

# [MS-OXOAB]: Offline Address Book (OAB) File Format and Schema

---

## 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](mailto: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.
- **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.

**Preliminary Documentation.** This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final

documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

## Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability.
04/25/2008	0.2		Revised and updated property names and other technical content.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Revised and edited technical content.
09/03/2008	1.02		Revised and edited technical content.
10/01/2008	1.03		Revised and edited technical content.
12/03/2008	1.04		Updated IP notice.
02/04/2009	1.05		Revised and edited technical content.
03/04/2009	1.06		Revised and edited technical content.
04/10/2009	2.0		Updated technical content for new product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	4.1.0	Minor	Updated the technical content.
05/05/2010	5.0.0	Major	Updated and revised the technical content.
08/04/2010	6.0	Major	Significantly changed the technical content.
11/03/2010	6.1	Minor	Clarified the meaning of the technical content.
03/18/2011	7.0	Major	Significantly changed the technical content.
08/05/2011	7.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/07/2011	7.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	8.0	Major	Significantly changed the technical content.

Preliminary

# Table of Contents

<b>1 Introduction</b>	<b>6</b>
1.1 Glossary	6
1.2 References	7
1.2.1 Normative References	7
1.2.2 Informative References	8
1.3 Overview	8
1.3.1 OAB Version 2 and OAB Version 3	8
1.3.2 OAB Version 4	9
1.4 Relationship to Protocols and Other Structures	10
1.5 Applicability Statement	10
1.6 Versioning and Localization	10
1.7 Vendor-Extensible Fields	10
<b>2 Structures</b>	<b>11</b>
2.1 X500 Distinguished Name	11
2.2 Uncompressed OAB Display Template File	12
2.2.1 OAB_HDR	12
2.2.2 TEMPL_ENTRY	13
2.2.3 NAMES_STRUCT	14
2.3 Uncompressed OAB Version 2 and OAB Version 3 Browse File	15
2.3.1 OAB_HDR	15
2.3.2 B2_REC	16
2.3.3 RDN Hash Computation	17
2.4 Uncompressed OAB Version 2 and OAB Version 3 RDN Index File	18
2.4.1 RDN_HDR	18
2.4.2 RDN2_REC	19
2.5 Uncompressed OAB Version 2 and OAB Version 3 ANR Index File	20
2.5.1 OAB_HDR	21
2.5.2 ANR_REC	21
2.6 Uncompressed OAB Version 2 and OAB Version 3 Details File	22
2.6.1 OAB_HDR	32
2.7 Uncompressed OAB Version 2 and OAB Version 3 Changes File	32
2.7.1 OAB_HDR	34
2.7.2 CHG_REC	34
2.7.3 Change-record	36
2.8 Compressed OAB Version 2 or OAB Version 3 File	39
2.8.1 MDI_HDR	39
2.8.2 MDI_BLK	40
2.9 Uncompressed OAB Version 4 Full Details File	40
2.9.1 OAB_HDR	41
2.9.2 OAB_META_DATA	42
2.9.2.1 rgHdrAtts	42
2.9.2.2 rgOabAtts	44
2.9.2.2.1 Truncated Properties	51
2.9.3 OAB_PROP_TABLE	53
2.9.4 OAB_PROP_REC	53
2.9.5 OAB_V4_REC	55
2.9.6 Data Encoding	55
2.9.6.1 PtypInteger32 (0x0003) Value Encoding	55
2.9.6.2 PtypBoolean (0x000B) Value Encoding	56

2.9.6.3	PtypString8 (0x001E) Value Encoding .....	56
2.9.6.4	PtypString (0x001F) Value Encoding .....	56
2.9.6.5	PtypBinary (0x0102) Value Encoding .....	56
2.9.6.6	PtypMultipleInteger32 (0x1003) Value Encoding .....	56
2.9.6.7	PtypMultipleString8 (0x101E) Value Encoding .....	57
2.9.6.8	PtypMultipleString (0x101F) Value Encoding .....	57
2.9.6.9	PtypMultipleBinary (0x1102) Value Encoding .....	57
2.10	Compressed OAB Version 4 Differential Patch File .....	57
2.10.1	PATCH_HDR .....	58
2.10.2	PATCH_BLK .....	59
2.11	Compressed OAB Version 4 Details File and Compressed OAB Template File .....	59
2.11.1	LZX_HDR .....	60
2.11.2	LZX_BLK .....	60
<b>3</b>	<b>Structure Examples .....</b>	<b>62</b>
3.1	Full OAB Version 2 Offline Address List .....	62
3.2	Full OAB Version 3 Offline Address List .....	66
3.3	Full OAB Version 4 Details File .....	70
<b>4</b>	<b>Security .....</b>	<b>74</b>
4.1	Security Considerations for Implementers .....	74
4.2	Index of Security Parameters .....	74
<b>5</b>	<b>Appendix A: Product Behavior .....</b>	<b>75</b>
<b>6</b>	<b>Change Tracking .....</b>	<b>77</b>
<b>7</b>	<b>Index .....</b>	<b>80</b>

# 1 Introduction

The Offline Address Book (OAB) File Format and Schema describes the **offline address book (OAB)** version 2, OAB version 3, and OAB version 4 file formats. OABs are files that store **address list** information on the client, so that the client can access the information when it does not have a network connection with the server or is working **offline**.

Sections 1.7 and 2 of this specification are normative and contain RFC 2119 language. All other sections and examples in this specification are informative.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**Active Directory**  
**ambiguous name resolution (ANR)**  
**American National Standards Institute (ANSI) character set**  
**ASCII**  
**Augmented Backus-Naur Form (ABNF)**  
**distinguished name (DN)**  
**domain**  
**GUID**  
**little-endian**  
**name service provider interface (NSPI)**  
**relative distinguished name (RDN)**  
**Unicode**  
**X.509**

The following terms are defined in [\[MS-OXGLOS\]](#):

**address book**  
**address book container**  
**Address Book object**  
**address creation template**  
**address list**  
**address type**  
**alias**  
**departmental group**  
**distribution list**  
**Lempel-Ziv Extended (LZX)**  
**Lempel-Ziv Extended Delta (LZXD)**  
**locale**  
**mail tip**  
**mail user**  
**mailbox**  
**offline**  
**offline address book (OAB)**  
**property tag**  
**property type**  
**public folder**  
**recipient**  
**Rich Text Format (RTF)**  
**Simple Mail Transfer Protocol (SMTP)**  
**store**  
**X500 DN**

The following terms are specific to this document:

**parent distinguished name (PDN):** A distinguished name (DN) of an object that is the next immediate object closer to the root of a tree of relative distinguished names (RDNs).

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

## 1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[ISO/IEC8802-3] ISO/IEC, "Information technology -- Telecommunications and information exchange between systems -- Local and metropolitan area networks -- Specific requirements -- Part 3", Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, ISO/IEC 8802-3:2000, [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=31002](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=31002)

[ISO/IEC8825-1] ISO/IEC, "Information technology -- ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ISO/IEC 8825-1:1998, [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=32306](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32306)

[MS-ADTS] Microsoft Corporation, "[Active Directory Technical Specification](#)".

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](#)".

[MS-MCI] Microsoft Corporation, "[MCI Compression and Decompression](#)".

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

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

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

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

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

[MS-PATCH] Microsoft Corporation, "[LZX DELTA Compression and Decompression](#)".

[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>

[RFC2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version 1.5", RFC 2315, March 1998, <http://www.ietf.org/rfc/rfc2315.txt>

[RFC2459] Housley, R., Ford, W., Polk, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January 1999, <http://www.ietf.org/rfc/rfc2459.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>

[RFC3629] Yergeau, F., "UTF-8, A Transformation Format of ISO 10646", STD 63, RFC 3629, November 2003, <http://www.ietf.org/rfc/rfc3629.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>

## 1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[MS-OXGLOS] Microsoft Corporation, "[Exchange Server Protocols Master Glossary](#)".

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

## 1.3 Overview

An **address book** contains user properties, such as job titles, addresses, and telephone numbers that a server makes available to its clients. Clients can browse or search the address book to look for **recipient (2)** information. To organize the contents of an address book, the server can divide recipients (2) into containers and the client can choose which container to browse or search.

Each **address book container** is known as an address list. The collection of available containers, or address lists, is the address book. When the client is unable to reach the server, which can be caused by working offline or having high network costs to access the server, the client can use a local copy of the address book or address lists to retrieve user information. The local copy of the address book is known as an offline address book (OAB).

An OAB is composed of three or more files that provide the full functionality of the online address book when the client is working offline. This protocol describes the structure of each of the files that a version 2, version 3 or version 4 OAB require.

### 1.3.1 OAB Version 2 and OAB Version 3

The OAB version 2 and OAB version 3 file format specifies the structure of files that a client downloads from a server to support an offline address book (OAB). OAB version 2 and OAB version 3 are very similar; OAB version 3 adds **Unicode** support and additional recipient properties.

A version 2 or version 3 OAB consists of the following files:

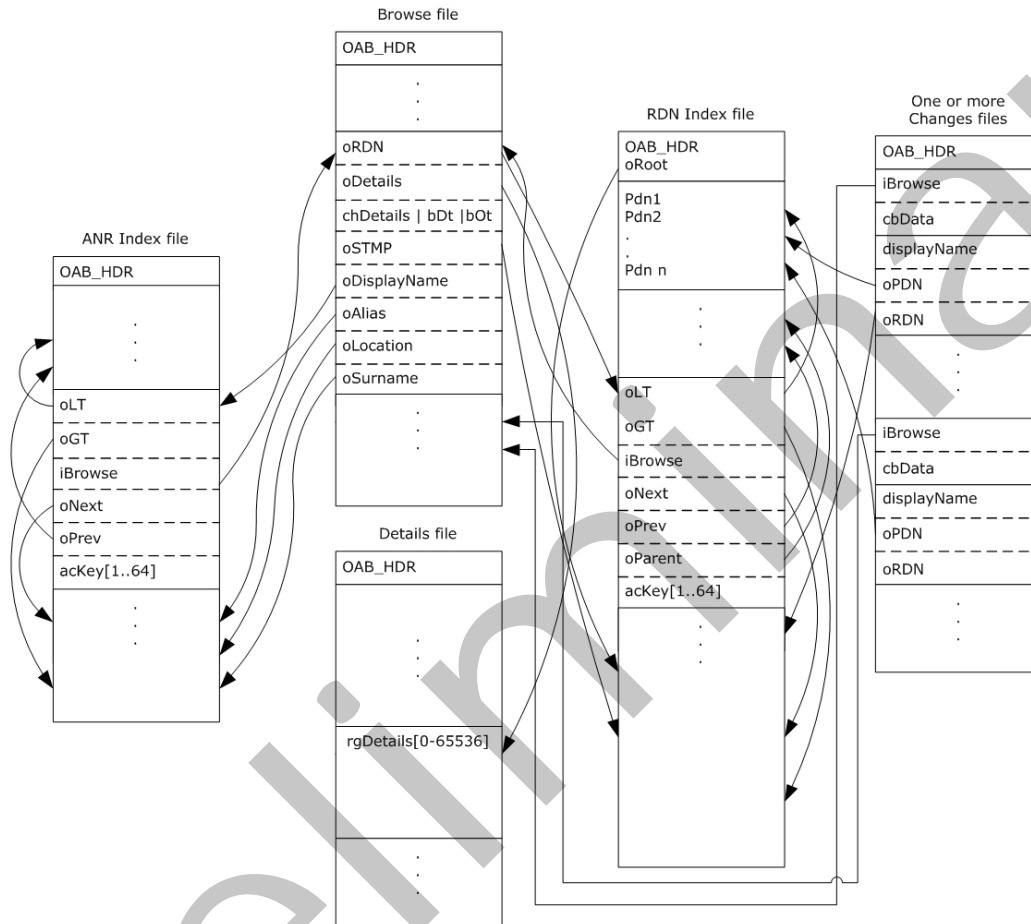
- Browse file.
- Relative Distinguished Name (RDN) Index file.
- Ambiguous Name Resolution (ANR) Index file.
- Details file.



- Display template files.
- Changes file.

The server compresses each of these files before synchronization to save network bandwidth.

The following diagram shows each of these OAB files and the indexes that point from one file to another. After the client downloads the OAB, the client can download incremental updates by using a Changes file.



**Figure 1: Relationship of the OAB version 2 and OAB version 3 ANR Index file, Browse file, Details file, RDN Index file, and Changes file**

### 1.3.2 OAB Version 4

A version 4 OAB consists of the following files:

- Full Details file.
- Differential Patch file.

- Display template file.

## 1.4 Relationship to Protocols and Other Structures

This specification assumes the reader has familiarity with the address book concepts and requirements of the Address Book Object Protocol, as described in [\[MS-OXOABK\]](#). Those concepts and requirements are not repeated in this specification.

Servers can distribute OABs to clients by using either **public folders** or a Web-based distribution method, as described in [\[MS-OXPFOAB\]](#) and [\[MS-OXWOAB\]](#) respectively.

In order to minimize communication costs, the server compresses the data in the OAB, as described in [\[MS-PATCH\]](#) and [\[MS-MCI\]](#).

The method of naming properties in the OAB is based on the **property tag** naming convention, as described in [\[MS-OXPROPS\]](#) section 1.3.3.

## 1.5 Applicability Statement

Clients use OAB structures to download information about **Address Book objects** for use when working offline or in cached mode.

## 1.6 Versioning and Localization

This document covers versioning issues in the following areas:

- **Structure Versions:** This document describes OAB version 2 and OAB Version 3 files in section [2.2](#), section [2.3](#), section [2.4](#), section [2.5](#), section [2.6](#), section [2.7](#), and section [2.8](#), and OAB version 4 files in section [2.2](#) and section [2.9](#).
- **Localization:** Localization dependent file format content is described in section [2.3.2](#) and in row 46 of the table in section [2.9.2.2](#).

## 1.7 Vendor-Extensible Fields

The OAB version 2, version 3 and version 4 structures make use of property tags, but OAB version 4 has an extensible schema. A vendor can add new properties to OAB version 4 by assigning property tags to **Active Directory** directory service properties, as described in [\[MS-ADTS\]](#).

## 2 Structures

**Note** Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.

All integer fields in the OAB structures are unsigned and use little-endian byte order.

The server calculates all CRC hash values by using the IEEE 802.3 CRC polynomial of 0xEDB88320 ( $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ ) which is seeded with the value 0xFFFFFFFF. For more details, see [\[ISO/IEC8802-3\]](#) section 3.2.8.

The server packs all structures on single byte boundaries.

The server measures all offsets in bytes from the beginning of the specified file. <1>

### 2.1 X500 Distinguished Name

**X500 DNs** uniquely identify Address Book objects in the OAB. Each Address Book object MUST have a unique X500 DN value. The server stores the X500 DN in the **PidTagEmailAddress** property, as specified in [\[MS-OXOABK\]](#) section 2.2.3.14. The following **Augmented Backus-Naur Form (ABNF)** definition, as specified in [\[RFC5234\]](#), specifies the format of an X500 DN.

```
x500-dn      =  org org-unit 0*13(container) object-rdn
                ; x500-dns are limited to 16 levels
template-x500-dn = org [org-unit] 0*13(container) object-rdn
org           =  "/o=" rdn
org-unit     =  "/ou=" rdn
container    =  "/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" / "|"

addresslist-x500-dn =  "/guid=" 32(HEXDIG) / "/" / x500-dn
```

## 2.2 Uncompressed OAB Display Template File

The display template file specifies how to display Address Book objects and e-mail addresses to the client. The display template file is a package that wraps the template for creating e-mail addresses, known as the **address creation template**, and the template for displaying Address Book object information. For more details about the address creation template and the template for displaying Address book object information, see [\[MS-OXOABKT\]](#).

The following ABNF definition specifies the format of an uncompressed display template file.

```

template-file      =  OAB_HDR mail-user-template
                    distribution-list-template
                    forum-template agent-template
                    organization-template
                    private-distributionlist-template
                    remote-mailuser-template
                    NAMES_STRUCT
                    address-templates data
mail-user-template =  TMPLT_ENTRY
                    ; display template for mailboxes
distribution-list-template = TMPLT_ENTRY
                    ; display template for distribution lists
forum-template     =  TMPLT_ENTRY
                    ; display template for public folders
agent-template     =  TMPLT_ENTRY
                    ; display template for mail agents
organization-template =  TMPLT_ENTRY
                    ; SHOULD be set to all zeros.
private-distributionlist-template = TMPLT_ENTRY
                    ; SHOULD be set to all zeros.
remote-mailuser-template = TMPLT_ENTRY
                    ; display template for external e-mail
                    ; addresses
address-templates  =  oot-count *(address-creation-template)
oot-count          =  %x00000000-%xFFFFFFFF
                    ; 32 bits of data
address-creation-template =  TMPLT_ENTRY
                    ; an address creation display template
data               =  *(OCTET)
                    ; unstructured data section

```

All the fields in this section that start with an 'o' indicate an offset from the beginning of the file into the unstructured data section.

### 2.2.1 OAB\_HDR

The **OAB\_HDR** structure specifies the OAB file format version.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
ulVersion																															
ulSerial																															

ulTotRecs
-----------

**ulVersion (4 bytes):** A 32-bit unsigned integer that specifies the file type. This field **MUST** be set to 0x00000007 for uncompressed display template files.

**ulSerial (4 bytes):** This field **MUST** be set to 0x00000000 when sent and **MUST** be ignored when received.

**ulTotRecs (4 bytes):** This field **SHOULD** be set to 0x00000000 when sent and **MUST** be ignored when received.

## 2.2.2 TMPLT\_ENTRY

The **TMPLT\_ENTRY** structure encodes properties of an individual display template.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
oDN																															
cbDN																															
oTmplt																															
cbTmplt																															
oScript																															
cbScript																															
oDispName																															
cbDispName																															

**oDN (4 bytes):** A 32-bit unsigned integer that specifies the absolute offset in the display template file to the template-X500-DN of the template. A value of 0x00000000 indicates that the data is not included in the file at the offset location and the value **MUST** be ignored.

**cbDN (4 bytes):** A 32-bit unsigned integer that specifies the length of the template-X500-DN value in bytes including the terminating null character.

**oTmplt (4 bytes):** A 32-bit unsigned integer that specifies the absolute offset in the display template file to the template structure data. For more details, see [\[MS-OXOABKT\]](#). A value of 0x00000000 indicates that the data is not included in the file at the offset location and the value **MUST** be ignored.

**cbTmplt (4 bytes):** A 32-bit unsigned integer that specifies the length of the template structure data, in bytes, which includes the template table, plus any stored strings. The stored strings are defined in the ABNF of section [2.2](#) as `data`.

**oScript (4 bytes):** A 32-bit unsigned integer that specifies the absolute offset in the display template file of the Script file for the template. For more details, see [\[MS-OXOABKT\]](#) section 2.2.2.2. A value of 0x00000000 indicates that the data is not included in the file at the offset location and the value MUST be ignored.

**cbScript (4 bytes):** A 32-bit unsigned integer that specifies the length of the Script file data in bytes.

**oDispName (4 bytes):** A 32-bit unsigned integer that specifies the absolute offset in the display template file to the display name for the template. A null-terminated string that uses the **American National Standards Institute (ANSI) character set**. A value of 0x00000000 indicates that the data is not included in the file at the offset location and the value MUST be ignored.

**cbDispName (4 bytes):** A 32-bit unsigned integer that specifies the length of the display name in bytes including the terminating null character.

### 2.2.3 NAMES\_STRUCT

The **NAMES\_STRUCT** structure maps **GUIDs** to and from property tags.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
cIDsNames																cGuids															
oIDs																															
oGuids																															
oNames																															

**cIDsNames (2 bytes):** A 16-bit unsigned integer that specifies both the count of property IDs, and the matching count of named properties.

**cGuids (2 bytes):** A 16-bit unsigned integer that specifies the count of GUIDs.

**oIDs (4 bytes):** A 32-bit unsigned integer that specifies the absolute offset in the display template file to the ID table. Each ID is a 4 byte integer that represents a property tag, as specified in [\[MS-OXCDATA\]](#) section 2.9. A value of 0x00000000 indicates that the data is not included in the file at the offset location and the value MUST be ignored.

**oGuids (4 bytes):** A 32-bit unsigned integer that specifies the absolute offset in the display template file to the **GUID** table. Each GUID is stored in binary format in 16 bytes, as specified in [\[MS-DTYP\]](#). A value of 0x00000000 indicates that the data is not included in the file at the offset location and the value MUST be ignored.

**oNames (4 bytes):** A 32-bit unsigned integer that specifies the absolute offset in the display template file to the PropertyName\_r structure table, as specified in [\[MS-OXCDATA\]](#) section 2.6.2. A value of 0x00000000 indicates that the data is not included in the file at the offset location and the value MUST be ignored.

## 2.3 Uncompressed OAB Version 2 and OAB Version 3 Browse File

**Note** Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.

The Browse file [<2>](#) contains one fixed size record per user, with members that point to offsets in the Relative Distinguished Name (RDN) Index file, the Ambiguous Name Resolution (ANR) Index, and the Details files. The fixed size record contains data and offsets that account for all of the user's data in the OAB version 2 and OAB version 3 structure.

The records in the Browse file are sorted in alphabetical order according to the Address Book object display names and they allow for fast paging of Address Book object data. The Browse file has offsets into the other files for the display name, the surname, the office location, the X500 DN, the **SMTP** address, the e-mail alias, and the details record. The Browse file also maintains values for the object type and Address Book object display type. Each record is a fixed size. Fetching an entire record requires that the client follow each link from the Browse file and retrieve data from the other files. The header of the Browse file includes a file type, a record count, and a serial number. The serial number is a rotating hash of the **RDN (1)** value of each record in the Browse file order.

OAB version 2 only includes support for encoding string data with characters in the ANSI character set code page of the Browse file. OAB version 3 added support for Unicode characters and additional properties to recipient (2) record data. If the client supports Unicode, the Unicode files of OAB version 3 SHOULD be used.

The following ABNF definition shows the format of an uncompressed OAB version 2 or OAB version 3 Browse file.

```
browse-file = OAB_HDR 1*16777213(B2_REC)
```

### 2.3.1 OAB\_HDR

The **OAB\_HDR** structure specifies the OAB file format version and the number of Address Book object records in the address list, and it contains a hash value for consistency checks.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
ulVersion																															
ulSerial																															
ulTotRecs																															

**ulVersion (4 bytes):** A 32-bit unsigned integer that specifies the file type. This field MUST be set to 0x0000000A for uncompressed version 2 OAB Browse files. This field MUST be set to 0x0000000E for uncompressed version 3 OAB Browse files.

**ulSerial (4 bytes):** A 32-bit hexadecimal string that specifies the hash of the RDN (1) values for the current set of OAB records. The value of this field is calculated as specified in section [2.3.3](#).

**ulTotRecs (4 bytes):** A 32-bit unsigned integer that specifies the number of **B2\_REC** records stored in the Browse file. This field **MUST** be 1 or larger and **MUST** be less than 16,777,213.

### 2.3.2 B2\_REC

The **B2\_REC** structure encodes an Address Book object in the Browse file. The Address Book objects are sorted in the Browse file by alphabetical display name order. The **locale** that is used by the server to sort the files **SHOULD** be stored on the public folder message that contains the files. The client **SHOULD** use the stored locale for string comparison when searching the files. For more details, see [\[MS-OXPFOAB\]](#) section 2.2.1.5.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
oRDN																															
oDetails																															
cbDetails																bDispType								a	bObjType						
oSMTP																															
oDispName																															
oAlias																															
oLocation																															
oSurname																															

**oRDN (4 bytes):** A 32-bit unsigned integer that specifies the offset of the RDN (1) record in the RDN Index file.

**oDetails (4 bytes):** A 32-bit unsigned integer that specifies the offset of the details record in the Details file.

**cbDetails (2 bytes):** An 16-bit unsigned integer that specifies the size of the details record in the Details file.

**bDispType (1 byte):** An unsigned 8-bit value that specifies The display type of the Address Book object. This field **MUST** be set to one of the values in the following table.

Value	Meaning
0x00	DT_MAILUSER
0x01	DT_DISTLIST
0x02	DT_FORUM
0x03	DT_AGENT



Value	Meaning
0x04	DT_ORGANIZATION
0x06	DT_REMOTE_MAILUSER

**a (1 bit):** A single bit value. This field SHOULD be set to 1 if the Address Book object can receive all message content, including **Rich Text Format (RTF)** and Object Linking and Embedding (OLE) objects. This field SHOULD be set to 0 if the Address Book object cannot receive all message content. For more details, see [\[MS-OXOABK\]](#) section 2.2.3.18.

**bObjType (7 bits):** A 7-bit value that specifies the object type of the Address Book object. This field MUST be set to one of the values in the following table.

Value	Meaning
0x03	MAPI-FOLDER
0x06	MAPI-MAILUSER
0x08	MAPI-DISTLIST

**oSMTP (4 bytes):** A 32-bit unsigned integer that specifies the offset of the SMTP address record in the RDN Index file.

**oDispName (4 bytes):** A 32-bit unsigned integer that specifies the offset of the display name record in the ANR Index file.

**oAlias (4 bytes):** A 32-bit unsigned integer that specifies the offset of the alias record in the ANR Index file.

**oLocation (4 bytes):** A 32-bit unsigned integer that specifies the offset of the office location record in the ANR Index file.

**oSurname (4 bytes):** A 32-bit unsigned integer that specifies the offset of the surname record in the ANR Index file.

### 2.3.3 RDN Hash Computation

The server creates the RDN (1) hash value, which it stores in the **ulSerial** field of the **OAB\_HDR** structure of the Browse file, as specified in section [2.3.1](#), by incorporating the RDN (1) value of each OAB record into a single value. The following steps produce the RDN (1) hash value:

1. Begin with a 4-byte integer value of 0x00000000; this is the current hash value when processing the first OAB record.
2. Process the OAB records in Browse-file order. See section [2.3](#) for an overview of the Browse file and the definition of Browse-file order. For each record in the OAB, perform steps 3 through 5.
3. Get the RDN (1) value of the OAB record. The RDN (1) value is the value that the server stores in the **acKey** field of the **RDN2\_REC** structure of the RDN Index file. For more details, see section [2.4.2](#). Pad the RDN (1) value with null characters (append null characters to the end of the RDN (1) string) to align it to a 4-byte boundary. The resulting value is a series of 4-byte values, each of which is treated as a little-endian integer.

4. XOR all of the 4-byte integers together with the current hash value, which is 0x00000000 for the first record. For each subsequent record, the current hash value is the result of the previous iteration.
5. Take the resulting 4-byte value and shift it left by 1 bit, rotating the high-order bit to the low-order bit. The resulting value is now the current hash value. Repeat steps 3 through 5 to process all records, and then move on to step 6.
6. After processing all OAB records, the current hash value of the last iteration is the RDN (1) hash value.

## 2.4 Uncompressed OAB Version 2 and OAB Version 3 RDN Index File

**Note** Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.

The client uses the RDN Index file <3> for primary key lookups based on the X500 DN and Simple Mail Transfer Protocol (SMTP) address properties of the Address Book object.

The RDN Index file is split into two sections: the **parent distinguished name (PDN)** table and the RDN (1) index. The PDN table contains the list of all PDN values for X500 DNs and all **domain** names used by SMTP addresses. The server stores the last RDN (1) of the X500 DNs and the local-part of SMTP addresses in the key field of the records in the RDN (1) index section.

For example, given the following **distinguished name (DN) (4)** value, /o=Adventure-Works/ou=New York/CN=recipients/CN=JohnH, the RDN (1) object is /CN=JohnH and the PDN is /o=Adventure-Works/ou=New York/CN=recipients/. The key field of the RDN (1) index, also known as the RDN (1) value, is simply JohnH.

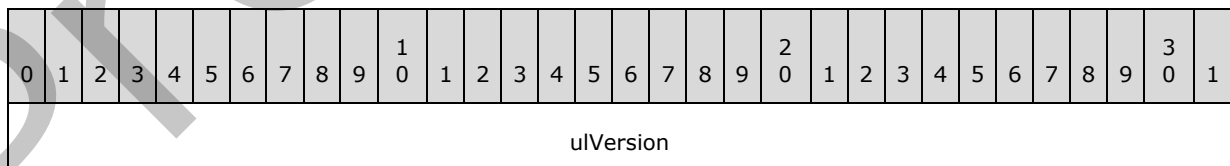
Records in the RDN (1) index part of the file are of variable size, contain the index key value, and have pointers to the record in the PDN table so that the original value of the X500 DN or SMTP address can be reconstructed. In the record is an index of the related browse record in the Browse file and four more offsets are stored to create a threaded tree structure within the RDN Index file. An offset in the header of the RDN Index file points past the end of the PDN table to the root of the RDN (1) index tree.

The following ABNF definition illustrates an uncompressed OAB version 2 or OAB version 3 RDN Index file.

```
rdn-file      = RDN_HDR 1*pdn-record 1*RDN2_REC
pdn-record   = 1*(CHAR) %x00
```

### 2.4.1 RDN\_HDR

The **RDN\_HDR** structure specifies the OAB file format version and the number of RDN (1) records in the RDN Index file, and it contains a hash value for consistency checks.



ulSerial
ulTotRecs
oRoot

**ulVersion (4 bytes):** A 32-bit unsigned integer that specifies the file type. This field SHOULD be set to 0x0000000A for uncompressed version 2 RDN Index files. This field SHOULD be set to 0x0000000E for uncompressed version 3 RDN Index files. <4>

**ulSerial (4 bytes):** This field SHOULD be set to zero when sent and MUST be ignored when received.

**ulTotRecs (4 bytes):** A 32-bit unsigned integer that specifies the number of **RDN2\_REC** records stored in the RDN Index file.

**oRoot (4 bytes):** A 32-bit unsigned integer that specifies the offset of the root **RDN2\_REC** node of the RDN (1) index tree. This record MUST be after the last **pdn-record** in the file. When parsing pdn-records, use this value to stop parsing pdn-records and start parsing RDN (1) records.

### 2.4.2 RDN2\_REC

Each **RDN2\_REC** structure corresponds to a node in the RDN (1) index tree. The server constructs the tree as a threaded tree so that searches and moving to the next and previous records are efficient.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
oLT																															
oGT																															
iBrowse																															
oPrev																															
oNext																															
oParentDN																															
acKey (variable)																															
...																															

**oLT (4 bytes):** A 32-bit unsigned integer that specifies the offset of the left **RDN2\_REC** child of the current node in the RDN Index file. The left child MUST sort to the same value as the current node or less. This field MUST be set to 0x00000000 to indicate that there is no left child node.

**oGT (4 bytes):** A 32-bit unsigned integer that specifies the offset of the right **RDN2\_REC** child of the current node in the RDN Index file. The right child MUST sort to the same value as the current node or greater. This field MUST be set to 0x00000000 to indicate that there is no right child node.

**iBrowse (4 bytes):** A 32-bit unsigned integer that specifies the index to the **B2\_REC** in the browse file that references this record. The values 0x00000000 through 0x00000002 are reserved and MUST NOT be used. The index value in the Browse file is computed by using the following equation:  $iBrowse - 0x00000003$ .

**oPrev (4 bytes):** A 32-bit unsigned integer that specifies the offset of the previous **RDN2\_REC** record in the RDN Index file when sorted as a flat list. This field MUST be set to 0x00000000 to indicate that this is the first node in the list.

**oNext (4 bytes):** A 32-bit unsigned integer that specifies the offset of the next **RDN2\_REC** record in the RDN Index file when sorted as a flat list. This field MUST be set to 0x00000000 to indicate that this is the last node in the list.

**oParentDN (4 bytes):** A 32-bit unsigned integer that specifies the offset of the null-terminated ANSI character set **pdn-record** string in the RDN Index file. This field MUST NOT be set to 0x00000000.

**acKey (variable):** The null-terminated ANSI character set string value of the record, as specified by RDN (1) in section 2.1, or the local portion of the SMTP address. This field MUST be 64 characters or fewer, plus the terminating null character.

For RDN (1) records, "/CN=" MUST be removed from the final RDN (1) before storing in the RDN Index file. The **oParentDN** points at the parent X500 DN; therefore, the actual value is computed by prepending the **acKey** value with "/CN=" then appending that result onto the end of the **parent** DN (4) value.

For SMTP records, the SMTP address is split after '@' and the local-part of the SMTP address including the '@' is stored in the **acKey** field. The domain name part of the SMTP address is pointed to by the **oParentDN** offset.

## 2.5 Uncompressed OAB Version 2 and OAB Version 3 ANR Index File

**Note** Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.

The **ANR** Index file <5> enables ambiguous name resolution (ANR). The server sorts values for the display name, surname, office location, and e-mail **alias** together into one structure so that a single search can find Address Book objects based on multiple properties.

The ANR Index file is structured similarly to the RDN Index file, but does not contain a PDN table. Each record is a variable size and has four offsets that construct a threaded tree structure. Records have an index of master records in the Browse file and the value portion is either an office location string, a surname string, an alias string, or a display name string. The root of the ANR index tree is always the first node in the file; therefore no root offset is required in the header.

The following ABNF definition shows the format of an uncompressed OAB version 2 or OAB version 3 ANR Index file.

```
anr-file = OAB_HDR 1*ANR_REC
```

## 2.5.1 OAB\_HDR

The **OAB\_HDR** structure specifies the OAB file format version and the number of ANR records in the ANR Index file, and it contains a hash value for consistency checks.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ulVersion																															
ulSerial																															
ulTotRecs																															

**ulVersion (4 bytes):** A 32-bit unsigned integer that specifies the file type. This field **MUST** be set to 0x0000000A for uncompressed OAB version 2 ANR Index files. This field **MUST** be set to 0x0000000E for uncompressed OAB version 3 ANR Index files.

**ulSerial (4 bytes):** This field **SHOULD** be set to zero when sent and **MUST** be ignored when received.

**ulTotRecs (4 bytes):** A 32-bit unsigned integer that specifies the number of **ANR\_REC** records stored in the ANR Index file.

## 2.5.2 ANR\_REC

Each **ANR\_REC** structure corresponds to a node in the ANR index tree. The server constructs the tree as a threaded tree so that searches are efficient, and traversing to the next and previous records is also efficient. The root of the tree **MUST** be the first **ANR\_REC** in the ANR Index file.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
oLT																															
oGT																															
iBrowse																								a	b						
oPrev																															
oNext																															
acKey (variable)																															
...																															

**oLT (4 bytes):** A 32-bit unsigned integer that specifies the offset of the left **ANR\_REC** child of the current node in the ANR Index file. The left child **MUST** sort to the same value as the

current node or less. This field MUST be set to 0x00000000 to indicate that there is no left child node.

**oGT (4 bytes):** A 32-bit unsigned integer that specifies the offset of the right **ANR\_REC** child of the current node in the ANR Index file. The right child MUST sort to the same value as the current node or greater. This field MUST be set to 0x00000000 to indicate that there is no right child node.

**iBrowse (3 bytes):** A 32-bit unsigned integer that specifies the index to the B2\_REC in the Browse file that references this record. The values 0x000000 through 0x000002 are reserved and MUST NOT be used. The index value in the browse file is computed by using the following equation:  $iBrowse - 0x000003$ .

**a (1 bit):** A single bit value. This field MUST be set to 1 for e-mail alias records. This field MUST be set to 0 for display name, office location, and surname records.

**b (7 bits):** A 7-bit value. This field MUST be all zeros.

**oPrev (4 bytes):** A 32-bit unsigned integer that specifies the offset of the previous **ANR\_REC** record in the ANR Index file when sorted as a flat list. This field MUST be set to 0x00000000 when this is the first node in the list.

**oNext (4 bytes):** A 32-bit unsigned integer that specifies the offset of the next **ANR\_REC** record in the ANR Index file when sorted as a flat list. This field MUST be set to 0x00000000 when this is the last node in the list.

**acKey (variable):** The null-terminated, ANSI character set string value, of the record for the OAB Version 2 ANR Index files or the null-terminated UTF-8 string value of the record for OAB Version 3 ANR Index files. This field MUST be 64 characters or fewer including the terminating null character.

## 2.6 Uncompressed OAB Version 2 and OAB Version 3 Details File

**Note** Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.

The Details file [<6>](#) contains all other properties for Address Book objects in OAB version 2 and OAB version 3. The server does not index the Details file. The client can choose not to download the Details file in order to save space and bandwidth since there is no information in there that the client requires for basic e-mail addressing.

The Details file contains variable size records that store a fixed set of properties for each Address Book object. Each record can be up to 65536 bytes long and all properties that the server stores for a single Address Book object have to fit into that record. The server does not index the data and there are no links from this file to any of the other files, but the Browse file does have links to this file.

The following ABNF definition shows the format of an uncompressed OAB version 2 and OAB version 3 Details file.

```
v2-details-file      = OAB_HDR 1*details-record
details-record       = user-certificate business-telephone
                       given-name initials street-address
                       city-locality state-province postal-code
```

```

country-region title company-name
assistant-name
department-name null home-telephone
business2-telephone home2-telephone
primary-fax mobile-telephone
assistant-telephone pager-telephone
comment proxy-addresses smime-certs
x509-certs
v3-details-file = OAB_HDR 1*v3-details-record
v3-details-record = user-certificate business-telephone
given-name initials street-address
city-locality state-province postal-code
country-region title company-name
assistant-name
department-name target-address
home-telephone
business2-telephone-mv home2-telephone-mv
primary-fax mobile-telephone
assistant-telephone pager-telephone
comment proxy-addresses smime-certs
x509-certs home-mdb manager
display-name-printable

user-certificate = binary-value
business-telephone = string-value
given-name = string-value
initials = string-value
street-address = string-value
city-locality = string-value
state-province = string-value
postal-code = string-value
country-region = string-value
title = string-value
company-name = string-value
assistant-name = string-value
department-name = string-value
home-telephone = string-value
business2-telephone = string-value
home2-telephone = string-value
business2-telephone-mv = multivalued-string
home2-telephone-mv = multivalued-string
primary-fax = string-value
mobile-telephone = string-value
assistant-telephone = string-value
pager-telephone = string-value
comment = string-value
proxy-addresses = multivalued-string
smime-certs = multivalued-binary
x509-certs = multivalued-binary
target-address = string-value
home-mdb = x500-dn
manager = x500-dn
display-name-printable = teletex-string
string-value = *(ansi-char) null / null
ansi-char = %x01-%xFF
; 8 bits of data
teletex-string = *(teletex-char) null / null
null = %x00
; 8 bits of data

```

```

multivalued-string = count 0*255(string-value) / null
count              = %x00-%xFF
                  ; 8 bits of data
binary-value      = byte-count 0*65535(OCTET) / null
byte-count        = %x0000-%xFFFF
                  ; 16 bits of data
multivalued-binary = count 0*255(binary-value) / null

```

Each Details record MUST fit into 65535 bytes. If a value is not present, a null byte MUST be encoded. All strings MUST be null-terminated strings. Multivalued-binary or multivalued-string encodings with one or more values MUST NOT have any zero length elements.

The details elements for OAB version 2 details files map directly to the following property tag table. For details about the following properties, see [\[MS-OXOABK\]](#).

Property tag name	Property tag	Property type	Description
<b>PidTagUserCertificate</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.64)	0x3A220102	<b>PtypBinary</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	The <b>user-certificate</b> property contains an ASN.1 authentication certificate for a messaging user. For more details, see <a href="#">[ISO/IEC8825-1]</a> . This property is deprecated and SHOULD NOT be set.
<b>PidTagBusinessTelephoneNumber</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.38)	0x3A08001F	<b>PtypString*</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	The <b>business-telephone</b> property contains the primary telephone number of the place of business of the Address Book object.
<b>PidTagGivenName</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.2)	0x3A06001F	<b>PtypString*</b>	The <b>given-name</b> property contains the given name of the Address Book object.
<b>PidTagInitials</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.5)	0x3A0A001F	<b>PtypString*</b>	The <b>initials</b> property contains the initials for parts of the full name of the Address Book object.
<b>PidTagStreetAddress</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.19)	0x3A29001F	<b>PtypString*</b>	The <b>street-address</b> property contains the street address of the Address Book object.
<b>PidTagLocality</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.21)	0x3A27001F	<b>PtypString*</b>	The <b>city-locality</b> property contains the name of the locality of the Address Book object, such as the town or city.
<b>PidTagStateOrProvince</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.22)	0x3A28001F	<b>PtypString*</b>	The <b>state-province</b> property contains the name of the state or province where the Address Book object is located.
<b>PidTagPostalCode</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.23)	0x3A2A001F	<b>PtypString*</b>	The <b>postal-code</b> property contains the postal code of the Address Book object.
<b>PidTagCountry</b> ( <a href="#">[MS-OXOABK]</a> )	0x3A26001	<b>PtypString*</b>	The <b>country-region</b> property



Property tag name	Property tag	Property type	Description
section 2.2.4.24)	F		contains the name of the country or region where the Address Book object is located.
<b>PidTagTitle</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.7)	0x3A17001 F	<b>PtypString*</b>	The <b>title</b> property contains the job title of the Address Book object.
<b>PidTagCompanyName</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.10)	0x3A16001 F	<b>PtypString*</b>	The <b>company-name</b> property contains the name of the company that employs the Address Book object.
<b>PidTagAssistant</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.11)	0x3A30001 F	<b>PtypString*</b>	The <b>assistant-name</b> property contains the name of the administrative assistant for the Address Book object.
<b>PidTagDepartmentName</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.9)	0x3A18001 F	<b>PtypString*</b>	The <b>department-name</b> property contains the department name in which the Address Book object works.
null	0x3A08001 F	<b>PtypString*</b>	Duplicate <b>PidTagBusinessTelephoneNumber</b> property. The client MUST ignore this value.
<b>PidTagHomeTelephoneNumber</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.39)	0x3A09001 F	<b>PtypString*</b>	The <b>home-telephone</b> property contains the primary home telephone number for the Address Book object.
<b>PidTagBusiness2TelephoneNumber</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.40)	0x3A1B001 F	<b>PtypString*</b>	The <b>business2-telephone</b> property contains a secondary business telephone number for the Address Book object. MUST be set to null in an OAB Version 2 Details file.
<b>PidTagHome2TelephoneNumber</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.42)	0x3A2F001 F	<b>PtypString*</b>	The <b>home2-telephone</b> property contains a secondary home telephone number for the Address Book object. MUST be set to null in an OAB Version 2 Details file.
<b>PidTagPrimaryFaxNumber</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.51)	0x3A23001 F	<b>PtypString*</b>	The <b>primary-fax</b> property contains the telephone number for the fax machine of the Address Book object.
<b>PidTagMobileTelephoneNumber</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.45)	0x3A1C001 F	<b>PtypString*</b>	The <b>mobile-telephone</b> property contains the mobile telephone number of the Address Book object.

Property tag name	Property tag	Property type	Description
<b>PidTagAssistantTelephoneNumber</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.57)	0x3A2E001F	<b>PtypString*</b>	The <b>assistant-telephone</b> property contains the telephone number for the administrative assistant of the Address Book object.
<b>PidTagPagerTelephoneNumber</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.49)	0x3A21001F	<b>PtypString*</b>	The <b>pager-telephone</b> property contains the pager telephone number of the Address Book object.
<b>PidTagComment</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.31)	0x3004001F	<b>PtypString*</b>	The <b>comment</b> property contains a description of the purpose or content of an object.
<b>PidTagAddressBookProxyAddresses</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.23)	0x800F101F	<b>PtypMultipleString*</b> ( <a href="#">[MS-OXCADATA]</a> section 2.11.1)	The <b>proxy-addresses</b> property contains a list of e-mail addresses that this Address Book object is known by. Each value MUST begin with an e-mail <b>address type</b> followed by a colon character then followed by the address value.
<b>PidTagUserX509Certificate</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.66)	0x3A701102	<b>PtypMultipleBinary</b> ( <a href="#">[MS-OXCADATA]</a> section 2.11.1)	The <b>smime-certs</b> property contains SMIME certificates that the server formats as PKCS-7 encodings. For more details, see <a href="#">[RFC2315]</a> .
<b>PidTagAddressBookX509Certificate</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.65)	0x8C6A1102	<b>PtypMultipleBinary</b>	The X509-certs property contains ASN.1 <a href="#">[ISO/IEC8825-1]</a> encoded <b>X.509</b> certificates. For more details, see <a href="#">[RFC3280]</a> .

\*The server encodes this property as an ANSI character set string, but the can interpret it as an ANSI character set or Unicode string.

The details elements for OAB version 3 details files map directly to the following property tag table.

Property tag name	Property tag	Property type	Description
<b>PidTagUserCertificate</b>	0x3A220102	<b>PtypBinary</b>	The <b>user-certificate</b> property contains an ASN.1 authentication certificate for a messaging user. For more details, see <a href="#">[ISO/IEC8825-1]</a> . This property is

Property tag name	Property tag	Property type	Description
			deprecated and SHOULD be set to a null entry.
<b>PidTagBusinessTelephoneNumber</b>	0x3A08001F	<b>PtypString</b>	The <b>business-telephone</b> property contains the primary telephone number of the place of business of the Address Book object.
<b>PidTagGivenName</b>	0x3A06001F	<b>PtypString</b>	The <b>given-name</b> property contains the given name of the Address Book object.
<b>PidTagInitials</b>	0x3A0A001F	<b>PtypString</b>	The <b>initials</b> property contains the initials for parts of the full name of the Address Book object.
<b>PidTagStreetAddress</b>	0x3A29001F	<b>PtypString</b>	The <b>street-address</b> property contains the street address of the Address Book object.
<b>PidTagLocality</b>	0x3A27001F	<b>PtypString</b>	The <b>city-locality</b> property contains the name of the locality of the Address Book object, such as the town or city.
<b>PidTagStateOrProvince</b>	0x3A28001F	<b>PtypString</b>	The <b>state-province</b> property contains the

Property tag name	Property tag	Property type	Description
			name of the state or province where the Address Book object is located.
<b>PidTagPostalCode</b>	0x3A2A001F	<b>PtypString</b>	The <b>postal-code</b> property contains the postal code of the Address Book object.
<b>PidTagCountry</b>	0x3A26001F	<b>PtypString</b>	The <b>country-region</b> property contains the name of the country or region where the Address Book object is located.
<b>PidTagTitle</b>	0x3A17001F	<b>PtypString</b>	The <b>title</b> property contains the job title of the Address Book object.
<b>PidTagCompanyName</b>	0x3A16001F	<b>PtypString</b>	The <b>company-name</b> property contains the name of the company that employs the Address Book object.
<b>PidTagAssistant</b>	0x3A30001F	<b>PtypString</b>	The <b>assistant-name</b> property contains the name of the administrative assistant for the Address Book object.
<b>PidTagDepartmentName</b>	0x3A18001F	<b>PtypString</b>	The <b>department-name</b>

Property tag name	Property tag	Property type	Description
			property contains the department name in which the Address Book object works.
<b>PidTagAddressBookTargetAddress</b> ( <a href="#">[MS-OXOAB]</a> section 2.2.3.22)	0x8011001F	<b>PtypString</b>	The <b>target-address</b> property contains the destination address for this object.
<b>PidTagHomeTelephoneNumber</b>	0x3A09001F	<b>PtypString</b>	The <b>home-telephone</b> property contains the primary home telephone number for the Address Book object.
<b>PidTagBusiness2TelephoneNumbers</b>	0x3A1B101F	<b>PtypMultipleString</b>	The <b>business2-telephone</b> property contains secondary business telephone numbers for the Address Book object.
<b>PidTagHome2TelephoneNumbers</b>	0x3A2F101F	<b>PtypMultipleString</b>	The <b>home2-telephone</b> property contains secondary home telephone numbers for the Address Book object.
<b>PidTagPrimaryFaxNumber</b>	0x3A23001F	<b>PtypString</b>	The <b>primary-fax</b> property contains the telephone number for the fax machine of the Address Book object.

Property tag name	Property tag	Property type	Description
<b>PidTagMobileTelephoneNumber</b>	0x3A1C001F	<b>PtypString</b>	The <b>mobile-telephone</b> property contains the mobile telephone number of the Address Book object.
<b>PidTagAssistantTelephoneNumber</b>	0x3A2E001F	<b>PtypString</b>	The <b>assistant-telephone</b> property contains the telephone number for the administrative assistant of the Address Book object.
<b>PidTagPagerTelephoneNumber</b>	0x3A21001F	<b>PtypString</b>	The <b>pager-telephone</b> property contains the pager telephone number of the Address Book object.
<b>PidTagComment</b>	0x3004001F	<b>PtypString</b>	The <b>comment</b> property contains a description of the purpose or content of an object.
<b>PidTagAddressBookProxyAddresses</b>	0x800F101F	<b>PtypMultipleString</b>	The <b>proxy-addresses</b> property contains a list of e-mail addresses that this Address Book object is known by. Each value <b>MUST</b> begin with an e-mail address type followed by a colon character then followed by

Property tag name	Property tag	Property type	Description
			the address value.
<b>PidTagUserX509Certificate</b>	0x3A701102	<b>PtypMultipleBinary</b>	The <b>smime-certs</b> property contains SMIME certificates formatted as PKCS-7 encodings. For more details, see <a href="#">[RFC2315]</a> .
<b>PidTagAddressBookX509Certificate</b>	0x8C6A1102	<b>PtypMultipleBinary</b>	The <b>X509-certs</b> property contains ASN.1 <a href="#">[ISO/IEC8825-1]</a> encoded X.509 certificates. For more details, see <a href="#">[RFC3280]</a> .
<b>PidTagAddressBookHomeMessageDatabase</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.67)	0x8006001F	<b>PtypString</b>	The <b>home-mdb</b> property contains the DN (4) of the <b>store</b> for this <b>mailbox</b> . This property value is not subject to truncation.
<b>PidTagAddressBookManager</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.5.1)	0x8005000D	<b>PtypComObject</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	The <b>manager</b> property contains the DN (4) of the manager of the recipient (2). The user object for the manager contains a <b>directReports</b> property that contains references to all user objects that have their manager property set to this DN (4).

Property tag name	Property tag	Property type	Description
<b>PidTagAddressBookDisplayNamePrintable</b> ( <a href="#">[MS-OXPROPS]</a> section 2.592)	0x39FF001F	<b>PtypString</b>	The <b>display-name-printable</b> property contains the printable string version of the display name. MUST contain Teletex characters only, as specified by the <code>teletex-char</code> rule in section <a href="#">2.1</a> .

### 2.6.1 OAB\_HDR

The **OAB\_HDR** structure specifies the OAB file format version and it contains a hash value for consistency checks.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ulVersion																															
ulSerial																															
ulTotRecs																															

**ulVersion (4 bytes):** A 32-bit unsigned integer that specifies the file type. This field MUST be set to 0x00000007 for uncompressed version 2 and version 3 Details files.

**ulSerial (4 bytes):** This field SHOULD be set to zero when sent and MUST be ignored when received.

**ulTotRecs (4 bytes):** This field SHOULD be set to zero when sent and MUST be ignored when received.

## 2.7 Uncompressed OAB Version 2 and OAB Version 3 Changes File

**Note** Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.

The Changes file [<7>](#) specifies the changes that need to be made to the other OAB files to produce a file set that represents the next generational version of the OAB version 2 and OAB version 3 files. The Changes file consists of a sequence of variable size records that contain data to update



individual records. Numerous change files can be required to make a set of OAB version 2 and OAB version 3 files current with the server.

The following ABNF definition shows the format of an uncompressed OAB version 2 or OAB version 3 Changes file.

```
changes-file      = OAB_HDR 1*change-record
change-record    = CHG_REC [display-name parent-dn-offset rdn]
                  [domain-name-offset local-portion]
                  [alias] [location] [surname]
                  [details]
                  [display-type] [object-type]
display-name     = string-value
parent-dn-offset = %x00000000-%xFFFFFFFF
                  ; little endian 32 bit value
                  ; offset of the pdn-record in the
                  ; rdn index file
domain-name-offset = %x00000000-%xFFFFFFFF
                  ; little endian 32 bit value
                  ; offset of the domain name record in the
                  ; rdn index file
local-portion   = 1*62(ansi-char) '@' null
alias           =      = 1*63(ansi-char) null
location       = 0*63(ansi-char) null
surname        = 0*63(ansi-char) null
details        = byte-count 0*65535(OCTET)
display-type    = DT-MAILUSER / DT-DISTLIST /
                  DT-FORUM / DT-AGENT / DT-ORGANIZATION /
                  DT-REMOTE-MAILUSER
                  ; 8 bit value
DT-MAILUSER    = %x00
                  ; mailbox display type
DT-DISTLIST    = %x01
                  ; distribution list display type
DT-FORUM       = %x02
                  ; public folder display type
DT-AGENT       = %x03
                  ; mail agent display type
DT-ORGANIZATION = %x04
                  ; department or organization display type
DT-REMOTE-MAILUSER = %x06
                  ; external e-mail address display type
object-type     = MAPI-FOLDER / MAPI-MAILUSER /
                  MAPI-DISTLIST
                  ; 8 bit value - high order bit is set to
                  ; 1 if the entry can receive all
                  ; message content, including Rich Text
                  ; Format (RTF) and OLE objects
                  ; For details, see section 2.786
                  ; in [MS-OXPROPS]
MAPI-FOLDER    = %x03
MAPI-MAILUSER  = %x06
MAPI-DISTLIST  = %x08
```

## 2.7.1 OAB\_HDR

The **OAB\_HDR** structure specifies the OAB file format version and the number of change records in the address list, and it contains a hash value for consistency checks.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
ulVersion																															
ulSerial																															
ulTotRecs																															

**ulVersion (4 bytes):** A 32-bit unsigned integer that specifies the file type. This field **MUST** be set to 0x0000000B for uncompressed version 2 Changes files. This field **MUST** be set to 0x0000000F for uncompressed version 3 Changes files.

**ulSerial (4 bytes):** A 32-bit hexadecimal string that contains the **ulSerial** value of the version 2 or version 3 OAB Browse file that these changes are to be applied against.

**ulTotRecs (4 bytes):** A 32-bit unsigned integer that specifies the count of the **change-record** structures in the Changes file.

## 2.7.2 CHG\_REC

The **CHG\_REC** structure specifies to the client which record to update and what attributes are included in the change record.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1																				
iBrowse																																																			
l				type				k				a				j				b				c				d				e				f				g				h				i			
cbData																																																			

**iBrowse (4 bytes):** A 32-bit unsigned integer that specifies the index of the record to be changed. The values 0x00000000 through 0x00000002 are reserved and **MUST NOT** be used. The index value in the Browse file is computed by using the following equation:  $iBrowse - 0x00000003$ .

If the change type is an addition, then the **iBrowse** **MUST** point at the record in the old file that the new record is inserted before. For example, if the record is to be inserted at the beginning of the file, then the **iBrowse** value will be 0x00000003. If the record is to be appended at the end of the file, then the **iBrowse** will be one plus the maximum **iBrowse** index in the old file. If the change type is a modification, then the **iBrowse** **MUST** point at the record in the old file to be modified. If the change type is a deletion, then the **iBrowse** **MUST** point at the record in the old file to be removed.

**l (5 bits):** This field MUST be set to zero when sent and MUST be ignored when received.

**type (3 bits):** A 3-bit value that specifies the type of change. This field MUST be set to 000, 001, or 010. A value of 000 indicates a record modification, a value of 001 indicates a record addition, and a value of 010 indicates a record deletion.

A value of 000 means that fields **a** through **i** are set according to the presence of the data fields in the change record, and that display-name, parent-DN-offset, and RDN (1) MUST NOT be present in the change record.

A value of 001 means that fields **a** through **k** MUST be set to 0, even if the values are present in the change-record structure, and that display-name, parent-DN-offset, and RDN (1) MUST be present in the change record. For addition records, even though values **a** through **k** are set to 0, they MUST be processed as if they are set to 1. If the corresponding value is not in the change-record, then a single space value is encoded when parsing the change-record.

A value of 010 means that fields **a** through **j** MUST be 0.

**k (1 byte):** This field MUST be set to zero when sent and MUST be ignored when received.

**a (1 bit):** A single bit value. This field is set to 1 to specify that the **object-type** field MUST be present in the change-record. This field is set to 0 to specify that the **object-type** field MUST NOT be present (if the type field is set to 010) or MUST be ignored (if the type field is set to 001).

**j (7 bits):** This field MUST be set to zero when sent and MUST be ignored when received.

**b (1 bit):** A single bit value. This field is set to 1 to specify that the **local-portion** field MUST be present in the change-record. This field is set to 0 to specify that the **local-portion** field MUST NOT be present (if the type field is set to 010) or MUST be ignored (if the type field is set to 001). The value of this field MUST be the same as field **c**.

**c (1 bit):** A single bit value. This field is set to 1 to specify that the **domain-name-offset** field MUST be present in the change-record. This field is set to 0 to specify that the **domain-name-offset** field MUST NOT be present (if the type field is set to 010) or MUST be ignored (if the type field is set to 001).

**d (1 bit):** A single bit value. This field is set to 1 to specify that the **alias** field MUST be present in the change-record. This field is set to 0 to specify that the **alias** field MUST NOT be present (if the type field is set to 010) or MUST be ignored (if the type field is set to 001).

**e (1 bit):** A single bit value. This field is set to 1 to specify that the **location** field MUST be present in the change-record. This field is set to 0 to specify that the **location** field MUST NOT be present (if the type field is set to 010) or MUST be ignored (if the type field is set to 001).

**f (1 bit):** A single bit value. This field is set to 1 to specify that the **surname** field MUST be present in the change-record. This field is set to 0 to specify that the **surname** field MUST NOT be present (if the type field is set to 010) or MUST be ignored (if the type field is set to 001).

**g (1 bit):** A single bit value. This field is set to 1 to specify that the **details** field MUST be present in the change-record. This field is set to 0 to specify that the **details** field MUST NOT be present (if the type field is set to 010) or MUST be ignored (if the type field is set to 001).

**h (1 bit):** A single bit value. This field is set to 1 to specify that the **details** field MUST be present in the change-record and that it is larger than the old details record in the old Details

file. This field is set to 0 to specify that the size of the **details** field is equal to or smaller than the old record in the Details file. If field **g** is set to 0 then field **h** MUST be also be set to 0.

**i (1 bit):** A single bit value. This field is set to 1 to specify that the **display-type** field MUST be present in the change-record. This field is set to 0 to specify that the **display-type** field MUST NOT be present.

**cbData (4 bytes):** A 32-bit unsigned integer that specifies the length of the **change-record** structure in bytes. This count does not include the **CHG\_REC** field.

### 2.7.3 Change-record

The following tables specify the default properties that the server populates in the OAB version 2 or OAB version 3 change-record.

Properties populated in the change-record for OAB version 2.

Index Number	Property tag name	Property type	Property size	Description
1	<b>PidTagDisplayName</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.1)	<b>PtypString*</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	Variable	The <b>display-name</b> property contains the display name for a given Address Book object.
2	ParentDnOffset	<b>PtypInteger32</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	4 bytes	The <b>parent-dn-offset</b> property contains the offset to the PDN in the RDN Index file. This property is present only if the <b>type</b> property is set to 001.
3	RDNRecordKey	<b>PtypString*</b>	Variable	The <b>rdn</b> property uniquely identifies the RDN (1) in the RDN Index file. This property is present only if the <b>type</b> property is set to 001. This is a null-terminated string. The maximum size of this property is 68 bytes.
4	ParentDnOffset ForSMTP	<b>PtypInteger32</b>	4 bytes	The <b>domain-name-offset</b> property contains the offset of the Parent DN (4) SMTP address entry in the RDN Index file. This property is present only if the <b>type</b> property is set to 000.
5	<b>PidTagSmtAddress</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.21)	<b>PtypString8</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	Variable	The <b>local-portion</b> property contains the SMTP mailing address of the sender.
6	<b>PidTagAccount</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.20)	<b>PtypString8</b>	Variable	The <b>alias</b> property contains the account name for the Address Book object.

Index Number	Property tag name	Property type	Property size	Description
7	<b>PidTagOfficeLocation</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.8)	<b>PtypString*</b>	Variable	The <b>location</b> property contains the office location of the Address Book object.
8	<b>PidTagSurname</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.1)	<b>PtypString*</b>	Variable	The <b>surname</b> property contains the family name of the Address Book object.
9	DetailsRecordSize	<b>PtypInteger16</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	2 bytes	The <b>details</b> property identifies the size of the modified user record, including the <b>DetailsRecordSize</b> and the terminating null character. This property is present only if the <b>type</b> property is set to 000 or 001. The maximum size of this property is limited to 64 kilobytes (KB).
10	DetailsRecords	Details record	Variable	Contains the <b>address-book-object-record</b> . This property is present only if the <b>type</b> property is set to 000 or 001.
11	<b>PidTagDisplayType</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.11)	1 byte integer	1 byte	The <b>display-type</b> property contains a value that is used to associate an icon with a particular row of a table.
12	<b>PidTagObjectType</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.10)	1 byte integer	1 byte	The <b>object-type</b> property contains the type of an object. The object type corresponds to the primary interface that is available for an object that is available through the OpenEntry interface. Set to 00 00 00 03 for a folder, 00 00 00 06 for a <b>mail user</b> , and 00 00 00 08 for a <b>distribution list</b> .

\*The server encodes this property as an ANSI character set string, but the client can interpret it as an ANSI character set or Unicode string.

Properties populated in the change-record for OAB version 3

Index Number	Property tag name	Property type	Property size	Description
1	<b>PidTagDisplayName</b>	<b>PtypString*</b>	Variable	The <b>display-name</b> property contains the display name for a given Address Book object encoded as UTF8.
2	ParentDNOffset	<b>PtypInteger32</b>	4 bytes	The <b>parent-dn-offset</b> property

Index Number	Property tag name	Property type	Property size	Description
				contains the offset to the PDN in the RDN Index file. This property is present only if the <b>type</b> property is set to 001.
3	RDNRecordKey	<b>PtypString*</b>	Variable	The <b>rdn</b> property uniquely identifies the RDN (1) in the RDN Index file. This property is present only if the <b>type</b> property is set to 001. This is a null-terminated string. The maximum size of this property is 68 bytes.
4	ParentDNOffsetForSMTP	<b>PtypInteger32</b>	4 bytes	The <b>domain-name-offset</b> property contains the offset of the Parent DN (4) SMTP address entry in the RDN Index file. This property is present only if the <b>type</b> property is set to 000.
5	<b>PidTagSmtAddress</b>	<b>PtypString8</b>	Variable	The <b>local-portion</b> property contains the SMTP mailing address of the sender encoded as UTF8.
6	<b>PidTagAccount</b>	<b>PtypString8</b>	Variable	The <b>alias</b> property contains the account name for the Address Book object encoded as UTF8.
7	<b>PidTagOfficeLocation</b>	<b>PtypString*</b>	Variable	The <b>location</b> property contains the office location of the Address Book object encoded as UTF8.
8	<b>PidTagSurname</b>	<b>PtypString*</b>	Variable	The <b>surname</b> property contains the family name of the Address Book object encoded as UTF8.
9	DetailsRecordSize	<b>PtypInteger16</b>	2 bytes	The <b>details</b> property identifies the size of the modified user record, including the <b>DetailsRecordSize</b> and the terminating null character. This property is present only if the <b>type</b> property is set to 000 or 001. The maximum size of this property is limited to 64 kilobytes (KB).
10	DetailsRecords	Details record	Variable	Contains the <b>address-book-object-record</b> . This property is present only if the <b>type</b> property is set to 000 or 001.
11	<b>PidTagDisplayType</b>	1 byte integer	1 byte	The <b>display-type</b> property

Index Number	Property tag name	Property type	Property size	Description
				contains a value that is used to associate an icon with a particular row of a table.
12	<b>PidTagObjectType</b>	1 byte integer	1 byte	The <b>object-type</b> property contains the type of an object. The object type corresponds to the primary interface that is available for an object that is available through the OpenEntry interface.  Set to 00 00 00 03 for a folder, 00 00 00 06 for a mail user, and 00 00 00 08 for a distribution list.

\*The server encodes this as an ANSI character set string, but the client can interpret it as an ANSI character set or Unicode string.

## 2.8 Compressed OAB Version 2 or OAB Version 3 File

**Note** Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.

The server compresses OAB version 2 and OAB version 3 files [<8>](#) before transferring them to the client. A compressed file starts with a header and then a sequence of compressed blocks. The server compresses all OAB version 2 and OAB version 3 files the same way. For more information about the compression of OAB version 2 and OAB version 3 files, see [\[MS-MCI\]](#).

A compressed OAB version 2 or OAB version 3 file is structured as the following ABNF definition illustrates.

```
v2-compressed-file = MDI_HDR 1*MDI_BLK
```

### 2.8.1 MDI\_HDR

The **MDI\_HDR** structure contains versioning information to indicate that it is an OAB version 2 or OAB version 3 compressed file. It contains the target file size value that the client SHOULD use to check that the final result is correct.

0	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	2	0	1	2	3	4	5	6	7	8	9	3	0	1
																		ulVersionHi																
																		ulVersionLo																
																		ulBlockMax																
																		ulTargetSize																

**ulVersionHi (4 bytes):** The high part of the file version information. This field MUST be set to 0x00000002.

**ulVersionLo (4 bytes):** The low part of the file version information. This field MUST be set to 0x00000001.

**ulBlockMax (4 bytes):** A 32-bit unsigned integer that indicates, in bytes, the largest sized block read from the source compressed input file or written to the target output file. This field is present so that the client can pre-allocate required buffers. This field MUST be set to 0x00008000.

**ulTargetSize (4 bytes):** A 32-bit unsigned integer value that specifies the expected length of the resulting output target file.

## 2.8.2 MDI\_BLK

The **MDI\_BLK** structure splits the decompression process into more easily handled smaller sized blocks.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ulFlags																															
ulCompSize																															
ulUncompSize																															
data (variable)																															
...																															

**ulFlags (4 bytes):** A 32-bit unsigned integer value that indicates whether the data field is compressed. MUST be either 0x00000000 to indicate the data field is not compressed and can be written out directly to the target file, or 0x00000001 to indicate the data field is compressed and ought to be decompressed using MCI decompression first.

**ulCompSize (4 bytes):** A 32-bit unsigned integer value that specifies the size of the data field in bytes.

**ulUncompSize (4 bytes):** A 32-bit unsigned integer value that specifies the size in bytes of the output target block to be written to the output file.

**data (variable):** A value that contains either a raw data stream or a compressed byte stream depending on the value of the **ulFlags** field. For more details, see [\[MS-MCI\]](#).

## 2.9 Uncompressed OAB Version 4 Full Details File

The Full Details file contains the entire OAB, including all Address Book objects, the list of properties that can be found in the address book, and information about the address book itself, including its name, a unique identity identifier, a version number, and a hash value.



Apart from the OAB header, the uncompressed Full Details file consists of the following three sections:

- OAB metadata record
- OAB header record
- One or more Address Book object records. Each Address Book object record starts with a little-endian 32 bit value that specifies the size of the record in bytes, including the record size field itself.

The OAB metadata record describes the schema of the OAB header record and Address Book object records. It starts with a record size value, then two schema tables: one for the OAB header record, and one for the Address Book object records. The server stores the tables sequentially after each other. The schema tables contain a 32 bit little-endian record count followed by the specified number of 32 bit property tag and 32 bit flag value pairs. The flag value tells the client which properties to index to match the behavior of a client working online.

The first property in the OAB header record and Address Book object records is the record size value, followed by a presence bit array, and then the property values. The property values appear in the order provided in the property table in the metadata record. The presence bit array indicates whether the property exists in the OAB header record or Address Book object records.

The OAB header record contains information about the address list itself, including the Unicode OAB name, the **ASCII** X500 DN (3) of the OAB, an integer sequence number, and the OAB GUID formatted as an ASCII string.

Address Book object records contain at minimum an ASCII SMTP address, an ASCII DN (3), a Unicode display name, an integer display type, and an integer object type. The number of Address Book object records matches the record count contained in the file header.

The Address Book object data in the Full Details file is not sorted in a predetermined manner, thus it is up to the client to decompress and index the file to enable fast retrieval and searches.

The following ABNF definition shows the format of an uncompressed OAB version 4 Details file.

```

v4-details-file      = OAB_HDR OAB_META_DATA
                       header-record
                       1*address-book-object-record
header-record        = OAB_V4_REC
address-book-object-record = OAB_V4_REC
    
```

### 2.9.1 OAB\_HDR

The **OAB\_HDR** structure specifies the OAB file format version and the number of Address Book object records in the address list, and it contains a hash value for consistency checks.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
ulVersion																															
ulSerial																															

ulTotRecs
-----------

**ulVersion (4 bytes):** A 32-bit unsigned integer that specifies the file type. Set to 0x00000020 for uncompressed version 4 OAB Full Details files. Set to 0x00000007 for uncompressed display template files.

**ulSerial (4 bytes):** A 32-bit unsigned integer that contains the CRC hash of the rest of the file not including this header structure. All CRC checksums are calculated with an initial seed of 0xFFFFFFFF and use the IEEE 802.3 CRC polynomial of 0xEDB88320, as specified in [\[ISO/IEC8802-3\]](#).

**ulTotRecs (4 bytes):** A 32-bit unsigned integer that specifies the number of **address-book-object-records** stored in the file.

## 2.9.2 OAB\_META\_DATA

The **OAB\_META\_DATA** structure contains information about the schema of all properties that can be represented in an OAB header or Address Book object record.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
cbSize																															
rgHdrAtts (variable)																															
...																															
rgOabAtts (variable)																															
...																															

**cbSize (4 bytes):** A 32-bit unsigned integer that specifies the length of the **OAB\_META\_DATA** structure in bytes. This count includes both the **cbSize** field and the combined length of the **rgHdrAtts** and **rgOabAtts** fields.

**rgHdrAtts (variable):** An **OAB\_PROP\_TABLE** structure that describes the properties that can be present in the **header-record**. MUST contain 4 or more header property records, as specified in section [2.9.2.1](#).

**rgOabAtts (variable):** An **OAB\_PROP\_TABLE** structure that describes the properties that can be present in any **address-book-object-record**. MUST contain 36 Address Book object property records, as specified in section [2.9.2.2](#).

### 2.9.2.1 rgHdrAtts

The **rgHdrAtts** table MUST have at least the four following attributes for compatibility with the client.

Index Number	Property tag name	Property tag	Property type	Description
1	<b>PidTagOfflineAddressBookName</b> ( <a href="#">[MS-OXPFOAB]</a> section 2.2.1.1)	0x680001F	<b>PtypString</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	Display name of the address list. Can change between generation versions of the same address list.
2	<b>PidTagOfflineAddressBookDistinguishedName</b> ( <a href="#">[MS-OXPFOAB]</a> section 2.2.1.4)	0x6804001E	<b>PtypString8</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	The AddressList-X500-DN of the address list container object. Can change between generation versions of the same address list. MUST contain Teletex characters only, as specified by the non-space-teletex rule in section <a href="#">2.1</a> .
3	<b>PidTagOfflineAddressBookSequence</b> ( <a href="#">[MS-OXPFOAB]</a> section 2.2.1.2)	0x68010003	<b>PtypInteger32</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	The sequence number of the OAB. This number increases by one between generation versions of the same address list.
4	<b>PidTagOfflineAddressBookContainerGuid</b> ( <a href="#">[MS-OXPFOAB]</a> section 2.2.1.3)	0x6802001E	<b>PtypString8</b>	A string formatted GUID that represents the address list container object. This value never changes between generation versions of the same address list.

Index Number	Property tag name	Property tag	Property type	Description
				This value is formatted as "xxxxxxx-xxxx-xxxx-xxxxxxxxxxx". MUST contain Teletex characters only, as specified by the non-space-teletex rule in section <a href="#">2.1</a>

The property in the following table is an optional property in the **rgHdrAtts** table. [<9>](#)

Property tag name	Property tag	Property type	Description
<b>PidTagAddressBookHierarchicalRootDepartment</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.7.2)	0x8C98001E	<b>PtypString8</b>	DN for the root <b>departmental group</b> in the department hierarchy for the organization. The DN (4) can change between generation versions of the same address list. MUST contain Teletex characters only, as specified by the non-space-teletex rule in section <a href="#">2.1</a> .

### 2.9.2.2 rgOabAtts

The rgOabAtts table MUST be present on all Address Book object records, and MUST have at least the following properties. If a required property is absent from the rgOabAtts table, then it is assumed that the property has no value for all recipients (2) in the address book.

**PidTagEmailAddress** ([\[MS-OXOABK\]](#) section 2.2.3.14) — this MUST be the first entry.

**PidTagSmtpAddress** ([\[MS-OXOABK\]](#) section 2.2.3.21) — this MUST be the second entry.

**PidTagDisplayName** ([\[MS-OXOABK\]](#) section 2.2.3.1)

**PidTagAccount** ([\[MS-OXOABK\]](#) section 2.2.3.20)

**PidTagSurname** ([\[MS-OXOABK\]](#) section 2.2.4.1)  
**PidTagGivenName** ([\[MS-OXOABK\]](#) section 2.2.4.2)  
**PidTagAddressBookProxyAddresses** ([\[MS-OXOABK\]](#) section 2.2.3.23)  
**PidTagOfficeLocation** ([\[MS-OXOABK\]](#) section 2.2.4.8)  
**PidTagDisplayType** ([\[MS-OXOABK\]](#) section 2.2.3.11)  
**PidTagObjectType** ([\[MS-OXOABK\]](#) section 2.2.3.10)  
**PidTagSendRichInfo** ([\[MS-OXOABK\]](#) section 2.2.3.18)  
**PidTagBusinessTelephoneNumber** ([\[MS-OXOABK\]](#) section 2.2.4.38)  
**PidTagInitials** ([\[MS-OXOABK\]](#) section 2.2.4.5)  
**PidTagStreetAddress** ([\[MS-OXOABK\]](#) section 2.2.4.19)  
**PidTagLocality** ([\[MS-OXOABK\]](#) section 2.2.4.21)  
**PidTagStateOrProvince** ([\[MS-OXOABK\]](#) section 2.2.4.22)  
**PidTagPostalCode** ([\[MS-OXOABK\]](#) section 2.2.4.23)  
**PidTagCountry** ([\[MS-OXOABK\]](#) section 2.2.4.24)  
**PidTagTitle** ([\[MS-OXOABK\]](#) section 2.2.4.7)  
**PidTagCompanyName** ([\[MS-OXOABK\]](#) section 2.2.4.10)  
**PidTagAssistant** ([\[MS-OXOABK\]](#) section 2.2.4.11)  
**PidTagDepartmentName** ([\[MS-OXOABK\]](#) section 2.2.4.9)  
**PidTagAddressBookTargetAddress** ([\[MS-OXOABK\]](#) section 2.2.3.22)  
**PidTagHomeTelephoneNumber** ([\[MS-OXOABK\]](#) section 2.2.4.39)  
**PidTagBusiness2TelephoneNumber** ([\[MS-OXOABK\]](#) section 2.2.4.41)  
**PidTagHome2TelephoneNumber** ([\[MS-OXOABK\]](#) section 2.2.4.43)  
**PidTagPrimaryFaxNumber** ([\[MS-OXOABK\]](#) section 2.2.4.50)  
**PidTagMobileTelephoneNumber** ([\[MS-OXOABK\]](#) section 2.2.4.45)  
**PidTagAssistantTelephoneNumber** ([\[MS-OXOABK\]](#) section 2.2.4.57)  
**PidTagPagerTelephoneNumber** ([\[MS-OXOABK\]](#) section 2.2.4.49)  
**PidTagComment** ([\[MS-OXOABK\]](#) section 2.2.3.31)  
**PidTagUserCertificate** ([\[MS-OXOABK\]](#) section 2.2.4.64)  
**PidTagUserX509Certificate** ([\[MS-OXOABK\]](#) section 2.2.4.66)  
**PidTagAddressBookX509Certificate** ([\[MS-OXOABK\]](#) section 2.2.4.65)

**PidTagAddressBookHomeMessageDatabase** ([\[MS-OXOABK\]](#) section 2.2.4.67)**PidTagAddressBookDisplayNamePrintable** ([\[MS-OXPROPS\]](#) section 2.592)

The following table lists the properties that the server populates by default on an Address Book object record. [<10><11>](#) All of these properties except **PidTagEmailAddress** ([\[MS-OXPROPS\]](#) section 2.759) and **PidTagSmtAddress** ([\[MS-OXPROPS\]](#) section 2.1093) are optional (the required properties are specified in the list that precedes this paragraph). Each of the properties is further specified in [\[MS-OXOABK\]](#).

Index Number	Property tag name	Property tag	Property type	Description
1	<b>PidTagEmailAddress</b>	0x3003001E	<b>PtypString8</b> ( <a href="#">[MS-OXCADATA]</a> section 2.11.1)	Contains the X500 DN. MUST contain Teletex characters only, as specified by the non-space-teletex rule in section <a href="#">2.1</a> .
2	<b>PidTagSmtAddress</b>	0x39fe001F	<b>PtypString</b> ( <a href="#">[MS-OXCADATA]</a> section 2.11.1)	Contains the SMTP mailing address of the sender.
3	<b>PidTagDisplayName</b>	0x3001001F	<b>PtypString</b>	Contains the display name for a given Address Book object.
4	<b>PidTagAddressBookPhoneticDisplayName</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.9)	0x8C92001F	<b>PtypString</b>	Contains the phonetic display name of an object.
5	<b>PidTagAccount</b>	0x3A00001F	<b>PtypString</b>	Contains the account name for the Address Book object.
6	<b>PidTagSurname</b>	0x3A11001F	<b>PtypString</b>	Contains the family name of the Address Book object.
7	<b>PidTagAddressBookPhoneticSurname</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.15)	0x8C8F001F	<b>PtypString</b>	Contains the phonetic spelling of the surname.
8	<b>PidTagGivenName</b>	0x3A06001F	<b>PtypString</b>	Contains the given name of the Address Book object.
9	<b>PidTagAddressBookPhoneticGivenName</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.14)	0x8C8E001F	<b>PtypString</b>	Contains the phonetic given name of the Address Book object.
10	<b>PidTagAddressBookProxyAddresses</b>	0x800f101F	<b>PtypMultipleString</b> ( <a href="#">[MS-OXCADATA]</a> section 2.11.1)	Contains the e-mail proxy addresses of the Address Book object. For example, SMTP:Laura.Miller@example.com or X400:c=US;a=;p=example;o=example;s=Miller;g=Laura;.

<b>Index Number</b>	<b>Property tag name</b>	<b>Property tag</b>	<b>Property type</b>	<b>Description</b>
11	<b>PidTagOfficeLocation</b>	0x3A19001F	<b>PtypString</b>	Contains the office location of the Address Book object.
12	<b>PidTagDisplayType</b>	0x39000003	<b>PtypInteger 32</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	Contains a value that is used to associate an icon with a particular row of a table.
13	<b>PidTagObjectType</b>	0x0FFE0003	<b>PtypInteger 32</b>	Contains the type of an object. The object type corresponds to the primary interface that is available for an object that is available through the OpenEntry interface.
14	<b>PidTagSendRichInfo</b>	0x3A40000B	<b>PtypBoolean</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	Contains "TRUE" if the entry can receive all message content, including RTF and OLE objects; otherwise, contains "FALSE".
15	<b>PidTagBusinessTelephoneNumber</b>	0x3A08001F	<b>PtypString</b>	Contains the primary business telephone for the Address Book object.
16	<b>PidTagInitials</b>	0x3A0A001F	<b>PtypString</b>	Contains the initials for parts of the full name of the Address Book object.
17	<b>PidTagStreetAddress</b>	0x3A29001F	<b>PtypString</b>	Contains the street address of the Address Book object.
18	<b>PidTagLocality</b>	0x3A27001F	<b>PtypString</b>	Contains the name of the locality of the Address Book object, such as the town or city.
19	<b>PidTagStateOrProvince</b>	0x3A28001F	<b>PtypString</b>	Contains the name of the state or province in which the Address Book object is located.
20	<b>PidTagPostalCode</b>	0x3A2A001F	<b>PtypString</b>	Contains the postal code for the postal address of the Address Book object.
21	<b>PidTagCountry</b>	0x3A26001F	<b>PtypString</b>	Contains the name of the country or region where the Address Book object is located.
22	<b>PidTagTitle</b>	0x3A17001F	<b>PtypString</b>	Contains the job title of the Address Book object.
23	<b>PidTagCompanyName</b>	0x3A16001F	<b>PtypString</b>	Contains the name of the company associated with the Address Book object.

<b>Index Number</b>	<b>Property tag name</b>	<b>Property tag</b>	<b>Property type</b>	<b>Description</b>
24	<b>PidTagAddressBookPhoneticCompanyName</b> ([MS-OXOABK] section 2.2.4.16)	0x8C91001F	<b>PtypString</b>	Contains the phonetic spelling of the company name of the Address Book object.
25	<b>PidTagAssistant</b>	0x3A30001F	<b>PtypString</b>	Contains the name of the administrative assistant of the Address Book object.
26	<b>PidTagDepartmentName</b>	0x3A18001F	<b>PtypString</b>	Contains the name of the department in which the Address Book object works.
27	<b>PidTagAddressBookPhoneticDepartmentName</b> ([MS-OXOABK] section 2.2.4.17)	0x8C90001F	<b>PtypString</b>	Contains the phonetic spelling of the name of the department in which the Address Book object works.
28	<b>PidTagAddressBookTargetAddress</b>	0x8011001F	<b>PtypString</b>	Contains the destination address for the Address Book object.
29	<b>PidTagHomeTelephoneNumber</b>	0x3A09001F	<b>PtypString</b>	Contains the primary home telephone number of the Address Book object.
30	<b>PidTagBusiness2TelephoneNumbers</b>	0x3A1B101F	<b>PtypMultipleString</b>	Contains the secondary business telephone numbers of the Address Book object.
31	<b>PidTagHome2TelephoneNumbers</b>	0x3A2F101F	<b>PtypMultipleString</b>	Contains the secondary home telephone numbers of the Address Book object.
32	<b>PidTagPrimaryFaxNumber</b>	0x3A23001F	<b>PtypString</b>	Contains the telephone number of the primary fax machine used by the Address Book object.
33	<b>PidTagMobileTelephoneNumber</b>	0x3A1C001F	<b>PtypString</b>	Contains the cellular telephone number of the Address Book object.
34	<b>PidTagAssistantTelephoneNumber</b>	0x3A2E001F	<b>PtypString</b>	Contains the telephone number of the administrative assistant of the Address Book object.
35	<b>PidTagPagerTelephoneNumber</b>	0x3A21001F	<b>PtypString</b>	Contains the pager telephone number of the Address Book object.
36	<b>PidTagComment</b>	0x3004001F	<b>PtypString</b>	Contains a comment about the purpose or content of an Address Book object.
37	<b>PidTagUserCertificate</b>	0x3A220102	<b>PtypBinary</b>	Contains an ASN.1 authentication certificate for a messaging user.



Index Number	Property tag name	Property tag	Property type	Description
38	<b>PidTagUserX509Certificate</b>	0x3A701102	<b>PtypMultiple Binary</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	Contains X.509 version 3 security certificates for the Address Book object, as specified in <a href="#">[RFC2459]</a> .
39	<b>PidTagAddressBookX509Certificate</b>	0x8C6A1102	<b>PtypMultiple Binary</b>	Contains ASN.1 encoded X.509 certificates, as specified in <a href="#">[RFC2459]</a> .
40	<b>PidTagAddressBookHomeMessageDatabase</b>	0x8006001E	<b>PtypString8</b>	Contains the X500 DN of the store for this mailbox. This property value is not subject to truncation. MUST contain Teletex characters only, as specified by the <code>teletex-char</code> rule in section <a href="#">2.1</a> .
41	<b>PidTagAddressBookDisplayName Printable</b>	0x39FF001E	<b>PtypString8</b>	Contains the printable string version of the display name of the Address Book object. MUST contain Teletex characters only, as specified by the <code>teletex-char</code> rule in section <a href="#">2.1</a> .
42	<b>PidTagDisplayTypeEx</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.12)	0x39050003	<b>PtypInteger 32</b>	Contains a value used to associate an icon with a particular row of a table.
43	<b>PidTagAddressBookSeniorityIndex</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.24)	0x8CA00003	<b>PtypInteger 32</b>	Contains the seniority index for the user or department. The value is used to sort users or departments by order of seniority.
44	<b>PidTagAddressBookHierarchicalIsHierarchicalGroup</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.6.5)	0x8CD0000B	<b>PtypBoolean</b>	Contains "TRUE" if the distribution list represents a departmental group; otherwise, contains "FALSE".
45	<b>PidTagAddressBookObjectGuid</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.25)	0x8C6D0102	<b>PtypBinary</b>	Contains the GUID that uniquely identifies the Address Book object.
46	<b>PidTagAddressBookSenderHintTranslations</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.26)	0x8CAC101F	<b>PtypMultiple String</b>	Contains the locale ID and translations of the default <b>mail tip</b> . For example, "en-US:Hello" "es:Hola".
47	<b>PidTagAddressBookDeliveryContentLength</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.27)	0x806A0003	<b>PtypInteger 32</b>	Specifies the maximum size of a message that a recipient (2) can receive.

Index Number	Property tag name	Property tag	Property type	Description
48	<b>PidTagAddressBookModerationEnabled</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.28)	0x8CB5000B	<b>PtypBoolean</b>	Contains "TRUE" if moderation is enabled for the mail user or distribution list; otherwise, contains "FALSE".
49	<b>PidTagAddressBookDistributionListMemberCount</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.29)	0x8CE20003	<b>PtypInteger32</b>	Contains the total number of recipients (2) in the distribution list. This value includes expanding all of the distribution lists that are members of the distribution list, and including their members in the total.
50	<b>PidTagAddressBookDistributionListExternalMemberCount</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.3.30)	0x8CE30003	<b>PtypInteger32</b>	Contains the number of external recipients (2) in the distribution list.
51	<b>PidTagAddressBookMember</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.6.1)	0x8009101E	<b>PtypEmbeddedTable</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1), encoded as <b>PtypMultipleString8</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1) as specified in section <a href="#">2.9.6.7</a> .	Contains the members of the distribution list. If the distribution list is also a departmental group (as specified by the <b>PidTagAddressBookHierarchicalIsHierarchicalGroup</b> property), then the <b>PidTagAddressBookMember</b> property contains the members of the department and the child departmental groups in the hierarchy of departments.
52	<b>PidTagAddressBookIsMemberOfDistributionList</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.5.3)	0x8008101E	<b>PtypEmbeddedTable</b> , encoded as <b>PtypMultipleString8</b> as specified in section <a href="#">2.9.6.7</a> .	Lists all of the distribution lists to which this object is a member.
53	<b>PidTagOfflineAddressBookTruncatedProperties</b> ( <a href="#">[MS-OXPROPS]</a> section 2.893)	0x68051003	<b>PtypMultipleInteger32</b>	Contains a list of the property tags that have been truncated or limited by the server. If no properties have been removed or limited, the attribute will not be present.  For more information about this property, see section <a href="#">2.9.2.2.1</a> .

### 2.9.2.2.1 Truncated Properties

The **PidTagOfflineAddressBookTruncatedProperties** ([\[MS-OXPROPS\]](#) section 2.893) property contains the list of property tags that have been truncated or dropped due to size limits. Clients SHOULD check the property being retrieved from the OAB record against the list of truncated properties for the record. If the property is included in the truncated property list, the value stored in the OAB file is not the same as the address book value that is available online through a corresponding address book on a **name service provider interface (NSPI)** server.

Properties are truncated in, or dropped from, the OAB file as follows:

- String property — A string in the OAB file is truncated to a size limit.
- Binary property — If the binary value exceeds the size limit, it is dropped from the OAB file.
- Multi-valued property (string or binary) — If the combined size of all values exceeds the size limit, individual values are dropped from the OAB file.
- **PtypObject** property — The value of a **PtypObject** property is absent in the OAB file. The **presenceBitArray**, specified in section [2.9.5](#), is set to 0 for a **PtypObject** property. A **PtypObject** property is included in the **PidTagOfflineAddressBookTruncatedProperties** whenever there is a value available online through a corresponding address book on an NSPI server. This document does not include value encodings (see section [2.9.6](#)) for properties of type **PtypObject**.

The following table defines the default minimum and maximum values of limit settings for string and binary data in the OAB file. The minimum limit value is the smallest value that a limit can be set to, rather than the smallest size that an actual value can be. The maximum limit value is the largest value that a size limit can be set to, and does reflect the largest size a property can be.

Limit	Type	Minimum Limit Value (in bytes)	Maximum Limit Value (in bytes)
String limit	DWORD	32	3400
Binary limit	DWORD	1024	32768
String multi-valued limit	DWORD	512	65536
Binary multi-valued limit	DWORD	2048	65536

The only properties that cannot be truncated are **PidTagOfflineAddressBookTruncatedProperties**, **PidTagEmailAddress**, and **PidTagAddressBookHomeMessageDatabase**. The **PidTagEmailAddress** and **PidTagAddressBookHomeMessageDatabase** properties are not limited because they are primary key values that uniquely identify an object.

The following table specifies the default attributes included in the **PidTagOfflineAddressBookTruncatedProperties** property. [<12>](#) Each property is further specified in [\[MS-OXOABK\]](#).

Index Number	Property tag name	Property tag	Property type	Description
1	<b>PidTagThumbnailPhoto</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.82)	0x8C9E0102	<b>PtypBinary</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	Contains an image of the mail user's photo in .jpg format.
2	<b>PidTagSpokenName</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.83)	0x8CC20102	<b>PtypBinary</b>	Contains a recording of the mail user's name pronunciation.
3	<b>PidTagAddressBookAuthorizedSenders</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.84)	0x8CD8000D	<b>PtypObject</b> ( <a href="#">[MS-OXCDATA]</a> section 2.11.1)	A value other than null indicates that delivery restrictions exist for this recipient (2). The address book does not contain the lists of senders that are allowed for this recipient (2); it only indicates whether or not such restrictions exist. The client can use <b>PidTagAddressBookAuthorizedSenders</b> , <b>PidTagAddressBookUnauthorizedSenders</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.85), <b>PidTagAddressBookDistributionListMemberSubmitAccepted</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.86), and <b>PidTagAddressBookDistributionListMemberSubmitRejected</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.87) to compute a Boolean value that indicates whether a distribution list has restrictions on it.
4	<b>PidTagAddressBookUnauthorizedSenders</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.85)	0x8CD9000D	<b>PtypObject</b>	A value other than null indicates that delivery restrictions exist for this recipient (2). The address book does not contain the lists of senders that are prohibited for this recipient (2); it only indicates whether or not such restrictions exist.
5	<b>PidTagAddressBookDistributionListMemberSubmitAccepted</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.86)	0x8073000D	<b>PtypObject</b>	A value other than null indicates that delivery restrictions exist for this recipient (2). The address book does not contain the lists of the group of senders that are allowed for this recipient (2); it only indicates whether or not such restrictions exist.

Index Number	Property tag name	Property tag	Property type	Description
6	<b>PidTagAddressBookDistributionListMemberSubmitRejected</b> ( <a href="#">[MS-OXOABK]</a> section 2.2.4.87)	0x8CD A00D	<b>PtypObject</b>	A value other than null indicates that delivery restrictions exist for this recipient (2). The address book does not contain the lists of the group of senders that are prohibited for this recipient (2); it only indicates whether or not such restrictions exist.
7	<b>PidTagAddressBookDistributionListRejectMessagesFromDLMembers</b> ( <a href="#">[MS-OXPROPS]</a> section 2.598)	0x8CD B00D	<b>PtypObject</b>	A value other than null indicates that delivery restrictions exist for this recipient (2). The address book does not contain the lists of distribution list senders that are prohibited for this recipient (2); it only indicates whether or not such restrictions exist.

### 2.9.3 OAB\_PROP\_TABLE

The **OAB\_PROP\_TABLE** structure represents the property schema of either the OAB header record or all the Address Book object records. It contains a list of **OAB\_PROP\_REC** structures.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
cAtts																															
rgProps (variable)																															
...																															

**cAtts (4 bytes):** A 32-bit integer that specifies the number of **OAB\_PROP\_REC** records in **rgProps**.

**rgProps (variable):** A list of **OAB\_PROP\_REC** structures.

### 2.9.4 OAB\_PROP\_REC

The **OAB\_PROP\_REC** structure defines a property that the server can store in an OAB header, or Address Book object record, and describes how the attribute is used online.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ulPropId																															

c	b	a
---	---	---

**ulPropId (4 bytes):** A 32-bit unsigned integer that specifies the property tag. The **property type** portion of the property tag MUST be one of the following values.

Value	Meaning
0x0003	<b>PtypInteger32</b> ([MS-OXCDATA] section 2.11.1)
0x000B	<b>PtypBoolean</b> ([MS-OXCDATA] section 2.11.1)
0x000D	<b>PtypObject</b> ([MS-OXCDATA] section 2.11.1)
0x001E	<b>PtypString8</b> ([MS-OXCDATA] section 2.11.1)
0x001F	<b>PtypString</b> ([MS-OXCDATA] section 2.11.1)
0x0102	<b>PtypBinary</b> ([MS-OXCDATA] section 2.11.1)
0x1003	<b>PtypMultipleInteger32</b> ([MS-OXCDATA] section 2.11.1)
0x101E	<b>PtypMultipleString8</b> ([MS-OXCDATA] section 2.11.1)
0x101F	<b>PtypMultipleString</b> ([MS-OXCDATA] section 2.11.1)
0x1102	<b>PtypMultipleBinary</b> ([MS-OXCDATA] section 2.11.1)

**c (30 bits):** This field MUST be set to zero when sent and MUST be ignored when received.

**b (1 bit):** A single bit value. This field is set to 1 to specify that the property is a primary key index when used online and a value MUST be present on every **address-book-object-record** in the OAB version 4 Full Details file. This field is set to 0 to specify that the property is not a primary key index — in this case, the property can be present, but is not required, on an **address-book-object-record**.

The server includes the following properties in the primary key index by default:

**PidTagEmailAddress** ([MS-OXOABK] section 2.2.3.14)

**PidTagSmtAddress** ([MS-OXOABK] section 2.2.3.21)

**a (1 bit):** A single bit value. This field is set to 1 to specify that the property is part of the ANR index online. This field is set to 0 to specify that it is not part of the ANR index online.

The server includes the following properties in the ANR index by default:

**PidTagDisplayName** ([MS-OXOABK] section 2.2.3.1)

**PidTagAddressBookPhoneticDisplayName** ([MS-OXOABK] section 2.2.3.9)

**PidTagAccount** ([MS-OXOABK] section 2.2.3.20)

**PidTagSurname** ([MS-OXOABK] section 2.2.4.1)

**PidTagAddressBookPhoneticSurname** ([MS-OXOABK] section 2.2.4.15)

**PidTagGivenName** ([MS-OXOABK] section 2.2.4.2)

**PidTagAddressBookPhoneticGivenName** ([MS-OXOABK] section 2.2.4.14)

**PidTagAddressBookProxyAddresses** ([MS-OXOABK] section 2.2.3.23)

**PidTagOfficeLocation<13>** ([MS-OXOABK] section 2.2.4.8)

## 2.9.5 OAB\_V4\_REC

The **OAB\_V4\_REC** structure represents either the OAB header record or an individual Address Book object record in an OAB file.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
cbSize																															
presenceBitArray (variable)																															
...																															
data (variable)																															
...																															

**cbSize (4 bytes):** A 32-bit unsigned integer that specifies the length of the **OAB\_V4\_REC** structure in bytes. This count includes both the **cbSize** field and the combined length of the **presenceBitArray** and **data** fields.

**presenceBitArray (variable):** A bit array that indicates whether a property specified in the **OAB\_PROP\_TABLE** structure is present in the data field. The first element of the bit array is the most significant bit of the first byte. The size of the **presenceBitArray** field in bytes MUST be the value of the **cAtts** field of the appropriate **OAB\_PROP\_TABLE** structure divided by 8 and rounded up to the nearest integer value. A 0 record in the **presenceBitArray** indicates that the property is not present in the data field. 1 indicates the property is present. The index of the property in the **OAB\_PROP\_TABLE** structure MUST match the index of the value in the **presenceBitArray**. Unused bits in the final byte MUST be set to 0.

**data (variable):** The set of property values for the **address-book-object-record** or **header-record**. The format of the **data** field is specified in section 2.9.6.

## 2.9.6 Data Encoding

Property values are encoded in the data field based on the property type and are packed on byte boundaries. The properties are laid out in the order that the property definition exists in the **OAB\_PROP\_TABLE** structure. If a property does not exist, the **presenceBitArray** value MUST be 0 and no value is encoded in the data field.

### 2.9.6.1 PtypInteger32 (0x0003) Value Encoding

All **PtypInteger32** ([MS-OXCDATA] section 2.11.1) values are considered unsigned and MUST fit in the range of a 32-bit integer (0---232-1). Integers equal to or less than 127 MUST be encoded as a single byte. Integers 128 or greater are encoded with first a byte count byte with the most significant bit set, then the **little-endian** value encoding. The byte count, if required, MUST be

0x81, 0x82, 0x83, or 0x84 representing 1, 2, 3, or 4 bytes. The most significant byte of the value representation MUST NOT be 0x00, a lower byte count MUST be used. For example, 0x0000007F is encoded as 0x7F, not as 0x81 0x7F, 0x82 0x7F 0x00, 0x83 0x7F 0x00 0x00, or 0x84 0x7F 0x00 0x00 0x00. Values 0x0100 through 0xFFFF are encoded as 0x82 0xLSB 0xMSB. Values 0x00010000 through 0x00FFFFFF are encoded as 0x83 0xLSB 0xXX 0xMSB, and values 0x01000000 through 0xFFFFFFFF are encoded as 0x84 0xLSB 0xXX 0xXX 0xMSB.

### 2.9.6.2 PtypBoolean (0x000B) Value Encoding

All **PtypBoolean** ([\[MS-OXCDATA\]](#) section 2.11.1) values are encoded as a single byte. "TRUE" MUST be encoded as 0x01 and "FALSE" MUST be encoded as 0x00.

### 2.9.6.3 PtypString8 (0x001E) Value Encoding

All narrow character set or multi-byte character set strings are encoded as byte sequences and MUST be terminated by a single 0x00 byte. A string sequence MUST NOT contain a 0x00 byte as part of the string itself. A zero length or empty string MUST NOT be encoded, but MUST be marked as not present in the **presenceBitArray**.

Properties whose data type is **PtypEmbeddedTable** ([\[MS-OXCDATA\]](#) section 2.11.1), and whose value represents a reference to at most one other Address Book object, are encoded using the **PtypString8** value encoding. The string value MUST be a distinguished name (DN) (4) for an Address Book object, which can be present in the OAB.

For more details about the **PtypString8** data type, see [\[MS-OXCDATA\]](#) section 2.11.1.

### 2.9.6.4 PtypString (0x001F) Value Encoding

ASCII strings are encoded as null-terminated strings.

All Unicode strings are encoded as UTF-8 byte sequences, as specified in [\[RFC3629\]](#), and MUST be terminated by a single 0x00 byte. A string encoding MUST NOT contain a 0x00 byte as part of the string itself. A zero length or empty string MUST NOT be encoded, but MUST be marked as not present in the **presenceBitArray**.

For more details about the **PtypString** data type, see [\[MS-OXCDATA\]](#) section 2.11.1.

### 2.9.6.5 PtypBinary (0x0102) Value Encoding

All raw byte sequences are encoded by a length value followed by the specified number of bytes. The length value is encoded as a **PtypInteger32** ([\[MS-OXCDATA\]](#) section 2.11.1) as shown in section [2.9.6.1](#). For example, the byte sequence 0x22 0xF8 0xFF 0x00 0x22 would be encoded as 0x05 0x22 0xF8 0xFF 0x00 0x22. A zero length **PtypBinary** ([\[MS-OXCDATA\]](#) section 2.11.1) value MUST NOT be encoded, but MUST be marked as not present in the **presenceBitArray**.

### 2.9.6.6 PtypMultipleInteger32 (0x1003) Value Encoding

Multi-valued integer encodings start with an integer count encoding followed by the specified number of integer value encodings. All integer encodings, including the value count, are encoded in the same way that **PtypInteger32** ([\[MS-OXCDATA\]](#) section 2.11.1) is encoded. All values MUST be unique. Values can appear in any order.

For more details about the **PtypMultipleInteger32** data type, see [\[MS-OXCDATA\]](#) section 2.11.1.



### 2.9.6.7 PtypMultipleString8 (0x101E) Value Encoding

Multi-valued string encodings start with an integer count encoding followed by the specified number of string value encodings. The count encoding is encoded in the same way that **PtypInteger32** ([\[MS-OXCADATA\]](#) section 2.11.1) is encoded. The individual string encodings are encoded in the same way that **PtypString8** ([\[MS-OXCADATA\]](#) section 2.11.1) is encoded. Strings MUST be case-insensitive. All values MUST be unique. Values can appear in any order. All strings MUST NOT be zero length or empty.

Properties whose data type is **PtypEmbeddedTable** ([\[MS-OXCADATA\]](#) section 2.11.1), and whose value represents references to any number of other Address Book objects, are encoded using the **PtypMultipleString8** ([\[MS-OXCADATA\]](#) section 2.11.1) value encoding. Each string value MUST be a distinguished name (DN) (4) to an Address Book object, which can be present in the OAB.

### 2.9.6.8 PtypMultipleString (0x101F) Value Encoding

Multi-valued Unicode string encodings start with an integer count encoding followed by the specified number of Unicode string value encodings. The count encoding is encoded in the same way that **PtypInteger32** ([\[MS-OXCADATA\]](#) section 2.11.1) is encoded. The individual string encodings are encoded in the same way that **PtypString** ([\[MS-OXCADATA\]](#) section 2.11.1) is encoded. Strings MUST be case-insensitive. All values MUST be unique. Values can appear in any order. All strings MUST NOT be zero length or empty.

For more details about the **PtypMultipleString** data type see [\[MS-OXCADATA\]](#) section 2.11.1.

### 2.9.6.9 PtypMultipleBinary (0x1102) Value Encoding

Multi-valued binary octet encodings start with an integer count encoding, followed by the specified number of binary value encodings. The count encoding is encoded in the same way that **PtypInteger32** ([\[MS-OXCADATA\]](#) section 2.11.1) is encoded. The individual binary encodings are encoded in the same way that **PtypBinary** ([\[MS-OXCADATA\]](#) section 2.11.1) is encoded. All values MUST be unique. Values can appear in any order. Any binary value MUST NOT be zero length.

For more details about the **PtypMultipleBinary** data type see [\[MS-OXCADATA\]](#) section 2.11.1.

## 2.10 Compressed OAB Version 4 Differential Patch File

The Differential Patch file describes how to transform an outdated Full Details file into another Full Details file. During transformation, the Differential Patch file is read by the client one block at a time to determine how large a block of the original Full Details file to read, how large the output block will be, and what the compressed patch data is. The patch file starts with a patch header that contains the file format version numbers, a maximum block size value, source and target file sizes, and the source and target file CRC hash codes. The maximum block size value tells the client the maximum size it can expect to be required to read from the original Full Details file, the maximum size it can expect to have to write to the output file, and the size of the largest patch record that will be produced. Following the patch header are a series of patch blocks. The patch block contains the patch size in bytes to be read from the patch file, the size in bytes of the target block that will be produced, the size in bytes of the block to be read from the original Full Details file, and the CRC hash that the resulting output block will have. The start and end of the source and output blocks do not necessarily fall on record boundaries of the source or output files.

The Differential Patch file cannot be uncompressed by itself as it requires the original Full Details file.

The following ABNF definition shows the format of a compressed OAB version 4 Differential Patch file.

```
patch-file = PATCH_HDR 1*PATCH_BLK
```

Patch files are only applied against OAB version 4 Full Details files to produce the next generation of the file.

### 2.10.1 PATCH\_HDR

The **PATCH\_HDR** structure contains versioning information to indicate that it is an OAB version 4 patch file. It contains source and target file hash and file size values.

0	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	2	0	1	2	3	4	5	6	7	8	9	3	0	1
ulVersionHi																																		
ulVersionLo																																		
ulBlockMax																																		
ulSourceSize																																		
ulTargetSize																																		
ulSourceCRC																																		
ulTargetCRC																																		

**ulVersionHi (4 bytes):** The high part of the file version information. This field MUST be set to 0x00000003.

**ulVersionLo (4 bytes):** The low part of the file version information. This field MUST be set to 0x00000002.

**ulBlockMax (4 bytes):** A 32-bit integer value that specifies in bytes the largest size of a block that will be read from the source OAB Details input file, written to the target OAB details output file, or read from the Differential Patch file. This field is here so that the client can pre-allocate required buffers.

**ulSourceSize (4 bytes):** A 32-bit integer value that specifies the length in bytes that the source input file is expected to be.

**ulTargetSize (4 bytes):** A 32-bit integer value that specifies the length that the resulting output target file is expected to be.

**ulSourceCRC (4 bytes):** A 32-bit integer value that represents the CRC hash of the source input file (excluding the **OAB\_HDR** structure).

**ulTargetCRC (4 bytes):** A 32-bit integer value that represents the CRC hash of the target output file (excluding the **OAB\_HDR** structure).

## 2.10.2 PATCH\_BLK

The **PATCH\_BLK** structure splits the patch process into more easily handled smaller-sized blocks.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ulPatchSize																															
ulTargetSize																															
ulSourceSize																															
ulCRC																															
data (variable)																															
...																															

**ulPatchSize (4 bytes):** A 32-bit integer value that specifies the size of the data field in bytes.

**ulTargetSize (4 bytes):** A 32-bit integer value that specifies the size in bytes of the output target block to be written to the output file.

**ulSourceSize (4 bytes):** A 32-bit integer value that specifies the size in bytes of the source input block to be read from the source input file and used to generate the output block.

**ulCRC (4 bytes):** A 32-bit integer value that specifies the CRC hash of the resulting target block.

**data (variable):** A byte stream of **Lempel-Ziv Extended Delta (LZXD)** compressed differences to apply to the source Verbatim block that results in the target Verbatim block. For more details, see [\[MS-PATCH\]](#).

## 2.11 Compressed OAB Version 4 Details File and Compressed OAB Template File

Uncompressed Details and display template files can be very large due to the amount of information stored. In order to reduce the network traffic between the client and the server, the server transmits these files in a compressed form. A compressed file always starts with a **LZX\_HDR** structure followed by one or more **LZX\_BLK** structures. The **LZX\_HDR** structure contains a maximum block size field that is used to tell the client the maximum size of a block it can expect to have to read from the compressed file and the maximum size of a block it can expect to have to write to an output file. It is passed so that the client can pre-allocate buffers before attempting to decompress a file. Also included in the compressed Details or display template file is a length field that indicates what the size of the resulting decompressed file will be. It is provided to help the client allocate disk storage and determine whether the resulting output file size is correct.

Each **LZX\_BLK** structure contains a flag indicating whether the data field is compressed. If the size of a compressed block is larger than the source data, the server might choose to not compress the block and just pass it verbatim. A CRC hash of the expected decompressed output block is passed to the client to help it determine if the results of decompression are valid.

The following ABNF definition shows the format of a compressed OAB version 4 Details file and a compressed OAB Template file.

```
v4-compressed-file = LZX_HDR 1*LZX_BLK
```

### 2.11.1 LZX\_HDR

The **LZX\_HDR** structure contains the target file size value and versioning information to indicate that it is an OAB version 4 compressed file.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ulVersionHi																															
ulVersionLo																															
ulBlockMax																															
ulTargetSize																															

**ulVersionHi (4 bytes):** A 32-bit integer value that MUST be set to 0x00000003.

**ulVersionLo (4 bytes):** A 32-bit integer value that MUST be set to 0x00000001.

**ulBlockMax (4 bytes):** A 32-bit integer value that indicates in bytes the maximum block size that will be read from the source compressed input file or written to the target output file. This field is provided so that the client can pre-allocate required buffers. The server sets this value to 0x00040000. [<14>](#)

**ulTargetSize (4 bytes):** A 32-bit integer value that specifies the expected length of the resulting output target file.

### 2.11.2 LZX\_BLK

The **LZX\_BLK** structure splits the decompression process into more easily handled smaller-sized blocks.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ulFlags																															
ulCompSize																															
ulUncompSize																															
ulCRC																															

data (variable)
...

**ulFlags (4 bytes):** A 32-bit integer value that indicates whether the data field is compressed. MUST be set to either 0x00000000 to indicate that the data field is not compressed and can be written out directly to the target file, or 0x00000001 to indicate that the data field is a compressed Verbatim block, as specified in [\[MS-PATCH\]](#), and ought to be decompressed using **LZX** decompression first.

**ulCompSize (4 bytes):** A 32-bit integer value that specifies the size of the data field in bytes.

**ulUncompSize (4 bytes):** A 32-bit integer value that specifies the size in bytes of the output target block to be written to the output file.

**ulCRC (4 bytes):** A 32-bit integer value that specifies the CRC hash of the resulting target block.

**data (variable):** Either a raw data stream or a compressed byte stream, depending on the value of the **ulFlags** field. For more details, see [\[MS-PATCH\]](#).

### 3 Structure Examples

The examples in this section illustrate the data after it is downloaded to the client and decompressed when they have an OAB installed. The client can use the data in these files to retrieve user information when working offline. The structure of the data in each file is specified in section 2.

#### 3.1 Full OAB Version 2 Offline Address List

The following data show the contents of a sample OAB version 2 Browse file. All data in this section is shown in actual byte order.

```
OAB_HDR
  ulVersion      0a 00 00 00
  ulSerial       bd 32 79 d3
  ulTotRecs      02 00 00 00

B2_REC
  oRDN           d2 00 00 00
  oDetails       0c 00 00 00
  cbDetails      39 00
  bDispType      00

; In bObjType, the high order bit is the a bit of the B2_REC.
bObjType        06

  oSmt           8c 00 00 00
  oDispName      69 00 00 00
  oAlias         2c 00 00 00
  oLocation      00 00 00 00
  oSurname       00 00 00 00

B2_REC
  oRDN           68 00 00 00
  oDetails       45 00 00 00
  cbDetails      35 00
  bDispType      00

; The high order bit is the a bit of the B2_REC.
bObjType        06

  oSmt           b3 00 00 00
  oDispName      0c 00 00 00
  oAlias         8b 00 00 00
  oLocation      00 00 00 00
  oSurname       4e 00 00 00
```

The following data show the contents of a sample OAB version 2 ANR Index file.

```
OAB_HDR
  ulVersion      0a 00 00 00
  ulSerial       00 00 00 00
  ulTotRecs      05 00 00 00

; The ANR record for "Lisa Miller".
ANR_REC (offset 0x0000000C)
```

```

oLT      2c 00 00 00
oGT      4e 00 00 00
iBrowse  04 00 00 00
oPrev    69 00 00 00
oNext    8b 00 00 00
acKey    4c 69 73 61 20 4d 69 6c 6c 65 72 00

; The ANR record for "Administrator".
ANR_REC (offset 0x0000002C)

; 0 indicates there is no left child.
oLT      00 00 00 00
oGT      69 00 00 00

; The high order bit of the last byte is the alias field.
iBrowse  03 00 00 80

; 0 indicates this is the left-most record.
oPrev    00 00 00 00
oNext    69 00 00 00
acKey    41 64 6d 69 6e 69 73 74 72 61 74 6f 72 00

; The ANR record for "Miller".
ANR_REC (offset 0x0000004E)
oLT      8b 00 00 00

; 0 indicates there is no right child.
oGT      00 00 00 00
iBrowse  04 00 00 00
oPrev    8b 00 00 00

; 0 indicates this is the right most record.
oNext    00 00 00 00
acKey    4d 69 6c 6c 65 72 00

; The ANR record for "Administrator".
ANR_REC (offset 0x00000069)

; 0 indicates there is no left child.
oLT      00 00 00 00

; 0 indicates there is no right child.
oGT      00 00 00 00
iBrowse  03 00 00 00
oPrev    2c 00 00 00
oNext    0c 00 00 00
acKey    41 64 6d 69 6e 69 73 74 72 61 74 6f 72 00

; The ANR record for "LisaMiller".
ANR_REC (offset 0x0000008E)

; 0 indicates there is no left child.
oLT      00 00 00 00

; 0 indicates there is no right child.
oGT      00 00 00 00

; The high order bit of last byte is the alias field.
iBrowse  04 00 00 80

```

```
oPrev      0c 00 00 00
oNext      4e 00 00 00
acKey      4c 69 73 61 4d 69 6c 6c 65 72 00
```

The following code shows the contents of a sample OAB version 2 RDN Index file.

```
OAB_HDR
  ulVersion 0a 00 00 00
  ulSerial  00 00 00 00
  ulTotRecs 04 00 00 00
  oRoot     68 00 00 00
pdn-record (offset 0x00000010) '/o=example/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients'
  2f 6f 3d 65 78 61 6d 70 6c 65 2f 6f 75 3d 45 78
  63 68 61 6e 67 65 20 41 64 6d 69 6e 69 73 74 72
  61 74 69 76 65 20 47 72 6f 75 70 20 28 46 59 44
  49 42 4f 48 46 32 33 53 50 44 4c 54 29 2f 63 6e
  3d 52 65 63 69 70 69 65 6e 74 73 00
pdn-record (offset 0x0000005C) 'example.com'
  65 78 61 6d 70 6c 65 2e 63 6f 6d 00

; RDN record for "Lisa Miller".
RDN2_REC (offset 0x00000068)
  oLT      8c 00 00 00
  oGT      b3 00 00 00
  iBrowse  04 00 00 00
  oPrev    8c 00 00 00
  oNext    b3 00 00 00
  oParentDN 10 00 00 00
  acKey    4c 69 73 61 20 4d 69 6c 6c 65 72 00

; RDN record for "Administrator@".
RDN2_REC (offset 0x0000008C)
  oLT      d2 00 00 00
  oGT      00 00 00 00
  iBrowse  03 00 00 00
  oPrev    d2 00 00 00
  oNext    68 00 00 00
  oParentDN 5c 00 00 00
  acKey    41 64 6d 69 6e 69 73 74 72 61 74 6f 72 40 00

; RDN record for "LisaM@".
RDN2_REC (offset 0x000000B3)
  oLT      00 00 00 00
  oGT      00 00 00 00
  iBrowse  04 00 00 00
  oPrev    68 00 00 00
  oNext    00 00 00 00
  oParentDN 5c 00 00 00
  acKey    4c 69 73 61 4d 40 00

; RDN record for "Administrator".
RDN2_REC (offset 0x000000d2)
  oLT      00 00 00 00
  oGT      00 00 00 00
  iBrowse  03 00 00 00
  oPrev    00 00 00 00
```



```
oNext          8c 00 00 00
oParentDN      10 00 00 00
acKey          41 64 6d 69 6e 69 73 74 72 61 74 6f 72 00
```

The following data show the contents of a sample OAB version 2 Details file.

```
OAB_HDR
  ulVersion     07 00 00 00
  ulSerial      00 00 00 00
  ulTotRecs    00 00 00 00

; The Details-Record contains empty values for first 22 properties.
Details-Record (offset 0x0000000C)

  ; Empty binary property.
  00 00
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

  ; Empty ANSI properties.
  00 00 00 00 00 00 00

  ; 1 value for the multivalued PidTagAddressBookProxyAddresses property.
  01

  ; "SMTP:Administrator@example.com".
  53 4d 54 50 3a 41 64 6d 69 6e 69 73 74 72 61 74
  6f 72 40 65 78 61 6d 70 6c 65 2e 63
  6f 6d 00

  ; An empty multivalued binary property.
  00

  ; An empty multivalued binary property.
  00

Details-Record (offset 0x00000045)

  ; An empty binary property.
  00 00

  ; An empty ANSI property.
  00

  ; The value for the PidTagGivenName property is "Lisa".
  4c 69 73 61 00
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

  ; Empty ANSI properties.
  00 00 00

  ; 1 value for the multivalued PidTagAddressBookProxyAddresses property.
  01

  ; "SMTP:LisaM@example.com"
  53 4d 54 50 3a 4c 69 73 61 4d 40 65 78 61 6d
  70 6c 65 2e 63 6f 6d 00
```

```
; An empty multivalued binary property.  
00  
  
; An empty multivalued binary property.  
00
```

### 3.2 Full OAB Version 3 Offline Address List

The following data show the contents of a sample OAB version 3 Browse file. All data in this section is shown in actual byte order.

```
OAB_HDR  
  ulVersion  0e 00 00 00  
  ulSerial   bf 62 4f 0b  
  ulTotRecs  02 00 00 00  
  
B2_REC  
  oRDN       c2 00 00 00  
  oDetails   0c 00 00 00  
  cbDetails  e6 00  
  bDispType  00  
  bObjType   06  
  oSmt       7c 00 00 00  
  oDispName  69 00 00 00  
  oAlias     2c 00 00 00  
  oLocation  00 00 00 00  
  oSurname   00 00 00 00  
  
B2_REC  
  oRDN       5e 00 00 00  
  oDetails   f2 00 00 00  
  cbDetails  e2 00  
  bDispType  00  
  bObjType   06  
  oSmt       a3 00 00 00  
  oDispName  0c 00 00 00  
  oAlias     8b 00 00 00  
  oLocation  00 00 00 00  
  oSurname   4e 00 00 00
```

The following data show the contents of a sample OAB version 3 ANR Index file.

```
OAB_HDR  
  ulVersion  0e 00 00 00  
  ulSerial   00 00 00 00  
  ulTotRecs  05 00 00 00  
  
; ANR record for "Lisa Miller".  
ANR_REC (offset 0x0000000C)  
  oLT        2c 00 00 00  
  oGT        4e 00 00 00  
  iBrowse    04 00 00 00  
  oPrev      69 00 00 00  
  oNext      8b 00 00 00  
  acKey      4c 69 73 61 20 4d 69 6c 6c 65 72 00
```

```

; ANR record for "Administrator"
ANR_REC (offset 0x0000002C)

; 0 indicates there is no left child.
oLT      00 00 00 00
oGT      69 00 00 00

; The high order bit of last byte is the alias field.
iBrowse  03 00 00 80

; 0 indicates this is the left-most record.
oPrev    00 00 00 00
oNext    69 00 00 00
acKey    41 64 6d 69 6e 69 73 74 72 61 74 6f 72 00

; ANR Record for "Miller".
ANR_REC (offset 0x0000004E)
oLT      8b 00 00 00

; 0 indicates there is no right child.
oGT      00 00 00 00
iBrowse  04 00 00 00
oPrev    8b 00 00 00

; 0 indicates this is the right most record.
oNext    00 00 00 00
acKey    4d 69 6c 6c 65 72 00

; ANR record for "Administrator".
ANR_REC (offset 0x00000069)

; 0 indicates there is no left child.
oLT      00 00 00 00

; 0 indicates there is no right child.
oGT      00 00 00 00
iBrowse  03 00 00 00
oPrev    2c 00 00 00
oNext    0c 00 00 00
acKey    41 64 6d 69 6e 69 73 74 72 61 74 6f 72 00

; ANR record for "LisaM".
ANR_REC (offset 0x0000008E)

; 0 indicates there is no left child.
oLT      00 00 00 00

; 0 indicates there is no right child.
oGT      00 00 00 00

; The high order bit of last byte is the alias field.
iBrowse  04 00 00 80
oPrev    0c 00 00 00
oNext    4e 00 00 00
acKey    4c 69 73 61 4d 00

```

The following code shows the contents of a sample OAB version 3 RDN Index file.

```

OAB_HDR
  ulVersion  0e 00 00 00
  ulSerial   00 00 00 00
  ulTotRecs  04 00 00 00
  oRoot      5e 00 00 00
pdn-record (offset 0x00000010) '/o=First Organization/ou=First Administrative
  Group/cn=Recipients'
      2f 6f 3d 46 69 72 73 74 20 4f 72 67 61 6e 69 7a
      61 74 69 6f 6e 2f 6f 75 3d 46 69 72 73 74 20 41
      64 6d 69 6e 69 73 74 72 61 74 69 76 65 20 47 72
      6f 75 70 2f 63 6e 3d 52 65 63 69 70 69 65 6e 74
      73 00
pdn-record (offset 0x00000052) 'example.com'
      65 78 61 6d 70 6c 65 2e 63 6f 6d 00

; RDN record for "LisaM".
RDN2_REC (offset 0x0000005e)
  oLT      7c 00 00 00
  oGT      a3 00 00 00
  iBrowse  04 00 00 00
  oPrev    7c 00 00 00
  oNext    a3 00 00 00
  oParentDN 10 00 00 00
  acKey    4c 69 73 61 4d 00

; RDN record for "Administrator@".
RDN2_REC (offset 0x0000007c)
  oLT      c2 00 00 00
  oGT      00 00 00 00
  iBrowse  03 00 00 00
  oPrev    c2 00 00 00
  oNext    5e 00 00 00
  oParentDN 52 00 00 00
  acKey    41 64 6d 69 6e 69 73 74 72 61 74 6f 72 40 00

; RDN record for "LisaM@".
RDN2_REC (offset 0x000000a3)
  oLT      00 00 00 00
  oGT      00 00 00 00
  iBrowse  04 00 00 00
  oPrev    5e 00 00 00
  oNext    00 00 00 00
  oParentDN 52 00 00 00
  acKey    4c 69 73 61 4d 40 00

; RDN record for "Administrator".
RDN2_REC (offset 0x000000c2)
  oLT      00 00 00 00
  oGT      00 00 00 00
  iBrowse  03 00 00 00
  oPrev    00 00 00 00
  oNext    7c 00 00 00
  oParentDN 10 00 00 00
  acKey    41 64 6d 69 6e 69 73 74 72 61 74 6f 72 00

```

The following data show the contents of a sample OAB version 3 Details file.

```
OAB_HDR
```

```

ulVersion  07 00 00 00
ulSerial   00 00 00 00
ulTotRecs  00 00 00 00
Details-Record (offset 0x0000000C)

; Empty binary property.
00 00

; Empty UTF8 properties.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

; Empty multivalued UTF8 properties.
00 00

; Empty UTF8 properties.
00 00 00 00 00

; 2 values for the multivalued PidTagAddressBookProxyAddresses property.
02

; "SMTP:Administrator@example.com"
53 4d 54 50 3a 41 64 6d 69 6e 69 73 74 72 61 74 6f 72 40 65 78 61 6d 70 6c
65 2e 63 6f 6d 00

; "X400:c=US;a= ;p=Example;o=Exchange;s=Administrator;"
58 34 30 30 3a 63 3d 55 53 3b 61 3d 20 3b 70 3d 45 78 61 6d 70 6c 65 3b
6f 3d 45 78 63 68 61 6e 67 65 3b 73 3d 41 64 6d 69 6e 69 73 74 72 61 74 6f
72 3b 00

; Empty multivalued binary property.
00

; Empty multivalued binary property.
00

; "/o=First
Organization/ou=First Administrative Group/cn=Configuration/cn=Servers/cn=
EXCH-H-977/cn=Microsoft Private MDB' PidTagAddressBookHomeMessageDatabase"
2f 6f 3d 46 69 72 73 74 20 4f 72 67 61 6e 69 7a 61 74 69 6f 6e 2f 6f 75
3d 46 69 72 73 74 20 41 64 6d 69 6e 69 73 74 72 61 74 69 76 65 20 47 72 6f
75 70 2f 63 6e 3d 43 6f 6e 66 69 67 75 72 61 74 69 6f 6e 2f 63 6e 3d 53 65
72 76 65 72 73 2f 63 6e 3d 45 58 43 48 2d 48 2d 39 37 37 2f 63 6e 3d 4d 69
63 72 6f 73 6f 66 74 20 50 72 69 76 61 74 65 20 4d 44 42 00

; Empty properties.
00 00

```

```

Details-Record (offset 0x000000f2)

```

```

; Empty binary property.
00 00

; Empty ANSI property.
00

; "Lisa" PidTagGivenName.
4c 69 73 61 00
00 00 00 00 00 00 00 00 00 00 00 00

```

```

; Empty multivalued UTF8 properties.
00 00

; Empty UTF8 properties.
00 00 00 00 00

; 2 values for the multivalued PidTagAddressBookProxyAddresses property.
02

; "SMTP:LisaM@example.com"
53 4d 54 50 3a 4c 69 73 61 4d 40 65 78 61 6d 70 6c 65 2e 63 6f 6d 00

; "X400:c=US;a= ;p=Example;o=Exchange;s=Miller;g=Lisa;"
58 34 30 30 3a 63 3d 55 53 3b 61 3d 20 3b 70 3d 45 78 61 6d 70 6c 65 3b
6f 3d 45 78 63 68 61 6e 67 65 3b 73 3d 4d 69 6c 6c 65 72 3b 67 3d 4c 69 73
61 3b 00

; Empty multivalued binary property.
00

; Empty multivalued binary property.
00

; "/o=First
Organization/ou=First Administrative Group/cn=Configuration/cn=Servers/cn=
EXCH-H-977/cn=Microsoft Private MDB' PidTagAddressBookHomeMessageDatabase"
2f 6f 3d 46 69 72 73 74 20 4f 72 67 61 6e 69 7a 61 74 69 6f 6e 2f 6f 75
3d 46 69 72 73 74 20 41 64 6d 69 6e 69 73 74 72 61 74 69 76 65 20 47 72 6f
75 70 2f 63 6e 3d 43 6f 6e 66 69 67 75 72 61 74 69 6f 6e 2f 63 6e 3d 53 65
72 76 65 72 73 2f 63 6e 3d 45 58 43 48 2d 48 2d 39 37 37 2f 63 6e 3d 4d 69
63 72 6f 73 6f 66 74 20 50 72 69 76 61 74 65 20 4d 44 42 00
00 00

```

### 3.3 Full OAB Version 4 Details File

The following code shows the contents of a sample OAB version 4 Details file. All data in this section is shown in actual byte order.

```

OAB_HDR
ulVersion      20 00 00 00
ulSerial       f7 da c0 7f
ulTotRecs     02 00 00 00
OAB_META_DATA
cbSize         5c 00 00 00
rgHdrAtts
cAtts         04 00 00 00
rgProps [0]
ulPropID      1f 00 00 68
; Combination of fields a,b,and c
ulFlags       00 00 00 00
rgProps [1]
ulPropID      1e 00 04 68
ulFlags       00 00 00 00
rgProps [2]
ulPropID      03 00 01 68

```

```

    ulFlags      00 00 00 00
    rgProps [3]
      ulPropID   1e 00 02 68
      ulFlags    00 00 00 00
    rgOabAtts
      cAtts      06 00 00 00
      rgProps [0]
        ulPropID 1e 00 03 30

    ; Combination of fields a,b,and c
    ulFlags      02 00 00 00
    rgProps [1]
      ulPropID   1f 00 fe 39
      ulFlags    02 00 00 00
    rgProps [2]
      ulPropID   1f 00 01 30
      ulFlags    01 00 00 00
    rgProps [3]
      ulPropID   03 00 fe 0f
      ulFlags    00 00 00 00
    rgProps [4]
      ulPropID   03 00 00 39
      ulFlags    00 00 00 00
    rgProps [5]
      ulPropID   03 10 05 68
      ulFlags    00 00 00 00

OAB_V4_REC (Header Properties)
  cbSize      42 00 00 00
  PresenceArray f0
  Att [0] (Utf8) 5c 47 6c 6f 62 61 6c 20
                41 64 64 72 65 73 73 20
                4c 69 73 74 00
  Att [1] (String) 2f 00
  Att [2] (Integer) 06
  Att [3] (String) 64 34 66 32 34 34 61 38
                  2d 61 38 65 63 2d 34 34
                  32 61 2d 38 37 61 33 2d
                  35 32 33 36 66 38 32 63
                  61 62 64 63 00

OAB_V4_REC (Address book object 0)
  cbSize      80 00 00 00
  PresenceArray f8
  Att [0] (string) 2f 6f 3d 65 78 61 6d 70
                  6c 65 2f 6f 75 3d 45 78
                  63 68 61 6e 67 65 20 41
                  64 6d 69 6e 69 73 74 72
                  61 74 69 76 65 20 47 72
                  6f 75 70 20 28 46 59 44
                  49 42 4f 48 46 32 33 53
                  50 44 4c 54 29 2f 63 6e
                  3d 52 65 63 69 70 69 65
                  6e 74 73 2f 63 6e 3d 4c
                  69 73 61 20 4d 69 6c 6c
                  65 72 00
  Att [1] (Utf8) 4c 69 73 61 4d 40 65 78
                  61 6d 70 6c 65 2e 63 6f
                  6d 00

```

```
Att [2] (Utf8)    4c 69 73 61 20 4d 69 6c
                6c 65 72 00
Att [3] (Integer) 06
Att [4] (Integer) 00
```

```
OAB_V4_REC (Address book object 1)
cbSize      8c 00 00 00
PresenceArray f8
Att [0] (string) 2f 6f 3d 65 78 61 6d 70
                6c 65 2f 6f 75 3d 45 78
                63 68 61 6e 67 65 20 41
                64 6d 69 6e 69 73 74 72
                61 74 69 76 65 20 47 72
                6f 75 70 20 28 46 59 44
                49 42 4f 48 46 32 33 53
                50 44 4c 54 29 2f 63 6e
                3d 52 65 63 69 70 69 65
                6e 74 73 2f 63 6e 3d 41
                64 6d 69 6e 69 73 74 72
                61 74 6f 72 00
Att [1] (Utf8)  41 64 6d 69 6e 69 73 74
                72 61 74 6f 72 40 65 78
                61 6d 70 6c 65 2e 63 6f
                6d 00
Att [2] (Utf8)  41 64 6d 69 6e 69 73 74
                72 61 74 6f 72 00
Att [3] (Integer) 06
Att [4] (Integer) 00
```

Flat OAB header version 32, serial 7FC0DAF7, records 2

-----  
Header Attributes

Property Flags

cAtts = 4

```
0x6800001F: 0  PidTagOfflineAddressBookName
0x6804001E: 0  PidTagOfflineAddressBookDistinguishedName
0x68010003: 0  PidTagOfflineAddressBookSequence
0x6802001E: 0  PidTagOfflineAddressBookContainerGuid
```

-----  
OAB Attributes

Property Flags

cAtts = 6

```
0x3003001E: 2  PidTagEmailAddress
0x39FE001F: 2  PidTagSmtpAddress
0x3001001F: 1  PidTagDisplayName
0x0FFE0003: 0  PidTagObjectType
0x39000003: 0  PidTagDisplayType
0x68051003: 0  PidTagOfflineAddressBookTruncatedProperties
```

-----  
OAB Meta Data

```
0x6800001F: \Global Address List
0x6804001E: /
0x68010003: 6
0x6802001E: d4f244a8-a8ec-442a-87a3-5236f82cabdc
```

-----  
Record 0  
-----



0x3003001E: /o=example/ou=Exchange Administrative Group  
(FYDIBOHF23SPDLT)/cn=Recipients/cn=Lisa Miller  
0x39FE001F: LisaM@example.com  
0x3001001F: Lisa Miller  
0x0FFE0003: 6  
0x39000003: 0

-----  
Record 1  
-----

0x3003001E: /o=example/ou=Exchange Administrative Group  
(FYDIBOHF23SPDLT)/cn=Recipients/cn=Administrator  
0x39FE001F: Administrator@example.com  
0x3001001F: Administrator  
0x0FFE0003: 6  
0x39000003: 0

-----

Preliminary

## 4 Security

### 4.1 Security Considerations for Implementers

Data stored in OAB files contain personally identifiable information. Implementers have to ensure that only authorized individuals have access to the data.

### 4.2 Index of Security Parameters

None.

Preliminary

## 5 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 15 Technical Preview
- Microsoft® Office Outlook® 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Outlook® 2010
- Microsoft® Outlook® 15 Technical Preview

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:](#) OAB version 2 and OAB version 3 files are supported by Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010. OAB version 4 is supported by Exchange 2003, Exchange 2007, Exchange 2010, Exchange 15 Technical Preview, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and Outlook 15 Technical Preview. There are some differences in the default behavior of Exchange 2003, Exchange 2007, and Exchange 2010, as described in this section.

[<2> Section 2.3:](#) All OAB version 2 and OAB version 3 files, including the Browse file, are not generated by Exchange 15 Technical Preview and are not supported by Outlook 15 Technical Preview.

[<3> Section 2.4:](#) All OAB version 2 and OAB version 3 files, including the RDN Index file, are not generated by Exchange 15 Technical Preview and are not supported by Outlook 15 Technical Preview.

[<4> Section 2.4.1:](#) An Office Outlook 2003 client connecting with an Exchange 2003 server will generate 0x0000000E as the **ulVersion** instead of 0x0000000A in the uncompressed **RDN** Index file.

[<5> Section 2.5:](#) All OAB version 2 and OAB version 3 files, including the ANR Index file, are not generated by Exchange 15 Technical Preview and are not supported by Outlook 15 Technical Preview.

[<6> Section 2.6:](#) All OAB version 2 and OAB version 3 files, including the Details file, are not generated by Exchange 15 Technical Preview and are not supported by Outlook 15 Technical Preview.

<7> [Section 2.7](#): All OAB version 2 and OAB version 3 files, including the Changes file, are not generated by Exchange 15 Technical Preview and are not supported by Outlook 15 Technical Preview.

<8> [Section 2.8](#): All OAB version 2 and OAB version 3 files, including compressed files, are not generated by Exchange 15 Technical Preview and are not supported by Outlook 15 Technical Preview.

<9> [Section 2.9.2.1](#): The **PidTagAddressBookHierarchicalRootDepartment** ([\[MS-OXOABK\]](#) section 2.2.7.2) property is not supported by Exchange 2003, Exchange 2007, Office Outlook 2003, or Office Outlook 2007.

<10> [Section 2.9.2.2](#): Exchange 2003 does not populate the following [properties](#) by default: **PidTagAddressBookPhoneticDisplayName** ([\[MS-OXOABK\]](#) section 2.2.3.9), **PidTagAddressBookPhoneticSurname** ([\[MS-OXOABK\]](#) section 2.2.4.15), **PidTagAddressBookPhoneticGivenName** ([\[MS-OXOABK\]](#) section 2.2.4.14), **PidTagAddressBookPhoneticCompanyName** ([\[MS-OXOABK\]](#) section 2.2.4.16), **PidTagAddressBookPhoneticDepartmentName** ([\[MS-OXOABK\]](#) section 2.2.4.17), and **PidTagDisplayTypeEx** ([\[MS-OXOABK\]](#) section 2.2.3.12).

<11> [Section 2.9.2.2](#): Exchange 2003 and Exchange 2007 do not populate the following properties by default: **PidTagAddressBookSeniorityIndex** ([\[MS-OXOABK\]](#) section 2.2.3.24), **PidTagAddressBookHierarchicalIsHierarchicalGroup** ([\[MS-OXOABK\]](#) section 2.2.6.5), **PidTagAddressBookObjectGuid** ([\[MS-OXOABK\]](#) section 2.2.3.25), **PidTagAddressBookSenderHintTranslations** ([\[MS-OXOABK\]](#) section 2.2.3.26), **PidTagAddressBookDeliveryContentLength** ([\[MS-OXOABK\]](#) section 2.2.3.27), **PidTagAddressBookModerationEnabled** ([\[MS-OXOABK\]](#) section 2.2.3.28), **PidTagAddressBookDistributionListMemberCount** ([\[MS-OXOABK\]](#) section 2.2.3.29), **PidTagAddressBookDistributionListExternalMemberCount** ([\[MS-OXOABK\]](#) section 2.2.3.30), **PidTagAddressBookMember** ([\[MS-OXOABK\]](#) section 2.2.6.1), and **PidTagAddressBookIsMemberOfDistributionList** ([\[MS-OXOABK\]](#) section 2.2.5.3).

<12> [Section 2.9.2.2.1](#): The **PidTagOfflineAddressBookTruncatedProperties** property ([\[MS-OXPROPS\]](#) section 2.893) contains no properties by default in Exchange 2003 and Exchange 2007.

<13> [Section 2.9.4](#): **PidTagOfficeLocation** ([\[MS-OXOABK\]](#) section 2.2.4.8) is not in the ANR index in Exchange 2007.

<14> [Section 2.11.1](#): Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not require that this value is set to 0x00040000.

## 6 Change Tracking

This section identifies changes that were made to the [MS-OXOAB] protocol document between the October 2011 and January 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">2 Structures</a>	Updated the product behavior note to include support for Exchange 15 Technical Preview and Outlook 15 Technical Preview.	Y	Product behavior note updated.
<a href="#">2.3 Uncompressed OAB Version 2 and OAB Version 3 Browse File</a>	Added a product behavior note to describe support for the Browse file in Exchange 15 Technical Preview and Outlook 15 Technical Preview.	Y	New product behavior note added.
<a href="#">2.4 Uncompressed OAB Version 2 and OAB Version 3 RDN Index File</a>	Added a product behavior note to describe support for the RDN Index file in Exchange 15 Technical Preview and Outlook 15 Technical Preview.	Y	New product behavior note added.
<a href="#">2.5 Uncompressed OAB Version 2 and OAB Version 3 ANR Index File</a>	Added a product behavior note to describe support for the ANR Index file in Exchange 15 Technical Preview and Outlook 15 Technical Preview.	Y	New product behavior note added.
<a href="#">2.6 Uncompressed OAB Version 2 and OAB Version 3 Details File</a>	Added a product behavior note to describe support for the Details file in Exchange 15 Technical Preview and Outlook 15 Technical Preview.	Y	New product behavior note added.
<a href="#">2.7 Uncompressed OAB Version 2 and OAB Version 3 Changes File</a>	Added a product behavior note to describe support for the Changes file in Exchange 15 Technical Preview and Outlook 15 Technical Preview.	Y	New product behavior note added.
<a href="#">2.8</a>	Added a product behavior note to describe	Y	New product

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">Compressed OAB Version 2 or OAB Version 3 File</a>	support for compressed OAB version 2 and OAB version 3 files in Exchange 15 Technical Preview and Outlook 15 Technical Preview.		behavior note added.
<a href="#">5 Appendix A: Product Behavior</a>	Added Exchange 15 Technical Preview and Outlook 15 Technical Preview to the list of applicable product versions.	Y	Content updated.

Preliminary

## 7 Index

### A

[Applicability](#) 10

### C

[Change tracking](#) 77

[Common data types and fields](#) 11

### D

[Data types and fields - common](#) 11

Details

[common data types and fields](#) 11

### E

[Examples](#) 62

[Full OAB Version 2 Offline Address List](#) 62

[Full OAB Version 3 Offline Address List](#) 66

[Full OAB Version 4 Details File](#) 70

### F

[Fields - vendor-extensible](#) 10

[Full OAB Version 2 Offline Address List example](#) 62

[Full OAB Version 3 Offline Address List example](#) 66

[Full OAB Version 4 Details File example](#) 70

### G

[Glossary](#) 6

### I

[Implementer - security considerations](#) 74

[Index of security parameters](#) 74

[Informative references](#) 8

[Introduction](#) 6

### L

[Localization](#) 10

### N

[Normative references](#) 7

### O

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

### P

[Parameters - security index](#) 74

[Product behavior](#) 75

### R

[References](#) 7

[informative](#) 8

[normative](#) 7

[Relationship to protocols and other structures](#) 10

### S

Security

[implementer considerations](#) 74

[parameter index](#) 74

Structures

[overview](#) 11

### T

[Tracking changes](#) 77

### V

[Vendor-extensible fields](#) 10

[Versioning](#) 10