14.3.2.7]
  2011|    13|      1|                           SEQUENCE :
  2013|     8|      1|                              OBJECT IDENTIFIER : rc2CBC
[1.2.840.113549.3.2]
  2023|     1|      1|                              INTEGER : 40
  2026|     7|      1|                           SEQUENCE :
  2028|     5|      1|                              OBJECT IDENTIFIER : sha1 [1.3.14.3.2.26]
  2035|    11|      1|                           SEQUENCE :
  2037|     9|      1|                              OBJECT IDENTIFIER :
[2.16.840.1.101.3.4.2.3]
  2048|    11|      1|                           SEQUENCE :
  2050|     9|      1|                              OBJECT IDENTIFIER :
[2.16.840.1.101.3.4.2.2]
  2061|    11|      1|                           SEQUENCE :
  2063|     9|      1|                              OBJECT IDENTIFIER :
[2.16.840.1.101.3.4.2.1]
  2074|    10|      1|                           SEQUENCE :
  2076|     8|      1|                              OBJECT IDENTIFIER : md5
[1.2.840.113549.2.5]
  2086|    13|      1|               SEQUENCE :
  2088|     9|      1|                  OBJECT IDENTIFIER : rsaEncryption
[1.2.840.113549.1.1.1]
  2099|     0|      1|                  NULL : ''
  2101|   128|      2|               OCTET STRING :
      |      |       |                  0902E60039438E3D884700475000C6625CCD712FD0
      |      |       |                  125C0CCED7C541B6806FD8E609DD29E6E9703672B8
      |      |       |                  F976304652EF10F7ED07FB235E6D4DC2661AD0C3F3
      |      |       |                  F09CA561F4B5D014A200201634C3563C01EC5B60B2
      |      |       |                  332BF2F5C4707211B599FEA7C15FD58EFD8724FC94
```

```
|       |       |                   D96F4A634D3EF21AB20FA2A2AE89225F41DD79776A
|       |       |                   81DC
```

4.3 PidTagUserX509Certificate Property in Binary Format

The following example shows the structure of a sample **PidTagUserX509Certificate** property (section 2.2.4.36) value in binary format.

Property Version Tag: 0x0001

Length of 8 bytes: 0x0008

Data: 0x00000001

KeyExchangeCertificate Tag: 0x0003

Length of 631 bytes: 0x0277

Data:

3082026F308201D8A00302010202103F924830235200BC496308CD3BAEB76B300D06092A864886F7
0D0101050500301531133011060355040313 0A525341434552545352563 01E170D303630333231303
5333132395A170D313630333231303534313131 5A3015311330110603550403130A52534143455254
53525630819F300D06092A864886F70D010101050003818D00308189028181 00B1AA1341E2AD0E3D
3E035B0F69FFD95CED15EBC9B85D74814F64E39D7F62E6CBB6627909ADE845ED578D56A7710849B
B3A084F0BD4AB5E9059448D365D667A5AC41B2B207344CF8EDB76FCCBDC0D80AF6A2228D42CD38
458542B80E21CC0780B9CAD962F6041820E02FBA4997E606DDFFD299A04D8C2C8B12DCE47199D46
935D0203010001A381BF3081BC300B0603551D0F040403020186300F0603551D130101FF040530030
101FF301D0603551D0E0416041471DDD1A9BB0D7EE05F82FC9DC59D7A5E098E41EF306B0603551D
1F046430623060A05EA05C862B687474703A2F2F72736163657274737276 2F43657274456E726F6C6
C2F525341434552545352562E63726C862D66696C653A2F2F5C5C72736163657274737276 5C436572
74456E726F6C6C5C525341434552545352562E63726C301006092B06010401823715010403020100 3
00D06092A864886F70D01010505000381810 01FECD3BBF82DA9AD38CE804BC535AB39ED90F9F3A2
BC73C79F43191CB250AFC4A77C43BA4D0D6DC108998551EC3127AC5598C805F4B7DD76CAEDB431F
8D37D53126F0FF9FE6D829B24F548B6CCA54361629D9B93EA9324AA06C2EA3A39D9A1D29ED7EC07
E9EE6E7C9D6805C1C7D07013A8B264216BBFCCF75330D6240A45F3D6

Encryption Type Tag: 0x0006

Length of 8 bytes: 0x0008

Data indicates S/MIME: 0x00000001

Defaults Tag: 0x0020

Length of 8 bytes: 0x0008

Data indicates an unspecified value: 0x0000007

AsymetricCapabilities: 0x0002

Length of 4 bytes: 0x0004

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to this protocol. Note, however, that general security considerations pertaining to the underlying NSPI RPC-based transport, as described in [MS-NSPI] and [MS-OXNSPI], do apply to this protocol.

The Messaging Application Programming Interface (MAPI) Extensions for HTTP, as described in [MS-OXCMAPIHTTP], do not introduce any security considerations beyond those of the underlying HTTP or **HTTPS** transport protocol.<77>

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

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

- Microsoft Exchange Server 2003
- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

<1> Section 2.1: Exchange 2003 and Exchange 2007 point the client to Active Directory Domain Services (AD DS), which implements the NSPI methods as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the NSPI methods as described in [MS-OXNSPI]. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory® global catalog server, in which case the server points the client to AD DS.

<2> Section 2.1: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The MAPI Extensions for HTTP were introduced in Microsoft Exchange Server 2013 Service Pack 1 (SP1) and Microsoft Outlook 2013 Service Pack 1 (SP1).

<3> Section 2.2: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The MAPI Extensions for HTTP were introduced in Exchange 2013 SP1 and Outlook 2013 SP1.

<4> Section 2.2.1.1: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **PermanentEntryId** structure as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **PermanentEntryId** structure as described in [MS-OXNSPI] section 2.2.9.3. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **PermanentEntryId** structure as described in [MS-NSPI].

<5> Section 2.2.1.1: The OAB version 4 implementation in Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 includes the following properties, which are not included in objects on an NSPI server: **PidTagAddressBookDistributionListExternalMemberCount** (section 2.2.3.30), **PidTagAddressBookDistributionListMemberCount** (section 2.2.3.29), and **PidTagOfflineAddressBookTruncatedProperties** property ([MS-OXOAB] section 2.9.2.2).

<6> Section 2.2.2: Retrieval of information about the address book hierarchy by using the OAB Public Folder Retrieval Protocol, as described in [MS-OXPFOAB], is supported by Exchange 2003, Exchange 2007, and Exchange 2010.

<7> Section 2.2.2.3: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **NspiBind** method as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **NspiBind** method as described in [MS-OXNSPI] section 3.1.4.1.1. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **NSPIBind** method as described in [MS-NSPI].

<8> Section 2.2.2.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the **Bind** request type response. The **Bind** request type response was introduced in Exchange 2013 SP1 and Outlook 2013 SP1.

<9> Section 2.2.3: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the NSPI methods as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the NSPI methods as described in [MS-OXNSPI]. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the NSPI methods as described in [MS-NSPI].

<10> Section 2.2.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the HTTP request types for an address book server endpoint. The HTTP request types for an address book server endpoint were introduced in Exchange 2013 SP1 and Outlook 2013 SP1.

<11> Section 2.2.3.2: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **PermanentEntryID** and **EphemeralEntryID** structures as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **PermanentEntryID** and **EphemeralEntryID** structures as described in [MS-OXNSPI]. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **PermanentEntryID** and **EphemeralEntryID** structures as described in [MS-NSPI].

<12> Section 2.2.3.9: The **PidTagAddressBookPhoneticDisplayName** property (section 2.2.3.9) is not used in Exchange 2003 servers or Office Outlook 2003 clients.

<13> Section 2.2.3.11: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the display type values as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the display type values as described in [MS-OXNSPI] section 2.2.1.3. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the display type values as described in [MS-NSPI].

<14> Section 2.2.3.12: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the display type values as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the display type values as described in [MS-OXNSPI] section 2.2.1.3. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which

case the server points the client to AD DS and implements the display type values as described in [MS-NSPI].

<15> Section 2.2.3.24: The **PidTagAddressBookSeniorityIndex** property (section 2.2.3.24) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<16> Section 2.2.3.25: The **PidTagAddressBookObjectGuid** property (section 2.2.3.25) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<17> Section 2.2.3.26: The **PidTagAddressBookSenderHintTranslations** property (section 2.2.3.26) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<18> Section 2.2.3.27: The **PidTagAddressBookDeliveryContentLength** property (section 2.2.3.27) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<19> Section 2.2.3.28: The **PidTagAddressBookModerationEnabled** property (section 2.2.3.28) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<20> Section 2.2.3.29: The **PidTagAddressBookDistributionListMemberCount** property (section 2.2.3.29) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<21> Section 2.2.3.30: The **PidTagAddressBookDistributionListExternalMemberCount** property (section 2.2.3.30) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<22> Section 2.2.3.35: The **PidTagAddressBookDisplayTypeExtended** property (section 2.2.3.35) is not used in Exchange 2003 servers or Office Outlook 2003 clients.

<23> Section 2.2.4.10: The **PidTagAddressBookPhoneticGivenName** property (section 2.2.4.10) is not used in Exchange 2003 servers or Office Outlook 2003 clients.

<24> Section 2.2.4.11: The **PidTagAddressBookPhoneticSurname** property (section 2.2.4.11) is not used in Exchange 2003 servers or Office Outlook 2003 clients.

<25> Section 2.2.4.12: The **PidTagAddressBookPhoneticCompanyName** property (section 2.2.4.12) is not used by Exchange 2003 servers or Office Outlook 2003 clients.

<26> Section 2.2.4.13: The **PidTagAddressBookPhoneticDepartmentName** property (section 2.2.4.13) is not used by Exchange 2003 servers or Office Outlook 2003 clients.

<27> Section 2.2.4.36.4: The **PidTagAddressBookPhoneticDisplayName** property (section 2.2.3.9) is not used in Exchange 2003 or Office Outlook 2003 servers.

<28> Section 2.2.4.36.5: The **PidTagAddressBookPhoneticDisplayName** property (section 2.2.3.9) is not used in Exchange 2003 or Office Outlook 2003 servers.

<29> Section 2.2.4.36.7: The **PidTagAddressBookPhoneticDisplayName** property (section 2.2.3.9) is not used in Exchange 2003 or Office Outlook 2003 servers.

<30> Section 2.2.4.36.9: The **PidTagAddressBookPhoneticDisplayName** property (section 2.2.3.9) is not used in Exchange 2003 or Office Outlook 2003 servers.

<31> Section 2.2.4.39: The **PidTagAddressBookOrganizationalUnitRootDistinguishedName** property (section 2.2.4.39) is not used by Office Outlook 2003 clients.

<32> Section 2.2.4.40: The **PidTagThumbnailPhoto** property (section 2.2.4.40) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<33> Section 2.2.4.41: The **PidTagSpokenName** property (section 2.2.4.41) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<34> Section 2.2.4.42: The **PidTagAddressBookAuthorizedSenders** property (section 2.2.4.42) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<35> Section 2.2.4.43: The **PidTagAddressBookUnauthorizedSenders** property (section 2.2.4.43) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<36> Section 2.2.4.44: The **PidTagAddressBookDistributionListMemberSubmitAccepted** property (section 2.2.4.44) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<37> Section 2.2.4.45: The **PidTagAddressBookDistributionListMemberSubmitRejected** property (section 2.2.4.45) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<38> Section 2.2.5: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the tables as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the tables as described in [MS-OXNSPI] section 3.1.4.4.2.2. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the tables as described in [MS-NSPI].

<39> Section 2.2.5: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **NspiGetMatches** and **NspiModLinkAtt** methods as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **NspiGetMatches** and **NspiModLinkAtt** methods, as described in [MS-OXNSPI]. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory® global catalog server, in which case the server points the client to AD DS and implements the **NspiGetMatches** and **NspiModLinkAtt** methods as described in [MS-NSPI].

<40> Section 2.2.5: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the **GetMatches** request type and the **ModLinkAtt** request type. The **GetMatches** request type and the **ModLinkAtt** request type were introduced in Exchange 2013 SP1 and Outlook 2013 SP1.

<41> Section 2.2.5.3: The **PidTagAddressBookIsMemberOfDistributionList** property (section 2.2.5.3) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<42> Section 2.2.6.1: The **PidTagAddressBookMember** property (section 2.2.6.1) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<43> Section 2.2.6.5: The **PidTagAddressBookHierarchicalIsHierarchicalGroup** property (section 2.2.6.5) is not used by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<44> Section 2.2.7: The **PidTagAddressBookOrganizationalUnitRootDistinguishedName** property (section 2.2.4.39) is not used by Office Outlook 2003 clients.

<45> Section 2.2.7.2: Office Outlook 2003 clients do not recognize departmental groups. If the department hierarchy is represented by using departmental groups, these clients run as if the **PidTagAddressBookHierarchicalRootDepartment** property (section 2.2.7.2) is not present, and there is no hierarchical representation of the departments.

<46> Section 2.2.8: If a server represents a department hierarchy by using both Department objects and departmental groups, Office Outlook 2003 only makes use of the hierarchy that is represented by the Department objects, and Office Outlook 2007, Outlook 2010, Outlook 2013, and Outlook 2016 only make use of the hierarchy that is represented by the departmental groups. Note that a single hierarchy cannot be represented by a combination of the two schemas.

<47> Section 2.2.10.1: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **NspiGetMatches** methods as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the NspiGetMatches methods as described in [MS-OXNSPI] section 3.1.4.1.10. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **NspiGetMatches** method as described in [MS-NSPI].

<48> Section 2.2.10.1: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the **GetMatches** request body. The **GetMatches** request body was introduced in Exchange 2013 SP1 and Outlook 2013 SP1.

<49> Section 2.2.10.1: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements ANR as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements ANR as described in [MS-OXNSPI] section 3.1.4.7. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the ANR as described in [MS-NSPI].

<50> Section 2.2.10.1: Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, and Outlook 2016 perform a lookup of SMTP addresses by constructing a specific target string for ambiguous name resolution (ANR) that is understood by Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 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 the **NspiGetMatches** method, as described in [MS-OXNSPI] section 3.1.4.1.10. Such a target string is also understood by the *paStr* parameter to the **NspiResolveNames** method, as specified in [MS-OXNSPI] section 3.1.4.1.16, or the *paWStr* parameter to the **NspiResolveNamesW** method, as specified in [MS-OXNSPI] section 3.1.4.1.17.

<51> Section 3.1.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<52> Section 3.1.3.1: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **NspiBind** method as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **NspiBind** method as described in [MS-OXNSPI] section 3.1.4.1.1. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **NspiBind** method as described in [MS-NSPI].

<53> Section 3.1.3.2: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<54> Section 3.1.3.3: Retrieval of address lists by using the OAB Public Folder Retrieval Protocol, as described in [MS-OXPFOAB], is supported by Exchange 2003, Exchange 2007, and Exchange 2010.

<55> Section 3.1.4.1: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **NspiGetSpecialTable** method as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **NspiGetSpecialTable** method as described in [MS-OXNSPI] section 3.1.4.1.3. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global

catalog server, in which case the server points the client to AD DS and implements the **NspiGetSpecialTable** method as described in [MS-NSPI].

<56> Section 3.1.4.1: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<57> Section 3.1.4.1: Retrieval of hierarchy tables by using the OAB Public Folder Retrieval Protocol, as described in [MS-OXPFOAB], is supported by Exchange 2003, Exchange 2007, and Exchange 2010.

<58> Section 3.1.4.2: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<59> Section 3.1.4.2: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **STAT** structure as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **STAT** structure as described in [MS-OXNSPI] section 2.2.8. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **STAT** structure as described in [MS-NSPI].

<60> Section 3.1.4.2: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **NspiUpdateStat** method as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **NspiUpdateStat** method, as described in [MS-OXNSPI] section 3.1.4.1.4. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **NspiUpdateStat** method as described in [MS-NSPI].

<61> Section 3.1.4.2: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<62> Section 3.1.4.2: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<63> Section 3.1.4.2: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<64> Section 3.1.4.3: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **NspiQueryRows** method as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **NspiQueryRows** method as described in [MS-OXNSPI] section 3.1.4.1.8. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **NspiQueryRows** method as described in [MS-NSPI].

<65> Section 3.1.4.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<66> Section 3.1.4.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<67> Section 3.1.4.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<68> Section 3.1.4.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<69> Section 3.1.4.3: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **PermanentEntryID** structure as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **PermanentEntryID** structure as described in [MS-OXNSPI] section 2.2.9.3. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **PermanentEntryID** structure as described in [MS-NSPI].

<70> Section 3.1.4.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<71> Section 3.1.4.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<72> Section 3.1.4.3: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<73> Section 3.1.4.4: Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the **NspiResolveNames**

method, as described in [MS-NSPI]. Exchange 2010, Exchange 2013, and Exchange 2016 point the client to the Exchange NSPI server, which implements the **NspiResolveNames** method, as described in [MS-OXNSPI] section 3.1.4.1.16. The only exception is when Exchange 2010, Exchange 2013, or Exchange 2016 is installed on an Active Directory global catalog server, in which case the server points the client to AD DS and implements the **NspiResolveName** methods, as described in [MS-NSPI].

<74> Section 3.1.4.4: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<75> Section 3.1.4.4: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP. The Messaging Application Programming Interface (MAPI) Extensions for HTTP were introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<76> Section 4: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP protocol. The Messaging Application Programming Interface (MAPI) Extensions for HTTP protocol was introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

<77> Section 5.1: Exchange 2003, Exchange 2007, Exchange 2010, the initial release of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release of Outlook 2013 do not support the Messaging Application Programming Interface (MAPI) Extensions for HTTP protocol. The Messaging Application Programming Interface (MAPI) Extensions for HTTP protocol was introduced in Outlook 2013 SP1 and Exchange 2013 SP1.

7 Change Tracking

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

8 Index

T

V