[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 ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.

- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.

- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.

- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

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

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability.
4/25/2008	0.2	Minor	Revised and updated property names and other technical content.
6/27/2008	1.0	Major	Initial Release.
8/6/2008	1.01	Minor	Revised and edited technical content.
9/3/2008	1.02	Minor	Revised and edited technical content.
10/1/2008	1.03	Minor	Revised and edited technical content.
12/3/2008	1.04	Minor	Updated IP notice.
2/4/2009	1.05	Minor	Revised and edited technical content.
3/4/2009	1.06	Minor	Revised and edited technical content.
4/10/2009	2.0	Major	Updated technical content for new product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	4.0.0	Major	Updated and revised the technical content.
2/10/2010	4.1.0	Minor	Updated the technical content.
5/5/2010	5.0.0	Major	Updated and revised the technical content.
8/4/2010	6.0	Major	Significantly changed the technical content.
11/3/2010	6.1	Minor	Clarified the meaning of the technical content.
3/18/2011	7.0	Major	Significantly changed the technical content.
8/5/2011	7.0	None	No changes to the meaning, language, or formatting of the technical content.
10/7/2011	7.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	8.0	Major	Significantly changed the technical content.
4/27/2012	9.0	Major	Significantly changed the technical content.
7/16/2012	9.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	9.1	Minor	Clarified the meaning of the technical content.
2/11/2013	9.2	Minor	Clarified the meaning of the technical content.
7/26/2013	9.3	Minor	Clarified the meaning of the technical content.
11/18/2013	9.3	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	9.3	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
4/30/2014	9.4	Minor	Clarified the meaning of the technical content.
7/31/2014	9.4	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	9.5	Minor	Clarified the meaning of the technical content.
3/16/2015	10.0	Major	Significantly changed the technical content.
5/26/2015	10.1	Minor	Clarified the meaning of the technical content.
9/14/2015	10.1	None	No changes to the meaning, language, or formatting of the technical content.
6/13/2016	10.1	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

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. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

Active Directory: A general-purpose network directory service. **Active Directory** also refers to the Windows implementation of a directory service. **Active Directory** stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in **Active Directory**. **Active Directory** is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS). [MS-ADTS] describes both forms. For more information, see [MS-AUTHSOD] section 1.1.1.5.2, Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Kerberos, and DNS.

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

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

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

address creation template: A template that describes how to present a dialog to a messaging user along with a script describing how to construct a new email address from the user's response.

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

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

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

ambiguous name resolution (ANR): A search algorithm that permits a client to search multiple naming-related attributes (2) on objects by way of a single clause of the form "(anr=value)" in a Lightweight Directory Access Protocol (LDAP) search filter. This permits a client to query for an object when the client possesses some identifying material related to the object but does not know which attribute of the object contains that identifying material.

American National Standards Institute (ANSI) character set: A character set defined by a code page approved by the American National Standards Institute (ANSI). The term "ANSI" as used to signify Windows code pages is a historical reference and a misnomer that persists in the Windows community. The source of this misnomer stems from the fact that the Windows code page 1252 was originally based on an ANSI draft, which became International Organization for Standardization (ISO) Standard 8859-1 [ISO/IEC-8859-1]. In Windows, the ANSI character set can be any of the following code pages: 1252, 1250, 1251, 1253, 1254, 1255, 1256, 1257, 1258, 874, 932, 936, 949, or 950. For example, "ANSI application" is usually a reference to a non-**Unicode** or code-page-based application. Therefore, "ANSI character set" is often misused to refer to one of the character sets defined by a Windows code page that can be used as an active system code page; for example, character sets defined by code page 1252 or character sets defined by code page 950. Windows is now based on **Unicode**, so the use of ANSI

character sets is strongly discouraged unless they are used to interoperate with legacy applications or legacy data.

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

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

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

distinguished name (DN): (1) A name that uniquely identifies an object by using the **relative distinguished name (RDN)** for the object, and the names of container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in a tree.

(2) In X.500, the globally unique name string that identifies an entity in an X.500 directory, as described in [X500]. The DN consists of several components and is used in X.509 certificates (2) to identify the subject and issuer principals, as described in [X509].

(3) In Lightweight Directory Access Protocol (LDAP), an LDAP Distinguished Name, as described in [RFC2251] section 4.1.3. The DN of an object is the DN of its parent, preceded by the RDN of the object. For example: CN=David Thompson, OU=Users, DC=Microsoft, DC=COM. For definitions of CN and OU, see [RFC2256] sections 5.4 and 5.12, respectively.

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

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a domain controller (DC) and host a member list that identifies all members of the domain, as well as optionally hosting the **Active Directory** service. The domain controller provides authentication (2) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

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

Lempel-Ziv Extended (LZX): An LZ77-based compression engine, as described in [UASDC], that is a universal lossless data compression algorithm. It performs no analysis on the data.

Lempel-Ziv Extended Delta (LZXD): A derivative of the Lempel-Ziv Extended (LZX) format with some modifications to facilitate efficient delta compression. Delta compression is a technique in which one set of data can be compressed within the context of a reference set of data that is supplied both to the compressor and decompressor. Delta compression is commonly used to encode updates to similar existing data sets so that the size of compressed data can be significantly reduced relative to ordinary non-delta compression techniques. Expanding a delta-compressed set of data requires that the exact same reference data be provided during decompression.

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

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

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

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

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

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

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

OAL data sequence number: An integer that is associated with **offline address list (OAL)** data that represents the generation number of this data. The value of the initial sequence number is "1". Each subsequent data generation process that produces a data set that is not identical to the previous data set is incremented by one.

Object Linking and Embedding (OLE): A technology for transferring and sharing information between applications by inserting a file or part of a file into a compound document. The inserted file can be either embedded or linked. See also embedded object and linked object.

offline: The condition of not being connected to or not being on a network or the Internet. Offline can also refer to a device, such as a printer that is not connected to a computer, and files that are stored on a computer that is not connected to or not on a network or the Internet.

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

offline address list (OAL): A portion of data that is in an **offline address book (OAB)** and is related to a single **address list**.

parent distinguished name (PDN): A **distinguished name (DN) (1)** of an object that is the next immediate object closer to the root of a tree of **relative distinguished names (RDNs)**.

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

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

public folder: A Folder object that is stored in a location that is publicly available.

recipient: An entity that is in an **address list**, can receive email messages, and contains a set of attributes (1). Each attribute has a set of associated values.

relative distinguished name (RDN): The name of an object relative to its parent. This is the leftmost attribute-value pair in the **distinguished name (DN)** of an object. For example, in the **DN** "cn=Peter Houston, ou=NTDEV, dc=microsoft, dc=com", the **RDN** is "cn=Peter Houston". For more information, see [RFC2251].

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

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

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

X.509: An ITU-T standard for public key infrastructure subsequently adapted by the IETF, as specified in [RFC3280].

X500 DN: A distinguished name (DN), in Teletex form, of an object that is in an **address book**. An X500 DN can be more limited in the size and number of relative distinguished names (RDNs) than a full DN.

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

1.2   References

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

1.2.1   Normative References

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

[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

[ITU-X.690-2008] ITU-T, "X.690 : Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", Recommendation X.690, November 2008, http://www.itu.int/rec/T-REC-X.690-200811-I/en

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

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

[MS-MCI] Microsoft Corporation, "Microsoft ZIP (MSZIP) Compression and Decompression Data Structure".

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

[MS-OXLDAP] Microsoft Corporation, "Lightweight Directory Access Protocol (LDAP) Version 3 Extensions".

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

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

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

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

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

[MS-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.rfc-editor.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-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** information. To organize the contents of an address book, the server can divide recipients 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.

Browse file

OAB_HDR

.
.
.

oRDN
oDetails
chDetails | bDt |bOt
oSTMP
oDisplayName
oAlias
oLocation
oSurname

.
.
.

RDN Index file

OAB_HDR
oRoot

Pdn1
Pdn2
.
.
Pdn n

.
.
.

oLT
oGT
iBrowse
oNext
oPrev
oParent
acKey[1..64]

.
.
.

One or mo
Changes fil

OAB_HDR
iBrowse
cbData
displayName
oPDN
oRDN

.
.
.

iBrowse
cbData
displayName
oPDN
oRDN

.
.
.

ANR Index file

OAB_HDR

.
.
.

oLT
oGT
iBrowse
oNext
oPrev
acKey[1..64]

.
.
.

Details file

OAB_HDR

.
.
.

rgDetails[0-65536]

.
.
.

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.

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

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

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 (x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + 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
```
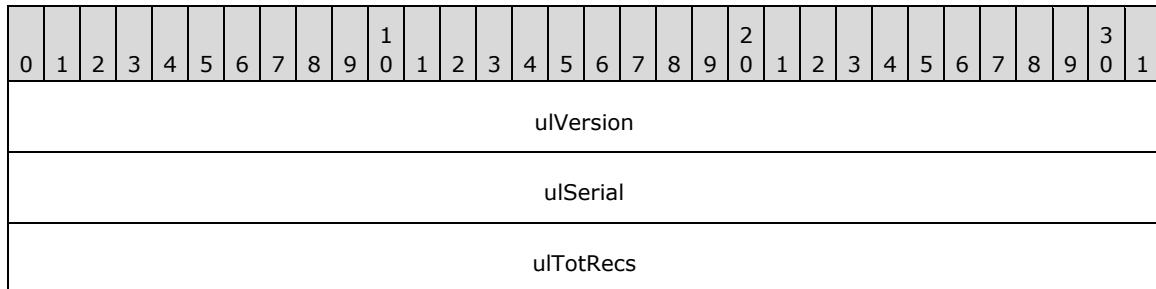
```
                           forum-template agent-template
                           organization-template
                           private-distributionlist-template
                           remote-mailuser-template
                           NAMES_STRUCT
                           address-templates 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 the subsections of 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	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
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): A 32-bit hexadecimal string. This field MUST be set to 0x00000000 when sent and MUST be ignored when received.

ulTotRecs (4 bytes): A 32-bit unsigned integer. 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	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
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] section 2.2.2.1. 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, the offset of stored strings is specified by the **ulString** field, as specified in [MS-OXOABKT] section 2.2.2.1.3, in the template table.

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	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
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

The Browse file<2> contains one fixed size record per user, with members that point to offsets in the RDN Index file, the ANR Index file, 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 **Simple Mail Transfer Protocol (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** 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 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	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
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 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 2.2.1.1.

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
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 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, as specified in [MS-OXNSPI] section 2.2.1.3.
0x01	DT_DISTLIST, as specified in [MS-OXNSPI] section 2.2.1.3.
0x02	DT_FORUM, as specified in [MS-OXNSPI] section 2.2.1.3.
0x03	DT_AGENT, as specified in [MS-OXNSPI] section 2.2.1.3.
0x06	DT_REMOTE_MAILUSER, as specified in [MS-OXNSPI] section 2.2.1.3.

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.

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, as specified in [MS-OXOABK] section 2.2.3.10.
0x06	MAPI-MAILUSER, as specified in [MS-OXOABK] section 2.2.3.10.
0x08	MAPI-DISTLIST, as specified in [MS-OXOABK] section 2.2.3.10.

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 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 value of each OAB record into a single value. The following steps produce the RDN 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 value of the OAB record. The RDN 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 value with null characters (append null characters to the end of the RDN 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 hash value.

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

 The client uses the RDN Index file<3> for primary key lookups based on the X500 DN and 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 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 of the X500 DNs and the local-part of SMTP addresses in the key field of the records in the RDN index section.

For example, given the following **distinguished name (DN) (3)** value, /o=Adventure-Works/ou=New York/CN=recipients/CN=JohnH, the RDN object is /CN=JohnH and the PDN is /o=Adventure-Works/ou=New York/CN=recipients/. The key field of the RDN index, also known as the RDN value, is simply JohnH.

Records in the RDN 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 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 records in the RDN Index file, and it contains a hash value for consistency checks and the offset of the root RDN record.

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
ulVersion																														
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.<4> This field SHOULD be set to 0x0000000E for uncompressed version 3 RDN Index files.

ulSerial (4 bytes): A 32-bit hexadecimal string that specifies the hash of the RDN 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 **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 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 records.

2.4.2 RDN2_REC

 Each **RDN2_REC** structure corresponds to a node in the RDN 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	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
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 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 records, "/CN=" MUST be removed from the final RDN 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 (3) 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

The ANR Index file<5> enables 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 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	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
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): A 32-bit hexadecimal string that specifies the hash of the RDN 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 **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	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
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

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
                           ; The teletex-char element is defined
                           ; in the ABNF definition of section 2.1.
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
PidTagUserCertificate ([MS-OXOABK] section 2.2.4.34)	0x3A22010 2	**PtypBinary** ([MS-OXCDATA] section	The **user-certificate** property contains an ASN.1 authentication certificate for a messaging user.

Property tag name	Property tag	Property type	Description
		2.11.1)	This property is deprecated and SHOULD NOT be set.
PidTagBusinessTelephoneNumber ([MS-OXOABK] section 2.2.4.21)	0x3A08001F	**PtypString*** ([MS-OXCDATA] section 2.11.1)	The **business-telephone** property contains the primary telephone number of the place of business of the Address Book object.
PidTagGivenName ([MS-OXOABK] section 2.2.4.2)	0x3A06001F	**PtypString***	The **given-name** property contains the given name of the Address Book object.
PidTagInitials ([MS-OXOABK] section 2.2.4.3)	0x3A0A001F	**PtypString***	The **initials** property contains the initials for parts of the full name of the Address Book object.
PidTagStreetAddress ([MS-OXOABK] section 2.2.4.14)	0x3A29001F	**PtypString***	The **street-address** property contains the street address of the Address Book object.
PidTagLocality ([MS-OXOABK] section 2.2.4.16)	0x3A27001F	**PtypString***	The **city-locality** property contains the name of the locality of the Address Book object, such as the town or city.
PidTagStateOrProvince ([MS-OXOABK] section 2.2.4.17)	0x3A28001F	**PtypString***	The **state-province** property contains the name of the state or province where the Address Book object is located.
PidTagPostalCode ([MS-OXOABK] section 2.2.4.18)	0x3A2A001F	**PtypString***	The **postal-code** property contains the postal code of the Address Book object.
PidTagCountry ([MS-OXOABK] section 2.2.4.19)	0x3A26001F	**PtypString***	The **country-region** property contains the name of the country or region where the Address Book object is located.
PidTagTitle ([MS-OXOABK] section 2.2.4.4)	0x3A17001F	**PtypString***	The **title** property contains the job title of the Address Book object.
PidTagCompanyName ([MS-OXOABK] section 2.2.4.7)	0x3A16001F	**PtypString***	The **company-name** property contains the name of the company that employs the Address Book object.
PidTagAssistant ([MS-OXOABK] section 2.2.4.8)	0x3A30001F	**PtypString***	The **assistant-name** property contains the name of the administrative assistant for the Address Book object.
PidTagDepartmentName ([MS-OXOABK] section 2.2.4.6)	0x3A18001F	**PtypString***	The **department-name** property contains the department name in which the Address Book object works.
null	0x3A08001F	**PtypString***	Duplicate **PidTagBusinessTelephoneNumber** property. The client MUST ignore this value.

Property tag name	Property tag	Property type	Description
PidTagHomeTelephoneNumber ([MS-OXOABK] section 2.2.4.22)	0x3A09001F	**PtypString***	The **home-telephone** property contains the primary home telephone number for the Address Book object.
PidTagBusiness2TelephoneNumber ([MS-OXOABK] section 2.2.4.23)	0x3A1B001F	**PtypString***	The **business2-telephone-mv** property contains a secondary business telephone number for the Address Book object. MUST be set to null in an OAB Version 2 Details file.
PidTagHome2TelephoneNumber ([MS-OXOABK] section 2.2.4.25)	0x3A2F001F	**PtypString***	The **home2-telephone-mv** property contains a secondary home telephone number for the Address Book object. MUST be set to null in an OAB Version 2 Details file.
PidTagPrimaryFaxNumber ([MS-OXOABK] section 2.2.4.29)	0x3A23001F	**PtypString***	The **primary-fax** property contains the telephone number for the fax machine of the Address Book object.
PidTagMobileTelephoneNumber ([MS-OXOABK] section 2.2.4.27)	0x3A1C001F	**PtypString***	The **mobile-telephone** property contains the mobile telephone number of the Address Book object.
PidTagAssistantTelephoneNumber ([MS-OXOABK] section 2.2.4.31)	0x3A2E001F	**PtypString***	The **assistant-telephone** property contains the telephone number for the administrative assistant of the Address Book object.
PidTagPagerTelephoneNumber ([MS-OXOABK] section 2.2.4.28)	0x3A21001F	**PtypString***	The **pager-telephone** property contains the pager telephone number of the Address Book object.
PidTagComment ([MS-OXOABK] section 2.2.3.31)	0x3004001F	**PtypString***	The **comment** property contains a description of the purpose or content of an object.
PidTagAddressBookProxyAddresses ([MS-OXOABK] section 2.2.3.23)	0x800F101F	**PtypMultipleString*** ([MS-OXCDATA] section 2.11.1)	The **proxy-addresses** property contains a list of e-mail addresses that this Address Book object is known by. Each value MUST begin with an e-mail **address type** followed by a colon character then followed by the address value.
PidTagUserX509Certificate ([MS-OXOABK] section 2.2.4.36)	0x3A701102	**PtypMultipleBinary** ([MS-OXCDATA] section 2.11.1)	The **smime-certs** property contains SMIME certificates that the server formats as PKCS-7 encodings. For more details, see [RFC2315].
PidTagAddressBookX509Certifica	0x8C6A110	**PtypMultipleBinar**	The **X509-certs** property contains **X.509** certificates encoded by

Property tag name	Property tag	Property type	Description
te ([MS-OXOABK] section 2.2.4.35)	2	**y**	using ASN.1 ([ITU-X.690-2008]). For more details, see [RFC3280].

*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.

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

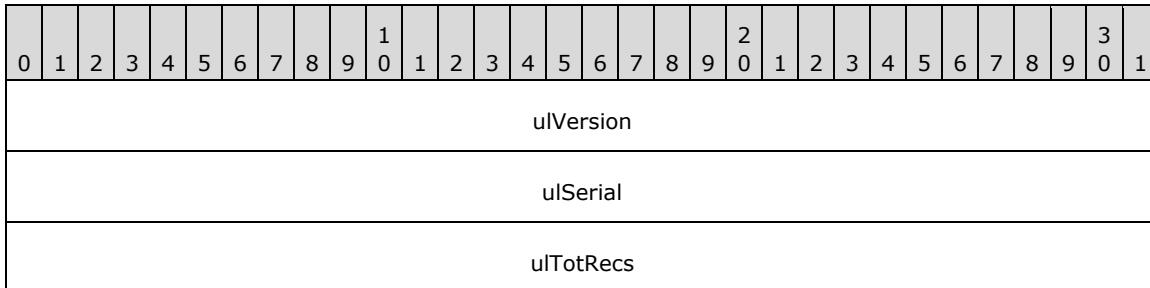| Property tag name	Property tag	Property type	Description
PidTagUserCertificate	0x3A220102	**PtypBinary**	The **user-certificate** property contains an ASN.1 authentication certificate for a messaging user. This property is deprecated and SHOULD be set to a null entry.
PidTagBusinessTelephoneNumber	0x3A08001F	**PtypString**	The **business-telephone** property contains the primary telephone number of the place of business of the Address Book object.
PidTagGivenName	0x3A06001F	**PtypString**	The **given-name** property contains the given name of the Address Book object.
PidTagInitials	0x3A0A001F	**PtypString**	The **initials** property contains the initials for parts of the full name of the Address Book object.
PidTagStreetAddress	0x3A29001F	**PtypString**	The **street-address** property contains the street address of the Address Book object.
PidTagLocality	0x3A27001F	**PtypString**	The **city-locality** property contains the name of the locality of the Address Book object, such as the town or city.
PidTagStateOrProvince	0x3A28001F	**PtypString**	The **state-province** property contains the name of the state or province where the Address Book object is located.
PidTagPostalCode	0x3A2A001F	**PtypString**	The **postal-code** property contains the

Property tag name	Property tag	Property type	Description
			postal code of the Address Book object.
PidTagCountry	0x3A26001F	**PtypString**	The **country-region** property contains the name of the country or region where the Address Book object is located.
PidTagTitle	0x3A17001F	**PtypString**	The **title** property contains the job title of the Address Book object.
PidTagCompanyName	0x3A16001F	**PtypString**	The **company-name** property contains the name of the company that employs the Address Book object.
PidTagAssistant	0x3A30001F	**PtypString**	The **assistant-name** property contains the name of the administrative assistant for the Address Book object.
PidTagDepartmentName	0x3A18001F	**PtypString**	The **department-name** property contains the department name in which the Address Book object works.
PidTagAddressBookTargetAddress ([MS-OXOABK] section 2.2.3.22)	0x8011001F	**PtypString**	The **target-address** property contains the destination address for this object.
PidTagHomeTelephoneNumber	0x3A09001F	**PtypString**	The **home-telephone** property contains the primary home telephone number for the Address Book object.
PidTagBusiness2TelephoneNumbers	0x3A1B101F	**PtypMultipleString**	The **business2-telephone** property contains secondary business telephone numbers for the Address Book object.
PidTagHome2TelephoneNumbers	0x3A2F101F	**PtypMultipleString**	The **home2-telephone** property contains secondary home telephone numbers for the Address Book object.
PidTagPrimaryFaxNumber	0x3A23001F	**PtypString**	The **primary-fax**

Property tag name	Property tag	Property type	Description
			property contains the telephone number for the fax machine of the Address Book object.
PidTagMobileTelephoneNumber	0x3A1C001F	**PtypString**	The **mobile-telephone** property contains the mobile telephone number of the Address Book object.
PidTagAssistantTelephoneNumber	0x3A2E001F	**PtypString**	The **assistant-telephone** property contains the telephone number for the administrative assistant of the Address Book object.
PidTagPagerTelephoneNumber	0x3A21001F	**PtypString**	The **pager-telephone** property contains the pager telephone number of the Address Book object.
PidTagComment	0x3004001F	**PtypString**	The **comment** property contains a description of the purpose or content of an object.
PidTagAddressBookProxyAddresses	0x800F101F	**PtypMultipleString**	The **proxy-addresses** property contains a list of e-mail addresses that this Address Book object is known by. Each value MUST begin with an e-mail address type followed by a colon character then followed by the address value.
PidTagUserX509Certificate	0x3A701102	**PtypMultipleBinary**	The **smime-certs** property contains SMIME certificates formatted as PKCS-7 encodings. For more details, see [RFC2315].
PidTagAddressBookX509Certificate	0x8C6A1102	**PtypMultipleBinary**	The **X509-certs** property contains X.509 certificates encoded by using ASN.1 ([ITU-X.690-2008]). For more details, see

Property tag name	Property tag	Property type	Description
			[RFC3280].
PidTagAddressBookHomeMessageDatabase ([MS-OXOABK] section 2.2.4.37)	0x8006001F	**PtypString**	The **home-mdb** property contains the DN (3) of the **message store** for this **mailbox**. This property value is not subject to truncation.
PidTagAddressBookManager ([MS-OXOABK] section 2.2.5.1)	0x8005000D	**PtypObject** ([MS-OXCDATA] section 2.11.1)	The **manager** property contains the DN (3) of the manager of the recipient. The user object for the manager contains a directReports property, as specified by [MS-OXLDAP] section 2.2.2.2, that contains references to all user objects that have their manager property set to this DN (3).
PidTagAddressBookDisplayNamePrintable ([MS-OXOABK] section 2.2.3.7)	0x39FF001F	**PtypString**	The **display-name-printable** property contains the printable string version of the display name. MUST contain Teletex characters only, as specified by the teletex-char rule in section 2.1.

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	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
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): A 32-bit hexadecimal string that specifies the hash of the RDN 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. 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

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
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" / "|"

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)
                        ; The byte-count element is defined in the
                        ; ABNF definition of section 2.6.
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 [MS-OXOABK] section 2.2.3.10.

MAPI-FOLDER          =     %x03
MAPI-MAILUSER        =     %x06
MAPI-DISTLIST        =     %x08
```
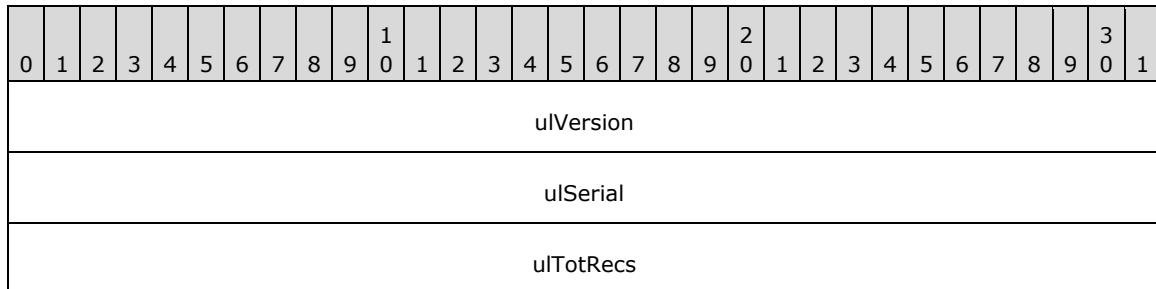
2.7.1 OAB_HDR

 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	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
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. 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 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	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
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 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 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 000), or this field 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 000), or this field 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 000), or this field 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 000), or this field 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 000), or this field 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 000), or this field 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 000), or this field 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	**PidTagDisplayName** ([MS-OXOABK] section 2.2.3.1)	**PtypString*** ([MS-OXCDATA] section 2.11.1)	Variable	The **display-name** property contains the display name for a given Address Book object. This property is present only if the **type** field of the **CHG_REC** structure, as specified in section 2.7.2, is set to 001.
2	ParentDNOffset	**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	4 bytes	The **parent-dn-offset** property contains the offset to the PDN in the RDN Index file. This property is present only if the **type** field of the **CHG_REC** structure, as specified in section 2.7.2, is set to 001.
3	RDNRecordKey	**PtypString***	Variable	The **rdn** property uniquely identifies the RDN in the RDN Index file. This property is present only if the **type** field of the **CHG_REC** structure is set to 001.

Index Number	Property tag name	Property type	Property size	Description
				This is a null-terminated string. The maximum size of this property is 68 bytes.
4	ParentDNOffset ForSMTP	**PtypInteger32**	4 bytes	The **domain-name-offset** property contains the offset of the Parent DN (3) SMTP address entry in the RDN Index file. This property is present only if the **type** field of the **CHG_REC** structure is set to 000.
5	**PidTagSmtpAddress** ([MS-OXOABK] section 2.2.3.21)	**PtypString8** ([MS-OXCDATA] section 2.11.1)	Variable	The **local-portion** property contains the SMTP mailing address of the sender. This property is present only if the **type** field of the **CHG_REC** structure is set to 000.
6	**PidTagAccount** ([MS-OXOABK] section 2.2.3.20)	**PtypString8**	Variable	The **alias** property contains the account name for the Address Book object. This property is present only if the **type** field of the **CHG_REC** structure is set to 000.
7	**PidTagOfficeLocation** ([MS-OXOABK] section 2.2.4.5)	**PtypString***	Variable	The **location** property contains the office location of the Address Book object.
8	**PidTagSurname** ([MS-OXOABK] section 2.2.4.1)	**PtypString***	Variable	The **surname** property contains the family name of the Address Book object.
9	DetailsRecordSize	**PtypInteger16** ([MS-OXCDATA] section 2.11.1)	2 bytes	The **details** property identifies the size of the modified user record, including the **DetailsRecordSize** and the terminating null character. This property is present only if the **type** field of the **CHG_REC** structure 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 **address-book-object-record**. This property is present only if the **type** field of the **CHG_REC** structure is set to 000 or 001.
11	**PidTagDisplayType** ([MS-OXOABK] section 2.2.3.11)	Integer	1 byte	The **display-type** property contains a value that is used to associate an icon with a particular row of a table.
12	**PidTagObjectType** ([MS-OXOABK] section 2.2.3.10)	Integer	1 byte	The **object-type** property contains the type of an object. 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 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	**PidTagDisplayName**	**PtypString***	Variable	The **display-name** property contains the display name for a given Address Book object encoded as UTF8. This property is present only if the **type** field of the **CHG_REC** structure, as specified in section 2.7.2, is set to 001.
2	ParentDNOffset	**PtypInteger32**	4 bytes	The **parent-dn-offset** property contains the offset to the PDN in the RDN Index file. This property is present only if the **type** field of the **CHG_REC** structure is set to 001.
3	RDNRecordKey	**PtypString***	Variable	The **rdn** property uniquely identifies the RDN in the RDN Index file. This property is present only if the **type** field of the **CHG_REC** structure is set to 001. This is a null-terminated string. The maximum size of this property is 68 bytes.
4	ParentDNOffset ForSMTP	**PtypInteger32**	4 bytes	The **domain-name-offset** property contains the offset of the Parent DN (3) SMTP address entry in the RDN Index file. This property is present only if the **type** field of the **CHG_REC** structure is set to 000.
5	**PidTagSmtpAddress**	**PtypString8**	Variable	The **local-portion** property contains the SMTP mailing address of the sender encoded as UTF8. This property is present only if the **type** field of the **CHG_REC** structure is set to 000.
6	**PidTagAccount**	**PtypString8**	Variable	The **alias** property contains the account name for the Address Book object encoded as UTF8. This property is present only if the **type** field of the **CHG_REC** structure is set to 000.
7	**PidTagOfficeLocation**	**PtypString***	Variable	The **location** property contains the office location of the Address Book object encoded as UTF8.
8	**PidTagSurname**	**PtypString***	Variable	The **surname** property contains the family name of the Address Book object encoded as UTF8.
9	DetailsRecordSize	**PtypInteger16**	2 bytes	The **details** property identifies the size of the modified user record, including the

Index Number	Property tag name	Property type	Property size	Description
				DetailsRecordSize and the terminating null character.
				This property is present only if the **type** field of the **CHG_REC** structure 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 **address-book-object-record**. This property is present only if the **type** field of the **CHG_REC** structure is set to 000 or 001.
11	**PidTagDisplayType**	1 byte integer	1 byte	The **display-type** property contains a value that is used to associate an icon with a particular row of a table.
12	**PidTagObjectType**	1 byte integer	1 byte	The **object-type** property contains the type of an object.
				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.

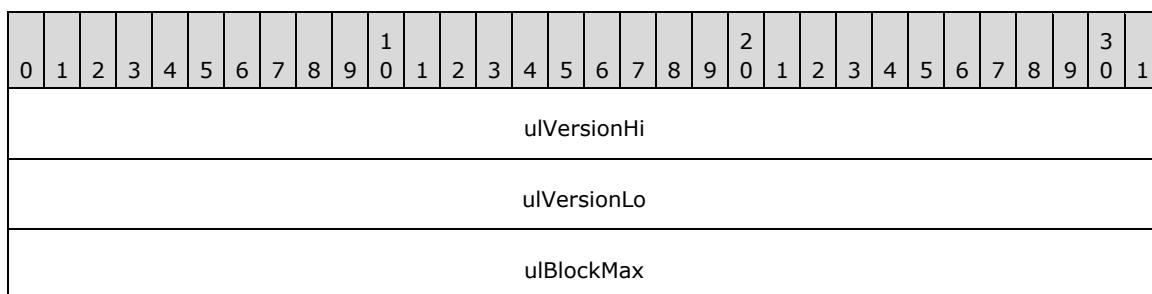## 2.8   Compressed OAB Version 2 or OAB Version 3 File

 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																															

										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

ulTargetSize

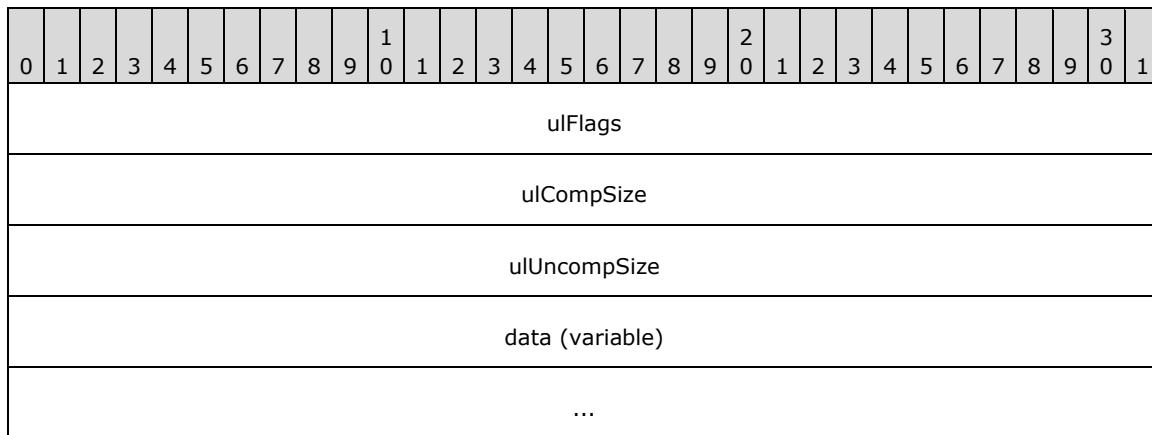**ulVersionHi (4 bytes):** The high part of the file version information. This field MUST be set to 0x00000002.

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	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
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 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 (2), 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
```
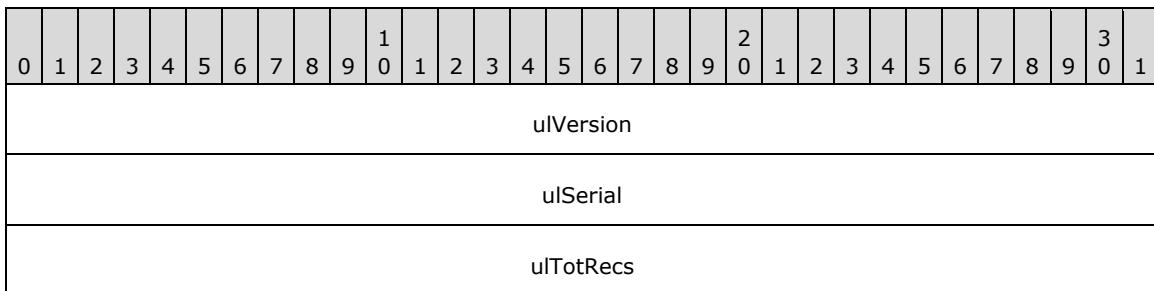
2.9.1 OAB_HDR

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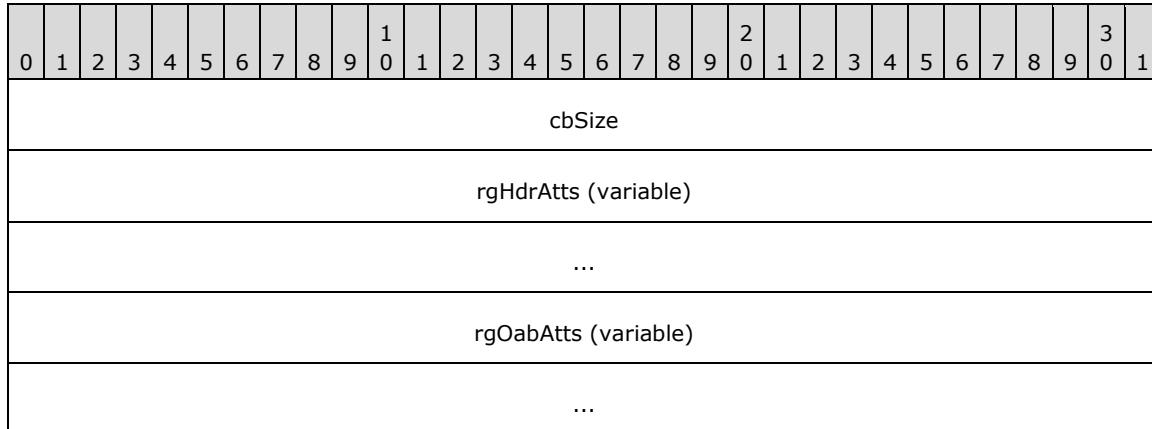
0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
ulVersion																																
ulSerial																																
ulTotRecs																																

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	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
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	**PidTagOfflineAddressBookName** (section 2.12.3)	0x6800001F	**PtypString** ([MS-OXCDATA] section 2.11.1)	Display name of the address list. Can change between generation versions of the same address list.

Index Number	Property tag name	Property tag	Property type	Description
2	**PidTagOfflineAddressBookDistinguishedName** (section 2.12.2)	0x6804001E	**PtypString8** ([MS-OXCDATA] 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 2.1.
3	**PidTagOfflineAddressBookSequence** (section 2.12.4)	0x68010003	**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	The sequence number of the OAB. This number increases by one between generation versions of the same address list.
4	**PidTagOfflineAddressBookContainerGuid** (section 2.12.1)	0x6802001E	**PtypString8**	A string formatted GUID that represents the address list container object. This value never changes between generation versions of the same address list. This value is formatted as "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx". MUST contain Teletex characters only, as specified by the non-space-teletex rule in section 2.1

The property in the following table is an optional property in the **rgHdrAtts** table.<9>

Property tag name	Property tag	Property type	Description
PidTagAddressBookHierarchicalRootDepartment ([MS-OXOABK] section 2.2.7.2)	0x8C98001E	**PtypString8**	DN for the root **departmental group** in the department hierarchy for the organization. The DN (3) 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 2.1.

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 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 2.2.4.5)

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 2.2.4.21)

PidTagInitials ([MS-OXOABK] section 2.2.4.3)

PidTagStreetAddress ([MS-OXOABK] section 2.2.4.14)

PidTagLocality ([MS-OXOABK] section 2.2.4.16)

PidTagStateOrProvince ([MS-OXOABK] section 2.2.4.17)

PidTagPostalCode ([MS-OXOABK] section 2.2.4.18)

PidTagCountry ([MS-OXOABK] section 2.2.4.19)

PidTagTitle ([MS-OXOABK] section 2.2.4.4)

PidTagCompanyName ([MS-OXOABK] section 2.2.4.7)

PidTagAssistant ([MS-OXOABK] section 2.2.4.8)

PidTagDepartmentName ([MS-OXOABK] section 2.2.4.6)

PidTagAddressBookTargetAddress ([MS-OXOABK] section 2.2.3.22)

PidTagHomeTelephoneNumber ([MS-OXOABK] section 2.2.4.22)

PidTagBusiness2TelephoneNumbers ([MS-OXOABK] section 2.2.4.24)

PidTagHome2TelephoneNumbers ([MS-OXOABK] section 2.2.4.26)

PidTagPrimaryFaxNumber ([MS-OXOABK] section 2.2.4.29)

PidTagMobileTelephoneNumber ([MS-OXOABK] section 2.2.4.27)

PidTagAssistantTelephoneNumber ([MS-OXOABK] section 2.2.4.31)

PidTagPagerTelephoneNumber ([MS-OXOABK] section 2.2.4.28)

PidTagComment ([MS-OXOABK] section 2.2.3.31)

PidTagUserCertificate ([MS-OXOABK] section 2.2.4.34)

PidTagUserX509Certificate ([MS-OXOABK] section 2.2.4.36)

PidTagAddressBookX509Certificate ([MS-OXOABK] section 2.2.4.35)

PidTagAddressBookHomeMessageDatabase ([MS-OXOABK] section 2.2.4.37)

PidTagAddressBookDisplayNamePrintable ([MS-OXPROPS] section 2.505)

The following table lists the properties that the server SHOULD<10><11> populate by default on an Address Book object. All of these properties except **PidTagEmailAddress** ([MS-OXOABK] section 2.2.3.14) and **PidTagSmtpAddress** ([MS-OXOABK] section 2.2.3.21) 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	**PidTagEmailAddress**	0x3003 001E	**PtypString8** ([MS-OXCDATA] section 2.11.1)	Contains the X500 DN. MUST contain Teletex characters only, as specified by the non-space-teletex rule in section 2.1.
2	**PidTagSmtpAddress**	0x39fe 001F	**PtypString** ([MS-OXCDATA] section 2.11.1)	Contains the SMTP mailing address of the sender.
3	**PidTagDisplayName**	0x3001 001F	**PtypString**	Contains the display name for a given Address Book object.
4	**PidTagAddressBookPhoneticDisplayName** ([MS-OXOABK] section 2.2.3.9)	0x8C92 001F	**PtypString**	Contains the phonetic display name of an object.
5	**PidTagAccount**	0x3A00	**PtypString**	Contains the account name for the

Index Number	Property tag name	Property tag	Property type	Description
		001F		Address Book object.
6	**PidTagSurname**	0x3A11 001F	**PtypString**	Contains the family name of the Address Book object.
7	**PidTagAddressBookPhoneticSurname** ([MS-OXOABK] section 2.2.4.11)	0x8C8F 001F	**PtypString**	Contains the phonetic spelling of the surname.
8	**PidTagGivenName**	0x3A06 001F	**PtypString**	Contains the given name of the Address Book object.
9	**PidTagAddressBookPhoneticGivenName** ([MS-OXOABK] section 2.2.4.10)	0x8C8E 001F	**PtypString**	Contains the phonetic given name of the Address Book object.
10	**PidTagAddressBookProxyAddresses**	0x800f 101F	**PtypMultipleString** ([MS-OXCDATA] 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;.
11	**PidTagOfficeLocation**	0x3A19 001F	**PtypString**	Contains the office location of the Address Book object.
12	**PidTagDisplayType**	0x3900 0003	**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	Contains a value that is used to associate an icon with a particular row of a table.
13	**PidTagObjectType**	0x0FFE 0003	**PtypInteger32**	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	**PidTagSendRichInfo**	0x3A40 000B	**PtypBoolean** ([MS-OXCDATA] section 2.11.1)	Contains "TRUE" if the entry can receive all message content, including RTF and OLE objects; otherwise, contains "FALSE".
15	**PidTagBusinessTelephoneNumber**	0x3A08 001F	**PtypString**	Contains the primary business telephone for the Address Book object.
16	**PidTagInitials**	0x3A0A 001F	**PtypString**	Contains the initials for parts of the full name of the Address Book object.
17	**PidTagStreetAddress**	0x3A29 001F	**PtypString**	Contains the street address of the Address Book object.
18	**PidTagLocality**	0x3A27 001F	**PtypString**	Contains the name of the locality of the Address Book object, such as the town or city.
19	**PidTagStateOrProvince**	0x3A28 001F	**PtypString**	Contains the name of the state or province in which the Address Book object is located.

Index Number	Property tag name	Property tag	Property type	Description
20	**PidTagPostalCode**	0x3A2A 001F	**PtypString**	Contains the postal code for the postal address of the Address Book object.
21	**PidTagCountry**	0x3A26 001F	**PtypString**	Contains the name of the country or region where the Address Book object is located.
22	**PidTagTitle**	0x3A17 001F	**PtypString**	Contains the job title of the Address Book object.
23	**PidTagCompanyName**	0x3A16 001F	**PtypString**	Contains the name of the company associated with the Address Book object.
24	**PidTagAddressBookPhoneticCompanyName** ([MS-OXOABK] section 2.2.4.12)	0x8C91 001F	**PtypString**	Contains the phonetic spelling of the company name of the Address Book object.
25	**PidTagAssistant**	0x3A30 001F	**PtypString**	Contains the name of the administrative assistant of the Address Book object.
26	**PidTagDepartmentName**	0x3A18 001F	**PtypString**	Contains the name of the department in which the Address Book object works.
27	**PidTagAddressBookPhoneticDepartmentName** ([MS-OXOABK] section 2.2.4.13)	0x8C90 001F	**PtypString**	Contains the phonetic spelling of the name of the department in which the Address Book object works.
28	**PidTagAddressBookTargetAddress**	0x8011 001F	**PtypString**	Contains the destination address for the Address Book object.
29	**PidTagHomeTelephoneNumber**	0x3A09 001F	**PtypString**	Contains the primary home telephone number of the Address Book object.
30	**PidTagBusiness2TelephoneNumbers**	0x3A1B 101F	**PtypMultipleString**	Contains the secondary business telephone numbers of the Address Book object.
31	**PidTagHome2TelephoneNumbers**	0x3A2F 101F	**PtypMultipleString**	Contains the secondary home telephone numbers of the Address Book object.
32	**PidTagPrimaryFaxNumber**	0x3A23 001F	**PtypString**	Contains the telephone number of the primary fax machine used by the Address Book object.
33	**PidTagMobileTelephoneNumber**	0x3A1C 001F	**PtypString**	Contains the cellular telephone number of the Address Book object.
34	**PidTagAssistantTelephoneNumber**	0x3A2E 001F	**PtypString**	Contains the telephone number of the administrative assistant of the Address Book object.
35	**PidTagPagerTelephoneNumber**	0x3A21	**PtypString**	Contains the pager telephone

Inde x Num ber	Property tag name	Proper ty tag	Property type	Description
		001F		number of the Address Book object.
36	**PidTagComment**	0x3004 001F	**PtypString**	Contains a comment about the purpose or content of an Address Book object.
37	**PidTagUserCertificate**	0x3A22 0102	**PtypBinary**	Contains an ASN.1 authentication certificate for a messaging user.
38	**PidTagUserX509Certificate**	0x3A70 1102	**PtypMultiple Binary** ([MS-OXCDATA] section 2.11.1)	Contains X.509 version 3 security certificates for the Address Book object, as specified in [RFC2459].
39	**PidTagAddressBookX509Certificat e**	0x8C6A 1102	**PtypMultiple Binary**	Contains ASN.1 encoded X.509 certificates, as specified in [RFC2459].
40	**PidTagAddressBookHomeMessage Database**	0x8006 001E	**PtypString8**	Contains the X500 DN of the message store for this mailbox. This property value is not subject to truncation. MUST contain Teletex characters only, as specified by the teletex-char rule in section 2.1.
41	**PidTagAddressBookDisplayNameP rintable**	0x39FF 001E	**PtypString8**	Contains the printable string version of the display name of the Address Book object. MUST contain Teletex characters only, as specified by the teletex-char rule in section 2.1.
42	**PidTagDisplayTypeEx** ([MS-OXOABK] section 2.2.3.12)	0x3905 0003	**PtypInteger3 2**	Contains a value used to associate an icon with a particular row of a table.
43	**PidTagAddressBookSeniorityIndex** ([MS-OXOABK] section 2.2.3.24)	0x8CA0 0003	**PtypInteger3 2**	Contains the seniority index for the user or department. The value is used to sort users or departments by order of seniority.
44	**PidTagAddressBookHierarchicalIs HierarchicalGroup** ([MS-OXOABK] section 2.2.6.5)	0x8CD D000B	**PtypBoolean**	Contains "TRUE" if the distribution list represents a departmental group; otherwise, contains "FALSE".
45	**PidTagAddressBookObjectGuid** ([MS-OXOABK] section 2.2.3.25)	0x8C6D 0102	**PtypBinary**	Contains the GUID that uniquely identifies the Address Book object.
46	**PidTagAddressBookSenderHintTra nslations** ([MS-OXOABK] section 2.2.3.26)	0x8CAC 101F	**PtypMultiple String**	Contains the locale ID and translations of the default **mail tip**. For example, "en-US:Hello" "es:Hola".
47	**PidTagAddressBookDeliveryConte ntLength** ([MS-OXOABK] section	0x806A 0003	**PtypInteger3 2**	Specifies the maximum size of a message that a recipient can

Index Number	Property tag name	Property tag	Property type	Description
	2.2.3.27)			receive.
48	**PidTagAddressBookModerationEnabled** ([MS-OXOABK] section 2.2.3.28)	0x8CB5 000B	**PtypBoolean**	Contains "TRUE" if moderation is enabled for the mail user or distribution list; otherwise, contains "FALSE".
49	**PidTagAddressBookDistributionListMemberCount** ([MS-OXOABK] section 2.2.3.29)	0x8CE2 0003	**PtypInteger32**	Contains the total number of recipients 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	**PidTagAddressBookDistributionListExternalMemberCount** ([MS-OXOABK] section 2.2.3.30)	0x8CE3 0003	**PtypInteger32**	Contains the number of external recipients in the distribution list.
51	**PidTagAddressBookMember** ([MS-OXOABK] section 2.2.6.1)	0x8009 101E	**PtypEmbeddedTable** ([MS-OXCDATA] section 2.11.1), encoded as **PtypMultipleString8** ([MS-OXCDATA] section 2.11.1) as specified in section 2.9.6.7.	Contains the members of the distribution list. If the distribution list is also a departmental group (as specified by the **PidTagAddressBookHierarchicalIsHierarchicalGroup** property), then the **PidTagAddressBookMember** property contains the members of the department and the child departmental groups in the hierarchy of departments.
52	**PidTagAddressBookIsMemberOfDistributionList** ([MS-OXOABK] section 2.2.5.3)	0x8008 101E	**PtypEmbeddedTable**, encoded as **PtypMultipleString8** as specified in section 2.9.6.7.	Lists all of the distribution lists to which this object is a member.
53	**PidTagOfflineAddressBookTruncatedProperties** ([MS-OXPROPS] section 2.811)	0x6805 1003	**PtypMultipleInteger32**	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 2.9.2.2.1.

2.9.2.2.1 Truncated Properties

The **PidTagOfflineAddressBookTruncatedProperties** ([MS-OXPROPS] section 2.811) 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, 32-bit integer, 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
32-bit integer multi-valued limit	INT32	2048	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 that SHOULD<12> be included in the **PidTagOfflineAddressBookTruncatedProperties** property. Each property is further specified in [MS-OXOABK] and [MS-OXPROPS].
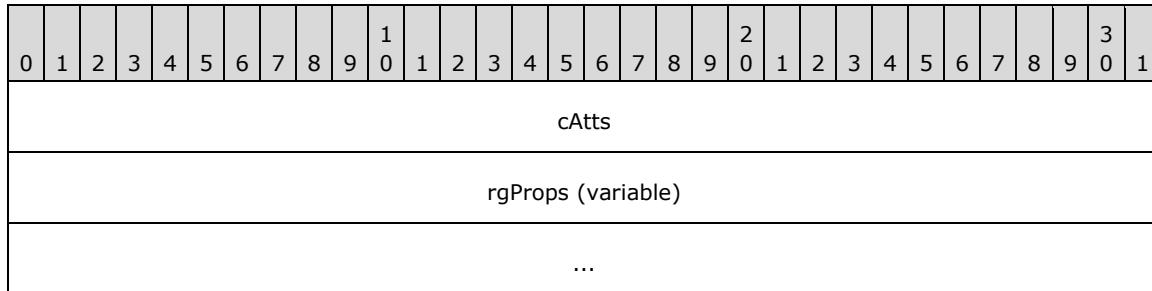
Index Number	Property tag name	Property tag	Property type	Description
1	**PidTagThumbnailPhoto** ([MS-OXOABK] section 2.2.4.40)	0x8C9E 0102	**PtypBinary** ([MS-OXCDATA]	Contains an image of the mail user's photo in .jpg format.

Index Number	Property tag name	Property tag	Property type	Description
			section 2.11.1)	
2	**PidTagSpokenName** ([MS-OXOABK] section 2.2.4.41)	0x8CC2 0102	**PtypBinary**	Contains a recording of the mail user's name pronunciation.
3	**PidTagAddressBookAuthorizedSenders** ([MS-OXOABK] section 2.2.4.42)	0x8CD 8000D	**PtypObject** ([MS-OXCDATA] section 2.11.1)	A value other than null indicates that delivery restrictions exist for this recipient. The address book does not contain the lists of senders that are allowed for this recipient; it only indicates whether or not such restrictions exist. The client can use **PidTagAddressBookAuthorizedSenders**, **PidTagAddressBookUnauthorizedSenders** ([MS-OXOABK] section 2.2.4.43), **PidTagAddressBookDistributionListMemberSubmitAccepted** ([MS-OXOABK] section 2.2.4.44), and **PidTagAddressBookDistributionListMemberSubmitRejected** ([MS-OXOABK] section 2.2.4.45) to compute a Boolean value that indicates whether a distribution list has restrictions on it.
4	**PidTagAddressBookUnauthorizedSenders** ([MS-OXOABK] section 2.2.4.43)	0x8CD 9000D	**PtypObject**	A value other than null indicates that delivery restrictions exist for this recipient. The address book does not contain the lists of senders that are prohibited for this recipient; it only indicates whether or not such restrictions exist.
5	**PidTagAddressBookDistributionListMemberSubmitAccepted** ([MS-OXOABK] section 2.2.4.44)	0x8073 000D	**PtypObject**	A value other than null indicates that delivery restrictions exist for this recipient. The address book does not contain the lists of the group of senders that are allowed for this recipient; it only indicates whether or not such restrictions exist.
6	**PidTagAddressBookDistributionListMemberSubmitRejected** ([MS-OXOABK] section 2.2.4.45)	0x8CD A000D	**PtypObject**	A value other than null indicates that delivery restrictions exist for this recipient. The address book does not contain the lists of the group of senders that are prohibited for this recipient; it only indicates whether or not such restrictions exist.
7	**PidTagAddressBookDistributionListRejectMessagesFromDLMembers** ([MS-OXPROPS] section 2.511)	0x8CD B000D	**PtypObject**	A value other than null indicates that delivery restrictions exist for this recipient. The address book does not contain the lists of distribution list senders that are prohibited for this recipient; it only indicates whether or

Index Number	Property tag name	Property tag	Property type	Description
				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	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
cAtts																															
rgProps (variable)																															
...																															
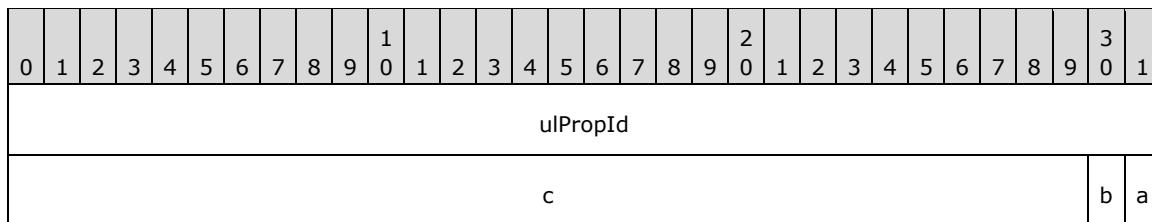
cAtts (4 bytes): A 32-bit integer that specifies the number of **OAB_PROP_REC** records in **rgProps**.

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	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
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	**PtypInteger32** ([MS-OXCDATA] section 2.11.1)
0x000B	**PtypBoolean** ([MS-OXCDATA] section 2.11.1)
0x000D	**PtypObject** ([MS-OXCDATA] section 2.11.1)
0x001E	**PtypString8** ([MS-OXCDATA] section 2.11.1)
0x001F	**PtypString** ([MS-OXCDATA] section 2.11.1)

Value	Meaning
0x0102	**PtypBinary** ([MS-OXCDATA] section 2.11.1)
0x1003	**PtypMultipleInteger32** ([MS-OXCDATA] section 2.11.1)
0x101E	**PtypMultipleString8** ([MS-OXCDATA] section 2.11.1)
0x101F	**PtypMultipleString** ([MS-OXCDATA] section 2.11.1)
0x1102	**PtypMultipleBinary** ([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)

PidTagSmtpAddress ([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 2.2.4.11)

PidTagGivenName ([MS-OXOABK] section 2.2.4.2)
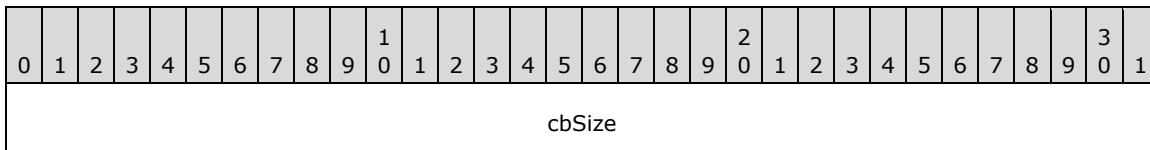
PidTagAddressBookPhoneticGivenName ([MS-OXOABK] section 2.2.4.10)

PidTagAddressBookProxyAddresses ([MS-OXOABK] section 2.2.3.23)

PidTagOfficeLocation<13> ([MS-OXOABK] section 2.2.4.5)

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	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
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----2^{32}-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 DN (3) 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-OXCDATA] section 2.11.1) is encoded. The individual string encodings are encoded in the same way that **PtypString8** ([MS-OXCDATA] 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-OXCDATA] section 2.11.1), and whose value represents references to any number of other Address Book objects, are encoded using the **PtypMultipleString8** ([MS-OXCDATA] section 2.11.1) value encoding. Each string value MUST be a DN (3) 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-OXCDATA] section 2.11.1) is encoded. The individual string encodings are

encoded in the same way that **PtypString** ([MS-OXCDATA] 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-OXCDATA] 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-OXCDATA] section 2.11.1) is encoded. The individual binary encodings are encoded in the same way that **PtypBinary** ([MS-OXCDATA] 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-OXCDATA] 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 unsigned 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 unsigned integer value that specifies the length in bytes that the source input file is expected to be.

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

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

ulTargetCRC (4 bytes): A 32-bit unsigned 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	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
ulPatchSize																															
ulTargetSize																															
ulSourceSize																															
ulCRC																															
data (variable)																															
...																															

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

ulTargetSize (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.

ulSourceSize (4 bytes): A 32-bit unsigned 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 unsigned 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 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	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): A 32-bit unsigned integer value that MUST be set to 0x00000003.
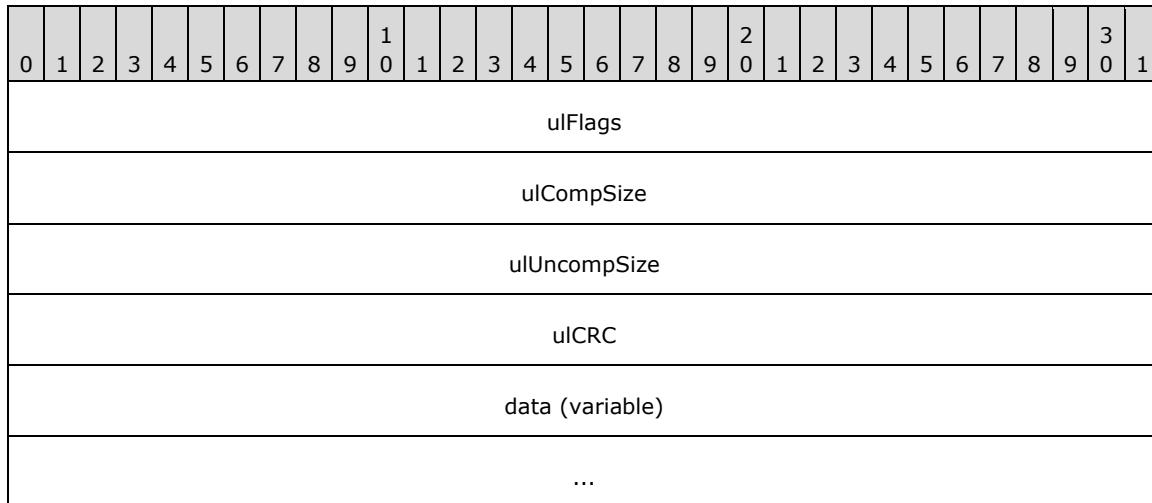
ulVersionLo (4 bytes): A 32-bit unsigned integer value that MUST be set to 0x00000001.

ulBlockMax (4 bytes): A 32-bit unsigned 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.

ulTargetSize (4 bytes): A 32-bit unsigned 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	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
ulFlags																															
ulCompSize																															
ulUncompSize																															
ulCRC																															
data (variable)																															
...																															

ulFlags (4 bytes): A 32-bit unsigned 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 **Lempel-Ziv Extended (LZX)** 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.

ulCRC (4 bytes): A 32-bit unsigned 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].

2.12 Properties

The properties that are common to both full and differential OAB messages are defined in the following subsections.

2.12.1 PidTagOfflineAddressBookContainerGuid

The **PidTagOfflineAddressBookContainerGuid** property ([MS-OXPROPS] section 2.806) contains a GUID that identifies a set of full and differential **OAL** files that form a complete OAL, ordered by their

OAL data sequence numbers, as specified by the **PidTagOfflineAddressBookSequence** property in section 2.12.4. A server MUST set this property to the same value for every full and differential OAB message in a sequence, but the server MUST set this to a different value for unrelated sequences of files.

2.12.2 PidTagOfflineAddressBookDistinguishedName

The **PidTagOfflineAddressBookDistinguishedName** property ([MS-OXPROPS] section 2.807) contains the distinguished name (DN) (1) of the address list that is contained in the OAB message. This DN (1) is addresslist-x500-DN, as specified in section 2.1. It MUST match the DN (1) that is returned by the name service provider interface (NSPI) protocol for the address list.

2.12.3 PidTagOfflineAddressBookName

The **PidTagOfflineAddressBookName** property ([MS-OXPROPS] section 2.809) contains the name of the address list that is contained in the OAB message.

2.12.4 PidTagOfflineAddressBookSequence

The **PidTagOfflineAddressBookSequence** property ([MS-OXPROPS] section 2.810) contains the OAL data sequence number of the attached full or differential OAL files.

An integer that is associated with offline address list (OAL) data that represents the generation number of this data. The value of the initial sequence number is "1". Each subsequent data generation process that produces a data set that is not identical to the previous data set is incremented by one.

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

    oSmtp           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

    oSmtp           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 0x0000008B)

        ; 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

        ; 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
        oSmtp       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
            oSmtp        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 0x0000008B)
```

```
            ; 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

        ; 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


-----------------------
```

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 Fields

None.

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 2013

- Microsoft Exchange Server 2016

- Microsoft Office Outlook 2003

- Microsoft Office Outlook 2007

- Microsoft Outlook 2010

- Microsoft Outlook 2013

- Microsoft Outlook 2016

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

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

<1> Section 2: 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 2013, Exchange 2016, Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, and Outlook 2016. There are some differences in the default behavior of Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016, 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 2013 and Exchange 2016 and are not supported by Outlook 2013 and Outlook 2016.

<3> Section 2.4: All OAB version 2 and OAB version 3 files, including the RDN Index file, are not generated by Exchange 2013 and Exchange 2016 and are not supported by Outlook 2013 and Outlook 2016.

<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 2013 and Exchange 2016 and are not supported by Outlook 2013 and Outlook 2016.

<6> Section 2.6: All OAB version 2 and OAB version 3 files, including the Details file, are not generated by Exchange 2013 and Exchange 2016 and are not supported by Outlook 2013 and Outlook 2016.

<7> Section 2.7: All OAB version 2 and OAB version 3 files, including the Changes file, are not generated by Exchange 2013 and Exchange 2016 and are not supported by Outlook 2013 and Outlook 2016.

<8> Section 2.8: All OAB version 2 and OAB version 3 files, including compressed files, are not generated by Exchange 2013 and Exchange 2016 and are not supported by Outlook 2013 and Outlook 2016.

<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 2.2.4.11), **PidTagAddressBookPhoneticGivenName** ([MS-OXOABK] section 2.2.4.10), **PidTagAddressBookPhoneticCompanyName** ([MS-OXOABK] section 2.2.4.12), **PidTagAddressBookPhoneticDepartmentName** ([MS-OXOABK] section 2.2.4.13), 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 2.811) contains no properties by default in Exchange 2003 and Exchange 2007.

<13> Section 2.9.4: **PidTagOfficeLocation** ([MS-OXOABK] section 2.2.4.5) is not in the ANR index in Exchange 2007.

6 Change Tracking

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

7 Index