

[MS-OXWOAB]: Offline Address Book (OAB) Retrieval Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the protocol documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

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

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

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability.
Microsoft Corporation	June 27, 2008	1.0	Initial Release.
Microsoft Corporation	August 6, 2008	1.01	Revised and edited technical content.
Microsoft Corporation	September 3, 2008	1.02	Revised and edited technical content.
Microsoft Corporation	December 3, 2008	1.03	Updated IP notice.

Table of Contents

1	<i>Introduction</i>	3
1.1	Glossary	3
1.2	References	4
1.2.1	Normative References	4
1.2.2	Informative References	4
1.3	Structure Overview	5
1.4	Relationship to Protocols and Other Structures	5
1.5	Applicability Statement.....	6
1.6	Versioning and Localization.....	6
1.7	Vendor-Extensible Fields	6
2	<i>Structures</i>	6
2.1	Manifest File Structure.....	6
2.1.1	oabElement	7
2.1.2	oalElement	7
2.1.3	fullElement	8
2.1.4	templateElement.....	8
2.1.5	diffElement	8
2.1.6	seq Element.....	9
3	<i>Structure Examples</i>	9
4	<i>Security Considerations</i>	11
5	<i>Appendix A: Office/Exchange Behavior</i>	11
	<i>Index</i>	12

1 Introduction

A server might choose to represent properties of known recipients and make them available in an address book to its clients. When the client cannot reach the server because it is offline or because there is a high network cost to access the server, the client might keep a local copy of the address book. This document specifies the **offline address book (OAB)** version 4 Web-based retrieval mechanism, which is a way of delivering an offline address book from the server to the client.

As part of OAB Web distribution, the server publishes an OAB manifest document. This document specifies the format of this manifest.

1.1 Glossary

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

- address book**
- Address Book object**
- Augmented Backus-Naur Form (ABNF)**
- address list**
- distinguished name (DN)**
- GUID**
- OAB data file**
- OAB manifest**
- OAB Web distribution**
- offline address book (OAB)**
- Uniform Resource Identifier (URI)**
- XML**

The following terms are specific to this document:

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

OAL data sequence number: The integer number associated with **Offline Address List (OAL)** data that represents the generation number of this data. The initial sequence number is 1. Every subsequent data generation that produces a data set that is not identical to the previous one increments the sequence number by one.

Web Distribution Point (WDP): The location on the server where **offline address book (OAB)** files are published for Web distribution. The **Uniform Resource Identifier (URI)** of the **WDP** is discoverable by the client via the Autodiscover Publishing and Lookup protocol, as specified in [MS-OXDSCLI].

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

1.2 References

1.2.1 Normative References

[FIP180-1] Federal Information Processing Standards Publication, "Secure Hash Standard", FIPS PUB 180-1, April 1995, <http://www.itl.nist.gov/fipspubs/fip180-1.htm>.

[MS-LCID] Microsoft Corporation, "Windows Language Code Identifier (LCID) Reference", March 2007, <http://go.microsoft.com/fwlink/?LinkId=112265>.

[MS-OXDCLI] Microsoft Corporation, "Autodiscover Publishing and Lookup Protocol Specification", June 2008.

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

[MS-OXOAB] Microsoft Corporation, "Offline Address Book (OAB) Format and Schema Protocol Specification", June 2008.

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

[MS-OXOABKT] Microsoft Corporation, "Address Book User Interface Templates Protocol Specification", June 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>.

[RFC2616] Fielding, R., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>.

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

1.2.2 Informative References

None.

1.3 Structure Overview

The **OAB manifest** is used by clients to identify the current version of data published by the server and build the **URIs** of data files to download. The format of the OAB manifest is **XML**, and contains it one entry for each data file in the OAB. The entries are organized hierarchically.

The client uses the Autodiscover Publishing and Lookup protocol, as specified in [MS-OXDSCLI], to discover the **Web Distribution Point (WDP)** URI, and then constructs a manifest URI by appending the well-known name "oab.xml", as specified in the following **ABNF**:

```
manifestURI = wdpUri "/oab.xml"
```

It then retrieves the manifest file by using the standard HTTP/1.1 protocol, as specified in [RFC2616]. The manifest file contains information about the **Offline Address List (OAL)** data sequence number, and the names of the data files that are published to a given WDP. The detailed structure of the manifest file is specified in section 2.

After the client retrieves and parses the manifest file, it finds out what OALs are associated with this OAB. Each OAL in the OAB can be retrieved by the client independently.

For each file that has to be retrieved, the client constructs the URI according to the following definition, and retrieves it using the HTTP/1.1 protocol [RFC2616]:

```
dataFileURI = wdpUri "/" file
```

1.4 Relationship to Protocols and Other Structures

- Clients discover the **URI** of the **WDP** by using the Autodiscover Publishing and Lookup protocol [MS-OXDSCLI].
- From the WDP URI, clients construct the manifest URI and use the HTTP/1.1 protocol [RFC2616] to retrieve the manifest file.
- Based on data in the manifest, clients use the **Offline Address Book (OAB)** Retrieval protocol to retrieve and consume **OAB data files** that are generated as specified in to [MS-OXOAB].
- The OAB Retrieval protocol relies on the HTTP 1.1 protocol, as specified in [RFC2616], to deliver the manifest and data OAB files from the server to the client. It also relies on HTTPS, as specified in [RFC2818], for data protection services.

1.5 Applicability Statement

In order to use the **OAB Web distribution** algorithm specified in this document, a set of OAB files has to be generated in the format specified in [MS-OXOAB], the files have to be published on an HTTP 1.1 server, and the **URI** of the **WDP** has to be published via the Autodiscover Publishing and Lookup protocol, as specified in [MS-OXDSCLI].

1.6 Versioning and Localization

The OAB retrieval protocol has only one version.

1.7 Vendor-Extensible Fields

None.

2 Structures

2.1 Manifest File Structure

The manifest file contains a well-formed **XML** document, as specified in [XML1], and has the following **ABNF** structure, as per the specification of Augumented BNF semantics [RFC4234]:

```
manifestDocument      = prolog oabElement
prolog                = "<?xml" VersionInfo UTF8EncodingDecl "?>"
VersionInfo           = "version" Eq (XMLQUOTE VersionNum XMLQUOTE)
VersionNum            = "1.0"
Eq                    = "="
UTF8EncodingDecl     = "encoding" Eq XMLQUOTE "UTF-8" XMLQUOTE
oabElement            = oabSTag oabContent oabETag
oabSTag               = "<OAB>"
oabETag              = "</OAB>"
oabContent            = 1* oalElement
oalElement           = oalSTag S oalAttributes *S ">" oalContent oalETag
oalSTag              = "<OAL"
oalETag              = "</OAL>"
oalAttributes        = idAttribute S dnAttribute S nameAttribute
idAttribute           = "id=" XMLQUOTE guidString XMLQUOTE
dnAttribute           = "dn=" XMLQUOTE addresslist-legacy-dn XMLQUOTE
nameAttribute        = "name=" XMLQUOTE nestedUnicodeRdn XMLQUOTE
guidString            = 8HEX "-" 4HEX "-" 4HEX "-" 4HEX "-" 12HEX
nestedUnicodeRdn     = 1*16 ( "\" unicodeRdn )
                        ; the total length is limited to
                        ; 1024 characters
unicodeRdn           = 1*1023 (NON-ZERO-OCTET)
NON-ZERO-OCTET       = %x01-FF ; Any octet (8-bit data unit) except for 0
oalContent           = fullElement 1*templateElement *diffElement
fullElement          = "<Full " S seq S ver S size S uncompressedSize S SHA
                        ">" file "</Full>"
templateElement      = "<Template " S seq S ver S size S uncompressedSize S
                        SHA S langid S type ">" file "</Template>"
diffElement          = "<Diff " S seq S ver S size S uncompressedSize S SHA
                        ">" file "</Diff>"
seq                  = "seq=" XMLQUOTE 1*DIGIT XMLQUOTE
                        ; limited to values from 0 to 2147483648
ver                  = "ver=" XMLQUOTE 1*DIGIT XMLQUOTE
```

```

; limited to values from 0 to 2147483648
size = "size=" XMLQUOTE 1*DIGIT XMLQUOTE
uncompressedsize = "uncompressedsize =" XMLQUOTE 1*DIGIT XMLQUOTE
SHA = "SHA=" XMLQUOTE 40HEX XMLQUOTE
langid = "langid =" XMLQUOTE 1*DIGIT XMLQUOTE
type = "type=" XMLQUOTE ("mac" / "windows") XMLQUOTE
file = *( NONDOT / DOT) 1* NONDOT
compressedfile = file ".lzx"
addresslist-legacy-dn = "/guid=" 32(HEX) / "/" / legacy-dn
legacy-dn = org org-unit 1*13(container) object-rdn
; legacy-dns are limited to 16 levels

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 and
; the number of rdns is limited to 16 but the
; total cumulative length of rdn characters in
; a legacy-dn is limited to 256.

teletex-char = " " / non-space-teletex
non-space-teletex = "!" / XMLQUOTE / "%" / "&" / "\" / "(" / ")" /
"*" / "+" / "," / "-" / "." / "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" / "|"

DIGIT = %x30-39
HEX = DIGIT
/ "A" / "B" / "C" / "D" / "E" / "F"
/ "a" / "b" / "c" / "d" / "e" / "f"

S = 1*(%x20 / %x09 / %x0D / %x0A)
ALPHA = %x41-5A / %x61-7A ; A-Z / a-z
XMLQUOTE = DQUOTE / "'"
DOT = "."
NONDOT = DIGIT / ALPHA / "-"

```

2.1.1 oabElement

The **oabElement** in the document structure represents a top-level container in the hierarchy of the XML document, and MUST contain one or more **oalElement** entities. The **oabElement** does not have any attributes.

2.1.2 oalElement

The **oalElement** is a container in the hierarchy of the XML document that contains XML nodes of types **fullElement**, **templateElement**, and **diffElement** and represents an OAL that is part of the OAB. The **oalElement** MUST have the following attributes:

- **id**, as specified in **idAttribute** element - A string representation of randomly chosen **GUIDs** that uniquely represents the current OAL. This id remains the same through all subsequent OAB generations.
- **dn** – The **distinguished name (DN)** of the OAL.
- **name** – The name of the **address list**, prepended with "\".

[MS-OXOABK] specifies address lists in detail.

2.1.3 fullElement

Each **oalElement** **MUST** contain exactly one **fullElement**. The **fullElement** provides information about the compressed full details file, as specified in section 1.3.1 of [MS-OXOAB]. The following elements **MUST** be specified:

- **Seq** – The **OAL data sequence number**.
- **Ver** – The version of the data file, as specified in [MS-OXOAB].
- **Size** – The size in bytes of the data file on the **WDP**.
- **Uncompressedsize** – The size in bytes of the data file after decompression.
- **SHA** – The SHA1 checksum of the compressed file, calculated as specified in [FIP180-1].
- **file** – The name of the data file on WDP.

2.1.4 templateElement

Each **oalElement** **MUST** contain at least one **templateElement**. The **templateElement** provides information about the compressed template file, as specified in [MS-OXOAB]. The following elements **MUST** be specified:

- **Seq** – This **OAL data sequence number**. It is kept in sync with sequence number of **fullElement**.
- **Ver** – The version of the data file, as specified in [MS-OXOAB].
- **Size** – The size in bytes of the data file on **WDP**.
- **Uncompressedsize** – The size in bytes of the data file after decompression.
- **SHA** – The SHA1 checksum of the compressed file, calculated as specified in [FIP180-1].
- **langid** – The template language identifier, as specified in [MS-LCID].
- **type** – A string representing the client platform, currently "windows" or "mac", as specified in [MS-OXOABKT].
- **file** – The name of the data file on WDP.

2.1.5 diffElement

Each **oalElement** **MAY** contain zero or more **diffElements**. The **diffElement** provides information about the OAB v4 differential details file, as specified in section 1.3.1 of [MS-OXOAB]. The following elements **MUST** be specified:

- **Seq** – This **OAL data sequence number**.

- **Ver** – The version of the resulting data file that will be produced by applying this differential file.
- **Size** – The size in bytes of data file on **WDP**.
- **Uncompressedsize** – The size in bytes of data file after decompression.
- **SHA** – The SHA1 checksum of the compressed file, calculated as specified in [FIP180-1].
- **file** – The name of the data file on WDP.

2.1.6 seq Element

As each **fullElement**, **templateElement**, and **diffElement** contains a **Seq** element, this enables certain optimizations for the client, as described in this section.

The client could internally maintain an integer value to store the sequence number of the last successfully downloaded **OAL** data. This enables it to determine whether the server has any data that is newer than the data that is available on the client side. This internal value is referred to as **clientSequenceNumber**. If the client implementation maintains **clientSequenceNumber**, the client also has to store the **OAL id** to identify **OAL** in future versions of the manifest, and the last downloaded full details file, to be able to build a new version of the data file by applying differential files to the older file.

As a result of parsing the manifest, the client finds the sequence number of the full **OAL** data file available on the server. This value is referred to as **serverSequenceNumber**. If **serverSequenceNumber** ≥ 2 , there are zero or more differential details files with sequential sequence numbers from **M** to **serverSequenceNumber**, where $2 \leq M \leq \text{serverSequenceNumber}$. The number of differential details files depends on the server implementation.

If for a particular **OAL** the client has a copy of the full details file with the data sequence number **clientSequenceNumber**, and the manifest has differential details files from **clientSequenceNumber** to **serverSequenceNumber**, the client can choose to download the differential detail files from **clientSequenceNumber** + 1 to **serverSequenceNumber** and apply them to the previously downloaded full details file to get to the latest version. Otherwise, the client can download a single full details file **serverSequenceNumber** to get up to date.

3 Structure Examples

The following is an example of the manifest file and corresponding **WDP** content. The **OAB** contains two **address lists**: "Global Address List," represented by the second **OAL** element, and "All Rooms," represented by the first **OAL** element. Both address lists include two templates, both for language with **id**=0409, such as English, for both "Mac" and "Windows" platforms. Both have a full details data file and a differential details files. The first **OAL**, however, has the data sequence number 2 and only one differential file. The second **OAL** has data sequence number 4 and three differential files.

```

<?xml version="1.0" encoding="UTF-8"?>
<OAB>
  <OAL id='f867b9e0-d01e-43e3-8708-ba86a1c77dff'
dn='/guid=F8E7206B268E404B9519453F0F184D24' name='\All Rooms'>
  <Full seq='2' ver='32' size='554' uncompressedsize='1165'
SHA='d626d8d782332b7e8d689eea266ee315c31f19da'>
    f867b9e0-d01e-43e3-8708-ba86a1c77dff-data-2.lzx
  </Full>
  <Template seq='2' ver='7' size='5794' uncompressedsize='25620'
SHA='53fb16d6dcdfla559b8649e9b269eee84b85c91b' langid='0409'
type='windows'>
    f867b9e0-d01e-43e3-8708-ba86a1c77dff-lng0409-2.lzx
  </Template>
  <Template seq='2' ver='7' size='5794' uncompressedsize='25620'
SHA='53fb16d6dcdfla559b8649e9b269eee84b85c91b' langid='0409' type='mac'>
    f867b9e0-d01e-43e3-8708-ba86a1c77dff-mac0409-2.lzx
  </Template>
  <Diff seq='2' ver='32' size='132' uncompressedsize='1165'
SHA='f53ec568b6fc3e4adce0e7d7dfd51ace604a9234'>
    f867b9e0-d01e-43e3-8708-ba86a1c77dff-binpatch-2.lzx
  </Diff>
</OAL>
  <OAL id='2e3eaccd-85a0-4abe-84f8-603a49801bb6' dn='/' name='\Global
Address List'>
  <Full seq='4' ver='32' size='574' uncompressedsize='1872'
SHA='91cld0fa378dc961f9e8aafb17a9569767e21c73'>
    2e3eaccd-85a0-4abe-84f8-603a49801bb6-data-4.lzx
  </Full>
  <Template seq='4' ver='7' size='5794' uncompressedsize='25620'
SHA='53fb16d6dcdfla559b8649e9b269eee84b85c91b' langid='0409'
type='windows'>
    2e3eaccd-85a0-4abe-84f8-603a49801bb6-lng0409-4.lzx
  </Template>
  <Template seq='4' ver='7' size='5794' uncompressedsize='25620'
SHA='53fb16d6dcdfla559b8649e9b269eee84b85c91b' langid='0409' type='mac'>
    2e3eaccd-85a0-4abe-84f8-603a49801bb6-mac0409-4.lzx
  </Template>
  <Diff seq='4' ver='32' size='132' uncompressedsize='1872'
SHA='49d0d0c8185dd93ba7df0fbc6b532049ba5a29c5'>
    2e3eaccd-85a0-4abe-84f8-603a49801bb6-binpatch-4.lzx
  </Diff>
  <Diff seq='2' ver='32' size='136' uncompressedsize='1197'
SHA='7e391a3fd934310489f87576ad6b6elfd6fc1590'>
    2e3eaccd-85a0-4abe-84f8-603a49801bb6-binpatch-2.lzx
  </Diff>
  <Diff seq='3' ver='32' size='138' uncompressedsize='1544'
SHA='3eb5108d87e366681eb27be395f3ef7d9525c63f'>
    2e3eaccd-85a0-4abe-84f8-603a49801bb6-binpatch-3.lzx
  </Diff>
</OAL>
</OAB>

```

4 Security Considerations

The manifest file contains the results of the SHA-1 hashing calculation; however, the SHA-1 hash value is used as an optional means of checksum verification of the downloaded file, and should not be used as a security feature. In particular, it does not prevent deliberate data tampering.

5 Appendix A: Office/Exchange Behavior

The information in this specification is applicable to the following versions of Office/Exchange:

- Office 2003 with Service Pack 3 applied
- Exchange 2003 with Service Pack 2 applied
- Office 2007 with Service Pack 1 applied
- Exchange 2007 with Service Pack 1 applied

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Office/Exchange behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies Office/Exchange does not follow the prescription.

Index

- Applicability statement, 6
- Examples, 9
- Glossary, 3
- Informative references, 4
- Introduction, 3
- Manifest file structure, 6
- Normative references, 4
- Office/Exchange behavior, 11
- References, 4
 - Informative references, 4
 - Normative references, 4
- Relationship to protocols and other structures, 5
- Security considerations, 11
- Structure overview, 5
- Structures, 6
 - Manifest file structure, 6
- Vendor-extensible fields, 6
- Versioning and localization, 6