**[MS-ABS]:**

**Address Book File Structure**

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](https://go.microsoft.com/fwlink/?LinkId=214445) or the [Microsoft Community Promise](https://go.microsoft.com/fwlink/?LinkId=214448). 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.
* **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](https://aka.ms/AA9ufj8).
* **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](https://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.

**Support.** For questions and support, please contact dochelp@microsoft.com.

**Revision Summary**

| Date | Revision History | Revision Class | Comments |
| --- | --- | --- | --- |
| 4/4/2008 | 0.1 | Major | Initial Availability |
| 4/25/2008 | 0.2 | Major | Revised and edited the technical content |
| 6/27/2008 | 1.0 | Major | Revised and edited the technical content |
| 8/15/2008 | 1.01 | Editorial | Edited the technical content |
| 12/12/2008 | 2.0 | Major | Revised and edited the technical content |
| 2/13/2009 | 2.01 | Editorial | Edited the technical content |
| 3/13/2009 | 2.02 | Editorial | Edited the technical content |
| 7/13/2009 | 2.03 | Major | Revised and edited the technical content |
| 8/28/2009 | 2.04 | Editorial | Revised and edited the technical content |
| 11/6/2009 | 2.05 | Editorial | Revised and edited the technical content |
| 2/19/2010 | 2.06 | Editorial | Revised and edited the technical content |
| 3/31/2010 | 2.07 | Major | Updated and revised the technical content |
| 4/30/2010 | 2.08 | Editorial | Revised and edited the technical content |
| 6/7/2010 | 2.09 | Editorial | Revised and edited the technical content |
| 6/29/2010 | 2.10 | Editorial | Changed language and formatting in the technical content. |
| 7/23/2010 | 2.10 | None | No changes to the meaning, language, or formatting of the technical content. |
| 9/27/2010 | 3.0 | Major | Significantly changed the technical content. |
| 11/15/2010 | 3.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 12/17/2010 | 3.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/18/2011 | 3.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/10/2011 | 3.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 1/20/2012 | 3.1 | Minor | Clarified the meaning of the technical content. |
| 4/11/2012 | 3.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/16/2012 | 3.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/8/2012 | 3.2 | Minor | Clarified the meaning of the technical content. |
| 2/11/2013 | 3.2 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/30/2013 | 3.3 | Minor | Clarified the meaning of the technical content. |
| 11/18/2013 | 3.3 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/10/2014 | 3.3.1 | Editorial | Changed language and formatting in the technical content. |
| 4/30/2014 | 3.3.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/31/2014 | 3.3.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/30/2014 | 3.3.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/30/2015 | 4.0 | Major | Significantly changed the technical content. |
| 9/4/2015 | 4.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/15/2016 | 4.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 9/14/2016 | 4.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 4/27/2018 | 5.0 | Major | Significantly changed the technical content. |
| 7/24/2018 | 6.0 | Major | Significantly changed the technical content. |
| 8/28/2018 | 7.0 | Major | Significantly changed the technical content. |
| 6/18/2019 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 8/17/2021 | 8.0 | Major | Significantly changed the technical content. |

Table of Contents

[1 Introduction 7](#_Toc79582248)

[1.1 Glossary 7](#_Toc79582249)

[1.2 References 9](#_Toc79582250)

[1.2.1 Normative References 9](#_Toc79582251)

[1.2.2 Informative References 9](#_Toc79582252)

[1.3 Structure Overview (Synopsis) 9](#_Toc79582253)

[1.3.1 Address Book Server Configuration 10](#_Toc79582254)

[1.3.2 Compressed File Format 11](#_Toc79582255)

[1.3.3 Decompressed File Format 11](#_Toc79582256)

[1.3.4 Byte Ordering 12](#_Toc79582257)

[1.4 Relationship to Protocols and Other Structures 13](#_Toc79582258)

[1.5 Applicability Statement 13](#_Toc79582259)

[1.6 Versioning and Localization 13](#_Toc79582260)

[1.7 Vendor-Extensible Fields 13](#_Toc79582261)

[2 Structures 14](#_Toc79582262)

[2.1 Address Book File Enumeration Tables 14](#_Toc79582263)

[2.1.1 ABF\_ATTRIBUTE\_FLAGS Enumeration 14](#_Toc79582264)

[2.1.2 ABF\_ATTRIBUTE\_TYPE Enumeration 15](#_Toc79582265)

[2.1.3 ABF\_ATTRIBUTE\_NORMALIZATION\_CONTROL Enumeration 16](#_Toc79582266)

[2.1.4 ABF\_ATTRIBUTE\_CLIENT\_FIELD Enumeration 16](#_Toc79582267)

[2.2 Address Book File Structures 17](#_Toc79582268)

[2.2.1 COMPRESSED\_BLOCK Structure 17](#_Toc79582269)

[2.2.2 COMPRESSED\_BLOCK\_HEADER Structure 18](#_Toc79582270)

[2.2.3 ABF\_FULL\_HEADER Structure 18](#_Toc79582271)

[2.2.4 ABF\_DELTA\_HEADER Structure 19](#_Toc79582272)

[2.2.5 ABF\_NORMALIZATION\_RULES Structure 20](#_Toc79582273)

[2.2.6 ABF\_ATTRIBUTES Structure 21](#_Toc79582274)

[2.2.7 ABF\_ATTRIBUTE Structure 21](#_Toc79582275)

[2.2.8 ABF\_CONTACTS Structure 21](#_Toc79582276)

[2.2.9 ABF\_DELETED\_CONTACTS Structure 22](#_Toc79582277)

[2.2.10 ABF\_CONTACT Structure 22](#_Toc79582278)

[2.2.11 ABF\_CONTACT\_ATTRIBUTE Structure 22](#_Toc79582279)

[2.2.12 ABF\_FULL\_TRAILER Structure 23](#_Toc79582280)

[2.2.13 ABF\_DELTA\_TRAILER Structure 23](#_Toc79582281)

[2.2.14 ABF\_TRAILER\_LENGTH Structure 24](#_Toc79582282)

[2.2.15 ABF\_CONTACTS\_CHANGES Structure 24](#_Toc79582283)

[2.2.16 ABF\_CONTACT\_CHANGES Structure 24](#_Toc79582284)

[3 Structure Examples 26](#_Toc79582285)

[3.1 Address Book File 26](#_Toc79582286)

[3.1.1 ABF\_DELTA\_HEADER 32](#_Toc79582287)

[3.1.2 ABF\_NORMALIZATION\_RULES 33](#_Toc79582288)

[3.1.3 Attribute Structure 35](#_Toc79582289)

[3.1.3.1 ABF\_ATTRIBUTE manager 35](#_Toc79582290)

[3.1.3.2 ABF\_ATTRIBUTE groupType 36](#_Toc79582291)

[3.1.3.3 ABF\_ATTRIBUTE proxyAddresses 37](#_Toc79582292)

[3.1.3.4 ABF\_ATTRIBUTE mail 37](#_Toc79582293)

[3.1.3.5 ABF\_ATTRIBUTE ipPhone 38](#_Toc79582294)

[3.1.3.6 ABF\_ATTRIBUTE otherTelephone 38](#_Toc79582295)

[3.1.3.7 ABF\_ATTRIBUTE otherMobile 39](#_Toc79582296)

[3.1.3.8 ABF\_ATTRIBUTE mobile 39](#_Toc79582297)

[3.1.3.9 ABF\_ATTRIBUTE otherHomePhone 40](#_Toc79582298)

[3.1.3.10 ABF\_ATTRIBUTE homePhone 40](#_Toc79582299)

[3.1.3.11 ABF\_ATTRIBUTE telephoneNumber 41](#_Toc79582300)

[3.1.3.12 ABF\_ATTRIBUTE msRTCSIP-PrimaryUserAddress 41](#_Toc79582301)

[3.1.3.13 ABF\_ATTRIBUTE physicalDeliveryOfficeName 42](#_Toc79582302)

[3.1.3.14 ABF\_ATTRIBUTE company 43](#_Toc79582303)

[3.1.3.15 ABF\_ATTRIBUTE mailNickname 43](#_Toc79582304)

[3.1.3.16 ABF\_ATTRIBUTE title 44](#_Toc79582305)

[3.1.3.17 ABF\_ATTRIBUTE displayName 44](#_Toc79582306)

[3.1.3.18 ABF\_ATTRIBUTE sn 45](#_Toc79582307)

[3.1.3.19 ABF\_ATTRIBUTE givenName 45](#_Toc79582308)

[3.1.3.20 ABF\_ATTRIBUTE msExchHideFromAddressLists 46](#_Toc79582309)

[3.1.4 ABSUser1 Contact 46](#_Toc79582310)

[3.1.4.1 ABF\_CONTACT 46](#_Toc79582311)

[3.1.4.2 ABF\_CONTACT\_ATTRIBUTE mail 47](#_Toc79582312)

[3.1.4.3 ABF\_CONTACT\_ATTRIBUTE otherTelephone 47](#_Toc79582313)

[3.1.4.4 ABF\_CONTACT\_ATTRIBUTE otherTelephone, normalized 47](#_Toc79582314)

[3.1.4.5 ABF\_CONTACT\_ATTRIBUTE mobile 48](#_Toc79582315)

[3.1.4.6 ABF\_CONTACT\_ATTRIBUTE mobile, normalized 48](#_Toc79582316)

[3.1.4.7 ABF\_CONTACT\_ATTRIBUTE otherHomePhone 48](#_Toc79582317)

[3.1.4.8 ABF\_CONTACT\_ATTRIBUTE otherHomePhone, normalized 49](#_Toc79582318)

[3.1.4.9 ABF\_CONTACT\_ATTRIBUTE homePhone 49](#_Toc79582319)

[3.1.4.10 ABF\_CONTACT\_ATTRIBUTE homePhone, normalized 49](#_Toc79582320)

[3.1.4.11 ABF\_CONTACT\_ATTRIBUTE telephoneNumber 50](#_Toc79582321)

[3.1.4.12 ABF\_CONTACT\_ATTRIBUTE telephoneNumber, normalized 50](#_Toc79582322)

[3.1.4.13 ABF\_CONTACT\_ATTRIBUTE msRTCSIP-PrimaryUserAddress 50](#_Toc79582323)

[3.1.4.14 ABF\_CONTACT\_ATTRIBUTE physicalDeliveryOfficeName 51](#_Toc79582324)

[3.1.4.15 ABF\_CONTACT\_ATTRIBUTE company 51](#_Toc79582325)

[3.1.4.16 ABF\_CONTACT\_ATTRIBUTE title 51](#_Toc79582326)

[3.1.4.17 ABF\_CONTACT\_ATTRIBUTE displayName 51](#_Toc79582327)

[3.1.4.18 ABF\_CONTACT\_ATTRIBUTE sn 52](#_Toc79582328)

[3.1.4.19 ABF\_CONTACT\_ATTRIBUTE givenName 52](#_Toc79582329)

[3.1.5 ABSUser5 Contact 52](#_Toc79582330)

[3.1.5.1 ABF\_CONTACT 52](#_Toc79582331)

[3.1.5.2 ABF\_CONTACT\_ATTRIBUTE telephoneNumber 53](#_Toc79582332)

[3.1.5.3 ABF\_CONTACT\_ATTRIBUTE telephoneNumber, normalized 53](#_Toc79582333)

[3.1.5.4 ABF\_CONTACT\_ATTRIBUTE msRTCSIP-PrimaryUserAddress 54](#_Toc79582334)

[3.1.5.5 ABF\_CONTACT\_ATTRIBUTE displayName 54](#_Toc79582335)

[3.1.5.6 ABF\_CONTACT\_ATTRIBUTE sn 54](#_Toc79582336)

[3.1.5.7 ABF\_CONTACT\_ATTRIBUTE givenName 55](#_Toc79582337)

[3.1.6 ABSUser2 Contact 55](#_Toc79582338)

[3.1.6.1 ABF\_CONTACT 55](#_Toc79582339)

[3.1.6.2 ABF\_CONTACT\_ATTRIBUTE telephoneNumber 56](#_Toc79582340)

[3.1.6.3 ABF\_CONTACT\_ATTRIBUTE telephoneNumber, normalized 56](#_Toc79582341)

[3.1.6.4 ABF\_CONTACT\_ATTRIBUTE msRTCSIP-PrimaryUserAddress 56](#_Toc79582342)

[3.1.6.5 ABF\_CONTACT\_ATTRIBUTE displayName 57](#_Toc79582343)

[3.1.6.6 ABF\_CONTACT\_ATTRIBUTE sn 57](#_Toc79582344)

[3.1.6.7 ABF\_CONTACT\_ATTRIBUTE givenName 57](#_Toc79582345)

[3.1.7 ABF\_CONTACT 57](#_Toc79582346)

[3.1.8 ABF\_DELTA\_TRAILER 58](#_Toc79582347)

[3.1.9 ABF\_TRAILER\_LENGTH 59](#_Toc79582348)

[3.2 Compressed Address Book File 59](#_Toc79582349)

[3.3 COMPRESSED\_BLOCK\_HEADER 60](#_Toc79582350)

[4 Security Considerations 62](#_Toc79582351)

[4.1 Security Considerations for Implementers 62](#_Toc79582352)

[4.2 Index of Security Fields 62](#_Toc79582353)

[5 Appendix A: Compression Format 63](#_Toc79582354)

[5.1 32-Bit CRC Algorithm 63](#_Toc79582355)

[5.2 Compressed Data Format 64](#_Toc79582356)

[5.3 Run Encoding 64](#_Toc79582357)

[5.4 Pseudo Code to Encode Offset and Length into a Token 65](#_Toc79582358)

[5.5 Pseudo Code to Decompress an Address Book File 66](#_Toc79582359)

[5.5.1 Function to Decompress a File 66](#_Toc79582360)

[5.5.2 Function to Decompress the Bytes in a Block 67](#_Toc79582361)

[6 Appendix B: Hash Function 70](#_Toc79582362)

[7 Appendix C: Active Directory Scanning Algorithm 72](#_Toc79582363)

[8 Appendix D: Product Behavior 74](#_Toc79582364)

[9 Change Tracking 76](#_Toc79582365)

[10 Index 77](#_Toc79582366)

# Introduction

The Address Book File Structure describes the format of the [**address book files**](#gt_b8d30ac8-0f5d-43d8-9ed8-f7067be3f800) that are produced daily by the [**Address Book Server (ABS)**](#gt_142ece82-2a9a-4f0f-8d8c-983e0d593037) and accessed by clients to get information about users, [**contacts**](#gt_48d3e923-3081-4b1c-a8b4-db07cc022128) and [**group objects**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) stored in [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90). Clients can use the contents of these address book files to provide an incremental search capability against the **users**, **contacts** and **groups** that were stored in Active Directory at the time the address book files were captured. Clients can also use this to provide reverse number lookup of incoming [**Voice over IP (VoIP)**](#gt_f8d6223d-5289-4966-9fc0-8ec7b7b42860) calls.

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

## Glossary

This document uses the following terms:

**Active Directory**: The Windows implementation of a general-purpose directory service, which uses LDAP as its primary access protocol. [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) stores information about a variety of objects in the network such as user accounts, computer accounts, groups, and all related credential information used by Kerberos [[MS-KILE]](%5BMS-KILE%5D.pdf#Section_2a32282edd484ad9a542609804b02cc9). [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is either deployed as [**Active Directory Domain Services (AD DS)**](#gt_2e72eeeb-aee9-4b0a-adc6-4476bacf5024) or Active Directory Lightweight Directory Services (AD LDS), which are both described in [[MS-ADOD]](%5BMS-ADOD%5D.pdf#Section_5ff67bf4c14548cb89cd4f5482d94664): Active Directory Protocols Overview.

**Active Directory Domain Services (AD DS)**: A directory service (DS) implemented by a domain controller (DC). The DS provides a data store for objects that is distributed across multiple DCs. The DCs interoperate as peers to ensure that a local change to an object replicates correctly across DCs. AD DS is a deployment of [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) [[MS-ADTS]](%5BMS-ADTS%5D.pdf#Section_d243592709994c628c6d13ba31a52e1a).

**address book contact**: A user, contact, or group object that is obtained from Active Directory Domain Services (AD DS), including a subset of the AD DS attributes that are associated with the object, and is stored in an address book file.

**address book file**: A file that contains a set of [**address book contact**](#gt_9f91b551-3c7f-4416-bd91-38f58f0b61f2) records.

**Address Book Server (ABS)**: A component that produces [**address book files**](#gt_b8d30ac8-0f5d-43d8-9ed8-f7067be3f800) on a daily basis.

**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]](https://go.microsoft.com/fwlink/?LinkId=90689). 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**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) 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**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8), so the use of ANSI character sets is strongly discouraged unless they are used to interoperate with legacy applications or legacy data.

**attribute**: An identifier for a single or multivalued data element that is associated with a directory object. An object consists of its [**attributes**](#gt_108a1419-49a9-4d19-b6ca-7206aa726b3f) and their values. For example, cn (common name), street (street address), and mail (email addresses) can all be [**attributes**](#gt_108a1419-49a9-4d19-b6ca-7206aa726b3f) of a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). An [**attribute's**](#gt_108a1419-49a9-4d19-b6ca-7206aa726b3f) schema, including the syntax of its values, is defined in an attributeSchema object.

**big-endian**: Multiple-byte values that are byte-ordered with the most significant byte stored in the memory location with the lowest address.

**contact**: A person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and [**attributes**](#gt_108a1419-49a9-4d19-b6ca-7206aa726b3f), such as an Internet message address or login name.

**cryptographic hash function**: A function that maps an input of any length to a short output bit string of fixed length, such that finding an input that maps to a particular bit string of the correct output length, or even finding two inputs that map to the same output bit string, is computationally infeasible. For more information, see [[SCHNEIER]](https://go.microsoft.com/fwlink/?LinkId=817338) chapters 2 and 18.

**cyclic redundancy check (CRC)**: An algorithm used to produce a checksum (a small, fixed number of bits) against a block of data, such as a packet of network traffic or a block of a computer file. The CRC is a broad class of functions used to detect errors after transmission or storage. A CRC is designed to catch random errors, as opposed to intentional errors. If errors might be introduced by a motivated and intelligent adversary, a [**cryptographic hash function**](#gt_a70d1a05-9add-4870-ab52-ec5d939e93e2) should be used instead.

**delta address book file**: An [**address book file**](#gt_b8d30ac8-0f5d-43d8-9ed8-f7067be3f800) that contains only the differences between two complete address book files. Differences can include changed values, added objects, and deleted objects.

**distinguished name (DN)**: In the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) directory service, the unique identifier of an object in [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90), as described in [MS-ADTS] and [[RFC2251]](https://go.microsoft.com/fwlink/?LinkId=90325).

**full address book file**: An [**address book file**](#gt_b8d30ac8-0f5d-43d8-9ed8-f7067be3f800) that contains a complete set of the [**address book contacts**](#gt_9f91b551-3c7f-4416-bd91-38f58f0b61f2) that existed when the file was generated by the user, contact, and groups objects in AD DS.

**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]](https://go.microsoft.com/fwlink/?LinkId=90460) or [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) must be used for generating the [**GUID**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1). See also universally unique identifier (UUID).

**group object**: In [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90), a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) has an object class group. A group has a forward link attribute member; the values of this [**attribute**](#gt_108a1419-49a9-4d19-b6ca-7206aa726b3f) either represent elements of the group (for example, objects of class user or computer) or subsets of the group (objects of class group). The representation of group subsets is called "nested group membership". The back link attribute memberOf enables navigation from group members to the groups containing them. Some groups represent groups of security principals and some do not and are, for instance, used to represent email distribution lists.

**hash**: A fixed-size result that is obtained by applying a one-way mathematical function, which is sometimes referred to as a hash algorithm, to an arbitrary amount of data. If the input data changes, the hash also changes. The hash can be used in many operations, including authentication and digital signing.

**hash code**: See [**hash**](#gt_b7e2b611-0af5-4fec-8af2-3f9ce7bad205).

**in-band provisioning**: A process in which a protocol client obtains configuration information from a protocol server.

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

**Session Initiation Protocol (SIP)**: An application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. [**SIP**](#gt_586971aa-3b65-4de3-be93-1a9756777d89) is defined in [[RFC3261]](https://go.microsoft.com/fwlink/?LinkId=90410).

**Unicode**: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) standard [[UNICODE5.0.0/2007]](https://go.microsoft.com/fwlink/?LinkId=154659) 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).

**user object**: An object of class user. A user object is a security principal object; the principal is a person or service entity running on the computer. The shared secret allows the person or service entity to authenticate itself, as described in ([[MS-AUTHSOD]](%5BMS-AUTHSOD%5D.pdf#Section_953d700a57cb4cf7b0c3a64f34581cc9) section 1.1.1.1).

**UTF-8**: A byte-oriented standard for encoding Unicode characters, defined in the Unicode standard. Unless specified otherwise, this term refers to the UTF-8 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

**Voice over IP (VoIP)**: The use of the Internet Protocol (IP) for transmitting voice communications. VoIP delivers digitized audio in packet form and can be used to transmit over intranets, extranets, and the Internet.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [[RFC2119]](https://go.microsoft.com/fwlink/?LinkId=90317). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 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](https://go.microsoft.com/fwlink/?linkid=850906).

### 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.

[E164] ITU-T, "The International Public Telecommunication Numbering Plan", Recommendation E.164, February 2005, [http://www.itu.int/rec/T-REC-E.164/e](https://go.microsoft.com/fwlink/?LinkId=89855)

**Note** There is a charge to download the specification.

[MC-RegEx] Microsoft Corporation, "Regular Expression Language Elements", [http://msdn.microsoft.com/en-us/library/az24scfc(VS.80).aspx](https://go.microsoft.com/fwlink/?LinkId=114616)

[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](https://go.microsoft.com/fwlink/?LinkId=90317)

### Informative References

[UASDC] Ziv, J. and Lempel, A., "A Universal Algorithm for Sequential Data Compression", May 1977, [http://www.cs.duke.edu/courses/spring03/cps296.5/papers/ziv\_lempel\_1977\_universal\_algorithm.pdf](https://go.microsoft.com/fwlink/?LinkId=90549)

[UNICODE] The Unicode Consortium, "The Unicode Consortium Home Page", [http://www.unicode.org/](https://go.microsoft.com/fwlink/?LinkId=90550)

## Structure Overview (Synopsis)

Address book files are produced daily by the Address Book Server. These files represent a daily snapshot of the [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17), [**contact**](#gt_48d3e923-3081-4b1c-a8b4-db07cc022128) and [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) in the [**Active Directory Domain Services (AD DS)**](#gt_2e72eeeb-aee9-4b0a-adc6-4476bacf5024), represented as a set of [**address book contacts**](#gt_9f91b551-3c7f-4416-bd91-38f58f0b61f2) in the address book files. The address book files are stored in a file system folder that contains the most recent N days of address book files, where N is greater than 0. Client applications can schedule daily downloads of the address book file for local, incremental searching of users, contacts, and groups based on various fields in the file. Client applications can also use the data in the address book file to perform reverse number lookup for incoming calls from other clients.

Clients access the address book files by using the HTTP GET method. Clients construct the URL of the latest address book file by concatenating the base URL they receive as part of [**in-band provisioning**](#gt_6deb2fc7-2825-42d5-b33f-6529ea1b01f4) with the address book file name constructed from the date. The file name is constructed as follows:

F-XXXX.lsabs

D-XXXX-YYYY.lsabs

C-XXXX-YYYY.lsabs[<1>](#Appendix_A_1" \o "Product behavior note 1)

F-XXXX.dabs

D-XXXX-YYYY.dabs

Where:

* XXXX and YYYY are 4-digit hexadecimal values that represent dates as the 0-based number of days after January 1, 2001, 00:00:00 UTC. For example, F-0A8C.lsabs would be the address book file for Saturday, May 24, 2008.
* File names that begin with F-XXXX are called full files and contain all user object, contact and group objects that exist on a given day.
* Files that begin with D-XXXX-YYYY are called delta files and represent the difference between two full files (F-XXXX.ext and F-YYYY.ext), where XXXX is less than YYYY. A delta file contains new and changed user object, contact and group objects, as well as deleted user objects, contact and group objects.
* Files that begin with C-XXXX-YYYY are called compact delta files and represent the difference between two full files (F-XXXX.lsabs and F-YYYY.lsabs), where XXXX is less than YYYY. A compact delta file contains new and deleted user objects, contacts (3), and group objects (2), as well as the specific attribute changes for changed user objects, contacts (3), and group objects (2).[<2>](#Appendix_A_2" \o "Product behavior note 2)
* Files with the .lsabs extension contain, for each user object, contacts, and group objects, all [**attributes**](#gt_108a1419-49a9-4d19-b6ca-7206aa726b3f) that would be of interest to an end user.
* Files with the .dabs extension contain a minimal subset of attributes that would be useful to a physical device with limited memory, such as a [**Session Initiation Protocol (SIP)**](#gt_586971aa-3b65-4de3-be93-1a9756777d89) phone client. Also, group objects (2) are never included in device-specific address book files, so setting this flag on the groupType attribute will have no effect.

### Address Book Server Configuration

The Address Book Server relies on a number of settings to control what goes into address book files. At a minimum, each address book file includes the following configuration settings:

**ABF\_ATTRIBUTES**: Controls which attributes for each **user**, **contact** and **group** are examined and how they are processed by the Address Book Server. A portion of this information is also used by client programs when interpreting downloaded address book files.

**ABF\_NORMALIZATION\_RULES**: Controls how the Address Book Server normalizes phone number [**attribute**](#gt_108a1419-49a9-4d19-b6ca-7206aa726b3f) values. Normalization is the process of taking an arbitrarily formatted phone number and converting it into a dial string, as described in [[E164]](https://go.microsoft.com/fwlink/?LinkId=89855). These normalization rules are also stored in the address book files for use by clients when normalizing numbers outside of the address book, such as from personal contacts repository or a phone number entered manually through the client UI.

**UseNormalizationRules**: Controls whether the Address Book Server puts normalized phone numbers in the address book file or leaves phone numbers as they are and lets the client perform the normalization if needed.

How these configuration settings are stored and accessed is at the discretion of the system implementer and is outside the scope of this document.

### Compressed File Format

Each address book file is compressed using the proprietary compression algorithm described in section [5](#Section_5c0d93f605b1421fb971ab547414a6d2). The file is stored on disk in compressed form and read over the wire using the HTTP GET method by client software. The client software then decompresses the file locally. The format of the compressed address book file is one or more COMPRESSED\_BLOCK records. Each COMPRESSED\_BLOCK record consists of a COMPRESSED\_BLOCK\_HEADER followed by the bytes for that COMPRESSED\_BLOCK. The decompression algorithm reads each block and decompresses it. The concatenation of all the decompressed blocks is the decompressed address book file.

### Decompressed File Format

After an address book file is decompressed, it can be examined. For full files, the overall layout is as follows:



Figure 1: Address book full file decompression layout

For delta files, the overall layout is as follows:



Figure 2: Address book full file decompression layout

For compact delta files, the overall layout is as follows:[<3>](#Appendix_A_3" \o "Product behavior note 3)



Figure 3: Address book compact delta file compression layout

### Byte Ordering

Data in **address book file** records are stored in [**little-endian**](#gt_079478cb-f4c5-4ce5-b72b-2144da5d2ce7) format.

Some computer architectures number bytes in a binary word from left to right, which is referred to as [**big-endian**](#gt_6f6f9e8e-5966-4727-8527-7e02fb864e7e). The bit diagram for this documentation is big-endian. Other architectures number the bytes in a binary word from right to left, which is referred to as little-endian. The underlying file format enumerations, objects and records are little-endian.

Using the big-endian and little-endian methods, the number 0x12345678 would be stored as shown in the following table.

| Byte order  | Byte 0  | Byte 1  | Byte 2  | Byte 3  |
| --- | --- | --- | --- | --- |
| big-endian | 0x12 | 0x34 | 0x56 | 0x78 |
| little-endian | 0x78 | 0x56 | 0x34 | 0x12 |

## Relationship to Protocols and Other Structures

None.

## Applicability Statement

Files that adhere to the **address book file** structure are suitable for use by clients listed in section [8](#Section_23412d36c1b8490bb1c1d700e0422bd4).

## Versioning and Localization

This document covers versioning issues in the following areas:

* Version: This document specifies the first version of the **address book file** structure. The **ABF\_TRAILER\_LENGTH** structure provides support for adding more information in a way that does not affect older clients of the file structure.
* Localization: This structure defines no locale-specific processes or data. All strings are encoded in [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF-8**](#gt_409411c4-b4ed-4ab6-b0ee-6d7815f85a35) [[UNICODE]](https://go.microsoft.com/fwlink/?LinkId=90550) format.

## Vendor-Extensible Fields

None.

# Structures

The following sections specify various types of address book file records and enumerations.

Note: All character strings specified in this section MUST be encoded in [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF-8**](#gt_409411c4-b4ed-4ab6-b0ee-6d7815f85a35) format.

## Address Book File Enumeration Tables

### ABF\_ATTRIBUTE\_FLAGS Enumeration

 The **ABF\_ATTRIBUTE\_FLAGS** enumeration defines the bit values stored in the Flags field of the **ABF\_ATTRIBUTE** structure.

1. typedef enum
2. {
3. ABF\_ATTRIBUTE\_FLAGS\_TYPE\_MASK = 0x000000FF,
4. ABF\_ATTRIBUTE\_FLAGS\_RESERVED = 0x00000100,
5. ABF\_ATTRIBUTE\_FLAGS\_EMAIL = 0x00000400,
6. ABF\_ATTRIBUTE\_FLAGS\_REQUIRED = 0x00000800,
7. ABF\_ATTRIBUTE\_FLAGS\_NC\_MASK = 0x00003000,
8. ABF\_ATTRIBUTE\_FLAGS\_EXCLUDE = 0x00004000,
9. ABF\_ATTRIBUTE\_FLAGS\_INCLUDE = 0x00008000,
10. ABF\_ATTRIBUTE\_FLAGS\_GROUPTYPE = 0x00010000,
11. ABF\_ATTRIBUTE\_FLAGS\_DEVICE = 0x00020000,
12. ABF\_ATTRIBUTE\_FLAGS\_UNUSED = 0x00FC0200,
13. ABF\_ATTRIBUTE\_FLAGS\_CLIENT\_MASK = 0xFF000000,
14. } ABF\_ATTRIBUTE\_FLAGS;

**ABF\_ATTRIBUTE\_FLAGS\_TYPE\_MASK**: Used to mask out the value type of an Active Directory attribute. Doing a bitwise AND of this value with the Flags field of the ABF\_ATTRIBUTE will yield a value in the ABF\_ATTRIBUTE\_TYPE enumeration.

**ABF\_ATTRIBUTE\_FLAGS\_RESERVED**: Reserved for future use. Value MUST be ignored.

**ABF\_ATTRIBUTE\_FLAGS\_EMAIL**: If a bitwise AND of this value with the Flags field of the ABF\_ATTRIBUTE structure yields a nonzero value, the associated attribute is an e-mail alias for an [**address book contact**](#gt_9f91b551-3c7f-4416-bd91-38f58f0b61f2). If this bit is not set, the associated attribute does not represent an e-mail alias for an address book contact.

**ABF\_ATTRIBUTE\_FLAGS\_REQUIRED**: If a bitwise AND of this value with the Flags field of the ABF\_ATTRIBUTE structure yields a nonzero value, the associated attribute is required to be present on a user, contact or group object in order for the corresponding address book contact to be included in the address book file by the Address Book Server. If this bit is not set, the presence or absence of the associated attribute on a user, contact or group object does not affect whether the address book contact is included in the address book file. For information about how this attribute flag interacts with the other attribute flags, see section [7](#Section_79750d26128e45649969aa94f4632b47).

**ABF\_ATTRIBUTE\_FLAGS\_NC\_MASK**: This value can be used to mask out the phone number normalization control for an Active Directory [**attribute**](#gt_108a1419-49a9-4d19-b6ca-7206aa726b3f). Doing a bitwise AND of this value with the Flags field of the ABF\_ATTRIBUTE yields a value from the ABF\_ATTRIBUTE\_NORMALIZATION\_CONTROL enumeration.

**ABF\_ATTRIBUTE\_FLAGS\_EXCLUDE**: If a bitwise AND of this value with the Flags field of the ABF\_ATTRIBUTE yields a nonzero value, any user, contact or group object that has the associated attribute will be excluded from the address book file by the Address Book Server. This bit setting overrides the setting of the ABF\_ATTRIBUTE\_FLAGS\_INCLUDE bit if both are specified for an attribute. If this bit is not set, the presence or absence of the associated attribute on a user, contact or group object does not affect whether the address book contact is included in the address book file. For information about how this attribute flag interacts with the other attribute flags, see section 7.

**ABF\_ATTRIBUTE\_FLAGS\_INCLUDE**: If a bitwise AND of this value with the Flags field of the ABF\_ATTRIBUTE yields a nonzero value, then any user, contact or group object that has the associated attribute is included as an address book contact in the address book file by the Address Book Server. If this bit is not set, the presence or absence of the associated attribute on a user, contact or group object does not affect whether the address book contact is included in the address book file. For information about how this attribute flag interacts with the other attribute flags, see section 7.

**ABF\_ATTRIBUTE\_FLAGS\_GROUPTYPE**: If a bitwise AND of this value with the Flags field of the ABF\_ATTRIBUTE yields a nonzero value, the Boolean value of the associated attribute indicates whether the address book contact that has the associated attribute is a distribution list. If this bit is not set, the associated attribute is not used to determine if an address book contact is a distribution list.

**ABF\_ATTRIBUTE\_FLAGS\_DEVICE**: If a bitwise AND of this value with the Flags field of the ABF\_ATTRIBUTE yields a nonzero value, the associated attribute is included in the device-specific address book files (.dabs extension). If this bit is not set, the attribute is not included in the device-specific address book files. Group objects are never included in device specific address book files, so setting this flag on the groupType attribute has no effect.

**ABF\_ATTRIBUTE\_FLAGS\_UNUSED**: These bits in the Flags field are unused and MUST be ignored.

**ABF\_ATTRIBUTE\_FLAGS\_CLIENT\_MASK**: Used to mask out the client mapping of an Active Directory attribute. Doing a bitwise AND of this value with the Flags field of the ABF\_ATTRIBUTE yields a value from the ABF\_ATTRIBUTE\_CLIENT\_FIELD enumeration.

### ABF\_ATTRIBUTE\_TYPE Enumeration

 The **ABF\_ATTRIBUTE\_TYPE** enumeration defines values that specify how the Address Book Server and clients process Active Directory attributes associated with address book contacts. These values are found in the low-order byte of the **Flags** field of the **ABF\_ATTRIBUTE** structure.

1. typedef enum
2. {
3. ABF\_TYPE\_STRING = 0x00,
4. ABF\_TYPE\_BINARY = 0x01,
5. ABF\_TYPE\_STRING\_PROXYADDRESS = 0x02
6. } ABF\_ATTRIBUTE\_TYPE;

**ABF\_TYPE\_STRING**: An Active Directory attribute whose value is a [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) string.

**ABF\_TYPE\_BINARY**: An Active Directory attribute whose value is a sequence of bytes.

**ABF\_TYPE\_STRING\_PROXYADDRESS**: An Active Directory proxyAddresses attribute whose value is a Unicode string that MUST begin with "TEL:" or "SMTP:". The case of "tel:" and "smtp:" is not important. A proxyAdddresses attribute whose value starts with "SMTP:" MUST contain the '@' character in the value. A proxyAdddresses attribute whose value starts with "TEL:" MUST have the format "tel:+nnnnnnnnnnnn;display-name=xxxxxxxxxx;ad-rdn=tttttt". proxyAddresses in Active Directory that begin with anything else (for example, "X500") are ignored and not stored in the Address Book Server output files. If **UseNormalizationRules** is set to "1", no **proxyAddress**es attribute values are included in the output files.

 String "nnnnnnnnnnnn" is up to 15 digits and represents the E164 form of the phone number, "xxxxxxxxxx" is arbitrary text to use as the display string for the phone number and "tttttt" gives the type of phone number and may be **telephoneNumber**, **homeNumber**, **mobile**, or **otherTelephone**.

### ABF\_ATTRIBUTE\_NORMALIZATION\_CONTROL Enumeration

 The **ABF\_ATTRIBUTE\_ NORMALIZATION\_CONTROL** enumeration defines values that specify how an Address Book Server normalizes phone number attributes associated with user, contact and group objects. These values are found in bits 18 and 19 of the **Flags** field of the **ABF\_ATTRIBUTE** structure(base index is 0). If normalization is enabled for a particular attribute, and that attribute is present multiple times for a given user or contact, only the first value of that attribute is normalized.

1. typedef enum
2. {
3. ABF\_NORMALIZATION\_CONTROL\_NONE = 0x0000,
4. ABF\_NORMALIZATION\_CONTROL\_ALWAYS = 0x1000,
5. ABF\_NORMALIZATION\_CONTROL\_PROXYADDRESS = 0x2000,
6. ABF\_NORMALIZATION\_CONTROL\_UNDEFINED = 0x3000
7. } ABF\_ATTRIBUTE\_NORMALIZATION\_CONTROL;

**ABF\_NORMALIZATION\_CONTROL\_NONE**: Servers MUST NOT normalize the attribute value.

**ABF\_NORMALIZATION\_CONTROL\_ALWAYS**: Servers MUST normalize the attribute value if matching the phone number normalization rules in Rules field of ABF\_NORMALIZATION\_RULES Structure; otherwise, the attribute value is not normalized.

**ABF\_NORMALIZATION\_CONTROL\_PROXYADDRESS**: Servers MUST extract the normalized phone number from the proxyAddress value if the UseNormalizationRules field in ABF\_FULL\_HEADER or in ABF\_DELTA\_HEADER is zero. Otherwise, they MUST behave as if ABF\_NORMALIZATION\_CONTROL\_ALWAYS is set.

 Normalized numbers are stored in the **proxyAddress** value using the following syntax:

 tel:+nnnnnnnnnnnn;display-name=xxxxxxxxxx;ad-rdn=tttttt

 where nnnnnnnnnnnn is up to 15 digits and represents the E164 form of the phone number, xxxxxxxxxx is arbitrary text to use as the display string for the phone number and tttttt gives the type of phone number and may be telephoneNumber, homeNumber, mobile, or otherTelephone.

**ABF\_NORMALIZATION\_CONTROL\_UNDEFINED**: This value is not used and MUST be ignored.

### ABF\_ATTRIBUTE\_CLIENT\_FIELD Enumeration

 The **ABF\_ATTRIBUTE\_CLIENT\_FIELD** enumeration defines values that specify how clients map attributes associated with address book contacts to attributes in the client database. These values are found in the high order byte of the **Flags** field of the **ABF\_ATTRIBUTE** structure.

1. typedef enum
2. {
3. ABF\_CLIENT\_FIELD\_PROXYADDRESSES = 0x00000000,
4. ABF\_CLIENT\_FIELD\_GIVENNAME = 0x01000000,
5. ABF\_CLIENT\_FIELD\_SN = 0x02000000,
6. ABF\_CLIENT\_FIELD\_DISPLAYNAME = 0x03000000,
7. ABF\_CLIENT\_FIELD\_TITLE = 0x04000000,
8. ABF\_CLIENT\_FIELD\_MAILNICKNAME = 0x05000000,
9. ABF\_CLIENT\_FIELD\_COMPANY = 0x06000000,
10. ABF\_CLIENT\_FIELD\_PHYSICALDELIVERYOFFICENAME = 0x07000000,
11. ABF\_CLIENT\_FIELD\_MSRTCSIP\_PRIMARYUSERADDRESS = 0x08000000,
12. ABF\_CLIENT\_FIELD\_TELEPHONENUMBER = 0x09000000,
13. ABF\_CLIENT\_FIELD\_HOMENUMBER = 0x0A000000,
14. ABF\_CLIENT\_FIELD\_MOBILE = 0x0B000000,
15. ABF\_CLIENT\_FIELD\_OTHERTELEPHONE = 0x0C000000,
16. ABF\_CLIENT\_FIELD\_IPPHONE = 0x0D000000,
17. ABF\_CLIENT\_FIELD\_MAIL = 0x0E000000,
18. ABF\_CLIENT\_FIELD\_GROUPTYPE = 0x0F000000,
19. ABF\_CLIENT\_FIELD\_MANAGER = 0x10000000,
20. ABF\_CLIENT\_FIELD\_IGNORE = 0xFF000000
21. } ABF\_ATTRIBUTE\_CLIENT\_FIELD;

**ABF\_CLIENT\_FIELD\_PROXYADDRESSES**: A proxyAddresses attribute of an address book contact.

**ABF\_CLIENT\_FIELD\_GIVENNAME**: The first name of an address book contact.

**ABF\_CLIENT\_FIELD\_SN**: The surname or last name of a user or contact.

**ABF\_CLIENT\_FIELD\_DISPLAYNAME**: The display name of an address book contact.

**ABF\_CLIENT\_FIELD\_TITLE**: The title of an address book contact.

**ABF\_CLIENT\_FIELD\_MAILNICKNAME**: The e-mail account name of an address book contact.

**ABF\_CLIENT\_FIELD\_COMPANY**: The company name of an address book contact.

**ABF\_CLIENT\_FIELD\_PHYSICALDELIVERYOFFICENAME**: The office name of an address book contact.

**ABF\_CLIENT\_FIELD\_MSRTCSIP\_PRIMARYUSERADDRESS**: The primary [**SIP**](#gt_586971aa-3b65-4de3-be93-1a9756777d89) address of an address book contact.

**ABF\_CLIENT\_FIELD\_TELEPHONENUMBER**: The work phone number of an address book contact.

**ABF\_CLIENT\_FIELD\_HOMENUMBER**: The home phone number of an address book contact.

**ABF\_CLIENT\_FIELD\_MOBILE**: The mobile phone number of an address book contact.

**ABF\_CLIENT\_FIELD\_OTHERTELEPHONE**: An alternate phone number of an address book contact.

**ABF\_CLIENT\_FIELD\_IPPHONE**: The IP phone number of an address book contact.

**ABF\_CLIENT\_FIELD\_MAIL**: The email address (user@host) for an address book contact.

**ABF\_CLIENT\_FIELD\_GROUPTYPE**: A Boolean indicator of whether an address book contact is a distribution list. If this attribute is present on an address book contact, the value MUST be 1, indicating that the address book contact is a distribution list.

**ABF\_CLIENT\_FIELD\_MANAGER**: The [**distinguished name (DN)**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) of the manager of an address book contact.[<4>](#Appendix_A_4" \o "Product behavior note 4)

**ABF\_CLIENT\_FIELD\_IGNORE**: Ignored by the client and not mapped into any field in the client's database.

## Address Book File Structures

### COMPRESSED\_BLOCK Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Header |
| ... |
| ... |
| Data (variable) |
| ... |

Header (12 bytes): A **COMPRESSED\_BLOCK\_HEADER** structure that gives the [**cyclic redundancy check (CRC)**](#gt_9cb45a36-92bb-4c14-b2fd-2ad7e2979bfd) for the original, uncompressed **Data** bytes, the length of the **Data** bytes, and the length of the decompressed **Data** bytes.

**Data (variable):** 1 or more bytes of compressed data. Length is defined in the **Header** structure.

### COMPRESSED\_BLOCK\_HEADER Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| CRC |
| CompressedSize |
| DecompressedSize |

CRC (4 bytes): A 32-bit unsigned integer that gives the [**CRC**](#gt_9cb45a36-92bb-4c14-b2fd-2ad7e2979bfd) for the original, uncompressed **Data** bytes. The value in this field MUST be calculated using the algorithm described in section [5](#Section_5c0d93f605b1421fb971ab547414a6d2).

**CompressedSize (4 bytes):** A 32-bit unsigned integer that gives the length of the **Data** bytes that follow the **Header** field. **CompressedSize** MUST be less than or equal to 64 kilobytes (0x10000).

**DecompressedSize (4 bytes):** A 32-bit unsigned integer that gives the length of the **Data** bytes after they are decompressed with the algorithm described in section 5. **DecompressedSize** MUST be greater than or equal to **CompressedSize**. If **CompressedSize** equals **DecompressedSize,** the block is not compressed, and the **Data** bytes represent the actual data. The block MAY be uncompressed.

### ABF\_FULL\_HEADER Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| FileType (16 bytes) |
| ... |
| ... |
| NumberOfAttributes | CreationDate |
| UseNormalizationRules | MaximumAttributeId |
| SourceStream (128 bytes) |
| ... |
| ... |

FileType (16 bytes): A 16-byte [**GUID**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1) that MUST equal 0x76 0x6c 0xe1 0x44 0xfd 0x0a 0xa9 0x40 0x8b 0x63 0x5f 0xe9 0xb0 0x81 0x73 0x8f. This value indicates that the file is a full file.

**CreationDate (2 bytes):** A 16-bit unsigned number that is the XXXX portion of the F-XXXX file name.

**NumberOfAttributes (2 bytes):** A 16-bit unsigned number that is the number of **ABF\_ATTRIBUTE** structures in the **ABF\_ATTRIBUTES** structure.

**MaximumAttribudeId (2 bytes):** A 16-bit unsigned number that is the largest value of the **Id** field in all of the **ABF\_ATTRIBUTE** structures in the **ABF\_ATTRIBUTES** structure.[<5>](#Appendix_A_5" \o "Product behavior note 5)

**UseNormalizationRules (2 bytes):** A 16-bit unsigned number that is 0 if the server did not use the phone normalization rules in the **ABF\_NORMALIZATION\_RULES** structure to normalize phone number strings. The value is nonzero if the Address Book Server did normalize phone numbers using the phone normalization rules.

**SourceStream (128 bytes):** A 128-byte field that SHOULD be set to all zeroes. The client does not validate this field.

### ABF\_DELTA\_HEADER Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| FileType (16 bytes) |
| CreationDate | BaseCreationDate |
| MaximumAttributeId | NumberOfAttributes |
| SourceStream (128 bytes) | UseNormalizationRules |
|  |

FileType (16 bytes): A 16-byte GUID. For delta files, it MUST equal 0x16 0xc1 0x4b 0xb5 0x08 0x90 0xc7 0x47 0xb9 0xbd 0xf3 0xbb 0x1a 0x0a 0xb6 0xeb. This value indicates that the file is a delta file.

For compact files, it MUST equal 0x34 0x17 0x7d 0xf7 0x87 0xae 0x2b 0x4d 0x09 0xa0 0x8e 0xe9 0xba 0x89 0x4a 0x04. This value indicates that the file is a compact delta file.[<6>](#Appendix_A_6" \o "Product behavior note 6)

**BaseCreationDate (2 bytes):** A 16-bit unsigned number that is the XXXX portion of the D-XXXX-YYYY file name.

**CreationDate (2 bytes):** A 16-bit unsigned number that is the YYYY portion of the D-XXXX-YYYY file name.

**NumberOfAttributes (2 bytes):** A 16-bit unsigned number that is the number of **ABF\_ATTRIBUTE** structures in the **ABF\_ATTRIBUTES** structure.

**MaximumAttributeId (2 bytes):** A 16-bit unsigned number that is the largest value of the **Id** field in all of the **ABF\_ATTRIBUTE** structures in the **ABF\_ATTRIBUTES** structure.[<7>](#Appendix_A_7" \o "Product behavior note 7)

**UseNormalizationRules (2 bytes):** A 16-bit unsigned number that is 0 if the server did not use the phone normalization rules in the **ABF\_NORMALIZATION\_RULES** structure to normalize phone number strings. The value is nonzero if the Address Book Server did normalize phone numbers using the phone normalization rules.

**SourceStream (128 bytes):** A 128-byte field that SHOULD be set to all zeroes. The client does not validate this field.

### ABF\_NORMALIZATION\_RULES Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Length |
| Rules (variable) |
| ... |

**Length (4 bytes):** A 32-bit unsigned number that is the number of bytes in the **Rules** field.

**Rules (variable):** A series of [**UTF-8**](#gt_409411c4-b4ed-4ab6-b0ee-6d7815f85a35) characters that contains phone number normalization rules. The number of characters in the series is specified by the **Length** field. If **Length** is nonzero, the last byte of the **Rules** field MUST be a zero byte. Each rule consists of two lines, and each line MUST be terminated by a carriage return, linefeed sequence (0xD, 0xA). The first line is a regular expression to match against a phone number string, using standard [[MC-RegEx]](%5BMC-RegEx%5D.pdf) regular expression syntax.[<8>](#Appendix_A_8" \o "Product behavior note 8) The second line is a replacement string to convert the matching phone number into a valid [[E164]](https://go.microsoft.com/fwlink/?LinkId=89855) number. When more than one rule is present, the rules MUST be processed in order, and the phone number is normalized using the first matching rule. If no rule matches a phone number, the phone number is not normalized.

There is a special built-in rule called E164. If the regular expression in the first line of one of the preceding pairs of lines is the string "E164," the second line is ignored and the rule matches any input that consists of 15 or fewer decimal digits, with any spaces, periods, hyphens, or parentheses ignored. If an input does match the built-in E164 rule, the result is "tel:+" followed by the digits that matched. The ignored characters are discarded.

### ABF\_ATTRIBUTES Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Attributes (variable) |
| ... |

**Attributes (variable):** One or more **ABF\_ATTRIBUTE** structures.

### ABF\_ATTRIBUTE Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id | Length |
| Flags |
| Name (variable) |

**Length (2 bytes):** A 16-bit unsigned integer that gives the length of the remaining bytes in the structure.

**Id (2 bytes):** A 16-bit unsigned integer that gives the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags (4 bytes):** A 32-bit unsigned integer that defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. The bits in this field are described by the **ABF\_ATTRIBUTE\_FLAGS** enumeration.

**Name (variable):** A zero-terminated UTF-8 string that is the name of the attribute in Active Directory. Name is not case-sensitive.

### ABF\_CONTACTS Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Contacts (variable) |
| ... |

**Contacts (variable):** Zero or more **ABF\_CONTACT** structures.

### ABF\_DELETED\_CONTACTS Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Contacts (variable) |
| ... |

**Contacts (variable):** Zero or more **ABF\_CONTACT** structures.

### ABF\_CONTACT Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (16 bytes) | Length |
| NumberOfAttributes | ... |
| Attributes (variable) | NumberOfDeletedAttributes (optional) |
|  |

**Length (2 bytes):** A 16-bit unsigned integer that gives the length of the remaining bytes in the structure.

**Id (16 bytes):** A 16 byte GUID that is the value of the **objectGUID** Active Directory attribute for this address book contact. If this field is all zeroes, this is a sentinel entry that marks the end of the **ABF\_CONTACTS** structure. The sentinel contact entry is not included in the count of **ABF\_CONTACTS** structures. The value of the **NumberOfAttributes** field in the sentinel contact entry MUST be 0, and the **Length** field MUST be 0x12.

**NumberOfAttributes (2 bytes):** A 16-bit unsigned integer that gives the number of **ABF\_CONTACT\_ATTRIBUTE** records in the **Attributes** field.

**NumberOfDeletedAttributes (2 bytes, optional):** If the **NumberOfAttributes** field is zero, this is a deleted Contact object, and the **NumberOfDeletedAttributes** field MUST be present if the **Length** field is greater than 20 (0x14).

**Attributes (variable):** Zero or more **ABF\_CONTACT\_ATTRIBUTE** structures that give the requested Active Directory attributes for this address book contact object.

### ABF\_CONTACT\_ATTRIBUTE Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Length (optional) | Id |
| Data (variable) |

**Id (1 or 2 bytes):** An 8-bit or 16-bit unsigned integer that gives the identifier of the Active Directory attribute. This field corresponds to the **Id** field in the **ABF\_ATTRIBUTE** structure. This field is 1 byte if the value of the **MaximumAttributeId** field in the **ABF\_FULL\_HEADER** or **ABF\_DELTA\_HEADER** structure is less than 256; otherwise, this field is 2 bytes.

**Length (2 bytes, optional):** A 16-bit unsigned integer that gives the length of the **Data** field if the **Id** field specifies a binary attribute. If the **Id** field does not specify a binary attribute, this field is not present.

**Data (variable):** A variable field. The preceding **Id** field is used to index the **ABF\_ATTRIBUTES** array for the associated **ABF\_ATTRIBUTE** structure. The **Flags** field of that structure, masked with the **ABF\_ATTRIBUTE\_FLAGS\_TYPE\_MASK,** specifies the type of this field. If the type is ABF\_TYPE\_BINARY, this field contains the bytes for that attribute. If the type is not ABF\_TYPE\_BINARY, this field contains a zero-terminated UTF-8 string.

### ABF\_FULL\_TRAILER Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| NumberOfContacts | Hash |
|  | NumberOfContacts (cont.) |

Hash (2 bytes): A 16-bit unsigned number that is a [**hash**](#gt_b7e2b611-0af5-4fec-8af2-3f9ce7bad205) of the contents of the file. The [**cryptographic hash function**](#gt_a70d1a05-9add-4870-ab52-ec5d939e93e2) used can be anything, but it MUST be the same function for all files. For an example of a hash function, see section [6](#Section_5af7bd1aa87f4f2a8485bd10d85534fa).

**NumberOfContacts (4 bytes):** A 32-bit unsigned number that is the number of **ABF\_CONTACT** structures in the file. Does not include the sentinel **ABF\_CONTACT** structure (**Id** and **NumberOfAttributes** fields all zeroes).

### ABF\_DELTA\_TRAILER Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| BaseFileHash | Hash |
| NumberOfContacts |
| NumberOfDeletedContacts |

Hash (2 bytes): A 16-bit unsigned number that is a [**hash**](#gt_b7e2b611-0af5-4fec-8af2-3f9ce7bad205) of the contents of the file. Any [**cryptographic hash function**](#gt_a70d1a05-9add-4870-ab52-ec5d939e93e2) can be used, as long as it is the same function for all files. For an example of a hash function, see section [6](#Section_5af7bd1aa87f4f2a8485bd10d85534fa).

BaseFileHash (2 bytes): A 16-bit unsigned number that is a hash of the contents of the base file that was used to calculate this delta file. Any hash function used can be used, but it MUST be the same function for all files. For an example of a hash function, see section 6.

**NumberOfContacts (4 bytes):** A 32-bit unsigned number that is the number of **ABF\_CONTACT** structures in the file. Does not include the sentinel **ABF\_CONTACT** structure (**Id** and **NumberOfAttributes** fields all zeroes).

**NumberOfDeletedContacts (4 bytes):** A 32-bit unsigned number that is the number of **ABF\_CONTACT** structures in the file that identify deleted contacts.

### ABF\_TRAILER\_LENGTH Structure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| LengthOfTrailer |

**LengthOfTrailer (4 bytes):** A 32-bit unsigned number that is the length of either the **ABF\_FULL\_TRAILER** or **ABF\_DELTA\_TRAILER** structure that precedes this structure. Additional information MAY be added to the file format. If additional information is added, it MUST be placed after the **ABF\_FULL\_TRAILER** or **ABF\_DELTA\_TRAILER** structure and before the **ABF\_TRAILER\_LENGTH** structure. To access the trailer structure, readers of the file seek to the end of the file less the size of the **ABF\_TRAILER\_LENGTH** structure to read the length of the trailer structure. Once the **LengthOfTrailer** field has been read, readers of the file can seek from the end of the file less **LengthOfTrailer** less 4 (the size of the **ABF\_TRAILER\_LENGTH** structure) to get to the beginning of the trailer structure. This allows future versions of the file format to contain more information if necessary. If more information is added to the file, the **LengthOfTrailer** MUST include the length of the added information. Using this mechanism prevents breaking old clients using the existing file format.

### ABF\_CONTACTS\_CHANGES Structure

The product behavior in this section replaces that in section [2.2.8](#Section_f4ce14fb9f80465a99ffbf1dd2241675).[<9>](#Appendix_A_9" \o "Product behavior note 9)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Changed Contacts (variable) |
| ... |

**Changed Contacts (variable):** Zero or more **ABF\_CONTACT\_CHANGES** structures. Compact delta files uses ABF\_CONTACTS\_CHANGES structure instead of ABF\_CONTACTS structure.

### ABF\_CONTACT\_CHANGES Structure

The product behavior in this section replaces that in section [2.2.10](#Section_0f063341df544eebaabeaf2cb67ab572).[<10>](#Appendix_A_10" \o "Product behavior note 10)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (16 bytes) | Length |
| NumberOfAttributes | ... |
| Attributes (variable) | NumberOfDeletedAttributes (optional) |
|  |

**Length (2 bytes):** A 16-bit unsigned integer that gives the length of the remaining bytes in the structure.

**Id (16 bytes):** A 16 byte GUID that is the value of the **objectGUID** Active Directory attribute for this **address book contact**. If this field is all zeroes, this is a sentinel entry that marks the end of the **ABF\_CONTACTS\_CHANGES** structure. The sentinel contact entry is not included in the count of **ABF\_CONTACTS\_CHANGES** structures. The value of the **NumberOfAttributes** field in the sentinel contact entry MUST be 0, and the **Length** field MUST be 0x12.

**NumberOfAttributes (2 bytes):** A 16-bit unsigned integer that gives the number of **ABF\_CONTACT\_ATTRIBUTE** records in the **Attributes** field.

**NumberOfDeletedAttributes (2 bytes, optional):** If the **NumberOfAttributes** field is zero, this is a deleted Contact object, and the **NumberOfDeletedAttributes** field MUST be present if the **Length** field is greater than 20 (0x14).

**Attributes (variable):** Zero or more **ABF\_CONTACT\_ATTRIBUTE** structures that give the requested Active Directory attributes for this **address book contact** object. These are only the attributes that have changed for the contact. If an attribute is deleted, it shows up here as empty. Missing attributes are unchanged.

# Structure Examples

## Address Book File

The primary function of the Address Book Server is to scan Active Directory for users, contacts and groups, and determine which objects to include in the Address Book Server files and, for each object included, which attributes to include. For pseudo-code that shows how users, contacts, and groups are chosen from Active Directory, see section [7](#Section_79750d26128e45649969aa94f4632b47).

This section describes an example of an address book output file. The scenario that produced this address book file is as follows:

* Monday, January 21, 2008, a [**full address book file**](#gt_dbc72c8e-9165-4ab3-88e7-bd18edf25608) is produced, F-0A10.lsabs.
* Later in the day, an administrator makes several edits to the Active Directory Domain Services. The **displayName** attribute of one user is changed, another user is deleted, and a new user is added.
* Tuesday, January 22, 2008, a full address book file is produced, F-0A11.lsabs, along with a [**delta address book file**](#gt_ee5b6fad-5c27-49d0-a20d-0eceda2baf3a), D-0A10-0A11.lsabs, which is the delta between the two full files. and a compact delta address book file, C-0A10-0A11.lsabs, which is the delta of the contact attribute changes between the two full files.[<11>](#Appendix_A_11" \o "Product behavior note 11)
* The compact delta file is identical to the delta file with two exceptions. 1) only the attributes that changed for a contact are included, so unchanged attributes are ignored, and 2) attributes for the contact that no longer exist are marked as empty. [<12>](#Appendix_A_12" \o "Product behavior note 12)

The file that will be used in the example is the delta address book file, D-0A10-0A11.lsabs.

The relevant portion of Active Directory contents when F-0A10.lsabs is generated:

1. sn: ABSUser1\_lastname;
2. title: Development Manager;
3. physicalDeliveryOfficeName: 12345;
4. telephoneNumber: 555 391 3224;
5. givenName: ABSUser1\_firstname;
6. displayName: ABSUser1\_displayname;
7. otherTelephone: 555-533-4312;
8. company: TestCompany;
9. proxyAddresses: sip:ABSUser1@urtest.rtmp.selfhost.corp.proseware.com;
10. otherHomePhone: 555-391-3042;
11. objectGUID: 8c36ad0a-5e97-46dd-8d5b-255140c52b00;
12. mail: ABSUser1@urtest.com;
13. homePhone: 555-566-4312;
14. mobile: 555-533-4313;
15. msRTCSIP-PrimaryUserAddress: sip:ABSUser1@urtest.rtmp.selfhost.corp.proseware.com;
16. -----------
17. Dn: CN=ABSUser2,CN=Users,...
18. sn: ABSUser2\_lastname;
19. telephoneNumber: 555-783-4756;
20. givenName: ABSUser2\_firstname;
21. displayName: ABSUser2\_displayname;
22. proxyAddresses: sip:ABSUser2@urtest.rtmp.selfhost.corp.proseware.com;
23. objectGUID: 477c251b-db42-4ef6-9c69-fabbecc67f31;
24. msRTCSIP-PrimaryUserAddress: sip:ABSUser2@urtest.rtmp.selfhost.corp.proseware.com;
25. -----------
26. Dn: CN=ABSUser3,CN=Users,...
27. sn: ABSUser3\_lastname;
28. title: Program Manager;
29. telephoneNumber: 555-555-1234;
30. givenName: ABSUser3\_firstname;
31. displayName: ABSUser3\_displayname;
32. company: TestCompany;
33. proxyAddresses: sip:ABSUser3@urtest.rtmp.selfhost.corp.proseware.com;
34. otherHomePhone: 555-555-5678;
35. objectGUID: e335ddce-8a89-4e38-a4eb-116965911f4a;
36. mail: ABSUser3@urtest.com;
37. homePhone: 555-555-4321;
38. msRTCSIP-PrimaryUserAddress: sip:ABSUser3@urtest.rtmp.selfhost.corp.proseware.com;
39. -----------
40. Dn: CN=ABSUser4,CN=Users,...
41. sn: ABSUser4\_lastname;
42. title: Program Manager;
43. telephoneNumber: 555-555-8765;
44. givenName: ABSUser4\_firstname;
45. displayName: ABSUser4\_displayname;
46. company: TestCompany;
47. proxyAddresses: sip:ABSUser4@urtest.rtmp.selfhost.corp.proseware.com;
48. objectGUID: f02291c7-4c80-4956-8708-6bd526f47be5;
49. mail: ABSUser4@urtest.com;
50. mobile: 555-555-1111;
51. msRTCSIP-PrimaryUserAddress: sip:ABSUser4@urtest.rtmp.selfhost.corp.proseware.com;
52. -----------

The relevant portion of the Active Directory contents when F-0A11.lsabs is generated:

1. Dn: CN=ABSUser1,CN=Users,...
2. sn: ABSUser1\_lastname;
3. title: Development Manager;
4. physicalDeliveryOfficeName: 12345;
5. telephoneNumber: 555 391 3224;
6. givenName: ABSUser1\_firstname;
7. displayName: ABSUser1\_displayname\_changed;
8. otherTelephone: 555-533-4312;
9. company: TestCompany;
10. proxyAddresses: sip:ABSUser1@urtest.rtmp.selfhost.corp.proseware.com;
11. otherHomePhone: 555-391-3042;
12. objectGUID: 8c36ad0a-5e97-46dd-8d5b-255140c52b00;
13. mail: ABSUser1@urtest.com;
14. homePhone: 555-566-4312;
15. mobile: 555-533-4313;
16. msRTCSIP-PrimaryUserAddress: sip:ABSUser1@urtest.rtmp.selfhost.corp.proseware.com;
17. -----------
18. Dn: CN=ABSUser3,CN=Users,...
19. sn: ABSUser3\_lastname;
20. title: Program Manager;
21. telephoneNumber: 555-555-1234;
22. givenName: ABSUser3\_firstname;
23. displayName: ABSUser3\_displayname;
24. company: TestCompany;
25. proxyAddresses: sip:ABSUser3@urtest.rtmp.selfhost.corp.proseware.com;
26. otherHomePhone: 555-555-5678;
27. objectGUID: e335ddce-8a89-4e38-a4eb-116965911f4a;
28. mail: ABSUser3@urtest.com;
29. homePhone: 555-555-4321;
30. msRTCSIP-PrimaryUserAddress: sip:ABSUser3@urtest.rtmp.selfhost.corp.proseware.com;
31. -----------
32. Dn: CN=ABSUser4,CN=Users,...
33. sn: ABSUser4\_lastname;
34. title: Program Manager;
35. telephoneNumber: 555-555-8765;
36. givenName: ABSUser4\_firstname;
37. displayName: ABSUser4\_displayname;
38. company: TestCompany;
39. proxyAddresses: sip:ABSUser4@urtest.rtmp.selfhost.corp.proseware.com;
40. objectGUID: f02291c7-4c80-4956-8708-6bd526f47be5;
41. mail: ABSUser4@urtest.com;
42. mobile: 555-555-1111;
43. msRTCSIP-PrimaryUserAddress: sip:ABSUser4@urtest.rtmp.selfhost.corp.proseware.com;
44. -----------
45. Dn: CN=ABSUser5,CN=Users,...
46. sn: ABSUser5\_lastname;
47. telephoneNumber: 555-789-6666;
48. givenName: ABSUser5\_firstname;
49. displayName: ABSUser5\_displayname;
50. proxyAddresses: sip:ABSUser5@urtest.rtmp.selfhost.corp.proseware.com;
51. objectGUID: e1e8410b-c8c4-4022-93ba-b2145ed6d134;
52. msRTCSIP-PrimaryUserAddress: sip:ABSUser5@urtest.rtmp.selfhost.corp.proseware.com;
53. -----------

The simple text difference of the two preceding Active Directory listings follows, to illustrate what was changed. The **displayName** attribute for ABSUser1 was changed, ABSUser2 was deleted, and a new user, ABSUser5, was added.

1. 7c7
2. < displayName: ABSUser1\_displayname;
3. ---
4. > displayName: ABSUser1\_displayname\_changed;
5. 19,28d18
6. < Dn: CN=ABSUser2,CN=Users,...
7. < sn: ABSUser2\_lastname;
8. < telephoneNumber: 555-783-4756;
9. < givenName: ABSUser2\_firstname;
10. < displayName: ABSUser2\_displayname;
11. < proxyAddresses: sip:ABSUser2@urtest.rtmp.selfhost.corp.proseware.com;
12. < objectGUID: 477c251b-db42-4ef6-9c69-fabbecc67f31;
13. < msRTCSIP-PrimaryUserAddress: sip:ABSUser2@urtest.rtmp.selfhost.corp.proseware.com;
14. < -----------
15. <
16. 56a47,56
17. >
18. > Dn: CN=ABSUser5,CN=Users,...
19. > sn: ABSUser5\_lastname;
20. > telephoneNumber: 555-789-6666;
21. > givenName: ABSUser5\_firstname;
22. > displayName: ABSUser5\_displayname;
23. > proxyAddresses: sip:ABSUser5@urtest.rtmp.selfhost.corp.proseware.com;
24. > objectGUID: e1e8410b-c8c4-4022-93ba-b2145ed6d134;
25. > msRTCSIP-PrimaryUserAddress: sip:ABSUser5@urtest.rtmp.selfhost.corp.proseware.com;
26. > -----------

The contents of this delta address book file example follow, in hexadecimal bytes. The far-left column is the byte count; the far-right characters are the interpretation of the bytes in the [**American National Standards Institute (ANSI) character set**](#gt_100cd8a6-5cb1-4895-9de6-e4a3c224a583). The sections that follow describe the structures that convey this series of bytes. This hexadecimal dump occurred after the actual D-0A10-0A11.lsabs file was decompressed. For a description of how the file is decompressed, see section [5](#Section_5c0d93f605b1421fb971ab547414a6d2).

1. 00000000: 16 c1 4b b5 08 90 c7 47 b9 bd f3 bb 1a 0a b6 eb ..K....G........
2. 00000010: 10 0a 11 0a 14 00 14 00 01 00 00 00 00 00 00 00 ................
3. 00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
4. 00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
5. 00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
6. 00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
7. 00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
8. 00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
9. 00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
10. 00000090: 00 00 00 00 00 00 00 00 00 00 5b 08 00 00 2e 2a ..........[....\*
11. 000000a0: 38 38 32 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2a 38 882[\s()\-\./]\*8
12. 000000b0: 30 38 30 2e 2a 5b 58 78 5d 2b 5b 5c 73 28 29 5c 080.\*[Xx]+[\s()\
13. 000000c0: 2d 5c 2e 2f 5d 2a 28 5c 64 5c 64 5c 64 5c 64 5c -\./]\*(\d\d\d\d\
14. 000000d0: 64 29 0d 0a 24 31 3b 70 68 6f 6e 65 2d 63 6f 6e d)..$1;phone-con
15. 000000e0: 74 65 78 74 3d 70 72 6F 73 65 77 61 72 65 2e 63 text=proseware.c
16. 000000f0: 6f 6d 0d 0a 2e 2a 38 38 32 5b 5c 73 28 29 5c 2d om...\*882[\s()\-
17. 00000100: 5c 2e 2f 5d 2a 38 30 38 30 2e 2a 0d 0a 6e 75 6c \./]\*8080.\*..nul
18. 00000110: 6c 0d 0a 5c 28 28 5c 2b 5c 73 2a 31 29 3f 5c 29 l..\((\+\s\*1)?\)
19. 00000120: 5b 5c 73 5c 2d 5c 2e 5d 2a 5c 28 3f 28 5c 64 5c [\s\-\.]\*\(?(\d\
20. 00000130: 64 5c 64 29 5c 73 2a 5c 29 3f 5b 5c 73 28 29 5c d\d)\s\*\)?[\s()\
21. 00000140: 2d 5c 2e 2f 5d 2a 28 5c 64 5c 64 5c 64 29 5b 5c -\./]\*(\d\d\d)[\
22. 00000150: 73 28 29 5c 2d 5c 2e 2f 5d 2a 28 5c 64 5c 64 5c s()\-\./]\*(\d\d\
23. 00000160: 64 5c 64 29 5c 73 2a 5b 58 78 5d 2b 5b 5c 73 28 d\d)\s\*[Xx]+[\s(
24. 00000170: 29 5c 2d 5c 2e 2f 5d 2a 28 5c 64 5c 64 5c 64 5c )\-\./]\*(\d\d\d\
25. 00000180: 64 5c 64 29 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2a d\d)[\s()\-\./]\*
26. 00000190: 0d 0a 2b 31 24 32 24 33 24 34 3b 65 78 74 3d 24 ..+1$2$3$4;ext=$
27. 000001a0: 35 0d 0a 28 5c 2b 5c 73 2a 31 29 3f 5b 5c 73 5c 5..(\+\s\*1)?[\s\
28. 000001b0: 2d 5c 2e 5d 2a 5c 28 3f 28 5c 64 5c 64 5c 64 29 -\.]\*\(?(\d\d\d)
29. 000001c0: 5c 73 2a 5c 29 3f 5b 5c 73 28 29 5c 2d 5c 2e 2f \s\*\)?[\s()\-\./
30. 000001d0: 5d 2a 28 5c 64 5c 64 5c 64 29 5b 5c 73 28 29 5c ]\*(\d\d\d)[\s()\
31. 000001e0: 2d 5c 2e 2f 5d 2a 28 5c 64 5c 64 5c 64 5c 64 29 -\./]\*(\d\d\d\d)
32. 000001f0: 5c 73 2a 45 58 54 5c 73 2a 28 5c 64 5c 64 5c 64 \s\*EXT\s\*(\d\d\d
33. 00000200: 5c 64 5c 64 29 0d 0a 2b 31 24 32 24 33 24 34 3b \d\d)..+1$2$3$4;
34. 00000210: 65 78 74 3d 24 35 0d 0a 28 5c 2b 5c 73 2a 31 29 ext=$5..(\+\s\*1)
35. 00000220: 3f 5b 5c 73 5c 2d 5c 2e 5d 2a 5c 28 3f 28 5c 64 ?[\s\-\.]\*\(?(\d
36. 00000230: 5c 64 5c 64 29 5c 73 2a 5c 29 3f 5b 5c 73 28 29 \d\d)\s\*\)?[\s()
37. 00000240: 5c 2d 5c 2e 2f 5d 2a 28 5c 64 5c 64 5c 64 29 5b \-\./]\*(\d\d\d)[
38. 00000250: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2a 28 5c 64 5c 64 \s()\-\./]\*(\d\d
39. 00000260: 5c 64 5c 64 29 5c 73 2a 5b 78 58 5d 5c 73 2a 28 \d\d)\s\*[xX]\s\*(
40. 00000270: 5c 64 5c 64 5c 64 5c 64 5c 64 29 0d 0a 2b 31 24 \d\d\d\d\d)..+1$
41. 00000280: 32 24 33 24 34 3b 65 78 74 3d 24 35 0d 0a 28 5c 2$3$4;ext=$5..(\
42. 00000290: 73 2a 31 29 5b 5c 73 5c 2d 5c 2e 5d 2a 5c 28 3f s\*1)[\s\-\.]\*\(?
43. 000002a0: 5c 73 2a 28 5c 64 5c 64 5c 64 29 5c 73 2a 5c 29 \s\*(\d\d\d)\s\*\)
44. 000002b0: 3f 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2a 28 5c 64 ?[\s()\-\./]\*(\d
45. 000002c0: 5c 64 5c 64 29 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d \d\d)[\s()\-\./]
46. 000002d0: 2a 28 5c 64 5c 64 5c 64 5c 64 29 5b 5c 73 5d 2a \*(\d\d\d\d)[\s]\*
47. 000002e0: 0d 0a 2b 31 24 32 24 33 24 34 0d 0a 28 5c 2b 5c ..+1$2$3$4..(\+\
48. 000002f0: 73 2a 31 29 3f 5b 5c 73 5c 2d 5c 2e 5d 2a 5c 28 s\*1)?[\s\-\.]\*\(
49. 00000300: 3f 28 5c 64 5c 64 5c 64 29 5c 73 2a 5c 29 3f 5b ?(\d\d\d)\s\*\)?[
50. 00000310: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2a 37 30 28 5c 64 \s()\-\./]\*70(\d
51. 00000320: 5c 64 5c 64 5c 64 5c 64 29 0d 0a 2b 31 24 32 37 \d\d\d\d)..+1$27
52. 00000330: 30 24 33 3b 65 78 74 3d 24 33 0d 0a 37 30 28 5c 0$3;ext=$3..70(\
53. 00000340: 64 29 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2a 28 5c d)[\s()\-\./]\*(\
54. 00000350: 64 5c 64 5c 64 5c 64 29 5c 73 2a 5b 58 78 5d 2b d\d\d\d)\s\*[Xx]+
55. 00000360: 28 5c 64 5c 64 5c 64 5c 64 5c 64 29 0d 0a 2b 31 (\d\d\d\d\d)..+1
56. 00000370: 34 32 35 37 30 24 31 24 32 3b 65 78 74 3d 24 33 42570$1$2;ext=$3
57. 00000380: 0d 0a 28 5c 64 5c 64 5c 64 29 5b 5c 73 28 29 5c ..(\d\d\d)[\s()\
58. 00000390: 2d 5c 2e 2f 5d 2a 28 5c 64 5c 64 5c 64 5c 64 29 -\./]\*(\d\d\d\d)
59. 000003a0: 5c 73 2a 5b 58 78 5d 2b 28 5c 64 5c 64 5c 64 5c \s\*[Xx]+(\d\d\d\
60. 000003b0: 64 5c 64 29 0d 0a 2b 31 34 32 35 24 31 24 32 3b d\d)..+1425$1$2;
61. 000003c0: 65 78 74 3d 24 33 0d 0a 28 5c 64 5c 64 5c 64 29 ext=$3..(\d\d\d)
62. 000003d0: 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2a 28 5c 64 5c [\s()\-\./]\*(\d\
63. 000003e0: 64 5c 64 5c 64 29 0d 0a 2b 31 34 32 35 24 31 24 d\d\d)..+1425$1$
64. 000003f0: 32 0d 0a 5b 58 78 5d 5c 73 2a 28 5c 64 5c 64 5c 2..[Xx]\s\*(\d\d\
65. 00000400: 64 5c 64 5c 64 29 0d 0a 24 31 3b 70 68 6f 6e 65 d\d\d)..$1;phone
66. 00000410: 2d 63 6f 6e 74 65 78 74 3d 70 72 6F 73 65 77 61 -context=prosewa
67. 00000420: 72 65 2e 63 6f 6d 0d 0a 5c 28 3f 28 5c 64 5c 64 re.com..\(?(\d\d
68. 00000430: 29 5c 73 2a 5c 29 3f 5c 73 2a 5b 58 78 5d 2b 5c )\s\*\)?\s\*[Xx]+\
69. 00000440: 73 2a 28 5c 64 5c 64 5c 64 5c 64 5c 64 29 0d 0a s\*(\d\d\d\d\d)..
70. 00000450: 2b 31 34 32 35 24 31 24 32 3b 65 78 74 3d 24 32 +1425$1$2;ext=$2
71. 00000460: 0d 0a 30 31 31 28 5c 64 2b 29 28 5b 5c 73 28 29 ..011(\d+)([\s()
72. 00000470: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
73. 00000480: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
74. 00000490: 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 )?([\s()\-\./]+(
75. 000004a0: 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e \d+))?([\s()\-\.
76. 000004b0: 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 /]+(\d+))?([\s()
77. 000004c0: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
78. 000004d0: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
79. 000004e0: 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 )?([\s()\-\./]+(
80. 000004f0: 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e \d+))?([\s()\-\.
81. 00000500: 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 /]+(\d+))?([\s()
82. 00000510: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
83. 00000520: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
84. 00000530: 29 3f 5c 73 2a 5b 58 78 5d 2b 28 5c 64 7b 31 2c )?\s\*[Xx]+(\d{1,
85. 00000540: 31 35 7d 29 5b 5c 73 5d 2a 0d 0a 2b 24 31 24 33 15})[\s]\*..+$1$3
86. 00000550: 24 35 24 37 24 39 24 31 31 24 31 33 24 31 35 24 $5$7$9$11$13$15$
87. 00000560: 31 37 24 31 39 24 32 31 3b 65 78 74 3d 24 32 32 17$19$21;ext=$22
88. 00000570: 0d 0a 30 31 31 28 5c 64 2b 29 28 5b 5c 73 28 29 ..011(\d+)([\s()
89. 00000580: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
90. 00000590: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
91. 000005a0: 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 )?([\s()\-\./]+(
92. 000005b0: 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e \d+))?([\s()\-\.
93. 000005c0: 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 /]+(\d+))?([\s()
94. 000005d0: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
95. 000005e0: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
96. 000005f0: 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 )?([\s()\-\./]+(
97. 00000600: 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e \d+))?([\s()\-\.
98. 00000610: 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 /]+(\d+))?([\s()
99. 00000620: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
100. 00000630: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
101. 00000640: 29 3f 5b 5c 73 5d 2a 0d 0a 2b 24 31 24 33 24 35 )?[\s]\*..+$1$3$5
102. 00000650: 24 37 24 39 24 31 31 24 31 33 24 31 35 24 31 37 $7$9$11$13$15$17
103. 00000660: 24 31 39 24 32 31 0d 0a 28 5c 64 2b 29 28 5b 5c $19$21..(\d+)([\
104. 00000670: 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 s()\-\./]+(\d+))
105. 00000680: 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 ([\s()\-\./]+(\d
106. 00000690: 2b 29 29 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b +))([\s()\-\./]+
107. 000006a0: 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 5c 2d 5c (\d+))?([\s()\-\
108. 000006b0: 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 ./]+(\d+))?([\s(
109. 000006c0: 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 5b )\-\./]+(\d+))?[
110. 000006d0: 5c 73 5d 2a 0d 0a 2b 24 31 24 33 24 35 24 37 24 \s]\*..+$1$3$5$7$
111. 000006e0: 39 24 31 31 0d 0a 45 31 36 34 0d 0a 6e 75 6c 6c 9$11..E164..null
112. 000006f0: 0d 0a 5c 2b 2b 28 5c 64 2b 29 28 5b 5c 73 28 29 ..\++(\d+)([\s()
113. 00000700: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
114. 00000710: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
115. 00000720: 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 )?([\s()\-\./]+(
116. 00000730: 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e \d+))?([\s()\-\.
117. 00000740: 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 /]+(\d+))?([\s()
118. 00000750: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
119. 00000760: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
120. 00000770: 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 )?([\s()\-\./]+(
121. 00000780: 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e \d+))?([\s()\-\.
122. 00000790: 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 /]+(\d+))?([\s()
123. 000007a0: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
124. 000007b0: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
125. 000007c0: 29 3f 5c 73 2a 5b 58 78 5d 2b 28 5c 64 7b 31 2c )?\s\*[Xx]+(\d{1,
126. 000007d0: 31 35 7d 29 5b 5c 73 5d 2a 0d 0a 2b 24 31 24 33 15})[\s]\*..+$1$3
127. 000007e0: 24 35 24 37 24 39 24 31 31 24 31 33 24 31 35 24 $5$7$9$11$13$15$
128. 000007f0: 31 37 24 31 39 24 32 31 3b 65 78 74 3d 24 32 32 17$19$21;ext=$22
129. 00000800: 0d 0a 5c 2b 2b 28 5c 64 2b 29 28 5b 5c 73 28 29 ..\++(\d+)([\s()
130. 00000810: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
131. 00000820: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
132. 00000830: 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 )?([\s()\-\./]+(
133. 00000840: 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e \d+))?([\s()\-\.
134. 00000850: 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 /]+(\d+))?([\s()
135. 00000860: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
136. 00000870: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
137. 00000880: 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 )?([\s()\-\./]+(
138. 00000890: 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 5c 2d 5c 2e \d+))?([\s()\-\.
139. 000008a0: 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b 5c 73 28 29 /]+(\d+))?([\s()
140. 000008b0: 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 29 3f 28 5b \-\./]+(\d+))?([
141. 000008c0: 5c 73 28 29 5c 2d 5c 2e 2f 5d 2b 28 5c 64 2b 29 \s()\-\./]+(\d+)
142. 000008d0: 29 3f 5b 5c 73 5d 2a 0d 0a 2b 24 31 24 33 24 35 )?[\s]\*..+$1$3$5
143. 000008e0: 24 37 24 39 24 31 31 24 31 33 24 31 35 24 31 37 $7$9$11$13$15$17
144. 000008f0: 24 31 39 24 32 31 0d 0a 00 0e 00 14 00 00 00 00 $19$21..........
145. 00000900: 10 6d 61 6e 61 67 65 72 00 10 00 13 00 01 08 01 .manager........
146. 00000910: 0f 67 72 6f 75 70 54 79 70 65 00 15 00 12 00 02 .groupType......
147. 00000920: 01 00 00 70 72 6f 78 79 41 64 64 72 65 73 73 65 ...proxyAddresse
148. 00000930: 73 00 0b 00 11 00 00 00 00 0e 6d 61 69 6c 00 0e s.........mail..
149. 00000940: 00 10 00 00 20 00 0d 69 70 50 68 6f 6e 65 00 15 .... ..ipPhone..
150. 00000950: 00 0f 00 00 20 00 0c 6f 74 68 65 72 54 65 6c 65 .... ..otherTele
151. 00000960: 70 68 6f 6e 65 00 12 00 0e 00 00 20 00 0b 6f 74 phone...... ..ot
152. 00000970: 68 65 72 4d 6f 62 69 6c 65 00 0d 00 0d 00 00 28 herMobile......(
153. 00000980: 02 0b 6d 6f 62 69 6c 65 00 15 00 0c 00 00 20 00 ..mobile...... .
154. 00000990: 0a 6f 74 68 65 72 48 6f 6d 65 50 68 6f 6e 65 00 .otherHomePhone.
155. 000009a0: 10 00 0b 00 00 28 00 0a 68 6f 6d 65 50 68 6f 6e .....(..homePhon
156. 000009b0: 65 00 16 00 0a 00 00 28 02 09 74 65 6c 65 70 68 e......(..teleph
157. 000009c0: 6f 6e 65 4e 75 6d 62 65 72 00 22 00 09 00 00 08 oneNumber.".....
158. 000009d0: 02 08 6d 73 52 54 43 53 49 50 2d 50 72 69 6d 61 ..msRTCSIP-Prima
159. 000009e0: 72 79 55 73 65 72 41 64 64 72 65 73 73 00 21 00 ryUserAddress.!.
160. 000009f0: 08 00 00 00 00 07 70 68 79 73 69 63 61 6c 44 65 ......physicalDe
161. 00000a00: 6c 69 76 65 72 79 4f 66 66 69 63 65 4e 61 6d 65 liveryOfficeName
162. 00000a10: 00 0e 00 07 00 00 00 00 06 63 6f 6d 70 61 6e 79 .........company
163. 00000a20: 00 13 00 06 00 00 04 00 05 6d 61 69 6c 4e 69 63 .........mailNic
164. 00000a30: 6b 6e 61 6d 65 00 0c 00 05 00 00 00 00 04 74 69 kname.........ti
165. 00000a40: 74 6c 65 00 12 00 04 00 00 00 02 03 64 69 73 70 tle.........disp
166. 00000a50: 6c 61 79 4e 61 6d 65 00 09 00 03 00 00 00 00 02 layName.........
167. 00000a60: 73 6e 00 10 00 02 00 00 00 00 01 67 69 76 65 6e sn.........given
168. 00000a70: 4e 61 6d 65 00 21 00 01 00 00 00 00 ff 6d 73 45 Name.!.......msE
169. 00000a80: 78 63 68 48 69 64 65 46 72 6f 6d 41 64 64 72 65 xchHideFromAddre
170. 00000a90: 73 73 4c 69 73 74 73 00 66 01 0a ad 36 8c 97 5e ssLists.f...6..^
171. 00000aa0: dd 46 8d 5b 25 51 40 c5 2b 00 12 00 11 41 42 53 .F.[%Q@.+....ABS
172. 00000ab0: 55 73 65 72 31 40 75 72 74 65 73 74 2e 63 6f 6d User1@urtest.com
173. 00000ac0: 00 0f 35 35 35 2d 35 33 33 2d 34 33 31 32 00 0f ..555-533-4312..
174. 00000ad0: 74 65 6c 3a 2b 35 35 35 35 33 33 34 33 31 32 00 tel:+5555334312.
175. 00000ae0: 0d 35 35 35 2d 35 33 33 2d 34 33 31 33 00 0d 74 .555-533-4313..t
176. 00000af0: 65 6c 3a 2b 35 35 35 35 33 33 34 33 31 33 00 0c el:+5555334313..
177. 00000b00: 35 35 35 2d 33 39 31 2d 33 30 34 32 00 0c 74 65 555-391-3042..te
178. 00000b10: 6c 3a 2b 35 35 35 33 39 31 33 30 34 32 00 0b 35 l:+5553913042..5
179. 00000b20: 35 35 2d 35 36 36 2d 34 33 31 32 00 0b 74 65 6c 55-566-4312..tel
180. 00000b30: 3a 2b 35 35 35 35 36 36 34 33 31 32 00 0a 35 35 :+5555664312..55
181. 00000b40: 35 20 33 39 31 20 33 32 32 34 00 0a 74 65 6c 3a 5 391 3224..tel:
182. 00000b50: 2b 35 35 35 33 39 31 33 32 32 34 00 09 73 69 70 +5553913224..sip
183. 00000b60: 3a 41 42 53 55 73 65 72 31 40 75 72 74 65 73 74 :ABSUser1@urtest
184. 00000b70: 2e 72 74 6d 70 2e 73 65 6c 66 68 6f 73 74 2e 63 .rtmp.selfhost.c
185. 00000b80: 6f 72 70 2e 70 72 6F 73 65 77 61 72 65 2e 63 6f orp.proseware.co
186. 00000b90: 6d 00 08 31 32 33 34 35 00 07 54 65 73 74 43 6f m..12345..TestCo
187. 00000ba0: 6d 70 61 6e 79 00 05 44 65 76 65 6c 6f 70 6d 65 mpany..Developme
188. 00000bb0: 6e 74 20 4d 61 6e 61 67 65 72 00 04 41 42 53 55 nt Manager..ABSU
189. 00000bc0: 73 65 72 31 5f 64 69 73 70 6c 61 79 6e 61 6d 65 ser1\_displayname
190. 00000bd0: 5f 63 68 61 6e 67 65 64 00 03 41 42 53 55 73 65 \_changed..ABSUse
191. 00000be0: 72 31 5f 6c 61 73 74 6e 61 6d 65 00 02 41 42 53 r1\_lastname..ABS
192. 00000bf0: 55 73 65 72 31 5f 66 69 72 73 74 6e 61 6d 65 00 User1\_firstname.
193. 00000c00: a4 00 0b 41 e8 e1 c4 c8 22 40 93 ba b2 14 5e d6 ...A...."@....^.
194. 00000c10: d1 34 06 00 0a 35 35 35 2d 37 38 39 2d 36 36 36 .4...555-789-666
195. 00000c20: 36 00 0a 74 65 6c 3a 2b 35 35 35 37 38 39 36 36 6..tel:+55578966
196. 00000c30: 36 36 00 09 73 69 70 3a 41 42 53 55 73 65 72 35 66..sip:ABSUser5
197. 00000c40: 40 75 72 74 65 73 74 2e 72 74 6d 70 2e 73 65 6c @urtest.rtmp.sel
198. 00000c50: 66 68 6f 73 74 2e 63 6f 72 70 2e 70 72 6F 73 65 fhost.corp.prose
199. 00000c60: 77 61 72 65 2e 63 6f 6d 00 04 41 42 53 55 73 65 ware.com..ABSUse
200. 00000c70: 72 35 5f 64 69 73 70 6c 61 79 6e 61 6d 65 00 03 r5\_displayname..
201. 00000c80: 41 42 53 55 73 65 72 35 5f 6c 61 73 74 6e 61 6d ABSUser5\_lastnam
202. 00000c90: 65 00 02 41 42 53 55 73 65 72 35 5f 66 69 72 73 e..ABSUser5\_firs
203. 00000ca0: 74 6e 61 6d 65 00 a6 00 1b 25 7c 47 42 db f6 4e tname....%|GB..N
204. 00000cb0: 9c 69 fa bb ec c6 7f 31 00 00 06 00 0a 35 35 35 .i.....1.....555
205. 00000cc0: 2d 37 38 33 2d 34 37 35 36 00 0a 74 65 6c 3a 2b -783-4756..tel:+
206. 00000cd0: 35 35 35 37 38 33 34 37 35 36 00 09 73 69 70 3a 5557834756..sip:
207. 00000ce0: 41 42 53 55 73 65 72 32 40 75 72 74 65 73 74 2e ABSUser2@urtest.
208. 00000cf0: 72 74 6d 70 2e 73 65 6c 66 68 6f 73 74 2e 63 6f rtmp.selfhost.co
209. 00000d00: 72 70 2e 70 72 6F 73 65 77 61 72 65 2e 63 6f 6d rp.proseware.com
210. 00000d10: 00 04 41 42 53 55 73 65 72 32 5f 64 69 73 70 6c ..ABSUser2\_displ
211. 00000d20: 61 79 6e 61 6d 65 00 03 41 42 53 55 73 65 72 32 ayname..ABSUser2
212. 00000d30: 5f 6c 61 73 74 6e 61 6d 65 00 02 41 42 53 55 73 \_lastname..ABSUs
213. 00000d40: 65 72 32 5f 66 69 72 73 74 6e 61 6d 65 00 12 00 er2\_firstname...
214. 00000d50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
215. 00000d60: 00 00 db df d2 dd 02 00 00 00 01 00 00 00 0c 00 ................
216. 00000d70: 00 00 ..

### ABF\_DELTA\_HEADER

This section provides an example of the address book file **ABF\_DELTA\_HEADER** structure specified in section [2.2.4](#Section_ec9e5b68d82a46b5adb8242ef10c005a).

1. 00000000: 16 C1 4B B5 08 90 C7 47 B9 BD F3 BB 1A 0A B6 EB ..K....G........
2. 00000010: 10 0A 11 0A 14 00 14 00 01 00 00 00 00 00 00 00 ................
3. 00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
4. 00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
5. 00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
6. 00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
7. 00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
8. 00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
9. 00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
10. 00000090: 00 00 00 00 00 00 00 00 00 00 ..........

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| FileType (0xB54BC116) |
| 0x47C79008 |
| 0xBBF3BDB9 |
| 0xEBB60A1A |
| CreationDate (0x0A11) | BaseCreationDate (0x0A10) |
| MaximumAttributeId (0x0014) | NumberOfAttributes (0x0014) |
| SourceStream (0x0000) | UseNormalizationRules (0x0001) |
| 126 more bytes of 0x00 |

FileType: The GUID value {b54bc116-9008-47c7-b9bd-f3bb1a0ab6eb} indicates that this is a [**delta address book file**](#gt_ee5b6fad-5c27-49d0-a20d-0eceda2baf3a).

**BaseCreationDate:** 0x0A10 specifies the base full file (F-0A10.lsabs) that was used to create this delta file.

**CreationDate:** 0x0A11 specifies the date that this delta file (D-0A10-0A11.lsabs) was created, as well as the full file (F-0A11.lsabs) that was compared against the base full file (F-0A10.lsabs) to produce this delta file.

**NumberOfAttributes:** 0x0014 specifies the number of **ABF\_ATTRIBUTE** structures in the **ABF\_ATTRIBUTES** structure.

**MaximumAttributeId:** 0x0014 specifies the largest value of the **Id** field in all of the **ABF\_ATTRIBUTE** structures in the **ABF\_ATTRIBUTES** structure.

**UseNormalizationRules:** 0x0001 indicates that the Address Book Server did normalize phone numbers using the phone normalization rules in the **ABF\_NORMALIZATION\_RULES** structure.

**SourceStream:** 128 bytes of 0x00 are ignored by the client.

### ABF\_NORMALIZATION\_RULES

This section provides an example of the address book file **ABF\_NORMALIZATION\_RULES** structure specified in section [2.2.5](#Section_1699d600cec8415cb59d05db90be0fb2).

1. 00000090: 5B 08 00 00 2E 2A [....\*
2. 000000A0: 38 38 32 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2A 38 882[\s()\-\./]\*8
3. 000000B0: 30 38 30 2E 2A 5B 58 78 5D 2B 5B 5C 73 28 29 5C 080.\*[Xx]+[\s()\
4. 000000C0: 2D 5C 2E 2F 5D 2A 28 5C 64 5C 64 5C 64 5C 64 5C -\./]\*(\d\d\d\d\
5. 000000D0: 64 29 0D 0A 24 31 3B 70 68 6F 6E 65 2D 63 6F 6E d)..$1;phone-con
6. 000000E0: 74 65 78 74 3D 70 72 6F 73 65 77 61 72 65 2E 63 text=proseware.c
7. 000000F0: 6F 6D 0D 0A 2E 2A 38 38 32 5B 5C 73 28 29 5C 2D om...\*882[\s()\-
8. 00000100: 5C 2E 2F 5D 2A 38 30 38 30 2E 2A 0D 0A 6E 75 6C \./]\*8080.\*..nul
9. 00000110: 6C 0D 0A 5C 28 28 5C 2B 5C 73 2A 31 29 3F 5C 29 l..\((\+\s\*1)?\)
10. 00000120: 5B 5C 73 5C 2D 5C 2E 5D 2A 5C 28 3F 28 5C 64 5C [\s\-\.]\*\(?(\d\
11. 00000130: 64 5C 64 29 5C 73 2A 5C 29 3F 5B 5C 73 28 29 5C d\d)\s\*\)?[\s()\
12. 00000140: 2D 5C 2E 2F 5D 2A 28 5C 64 5C 64 5C 64 29 5B 5C -\./]\*(\d\d\d)[\
13. 00000150: 73 28 29 5C 2D 5C 2E 2F 5D 2A 28 5C 64 5C 64 5C s()\-\./]\*(\d\d\
14. 00000160: 64 5C 64 29 5C 73 2A 5B 58 78 5D 2B 5B 5C 73 28 d\d)\s\*[Xx]+[\s(
15. 00000170: 29 5C 2D 5C 2E 2F 5D 2A 28 5C 64 5C 64 5C 64 5C )\-\./]\*(\d\d\d\
16. 00000180: 64 5C 64 29 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2A d\d)[\s()\-\./]\*
17. 00000190: 0D 0A 2B 31 24 32 24 33 24 34 3B 65 78 74 3D 24 ..+1$2$3$4;ext=$
18. 000001A0: 35 0D 0A 28 5C 2B 5C 73 2A 31 29 3F 5B 5C 73 5C 5..(\+\s\*1)?[\s\
19. 000001B0: 2D 5C 2E 5D 2A 5C 28 3F 28 5C 64 5C 64 5C 64 29 -\.]\*\(?(\d\d\d)
20. 000001C0: 5C 73 2A 5C 29 3F 5B 5C 73 28 29 5C 2D 5C 2E 2F \s\*\)?[\s()\-\./
21. 000001D0: 5D 2A 28 5C 64 5C 64 5C 64 29 5B 5C 73 28 29 5C ]\*(\d\d\d)[\s()\
22. 000001E0: 2D 5C 2E 2F 5D 2A 28 5C 64 5C 64 5C 64 5C 64 29 -\./]\*(\d\d\d\d)
23. 000001F0: 5C 73 2A 45 58 54 5C 73 2A 28 5C 64 5C 64 5C 64 \s\*EXT\s\*(\d\d\d
24. 00000200: 5C 64 5C 64 29 0D 0A 2B 31 24 32 24 33 24 34 3B \d\d)..+1$2$3$4;
25. 00000210: 65 78 74 3D 24 35 0D 0A 28 5C 2B 5C 73 2A 31 29 ext=$5..(\+\s\*1)
26. 00000220: 3F 5B 5C 73 5C 2D 5C 2E 5D 2A 5C 28 3F 28 5C 64 ?[\s\-\.]\*\(?(\d
27. 00000230: 5C 64 5C 64 29 5C 73 2A 5C 29 3F 5B 5C 73 28 29 \d\d)\s\*\)?[\s()
28. 00000240: 5C 2D 5C 2E 2F 5D 2A 28 5C 64 5C 64 5C 64 29 5B \-\./]\*(\d\d\d)[
29. 00000250: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2A 28 5C 64 5C 64 \s()\-\./]\*(\d\d
30. 00000260: 5C 64 5C 64 29 5C 73 2A 5B 78 58 5D 5C 73 2A 28 \d\d)\s\*[xX]\s\*(
31. 00000270: 5C 64 5C 64 5C 64 5C 64 5C 64 29 0D 0A 2B 31 24 \d\d\d\d\d)..+1$
32. 00000280: 32 24 33 24 34 3B 65 78 74 3D 24 35 0D 0A 28 5C 2$3$4;ext=$5..(\
33. 00000290: 73 2A 31 29 5B 5C 73 5C 2D 5C 2E 5D 2A 5C 28 3F s\*1)[\s\-\.]\*\(?
34. 000002A0: 5C 73 2A 28 5C 64 5C 64 5C 64 29 5C 73 2A 5C 29 \s\*(\d\d\d)\s\*\)
35. 000002B0: 3F 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2A 28 5C 64 ?[\s()\-\./]\*(\d
36. 000002C0: 5C 64 5C 64 29 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D \d\d)[\s()\-\./]
37. 000002D0: 2A 28 5C 64 5C 64 5C 64 5C 64 29 5B 5C 73 5D 2A \*(\d\d\d\d)[\s]\*
38. 000002E0: 0D 0A 2B 31 24 32 24 33 24 34 0D 0A 28 5C 2B 5C ..+1$2$3$4..(\+\
39. 000002F0: 73 2A 31 29 3F 5B 5C 73 5C 2D 5C 2E 5D 2A 5C 28 s\*1)?[\s\-\.]\*\(
40. 00000300: 3F 28 5C 64 5C 64 5C 64 29 5C 73 2A 5C 29 3F 5B ?(\d\d\d)\s\*\)?[
41. 00000310: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2A 37 30 28 5C 64 \s()\-\./]\*70(\d
42. 00000320: 5C 64 5C 64 5C 64 5C 64 29 0D 0A 2B 31 24 32 37 \d\d\d\d)..+1$27
43. 00000330: 30 24 33 3B 65 78 74 3D 24 33 0D 0A 37 30 28 5C 0$3;ext=$3..70(\
44. 00000340: 64 29 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2A 28 5C d)[\s()\-\./]\*(\
45. 00000350: 64 5C 64 5C 64 5C 64 29 5C 73 2A 5B 58 78 5D 2B d\d\d\d)\s\*[Xx]+
46. 00000360: 28 5C 64 5C 64 5C 64 5C 64 5C 64 29 0D 0A 2B 31 (\d\d\d\d\d)..+1
47. 00000370: 34 32 35 37 30 24 31 24 32 3B 65 78 74 3D 24 33 42570$1$2;ext=$3
48. 00000380: 0D 0A 28 5C 64 5C 64 5C 64 29 5B 5C 73 28 29 5C ..(\d\d\d)[\s()\
49. 00000390: 2D 5C 2E 2F 5D 2A 28 5C 64 5C 64 5C 64 5C 64 29 -\./]\*(\d\d\d\d)
50. 000003A0: 5C 73 2A 5B 58 78 5D 2B 28 5C 64 5C 64 5C 64 5C \s\*[Xx]+(\d\d\d\
51. 000003B0: 64 5C 64 29 0D 0A 2B 31 34 32 35 24 31 24 32 3B d\d)..+1425$1$2;
52. 000003C0: 65 78 74 3D 24 33 0D 0A 28 5C 64 5C 64 5C 64 29 ext=$3..(\d\d\d)
53. 000003D0: 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2A 28 5C 64 5C [\s()\-\./]\*(\d\
54. 000003E0: 64 5C 64 5C 64 29 0D 0A 2B 31 34 32 35 24 31 24 d\d\d)..+1425$1$
55. 000003F0: 32 0D 0A 5B 58 78 5D 5C 73 2A 28 5C 64 5C 64 5C 2..[Xx]\s\*(\d\d\
56. 00000400: 64 5C 64 5C 64 29 0D 0A 24 31 3B 70 68 6F 6E 65 d\d\d)..$1;phone
57. 00000410: 2D 63 6F 6E 74 65 78 74 3D 70 72 6F 73 65 77 61 -context=prosewa
58. 00000420: 72 65 2E 63 6F 6D 0D 0A 5C 28 3F 28 5C 64 5C 64 re.com..\(?(\d\d
59. 00000430: 29 5C 73 2A 5C 29 3F 5C 73 2A 5B 58 78 5D 2B 5C )\s\*\)?\s\*[Xx]+\
60. 00000440: 73 2A 28 5C 64 5C 64 5C 64 5C 64 5C 64 29 0D 0A s\*(\d\d\d\d\d)..
61. 00000450: 2B 31 34 32 35 24 31 24 32 3B 65 78 74 3D 24 32 +1425$1$2;ext=$2
62. 00000460: 0D 0A 30 31 31 28 5C 64 2B 29 28 5B 5C 73 28 29 ..011(\d+)([\s()
63. 00000470: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
64. 00000480: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
65. 00000490: 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 )?([\s()\-\./]+(
66. 000004A0: 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E \d+))?([\s()\-\.
67. 000004B0: 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 /]+(\d+))?([\s()
68. 000004C0: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
69. 000004D0: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
70. 000004E0: 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 )?([\s()\-\./]+(
71. 000004F0: 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E \d+))?([\s()\-\.
72. 00000500: 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 /]+(\d+))?([\s()
73. 00000510: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
74. 00000520: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
75. 00000530: 29 3F 5C 73 2A 5B 58 78 5D 2B 28 5C 64 7B 31 2C )?\s\*[Xx]+(\d{1,
76. 00000540: 31 35 7D 29 5B 5C 73 5D 2A 0D 0A 2B 24 31 24 33 15})[\s]\*..+$1$3
77. 00000550: 24 35 24 37 24 39 24 31 31 24 31 33 24 31 35 24 $5$7$9$11$13$15$
78. 00000560: 31 37 24 31 39 24 32 31 3B 65 78 74 3D 24 32 32 17$19$21;ext=$22
79. 00000570: 0D 0A 30 31 31 28 5C 64 2B 29 28 5B 5C 73 28 29 ..011(\d+)([\s()
80. 00000580: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
81. 00000590: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
82. 000005A0: 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 )?([\s()\-\./]+(
83. 000005B0: 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E \d+))?([\s()\-\.
84. 000005C0: 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 /]+(\d+))?([\s()
85. 000005D0: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
86. 000005E0: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
87. 000005F0: 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 )?([\s()\-\./]+(
88. 00000600: 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E \d+))?([\s()\-\.
89. 00000610: 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 /]+(\d+))?([\s()
90. 00000620: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
91. 00000630: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
92. 00000640: 29 3F 5B 5C 73 5D 2A 0D 0A 2B 24 31 24 33 24 35 )?[\s]\*..+$1$3$5
93. 00000650: 24 37 24 39 24 31 31 24 31 33 24 31 35 24 31 37 $7$9$11$13$15$17
94. 00000660: 24 31 39 24 32 31 0D 0A 28 5C 64 2B 29 28 5B 5C $19$21..(\d+)([\
95. 00000670: 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 s()\-\./]+(\d+))
96. 00000680: 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 ([\s()\-\./]+(\d
97. 00000690: 2B 29 29 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B +))([\s()\-\./]+
98. 000006A0: 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 5C 2D 5C (\d+))?([\s()\-\
99. 000006B0: 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 ./]+(\d+))?([\s(
100. 000006C0: 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 5B )\-\./]+(\d+))?[
101. 000006D0: 5C 73 5D 2A 0D 0A 2B 24 31 24 33 24 35 24 37 24 \s]\*..+$1$3$5$7$
102. 000006E0: 39 24 31 31 0D 0A 45 31 36 34 0D 0A 6E 75 6C 6C 9$11..E164..null
103. 000006F0: 0D 0A 5C 2B 2B 28 5C 64 2B 29 28 5B 5C 73 28 29 ..\++(\d+)([\s()
104. 00000700: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
105. 00000710: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
106. 00000720: 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 )?([\s()\-\./]+(
107. 00000730: 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E \d+))?([\s()\-\.
108. 00000740: 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 /]+(\d+))?([\s()
109. 00000750: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
110. 00000760: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
111. 00000770: 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 )?([\s()\-\./]+(
112. 00000780: 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E \d+))?([\s()\-\.
113. 00000790: 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 /]+(\d+))?([\s()
114. 000007A0: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
115. 000007B0: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
116. 000007C0: 29 3F 5C 73 2A 5B 58 78 5D 2B 28 5C 64 7B 31 2C )?\s\*[Xx]+(\d{1,
117. 000007D0: 31 35 7D 29 5B 5C 73 5D 2A 0D 0A 2B 24 31 24 33 15})[\s]\*..+$1$3
118. 000007E0: 24 35 24 37 24 39 24 31 31 24 31 33 24 31 35 24 $5$7$9$11$13$15$
119. 000007F0: 31 37 24 31 39 24 32 31 3B 65 78 74 3D 24 32 32 17$19$21;ext=$22
120. 00000800: 0D 0A 5C 2B 2B 28 5C 64 2B 29 28 5B 5C 73 28 29 ..\++(\d+)([\s()
121. 00000810: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
122. 00000820: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
123. 00000830: 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 )?([\s()\-\./]+(
124. 00000840: 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E \d+))?([\s()\-\.
125. 00000850: 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 /]+(\d+))?([\s()
126. 00000860: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
127. 00000870: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
128. 00000880: 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 )?([\s()\-\./]+(
129. 00000890: 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 5C 2D 5C 2E \d+))?([\s()\-\.
130. 000008A0: 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B 5C 73 28 29 /]+(\d+))?([\s()
131. 000008B0: 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 29 3F 28 5B \-\./]+(\d+))?([
132. 000008C0: 5C 73 28 29 5C 2D 5C 2E 2F 5D 2B 28 5C 64 2B 29 \s()\-\./]+(\d+)
133. 000008D0: 29 3F 5B 5C 73 5D 2A 0D 0A 2B 24 31 24 33 24 35 )?[\s]\*..+$1$3$5
134. 000008E0: 24 37 24 39 24 31 31 24 31 33 24 31 35 24 31 37 $7$9$11$13$15$17
135. 000008F0: 24 31 39 24 32 31 0D 0A 00 $19$21...

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Length |
| Rules (variable) |
| ... |

**Length (4 bytes):** 0x0000085B specifies the length of the **Rules** field, in bytes.

**Rules (variable):** Normalization rules used by the Address Book Server to normalize the phone numbers in this file. This is a zero-terminated UTF-8 string, where the string contains one or more pairs of lines.

1. .\*882[\s()\-\./]\*8080.\*[Xx]+[\s()\-\./]\*(\d\d\d\d\d)$1;phone-
2. context=proseware.com.\*882[\s()\-\./]\*8080.\*null\((\+\s\*1)?\)[\s\-
3. \.]\*\(?(\d\d\d)\s\*\)?[\s()\-\./]\*(\d\d\d)[\s()\-\./]\*(\d\d\d\d)\s\*[Xx]+[\s()\-
4. \./]\*(\d\d\d\d\d)[\s()\-\./]\*+1$2$3$4;ext=$5 (\+\s\*1)?[\s\-
5. \.]\*\(?(\d\d\d)\s\*\)?[\s()\-\./]\*(\d\d\d)[\s()\-
6. \./]\*(\d\d\d\d)\s\*EXT\s\*(\d\d\d\d\d)+1$2$3$4;ext=$5 (\+\s\*1)?[\s\-
7. \.]\*\(?(\d\d\d)\s\*\)?[\s()\-\./]\*(\d\d\d)[\s()\-
8. \./]\*(\d\d\d\d)\s\*[xX]\s\*(\d\d\d\d\d)+1$2$3$4;ext=$5 (\s\*1)[\s\-
9. \.]\*\(?\s\*(\d\d\d)\s\*\)?[\s()\-\./]\*(\d\d\d)[\s()\-\./]\*(\d\d\d\d)[\s]\*+1$2$3$4
10. (\+\s\*1)?[\s\-\.]\*\(?(\d\d\d)\s\*\)?[\s()\-
11. \./]\*70(\d\d\d\d\d)+1$270$3;ext=$370(\d)[\s()\-
12. \./]\*(\d\d\d\d)\s\*[Xx]+(\d\d\d\d\d)+142570$1$2;ext=$3 (\d\d\d)[\s()\-
13. \./]\*(\d\d\d\d)\s\*[Xx]+(\d\d\d\d\d)+1425$1$2;ext=$3 (\d\d\d)[\s()\-
14. \./]\*(\d\d\d\d)+1425$1$2 [Xx]\s\*(\d\d\d\d\d)$1;phone-
15. context=proseware.com\(?(\d\d)\s\*\)?\s\*[Xx]+\s\*(\d\d\d\d\d)+1425$1$2;ext=$2011(\d+)([
16. \s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
17. \./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
18. \./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
19. \./]+(\d+))?\s\*[Xx]+(\d{1,15})[\s]\*+$1$3$5$7$9$11$13$15$17$19$21;ext=$22011(\d+)([\s(
20. )\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
21. \./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
22. \./]+(\d+))?([\s()\-\./]+(\d+))?[\s]\*+$1$3$5$7$9$11$13$15$17$19$21 (\d+)([\s()\-
23. \./]+(\d+))([\s()\-\./]+(\d+))([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
24. \./]+(\d+))?[\s]\*+$1$3$5$7$9$11E164null\++(\d+)([\s()\-\./]+(\d+))?([\s()\-
25. \./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
26. \./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
27. \./]+(\d+))?\s\*[Xx]+(\d{1,15})[\s]\*+$1$3$5$7$9$11$13$15$17$19$21;ext=$22\++(\d+)([\s(
28. )\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
29. \./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-\./]+(\d+))?([\s()\-
30. \./]+(\d+))?([\s()\-\./]+(\d+))?[\s]\*+$1$3$5$7$9$11$13$15$17$19$21

### Attribute Structure

#### ABF\_ATTRIBUTE manager

1. 000008F0: 0E 00 14 00 00 00 00 .......
2. 00000900: 10 6D 61 6E 61 67 65 72 00 .manager.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0014) | Length (0x000E) |
| Flags (0x10000000) |
| Name ("manager") |

**Length:** 0x000E specifies the length of the remaining bytes in this structure.

**Id:** 0x0014 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x10000000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_CLIENT\_FIELD\_MANAGER.

**Name:** Specifies that this attribute corresponds to the **manager** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE groupType

1. 00000900: 10 00 13 00 01 08 01 .......
2. 00000910: 0F 67 72 6F 75 70 54 79 70 65 00 .groupType.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0013) | Length (0x0010) |
| Flags (0x0F010801) |
| Name ("groupType") |

**Length:** 0x0010 specifies the length of the remaining bytes in this structure.

**Id:** 0x0013 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x0F010801 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_BINARY OR ABF\_ATTRIBUTE\_FLAGS\_REQUIRED OR ABF\_ATTRIBUTE\_FLAGS\_GROUPTYPE OR ABF\_CLIENT\_FIELD\_GROUPTYPE.

**Name:** Specifies that this attribute corresponds to the **groupType** attribute associated with Active Directory group objects.

#### ABF\_ATTRIBUTE proxyAddresses

1. 00000910: 15 00 12 00 02 .....
2. 00000920: 01 00 00 70 72 6F 78 79 41 64 64 72 65 73 73 65 ...proxyAddresse
3. 00000930: 73 00 s.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0012) | Length (0x0015) |
| Flags (0x00000102) |
| Name ("proxyAddresses") |

**Length:** 0x0015 specifies the length of the remaining bytes in this structure.

**Id:** 0x0012 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x00000102 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING\_PROXYADDRESS OR ABF\_ATTRIBUTE\_FLAGS\_RESERVED OR ABF\_CLIENT\_FIELD\_PROXYADDRESSES.

**Name:** Specifies that this attribute corresponds to the **proxyAddress**es attribute associated with Active Directory user, contact and group objects.

#### ABF\_ATTRIBUTE mail

1. 00000930: 0B 00 11 00 00 00 00 0E 6D 61 69 6C 00 ........mail.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0011) | Length (0x000B) |
| Flags (0x0E000000) |
| Name ("mail") |

**Length:** 0x000B specifies the length of the remaining bytes in this structure.

**Id:** 0x0011 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x0E000000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_CLIENT\_FIELD\_MAIL.

**Name:** Specifies that this attribute corresponds to the **mail** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE ipPhone

1. 00000930: 0E .
2. 00000940: 00 10 00 00 20 00 0D 69 70 50 68 6F 6E 65 00 .......ipPhone.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0010) | Length (0x000E) |
| Flags (0x0D002000) |
| Name ("ipPhone") |

**Length:** 0x000E specifies the length of the remaining bytes in this structure.

**Id:** 0x0010 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x0D002000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_NORMALIZATION\_CONTROL\_PROXYADDRESS OR ABF\_CLIENT\_FIELD\_IPPHONE.

**Name:** Specifies that this attribute corresponds to the **ipPhone** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE otherTelephone

1. 00000940: 15 .
2. 00000950: 00 0F 00 00 20 00 0C 6F 74 68 65 72 54 65 6C 65 .......otherTele
3. 00000960: 70 68 6F 6E 65 00 phone.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x000F) | Length (0x0015) |
| Flags (0x0C002000) |
| Name ("otherTelephone") |

**Length:** 0x0015 specifies the length of the remaining bytes in this structure.

**Id:** 0x000F specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x0C002000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_NORMALIZATION\_CONTROL\_PROXYADDRESS OR ABF\_CLIENT\_FIELD\_OTHERTELEPHONE

**Name:** Specifies that this attribute corresponds to the **otherTelephone** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE otherMobile

1. 00000960: 12 00 0E 00 00 20 00 0B 6F 74 ........ot
2. 00000970: 68 65 72 4D 6F 62 69 6C 65 00 herMobile.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x000E) | Length (0x0012) |
| Flags (0x0B002000) |
| Name ("otherMobile") |

**Length:** 0x0012 specifies the length of the remaining bytes in this structure.

**Id:** 0x000E specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x0B002000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_NORMALIZATION\_CONTROL\_PROXYADDRESS OR ABF\_CLIENT\_FIELD\_MOBILE.

**Name:** Specifies that this attribute corresponds to the **otherMobile** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE mobile

1. 00000970: 0D 00 0D 00 00 28 .....(
2. 00000980: 02 0B 6D 6F 62 69 6C 65 00 ..mobile.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x000D) | Length (0x000D) |
| Flags (0x0B022800) |
| Name ("mobile") |

**Length:** 0x000D specifies the length of the remaining bytes in this structure.

**Id:** 0x000D specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x0B022800 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_ATTRIBUTE\_FLAGS\_REQUIRED OR ABF\_NORMALIZATION\_CONTROL\_PROXYADDRESS OR ABF\_ATTRIBUTE\_FLAGS\_DEVICE OR ABF\_CLIENT\_FIELD\_MOBILE.

**Name:** Specifies that this attribute corresponds to the **mobile** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE otherHomePhone

1. 00000980: 15 00 0C 00 00 20 00 .......
2. 00000990: 0A 6F 74 68 65 72 48 6F 6D 65 50 68 6F 6E 65 00 .otherHomePhone.
3. 000009A0:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x000C) | Length (0x0015) |
| Flags (0x0A002000) |
| Name ("otherHomePhone") |

**Length:** 0x0015 specifies the length of the remaining bytes in this structure.

**Id:** 0x000C specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x0A002000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_NORMALIZATION\_CONTROL\_PROXYADDRESS OR ABF\_CLIENT\_FIELD\_HOMENUMBER.

**Name:** Specifies that this attribute corresponds to the **otherHomePhone** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE homePhone

1. 000009a0: 10 00 0b 00 00 28 00 0a 68 6f 6d 65 50 68 6f 6e .....(..homePhon
2. 000009b0: 65 00 e.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x000B) | Length (0x0010) |
| Flags (0x0A002800) |
| Name ("homePhone") |

**Length:** 0x0010 specifies the length of the remaining bytes in this structure.

**Id:** 0x000B specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x0A002800 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_ATTRIBUTE\_FLAGS\_REQUIRED OR ABF\_NORMALIZATION\_CONTROL\_PROXYADDRESS OR ABF\_CLIENT\_FIELD\_HOMENUMBER.

**Name:** Specifies that this attribute corresponds to the **homePhone** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE telephoneNumber

1. 000009B0: 16 00 0A 00 00 28 02 09 74 65 6C 65 70 68 .....(..teleph
2. 000009C0: 6F 6E 65 4E 75 6D 62 65 72 00 oneNumber.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x000A) | Length (0x0016) |
| Flags (0x09022800) |
| Name ("telephoneNumber") |

**Length:** 0x0016 specifies the length of the remaining bytes in this structure.

**Id:** 0x000A specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x09022800 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_ATTRIBUTE\_FLAGS\_REQUIRED OR ABF\_NORMALIZATION\_CONTROL\_PROXYADDRESS OR ABF\_ATTRIBUTE\_FLAGS\_DEVICE OR ABF\_CLIENT\_FIELD\_TELEPHONENUMBER.

**Name:** Specifies that this attribute corresponds to the **telephoneNumber** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE msRTCSIP-PrimaryUserAddress

1. 000009C0: 22 00 09 00 00 08 ".....
2. 000009D0: 02 08 6D 73 52 54 43 53 49 50 2D 50 72 69 6D 61 ..msRTCSIP-Prima
3. 000009E0: 72 79 55 73 65 72 41 64 64 72 65 73 73 00 ryUserAddress.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0009) | Length (0x0022) |
| Flags (0x08020800) |
| Name ("msRTCSIP-PrimaryUserAddress") |

**Length:** 0x0022 specifies the length of the remaining bytes in this structure.

**Id:** 0x0009 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x08020800 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_ATTRIBUTE\_FLAGS\_REQUIRED OR ABF\_ATTRIBUTE\_FLAGS\_DEVICE OR ABF\_CLIENT\_FIELD\_MSRTCSIP\_PRIMARYUSERADDRESS.

**Name:** Specifies that this attribute corresponds to the **msRTCSIP-PrimaryUserAddress** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE physicalDeliveryOfficeName

1. 000009E0: 21 00 !.
2. 000009F0: 08 00 00 00 00 07 70 68 79 73 69 63 61 6C 44 65 ......physicalDe
3. 00000A00: 6C 69 76 65 72 79 4F 66 66 69 63 65 4E 61 6D 65 liveryOfficeName
4. 00000A10: 00 .

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0008) | Length (0x0021) |
| Flags (0x07000000) |
| Name ("physicalDeliveryOfficeName") |

**Length:** 0x0021 specifies the length of the remaining bytes in this structure.

**Id:** 0x0008 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x07000000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_CLIENT\_FIELD\_PHYSICALDELIVERYOFFICENAME.

**Name:** Specifies that this attribute corresponds to the **physicalDeliveryOfficeName** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE company

1. 00000A10: 0E 00 07 00 00 00 00 06 63 6F 6D 70 61 6E 79 ........company
2. 00000A20: 00 .

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0007) | Length (0x000E) |
| Flags (0x06000000) |
| Name ("company") |

**Length:** 0x000E specifies the length of the remaining bytes in this structure.

**Id:** 0x0007 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x06000000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_CLIENT\_FIELD\_COMPANY.

**Name:** Specifies that this attribute corresponds to the **company** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE mailNickname

1. 00000A20: 13 00 06 00 00 04 00 05 6D 61 69 6C 4E 69 63 ........mailNic
2. 00000A30: 6B 6E 61 6D 65 00 kname.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0006) | Length (0x0013) |
| Flags (0x05000400) |
| Name ("mailNickname") |
|  |

**Length:** 0x0013 specifies the length of the remaining bytes in this structure.

**Id:** 0x0006 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x05000400 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_ATTRIBUTE\_FLAGS\_EMAIL OR ABF\_CLIENT\_FIELD\_MAILNICKNAME.

**Name:** Specifies that this attribute corresponds to the **mailNickname** attribute associated with Active Directory user, contact and group objects.

#### ABF\_ATTRIBUTE title

1. 00000A30: 0C 00 05 00 00 00 00 04 74 69 ........ti
2. 00000A40: 74 6C 65 00 tle.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0005) | Length (0x000C) |
| Flags (0x04000000) |
| Name ("title") |

**Length:** 0x000C specifies the length of the remaining bytes in this structure.

**Id:** 0x0005 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x04000000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_CLIENT\_FIELD\_TITLE.

**Name:** Specifies that this attribute corresponds to the **title** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE displayName

1. 00000A40: 12 00 04 00 00 00 02 03 64 69 73 70 ........disp
2. 00000A50: 6C 61 79 4E 61 6D 65 00 layName.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0004) | Length (0x0012) |
| Flags (0x03020000) |
| Name ("displayName") |

**Length:** 0x0012 specifies the length of the remaining bytes in this structure.

**Id:** 0x0004 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x03020000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_ATTRIBUTE\_FLAGS\_DEVICE OR ABF\_CLIENT\_FIELD\_DISPLAYNAME

**Name:** Specifies that this attribute corresponds to the **displayName** attribute associated with Active Directory user, contact and group objects.

#### ABF\_ATTRIBUTE sn

1. 00000A50: 09 00 03 00 00 00 00 02 ........
2. 00000A60: 73 6E 00 sn.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0003) | Length (0x0009) |
| Flags (0x02000000) |
| Name ("sn") |

**Length:** 0x0009 specifies the length of the remaining bytes in this structure.

**Id:** 0x0003 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x02000000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_CLIENT\_FIELD\_SN.

**Name:** Specifies that this attribute corresponds to the **sn** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE givenName

1. 00000A60: 10 00 02 00 00 00 00 01 67 69 76 65 6E ........given
2. 00000A70: 4E 61 6D 65 00 Name.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0002) | Length (0x0010) |
| Flags (0x01000000) |
| Name ("givenName") |

**Length:** 0x0010 specifies the length of the remaining bytes in this structure.

**Id:** 0x0002 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0x01000000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_CLIENT\_FIELD\_GIVENNAME.

**Name:** Specifies that this attribute corresponds to the **givenName** attribute associated with Active Directory user and contact objects.

#### ABF\_ATTRIBUTE msExchHideFromAddressLists

1. 00000A70: 21 00 01 00 00 00 00 FF 6D 73 45 !.......msE
2. 00000A80: 78 63 68 48 69 64 65 46 72 6F 6D 41 64 64 72 65 xchHideFromAddre
3. 00000A90: 73 73 4C 69 73 74 73 00 ssLists.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0001) | Length (0x0021) |
| Flags (0xFF000000) |
| Name ("msExchHideFromAddressLists") |

**Length:** 0x0021 specifies the length of the remaining bytes in this structure.

**Id:** 0x0001 specifies the identifier of this attribute as referenced by **ABF\_CONTACT** structures and the **MaximumAttributeId** field of the **ABF\_FULL\_HEADER** and **ABF\_DELTA\_HEADER** structures.

**Flags:** 0xFF000000 defines the type of attribute, how it is mapped by the client, and how it is processed by the Address Book Server. These flag settings correspond to the Boolean expression ABF\_TYPE\_STRING OR ABF\_CLIENT\_FIELD\_IGNORE.

**Name:** Specifies that this attribute corresponds to the **msExchHideFromAddressLists** attribute associated with Active Directory user, contact and group objects.

### ABSUser1 Contact

#### ABF\_CONTACT

1. 00000A90: 66 01 0A AD 36 8C 97 5E f...6..^
2. 00000AA0: DD 46 8D 5B 25 51 40 C5 2B 00 12 00 .F.[%Q@.+...

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0xAD0A) | Length (0x0166) |
| 0x5E978C36 |
| 0x5B8D46DD |
| 0xC5405125 |
| NumberOfAttributes (0x0012) | 0x002B |

**Length:** 0x0166 specifies the length of the remaining bytes in the structure.

**Id:** {8c36ad0a-5e97-46dd-8d5b-255140c52b00} is the **objectGUID** value associated with ABSUser1 in the sample Active Directory data.

**NumberOfAttributes:** 0x0012 specifies the number of **ABF\_CONTACT\_ATTRIBUTE** structures that follow this structure.

#### ABF\_CONTACT\_ATTRIBUTE mail

1. 00000AA0: 11 41 42 53 .ABS
2. 00000AB0: 55 73 65 72 31 40 75 72 74 65 73 74 2E 63 6F 6D User1@urtest.com
3. 00000AC0: 00 .

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser1@urtest.com" | Id (0x11) |

**Id:** 0x11 corresponds to the **mail** **ABF\_ATTRIBUTE** structure.

**Data:** "ABSUser1@urtest.com" is a null-terminated string that is the value of the **mail** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE otherTelephone

1. 00000AC0: 0F 35 35 35 2D 35 33 33 2D 34 33 31 32 00 .555-533-4312.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "555-533-4312" | Id (0x0F) |

**Id:** 0x0F corresponds to the **otherTelephone** **ABF\_ATTRIBUTE** structure.

**Data:** "555-533-4312" is a null-terminated string that is the value of the **otherTelephone** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE otherTelephone, normalized

1. 00000AC0: 0F .
2. 00000AD0: 74 65 6C 3A 2B 35 35 35 35 33 33 34 33 31 32 00 tel:+5555334312.
3. 00000AE0:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| " tel:+5555334312" | Id (0x0F) |

**Id:** 0x0F corresponds to the **otherTelephone** **ABF\_ATTRIBUTE** structure.

**Data:** "tel:+5555334312" is a null-terminated string that is the value of the **otherTelephone** attribute for this contact. This attribute value is not present in Active Directory, because this is the normalized version of the actual value of the **otherTelephone** attribute stored in Active Directory that was reported in the previous **ABF\_CONTACT\_ATTRIBUTE** structure.

#### ABF\_CONTACT\_ATTRIBUTE mobile

1. 00000AE0: 0D 35 35 35 2D 35 33 33 2D 34 33 31 33 00 .555-533-4313.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "555-533-4313" | Id (0x0D) |

**Id:** 0x0D corresponds to the **mobile** **ABF\_ATTRIBUTE** structure.

**Data:** "555-533-4313" is a null-terminated string that is the value of the **mobile** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE mobile, normalized

1. 00000AE0: 0D 74 .t
2. 00000AF0: 65 6C 3A 2B 35 35 35 35 33 33 34 33 31 33 00 el:+5555334313.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "tel:+5555334313" | Id (0x0D) |

**Id:** 0x0D corresponds to the **mobile** **ABF\_ATTRIBUTE** structure.

**Data:** "tel:+5555334313" is a null-terminated string that is the normalized version of the **mobile** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE otherHomePhone

1. 00000AF0: 0C .
2. 00000B00: 35 35 35 2D 33 39 31 2D 33 30 34 32 00 555-391-3042.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "555-391-3042" | Id (0x0C) |

**Id:** 0x0C corresponds to the **otherHomePhone** **ABF\_ATTRIBUTE** structure.

**Data:** "555-391-3042" is a null-terminated string that is the value of the "otherHomePhone" attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE otherHomePhone, normalized

1. 00000B00: 0C 74 65 .te
2. 00000B10: 6C 3A 2B 35 35 35 33 39 31 33 30 34 32 00 l:+5553913042.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "tel:+5553913042" | Id (0x0C) |

**Id:** 0x0C corresponds to the **otherHomePhone** **ABF\_ATTRIBUTE** structure.

**Data:** "tel:+5553913042" is a null-terminated string that is the normalized version of the "otherHomePhone" attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE homePhone

1. 00000B10: 0B 35 .5
2. 00000B20: 35 35 2D 35 36 36 2D 34 33 31 32 00 55-566-4312.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "555-566-4312" | Id (0x0B) |

**Id:** 0x0B corresponds to the **homePhone** **ABF\_ATTRIBUTE** structure.

**Data:** "555-566-4312" is a null-terminated string that is the value of the **homePhone** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE homePhone, normalized

1. 00000B20: 0B 74 65 6C .tel
2. 00000B30: 3A 2B 35 35 35 35 36 36 34 33 31 32 00 :+5555664312.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "tel:+5555664312" | Id (0x0B) |

**Id:** 0x0B corresponds to the **homePhone** **ABF\_ATTRIBUTE** structure.

**Data:** "tel:+5555664312" is a null-terminated string that is the normalized value of the **homePhone** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE telephoneNumber

1. 00000B30: 0A 35 35 .55
2. 00000B40: 35 20 33 39 31 20 33 32 32 34 00 5 391 3224.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "555 391 3224" | Id (0x0A) |

**Id:** 0x0A corresponds to the **telephoneNumber** **ABF\_ATTRIBUTE** structure.

**Data:** "555 391 3224" is a null-terminated string that is the value of the "telephoneNumber" attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE telephoneNumber, normalized

1. 00000B40: 0A 74 65 6C 3A .tel:
2. 00000B50: 2B 35 35 35 33 39 31 33 32 32 34 00 +5553913224.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "tel:+5553913224" | Id (0x0A) |

**Id:** 0x0A corresponds to the **telephoneNumber** **ABF\_ATTRIBUTE** structure.

**Data:** "tel:+5553913224" is a null-terminated string that is the normalized value of the **telephoneNumber** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE msRTCSIP-PrimaryUserAddress

1. 00000B50: 09 73 69 70 .sip
2. 00000B60: 3A 41 42 53 55 73 65 72 31 40 75 72 74 65 73 74 :ABSUser1@urtest
3. 00000B70: 2E 72 74 6D 70 2E 73 65 6C 66 68 6F 73 74 2E 63 .rtmp.selfhost.c
4. 00000B80: 6F 72 70 2E 70 72 6F 73 65 77 61 72 65 2E 63 6F orp.proseware.co
5. 00000B90: 6D 00 m.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "sip:ABSUser1@urtest.rtmp.selfhost.corp.proseware.com" | Id (0x09) |

**Id:** 0x09 corresponds to the **msRTCSIP-PrimaryUserAddress** **ABF\_ATTRIBUTE** structure.

**Data:** "sip:ABSUser1@urtest.rtmp.selfhost.corp.proseware.com" is a null-terminated string that is the value of the **msRTCSIP-PrimaryUserAddress** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE physicalDeliveryOfficeName

1. 00000B90: 08 31 32 33 34 35 00 .12345.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "12345" | Id (0x08) |

**Id:** 0x08 corresponds to the **physicalDeliveryOfficeName** **ABF\_ATTRIBUTE** structure.

**Data:** "12345" is a null-terminated string that is the value of the **physicalDeliveryOfficeName** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE company

1. 00000B90: 07 54 65 73 74 43 6F .TestCo
2. 00000BA0: 6D 70 61 6E 79 00 mpany.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "TestCompany" | Id (0x07) |

**Id:** 0x07 corresponds to the **company** **ABF\_ATTRIBUTE** structure.

**Data:** "TestCompany" is a null-terminated string that is the value of the **company** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE title

1. 00000BA0: 05 44 65 76 65 6C 6F 70 6D 65 .Developme
2. 00000BB0: 6E 74 20 4D 61 6E 61 67 65 72 00 nt.Manager.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "Development Manager" | Id (0x05) |

**Id:** 0x05 corresponds to the **title** **ABF\_ATTRIBUTE** structure.

**Data:** "Development Manager" is a null-terminated string that is the value of the **title** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE displayName

1. 00000BB0: 04 41 42 53 55 .ABSU
2. 00000BC0: 73 65 72 31 5F 64 69 73 70 6C 61 79 6E 61 6D 65 ser1\_displayname
3. 00000BD0: 5F 63 68 61 6E 67 65 64 00 \_changed.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser1\_displayname\_changed" | Id (0x04) |

**Id:** 0x04 corresponds to the **displayName** **ABF\_ATTRIBUTE** structure.

**Data:** "ABSUser1\_displayname\_changed" is a null-terminated string that is the value of the **displayName** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE sn

1. 00000BD0: 03 41 42 53 55 73 65 .ABSUse
2. 00000BE0: 72 31 5F 6C 61 73 74 6E 61 6D 65 00 r1\_lastname.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser1\_lastname" | Id (0x03) |

**Id:** 0x03 corresponds to the **sn** **ABF\_ATTRIBUTE** structure.

**Data:** "ABSUser1\_lastname" is a null-terminated string that is the value of the **sn** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE givenName

00000BE0: 02 41 42 53 .ABS

00000BF0: 55 73 65 72 31 5F 66 69 72 73 74 6E 61 6D 65 00 User1\_firstname.

00000C00:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser1\_firstname" | Id (0x02) |

**Id:** 0x02 corresponds to the **givenName** **ABF\_ATTRIBUTE** structure.

**Data:** "ABSUser1\_firstname" is a null-terminated string that is the value of the **givenName** attribute for this contact.

### ABSUser5 Contact

#### ABF\_CONTACT

1. 00000C00: A4 00 0B 41 E8 E1 C4 C8 22 40 93 BA B2 14 5E D6 ...A...."@....^.
2. 00000C10: D1 34 06 00 .4..

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x410B) | Length (0x00A4) |
| 0xC8C4E1E8 |
| 0xBA934022 |
| 0xD65E14B2 |
| NumberOfAttributes (0x0006) | 0x34D1 |

**Length:** 0x00A4 specifies the length of the remaining bytes in the structure.

**Id:** {e1e8410b-c8c4-4022-93ba-b2145ed6d134} is the **objectGUID** value associated with ABSUser5 in the sample Active Directory data.

**NumberOfAttributes:** 0x0006 specifies the number of **ABF\_CONTACT\_ATTRIBUTE** structures that follow this structure.

#### ABF\_CONTACT\_ATTRIBUTE telephoneNumber

1. 00000C10: 0A 35 35 35 2D 37 38 39 2D 36 36 36 .555-789-666
2. 00000C20: 36 00 6.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "555-789-6666" | Id (0x0A) |

**Id:** 0x0A corresponds to the **telephoneNumber** **ABF\_ATTRIBUTE** structure.

**Data:** "555-789-6666" is a null-terminated string that is the value of the **telephoneNumber** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE telephoneNumber, normalized

1. 00000C20: 0A 74 65 6C 3A 2B 35 35 35 37 38 39 36 36 .tel:+55578966
2. 00000C30: 36 36 00 66.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "tel:+5557896666 " | Id (0x0A) |

**Id:** 0x0A corresponds to the **telephoneNumber** **ABF\_ATTRIBUTE** structure.

**Data:** "tel:+5557896666 " is a null-terminated string that is the normalized value of the **telephoneNumber** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE msRTCSIP-PrimaryUserAddress

1. 00000C30: 09 73 69 70 3A 41 42 53 55 73 65 72 35 .sip:ABSUser5
2. 00000C40: 40 75 72 74 65 73 74 2E 72 74 6D 70 2E 73 65 6C @urtest.rtmp.sel
3. 00000C50: 66 68 6F 73 74 2E 63 6F 72 70 2E 70 72 6F 73 65 fhost.corp.prose
4. 00000C60: 77 61 72 65 2E 63 6F 6D 00 ware.com.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "sip:ABSUser5@urtest.rtmp.selfhost.corp.proseware.com" | Id (0x09) |

**Id:** 0x09 corresponds to the **msRTCSIP-PrimaryUserAddress** **ABF\_ATTRIBUTE** structure.

**Data:** "sip:ABSUser5@urtest.rtmp.selfhost.corp.proseware.com" is a null-terminated string that is the value of the **msRTCSIP-PrimaryUserAddress** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE displayName

1. 00000C60: 04 41 42 53 55 73 65 .ABSUse
2. 00000C70: 72 35 5F 64 69 73 70 6C 61 79 6E 61 6D 65 00 r5\_displayname.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser5\_displayname" | Id (0x04) |

**Id:** 0x04 corresponds to the **displayName** **ABF\_ATTRIBUTE** structure.

**Data:** "ABSUser5\_displayname" is a null-terminated string that is the value of the **displayName** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE sn

1. 00000C70: 03 .
2. 00000C80: 41 42 53 55 73 65 72 35 5F 6C 61 73 74 6E 61 6D ABSUser5\_lastnam
3. 00000C90: 65 00 e.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser5\_lastname" | Id (0x03) |

**Id:** 0x03 corresponds to the **sn** **ABF\_ATTRIBUTE** structure.

**Data:** "ABSUser5\_lastname" is a null-terminated string that is the value of the **sn** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE givenName

1. 00000C90: 02 41 42 53 55 73 65 72 35 5F 66 69 72 73 .ABSUser5\_firs
2. 00000CA0: 74 6E 61 6D 65 00 tname.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser5\_firstname" | Id (0x02) |

**Id:** 0x02 corresponds to the **givenName** ABF\_ATTRIBUTE structure.

**Data:** "ABSUser5\_firstname" is a null-terminated string that is the value of the **givenName** attribute for this contact.

### ABSUser2 Contact

#### ABF\_CONTACT

1. 00000CA0: A6 00 1B 25 7C 47 42 DB F6 4E ...%|GB..N
2. 00000CB0: 9C 69 FA BB EC C6 7F 31 00 00 06 00 .i.....1....

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x251B) | Length (0x00A6) |
| 0xDB42477C |
| 0x699C4EF6 |
| 0xC6ECBBFA |
| NumberOfAttributes (0x0000) | 0x317F |
|  | NumberOfDeletedAttributes (0x0006) |

**Length:** 0x00A6 specifies the length of the remaining bytes in the structure.

**Id:** {477c251b-db42-4ef6-9c69-fabbecc67f31} is the **objectGUID** value associated with ABSUser2 in the sample Active Directory data.

**NumberOfAttributes:** 0x0000 specifies that this contact was deleted. Because the **Length** field specifies a length greater than 18, the next field is also present.

**NumberOfDeletedAttributes:** 0x0006 specifies the number of **ABF\_CONTACT\_ATTRIBUTE** structures that follow this structure.

#### ABF\_CONTACT\_ATTRIBUTE telephoneNumber

1. 00000CB0: 0A 35 35 35 .555
2. 00000CC0: 2D 37 38 33 2D 34 37 35 36 00 -783-4756.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "555-783-4756" | Id (0x0A) |

**Id:** 0x0A corresponds to the **telephoneNumber** **ABF\_ATTRIBUTE** structure.

**Data:** "555-783-4756" is a null-terminated string that is the value of the **telephoneNumber** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE telephoneNumber, normalized

1. 00000CC0: 0A 74 65 6C 3A 2B .tel:+
2. 00000CD0: 35 35 35 37 38 33 34 37 35 36 00 5557834756 .

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "tel:+5557834756 " | Id (0x0A) |

**Id:** 0x0A corresponds to the **telephoneNumber** **ABF\_ATTRIBUTE** structure.

**Data:** "tel:+5557834756 " is a null-terminated string that is the value of the "telephoneNumber" attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE msRTCSIP-PrimaryUserAddress

1. 00000CD0: 09 73 69 70 3A .sip:
2. 00000CE0: 41 42 53 55 73 65 72 32 40 75 72 74 65 73 74 2E ABSUser2@urtest.
3. 00000CF0: 72 74 6D 70 2E 73 65 6C 66 68 6F 73 74 2E 63 6F rtmp.selfhost.co
4. 00000D00: 72 70 2E 70 72 6F 73 65 77 61 72 65 2E 63 6F 6D rp.proseware.com
5. 00000D10: 00 .

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "sip:ABSUser2@urtest.rtmp.selfhost.corp.proseware.com" | Id (0x09) |

**Id:** 0x09 corresponds to the **msRTCSIP-PrimaryUserAddress** **ABF\_ATTRIBUTE** structure.

**Data:** "sip:ABSUser2@urtest.rtmp.selfhost.corp.proseware.com" is a null-terminated string that is the value of the **msRTCSIP-PrimaryUserAddress** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE displayName

1. 00000D10: 04 41 42 53 55 73 65 72 32 5F 64 69 73 70 6C .ABSUser2\_displ
2. 00000D20: 61 79 6E 61 6D 65 00 ayname.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser2\_displayname" | Id (0x04) |

**Id:** 0x04 corresponds to the **displayName** **ABF\_ATTRIBUTE** structure.

**Data:** "ABSUser2\_displayname" is a null-terminated string that is the value of the **displayName** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE sn

1. 00000D20: 03 41 42 53 55 73 65 72 32 .ABSUser2
2. 00000D30: 5F 6C 61 73 74 6E 61 6D 65 00 \_lastname.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser2\_lastname" | Id (0x03) |

**Id:** 0x03 corresponds to the **sn** **ABF\_ATTRIBUTE** structure.

**Data:** "ABSUser2\_lastname" is a null-terminated string that is the value of the **sn** attribute for this contact.

#### ABF\_CONTACT\_ATTRIBUTE givenName

1. 00000D30: 02 41 42 53 55 73 .ABSUs
2. 00000D40: 65 72 32 5F 66 69 72 73 74 6E 61 6D 65 00 er2\_firstname.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| "ABSUser2\_firstname" | Id (0x02) |

**Id:** 0x02 corresponds to the **givenName** **ABF\_ATTRIBUTE** structure.

**Data:** "ABSUser2\_firstname" is a null-terminated string that is the value of the **givenName** attribute for this contact.

### ABF\_CONTACT

This section provides an example of the address book file **ABF\_CONTACT** structure specified in section [2.2.10](#Section_0f063341df544eebaabeaf2cb67ab572).

1. 00000D40: 12 00 ..
2. 00000D50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
3. 00000D60: 00 00 ....

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| Id (0x0000) | Length (0x0012) |
| 0x00000000 |
| 0x00000000 |
| 0x00000000 |
| NumberOfAttributes (0x0000) | 0x0000 |

**Length:** 0x0012 specifies the length of the remaining bytes in the structure.

**Id:** {00000000-0000-0000-0000-000000000000} is a GUID with a value of 0, which indicates the end of the **ABF\_CONTACT** array.

**NumberOfAttributes:** 0x0000 specifies that this contact was deleted, unless the GUID in the **Id** field has a value of 0, in which case this structure indicates the end of the ABF\_CONTACT array.

### ABF\_DELTA\_TRAILER

This section provides an example of the address book file **ABF\_DELTA\_TRAILER** structure specified in section [2.2.13](#Section_04c0e6760fa7437186360781b89c9a55).

1. 00000D60: DB DF D2 DD 02 00 00 00 01 00 00 00 ............

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| BaseFileHash (0xDDD2) | Hash (0xDFDB) |
| NumberOfContacts (0x00000002) |
| NumberOfDeletedContacts (0x00000001) |

Hash: 0xDFDB is the [**hash**](#gt_b7e2b611-0af5-4fec-8af2-3f9ce7bad205) of the contents of the file. The [**cryptographic hash function**](#gt_a70d1a05-9add-4870-ab52-ec5d939e93e2) that is used can be anything, as long as it is the same function for all files.

BaseFileHash: 0xDDD2 is the hash of the contents of the base file that was used to calculate this delta file. The hash function that is used can be anything, as long as it is the same function for all files.

**NumberOfContacts:** 0x00000002 specifies the number of **ABF\_CONTACT** structures in the file that have a nonzero value in the **NumberOfAttributes** field.

**NumberOfDeletedContacts:** 0x00000001 specifies the number of **ABF\_CONTACT** structures in the file that have a zero value in the **NumberOfAttributes** field.

### ABF\_TRAILER\_LENGTH

This section provides an example of the address book file **ABF\_TRAILER\_LENGTH** structure specified in section [2.2.14](#Section_63d7ef3d524040bdb169ab4bd097c70e).

1. 00000D60: 0C 00 ..
2. 00000D70: 00 00 ..

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| LengthOfTrailer (0x0000000C) |

**LengthOfTrailer:** 0x0000000C length of the **ABF\_DELTA\_TRAILER** structure in this file. This allows additional information to be placed in the file after the **ABF\_DELTA\_TRAILER** structure and before the **ABF\_TRAILER\_LENGTH** structure without affecting older clients.

## Compressed Address Book File

Using the compressed form of the D-0A10-0A11.lsabs file seen in [Address Book File Example](#Section_d714352cfb9d4c9d93cabd40f8fd6ad0), the hexadecimal dump of the compressed file is as follows:

1. 00000000: 6e 65 5c a2 83 04 00 00 72 0d 00 00 20 00 00 00 ne\.....r... ...
2. 00000010: 16 c1 4b b5 08 90 c7 47 b9 bd f3 bb 1a 0a b6 eb ..K....G........
3. 00000020: 10 0a 11 0a 14 00 14 00 01 00 07 00 2f 67 5b 08 ............/g[.
4. 00000030: 00 00 2e 11 00 00 00 2a 38 38 32 5b 5c 73 28 29 .......\*882[\s()
5. 00000040: 5c 2d 5c 2e 2f 5d 2a 38 30 38 30 2e 2a 5b 58 78 \-\./]\*8080.\*[Xx
6. 00000050: 5d 2b b7 00 28 5c 64 0d 00 00 00 00 00 29 0d 0a ]+..(\d......)..
7. 00000060: 24 31 3b 70 68 6f 6e 65 2d 63 6f 6e 74 65 78 74 $1;phone-context
8. 00000070: 3d 70 72 6F 73 65 77 61 72 65 2e 63 6f 60 00 00 =proseware.co`..
9. 00000080: 10 6d 0d 0a af 02 9d 0d 0a 6e 75 6c 6c 0d 0a 5c .m.......null..\
10. 00000090: 28 28 5c 2b 5c 73 2a 31 29 3f 5c 29 30 01 21 01 ((\+\s\*1)?\)0.!.
11. 000000a0: 5d 2a 5c 28 3f 0c 01 f0 a3 2c 03 29 d8 00 5c 29 ]\*\(?....,.)..\)
12. 000000b0: 3f ff 03 69 01 9f 04 f8 79 01 8f 05 04 bf 00 f2 ?..i....y.......
13. 000000c0: 0d 0a 2b 31 24 32 24 33 24 34 3b c9 05 24 35 0d ..+1$2$3$4;..$5.
14. 000000d0: 0a 6e 04 5f 04 2e 45 58 1c 21 ff 70 54 28 00 97 .n.\_..EX.!.pT(..
15. 000000e0: 09 f4 a7 03 48 5b 78 58 5d af 03 2f 0b 99 03 97 ....H[xX]../....
16. 000000f0: 03 9f 01 f0 af 03 18 a8 00 7f 0a f2 9f 06 18 37 ...............7
17. 00000100: 30 6f 05 f8 37 30 24 33 6b 05 33 0d 0a 02 01 9f 0o..70$3k.3.....
18. 00000110: 0f 06 17 02 f6 34 32 30 e2 3f 7f 35 18 02 48 02 .....420.?.5..H.
19. 00000120: 2e 02 df 11 0d 3f 02 f9 2f 02 12 8f 01 71 0d 0a .....?../....q..
20. 00000130: 79 02 57 0c 9f 19 9f 07 4d 09 3c 09 b5 04 97 0e y.W.....M.<.....
21. 00000140: cf 04 13 70 03 30 31 31 10 01 2b 29 28 d7 04 71 ...p.011..+)(..q
22. 00000150: 06 2b 29 29 3f 40 00 c0 c0 9f 00 1f 9b 8f 0c 7b .+))?@.........{
23. 00000160: 31 2c 31 35 7d 46 13 b0 07 33 24 35 24 37 24 39 1,15}F...3$5$7$9
24. 00000170: 24 31 31 24 31 33 24 31 48 08 37 24 31 39 24 32 $11$13$1H.7$19$2
25. 00000180: 80 f9 06 7f 31 74 08 7f 08 ff ba ef 07 0b fa 14 ....1t..........
26. 00000190: 97 07 fb 97 00 0d 67 04 ff 25 0d 0a 45 31 36 f8 ......g..%..E16.
27. 000001a0: 1f f4 2e 2b 63 01 5f 0b 9b 7f 14 ff 3b 7f 08 b7 ...+c.\_.....;...
28. 000001b0: 7f 14 1f 0d 00 0e 20 47 00 00 10 6d 61 6e 61 67 ...... G...manag
29. 000001c0: 65 40 00 00 00 72 00 10 00 13 00 01 08 01 0f 67 e@...r.........g
30. 000001d0: 72 6f 75 70 54 79 70 65 00 15 00 12 00 02 38 48 roupType......8H
31. 000001e0: 70 72 6f 78 79 41 0d 20 08 00 64 64 72 65 73 73 proxyA. ..ddress
32. 000001f0: 65 73 00 0b 00 11 c1 01 0e 6d 61 69 6c 28 02 10 es.......mail(..
33. 00000200: 00 00 20 00 0d 69 70 50 e9 29 98 01 0f 79 00 82 .. ..ipP.)...y..
34. 00000210: 00 35 00 0c 6f 74 68 65 72 54 65 6c 65 a2 2a 40 .5..otherTele.\*@
35. 00000220: 02 0e b1 00 0b b2 00 4d 6f 62 69 6c 65 00 0d 08 .......Mobile...
36. 00000230: 00 00 28 02 0b 6d 73 00 15 84 44 50 28 00 0c 11 ..(..ms...DP(...
37. 00000240: 01 0a 12 01 48 6f 6d 65 83 02 10 78 03 00 28 00 ....Home...x..(.
38. 00000250: 0a 68 8e 00 16 00 0a b9 01 09 74 ed 02 4e 75 6d .h........t..Num
39. 00000260: 62 00 06 22 00 22 00 00 00 09 00 00 08 02 08 6d b..".".........m
40. 00000270: 73 52 54 43 53 49 50 2d 50 72 69 6d 61 72 79 55 sRTCSIP-PrimaryU
41. 00000280: 73 65 72 ec 05 00 21 00 a0 4a 00 0a 00 00 00 00 ser...!..J......
42. 00000290: 07 70 68 79 73 69 63 61 6c 44 65 6c 69 76 65 72 .physicalDeliver
43. 000002a0: 79 4f 66 66 69 63 65 4e 61 6d 65 40 05 07 11 01 yOfficeName@....
44. 000002b0: 06 14 22 08 84 a8 2f 70 61 6e 79 a8 08 06 00 00 ..".../pany.....
45. 000002c0: 04 00 05 71 07 4e 69 63 6b 6e 21 01 0c 00 05 21 ...q.Nickn!....!
46. 000002d0: 01 04 74 69 74 d0 05 12 f8 00 00 00 a0 52 44 00 ..tit........RD.
47. 000002e0: 02 03 64 69 73 70 6c 61 79 32 02 09 00 03 09 01 ..display2......
48. 000002f0: 02 73 6e 10 06 02 51 00 01 67 50 03 6e e2 00 21 .sn...Q..gP.n..!
49. 00000300: f3 52 ff 6d 73 45 78 00 00 20 00 63 68 48 69 64 .R.msEx.. .chHid
50. 00000310: 65 46 72 6f 6d 24 05 4c 69 73 74 73 00 66 01 0a eFrom$.Lists.f..
51. 00000320: ad 36 8c 97 5e dd 46 8d 5b 25 51 40 00 00 01 21 .6..^.F.[%Q@...!
52. 00000330: c5 2b 28 03 11 41 42 53 69 06 31 40 75 72 74 65 .+(..ABSi.1@urte
53. 00000340: 73 ca 34 00 0f 35 35 35 2d 35 33 33 2d 34 33 31 s.4..555-533-431
54. 00000350: 32 00 0f 3a 00 11 9d a8 08 3a 2b 90 00 88 00 82 2..:.....:+.....
55. 00000360: 00 0d f7 00 33 00 0d f7 00 24 33 00 0c f1 00 33 ....3....$3....3
56. 00000370: 39 31 2d 33 30 34 32 00 0c f5 00 88 00 82 00 0b 91-0003.........
57. 00000380: ea 01 36 40 30 a0 52 36 e3 02 0b ee 01 36 36 82 ..6@0.R6.....66.
58. 00000390: 00 0a 50 00 20 58 01 20 33 32 32 34 00 0a ef 01 ..P. X. 0001....
59. 000003a0: 81 00 09 73 69 70 3a 9f 05 36 72 74 6d 70 2e 73 ...sip:..6rtmp.s
60. 000003b0: 00 0a 40 04 65 6c 66 68 6f 0a 06 72 70 2e 57 3b ..@.elfho..rp.W;
61. 000003c0: 00 08 31 32 33 34 35 00 07 54 68 01 43 24 0c 05 ..12345..Th.C$..
62. 000003d0: 44 65 76 65 6c 6f 70 6d 28 04 60 05 65 6e 74 20 Developm(.`.ent
63. 000003e0: 4d 8c 15 04 d5 02 5f c4 0b d1 0c 5f 63 68 61 6e M.....\_....\_chan
64. 000003f0: 67 65 64 00 03 ee 00 6c 61 73 74 aa 0d 02 96 00 ged....last.....
65. 00000400: 66 69 72 1f 08 00 80 9c 00 a4 00 0b 41 e8 e1 c4 fir.........A...
66. 00000410: c8 22 40 93 ba b2 14 5e d6 d1 34 06 b2 06 2d 37 ."@....^..4...-7
67. 00000420: 38 39 2d 36 00 00 b7 06 20 88 00 82 00 b7 06 0e 89-6.... .......
68. 00000430: 00 40 6d 35 b7 06 2f 10 6d 05 35 6f 05 2e 05 35 .@m5../.m.5o...5
69. 00000440: 2f 05 18 35 2f 05 a6 00 1b 25 7c 47 42 db f6 4e /..5/....%|GB..N
70. 00000450: 9c 69 fa bb ec c6 7f 31 08 15 3d 05 d8 0e 37 07 .i.....1..=...7.
71. 00000460: 08 ab 5a 35 3f 05 23 33 82 00 3f 05 32 3f 05 bf ..Z5?.#3..?.2?..
72. 00000470: 18 32 3f 05 32 3f 05 18 32 3f 05 12 ee 65 07 00 .2?.2?..2?...e..
73. 00000480: 00 db df d2 dd 01 18 01 00 00 00 0c 00 00 00 ...............

## COMPRESSED\_BLOCK\_HEADER

Each **COMPRESSED\_BLOCK** structure in a compressed address book file begins with a **COMPRESSED\_BLOCK\_HEADER** structure that gives the size of the remaining bytes in the **COMPRESS\_BLOCK** structure.

1. 00000000: 6e 65 5c a2 83 04 00 00 72 0d 00 00 ne\.....r...

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| CRC (0xA25C656E) |
| CompressedSize (0x00000483) |
| DecompressedSize (0x00000D72) |

CRC: 0xA25C656E is the [**CRC**](#gt_9cb45a36-92bb-4c14-b2fd-2ad7e2979bfd) of the decompressed **Data** bytes in the block. The algorithm that is used to compute this value is described in the next section.

**CompressedSize:** 0x00000483 is the length, in bytes, of the **Data** field that follows the **COMPRESSED\_BLOCK\_HEADER**.

**DecompressedSize:** 0x00000D72 is the expected length, in bytes, of the decompressed data.

# Security Considerations

## Security Considerations for Implementers

None.

## Index of Security Fields

None.

# Appendix A: Compression Format

The information in this section describes the format of compressed Address Book Files with examples and pseudo code to decompress a compressed file.

## 32-Bit CRC Algorithm

This section uses pseudo code to explain how to generate the 32-bit [**CRC**](#gt_9cb45a36-92bb-4c14-b2fd-2ad7e2979bfd) value stored in the **CRC** field of the **COMPRESSED\_BLOCK\_HEADER** structure.

1. Define CalculateCRC32 as Function With Parameters (
2. buffer as unsigned char []
3. length as unsigned integer
4. )
5. Begin
6. Define crc as unsigned integer
7. Define index as integer
8. crc = 0xFFFFFFFF
9. index := 0
10. While (index < length)
11. Begin
12. crc := crc32\_table [(crc AND 0xFF) XOR buffer [index]] ^ (crc / 256)
13. index := index + 1
14. End
15. return crc XOR 0xFFFFFFFF
16. End
17. Define crc32\_table as unsigned integer [256] =
18. Begin
19. 0x00000000, 0x77073096, 0xEE0E612C, 0x990951BA,
20. 0x076DC419, 0x706AF48F, 0xE963A535, 0x9E6495A3,
21. 0x0EDB8832, 0x79DCB8A4, 0xE0D5E91E, 0x97D2D988,
22. 0x09B64C2B, 0x7EB17CBD, 0xE7B82D07, 0x90BF1D91,
23. 0x1DB71064, 0x6AB020F2, 0xF3B97148, 0x84BE41DE,
24. 0x1ADAD47D, 0x6DDDE4EB, 0xF4D4B551, 0x83D385C7,
25. 0x136C9856, 0x646BA8C0, 0xFD62F97A, 0x8A65C9EC,
26. 0x14015C4F, 0x63066CD9, 0xFA0F3D63, 0x8D080DF5,
27. 0x3B6E20C8, 0x4C69105E, 0xD56041E4, 0xA2677172,
28. 0x3C03E4D1, 0x4B04D447, 0xD20D85FD, 0xA50AB56B,
29. 0x35B5A8FA, 0x42B2986C, 0xDBBBC9D6, 0xACBCF940,
30. 0x32D86CE3, 0x45DF5C75, 0xDCD60DCF, 0xABD13D59,
31. 0x26D930AC, 0x51DE003A, 0xC8D75180, 0xBFD06116,
32. 0x21B4F4B5, 0x56B3C423, 0xCFBA9599, 0xB8BDA50F,
33. 0x2802B89E, 0x5F058808, 0xC60CD9B2, 0xB10BE924,
34. 0x2F6F7C87, 0x58684C11, 0xC1611DAB, 0xB6662D3D,
35. 0x76DC4190, 0x01DB7106, 0x98D220BC, 0xEFD5102A,
36. 0x71B18589, 0x06B6B51F, 0x9FBFE4A5, 0xE8B8D433,
37. 0x7807C9A2, 0x0F00F934, 0x9609A88E, 0xE10E9818,
38. 0x7F6A0DBB, 0x086D3D2D, 0x91646C97, 0xE6635C01,
39. 0x6B6B51F4, 0x1C6C6162, 0x856530D8, 0xF262004E,
40. 0x6C0695ED, 0x1B01A57B, 0x8208F4C1, 0xF50FC457,
41. 0x65B0D9C6, 0x12B7E950, 0x8BBEB8EA, 0xFCB9887C,
42. 0x62DD1DDF, 0x15DA2D49, 0x8CD37CF3, 0xFBD44C65,
43. 0x4DB26158, 0x3AB551CE, 0xA3BC0074, 0xD4BB30E2,
44. 0x4ADFA541, 0x3DD895D7, 0xA4D1C46D, 0xD3D6F4FB,
45. 0x4369E96A, 0x346ED9FC, 0xAD678846, 0xDA60B8D0,
46. 0x44042D73, 0x33031DE5, 0xAA0A4C5F, 0xDD0D7CC9,
47. 0x5005713C, 0x270241AA, 0xBE0B1010, 0xC90C2086,
48. 0x5768B525, 0x206F85B3, 0xB966D409, 0xCE61E49F,
49. 0x5EDEF90E, 0x29D9C998, 0xB0D09822, 0xC7D7A8B4,
50. 0x59B33D17, 0x2EB40D81, 0xB7BD5C3B, 0xC0BA6CAD,
51. 0xEDB88320, 0x9ABFB3B6, 0x03B6E20C, 0x74B1D29A,
52. 0xEAD54739, 0x9DD277AF, 0x04DB2615, 0x73DC1683,
53. 0xE3630B12, 0x94643B84, 0x0D6D6A3E, 0x7A6A5AA8,
54. 0xE40ECF0B, 0x9309FF9D, 0x0A00AE27, 0x7D079EB1,
55. 0xF00F9344, 0x8708A3D2, 0x1E01F268, 0x6906C2FE,
56. 0xF762575D, 0x806567CB, 0x196C3671, 0x6E6B06E7,
57. 0xFED41B76, 0x89D32BE0, 0x10DA7A5A, 0x67DD4ACC,
58. 0xF9B9DF6F, 0x8EBEEFF9, 0x17B7BE43, 0x60B08ED5,
59. 0xD6D6A3E8, 0xA1D1937E, 0x38D8C2C4, 0x4FDFF252,
60. 0xD1BB67F1, 0xA6BC5767, 0x3FB506DD, 0x48B2364B,
61. 0xD80D2BDA, 0xAF0A1B4C, 0x36034AF6, 0x41047A60,
62. 0xDF60EFC3, 0xA867DF55, 0x316E8EEF, 0x4669BE79,
63. 0xCB61B38C, 0xBC66831A, 0x256FD2A0, 0x5268E236,
64. 0xCC0C7795, 0xBB0B4703, 0x220216B9, 0x5505262F,
65. 0xC5BA3BBE, 0xB2BD0B28, 0x2BB45A92, 0x5CB36A04,
66. 0xC2D7FFA7, 0xB5D0CF31, 0x2CD99E8B, 0x5BDEAE1D,
67. 0x9B64C2B0, 0xEC63F226, 0x756AA39C, 0x026D930A,
68. 0x9C0906A9, 0xEB0E363F, 0x72076785, 0x05005713,
69. 0x95BF4A82, 0xE2B87A14, 0x7BB12BAE, 0x0CB61B38,
70. 0x92D28E9B, 0xE5D5BE0D, 0x7CDCEFB7, 0x0BDBDF21,
71. 0x86D3D2D4, 0xF1D4E242, 0x68DDB3F8, 0x1FDA836E,
72. 0x81BE16CD, 0xF6B9265B, 0x6FB077E1, 0x18B74777,
73. 0x88085AE6, 0xFF0F6A70, 0x66063BCA, 0x11010B5C,
74. 0x8F659EFF, 0xF862AE69, 0x616BFFD3, 0x166CCF45,
75. 0xA00AE278, 0xD70DD2EE, 0x4E048354, 0x3903B3C2,
76. 0xA7672661, 0xD06016F7, 0x4969474D, 0x3E6E77DB,
77. 0xAED16A4A, 0xD9D65ADC, 0x40DF0B66, 0x37D83BF0,
78. 0xA9BCAE53, 0xDEBB9EC5, 0x47B2CF7F, 0x30B5FFE9,
79. 0xBDBDF21C, 0xCABAC28A, 0x53B39330, 0x24B4A3A6,
80. 0xBAD03605, 0xCDD70693, 0x54DE5729, 0x23D967BF,
81. 0xB3667A2E, 0xC4614AB8, 0x5D681B02, 0x2A6F2B94,
82. 0xB40BBE37, 0xC30C8EA1, 0x5A05DF1B, 0x2D02EF8D
83. End

## Compressed Data Format

The compression technique used to compress the output files belongs to the LZ77 family, as described in [[UASDC]](https://go.microsoft.com/fwlink/?LinkId=90549), and is based on a simple idea: compression is achieved by substitution of substrings with pointers to previous occurrence of exactly the same substring in form of **(offset, length)** where **offset** is offset to the beginning of matching substring and **length** is its length

At the highest level, each block of compressed data consists of one or more token groups. All but the last token group has 32 tokens. Each token group starts out with a 32-bit unsigned integer, where each bit indicates the type of one token. Bit 0 gives the type of the first token, Bit 1 the type of the second token, and so on. The data for each token is serialized after the 32-bit unsigned integer. If a bit in the unsigned integer is 0, then the corresponding token is a literal byte, which is copied directly from the input buffer to the output buffer. If the bit is 1, then the corresponding token is a reference to a sequence of 3 or more bytes from a previous position in the output buffer. The length and offset of the data to be copied are encoded in the data for the token. This encoding occupies a minimum of 2 bytes and a maximum of 6 bytes, depending on the length of the data being copied.

## Run Encoding

As described in the previous section, when a run of characters is encoded, a length and offset are stored in a minimum of 2 bytes and a maximum of 6. This encoding works by storing the offset in the high 13 bits of the first two bytes and the length in the low 3 bits. Because the compression algorithm does not generate runs shorter than 3 bytes, the length is biased by -3.

The first 2 bytes are treated as an unsigned 16-bit integer, with the high 13 bits being the offset and the low 3 bits being the length. The offset has been biased by -1, so in order for the number to be useful, 1 must be added back to it. The offset is a backwards offset from the current position in the output buffer. So after adjusting for the -1 bias, an offset of zero in the high 13 bits would be 1, and that value (1) would then be subtracted from the current position in the output buffer to get the location of the previous character in the buffer that starts the run.

The length field is biased by -3, because no runs of length less than 3 can be emitted by the compression algorithm. So a value of 0 in the low 3 bits is really a length of 3. This bias makes it possible to encode lengths of 3 to 10 (0 through 7) in 3 bits. Because the length will often be greater than 10 bits, there is an additional mechanism: If all 3 bits are set (0x7), then the next byte is examined. The first time, the low 4 bits of that byte are extracted as the length, and the location of the byte is stored. This 4-bit length is biased by -10 (-(3+7)), so it is now possible to represent lengths of 10 through 25 using the three-byte notation. In fact, only two and a half bytes are available, because each run of length 10 to 25 uses either the low or high four bits of the byte. Again, to accommodate lengths greater than 25, there is another escape mechanism: If all 4 bits are 1 (0xF), then the next byte is also examined. If the next byte is not equal to 255 (0xFF), then that is the length, biased by -25 (-(3+7+15)), so it is possible to encode lengths of 25 through 279. Finally, if the fourth byte is 255 (0xFF), then the next two bytes contain the length, biased by -3. The following section contains pseudo code to demonstrate this encoding algorithm. The decompression pseudo code in the section after that shows how to decode the encoded offset and length.

## Pseudo Code to Encode Offset and Length into a Token

While this code is necessary to generate a well formed compressed file, the bulk of the complexity in any LZ77 compression algorithm is the logic to find duplicate runs of bytes in the input. That logic is beyond the scope of this specification.

1. Define NibbleIndex as integer := -1
2. Define EncodeOffsetLength as Function With Parameters (
3. offset as unsigned integer
4. length as unsigned integer
5. outputBuffer as unsigned char array
6. outputIndex as unsigned integer
7. )
8. Begin
9. Define nibbleValue as unsigned integer
10. offset := offset - 1
11. length := length - 3
12. offset := offset times 8
13. If (length < 7)
14. Begin
15. offset := offset OR length
16. outputBuffer [outputIndex] = offset
17. outputBuffer [outputIndex+1] = offset / 256
18. return outputIndex + 2
19. End
20. offset := offset OR 0x7
21. length -= 7;
22. outputBuffer [outputIndex] = offset
23. outputBuffer [outputIndex+1] = offset / 256
24. outputIndex := outputIndex + 2
25. nibbleValue = length
26. If (nibbleValue >= 15)
27. Begin
28. nibbleValue = 15
29. End
30. If (NibbleIndex equals -1)
31. Begin
32. NibbleIndex := outputIndex
33. outputBuffer [NibbleIndex] := nibbleValue
34. outputIndex := outputIndex + 1
35. End
36. Else
37. Begin
38. outputBuffer [NibbleIndex] := outputBuffer [NibbleIndex] OR (nibbleValue times 16)
39. NibbleIndex := -1
40. End
41. If (nibbleValue equals 15)
42. Begin
43. length := length - 15
44. If (length >= 255)
45. Begin
46. length := length + 7 + 15
47. outputBuffer [outputIndex] := 255
48. outputBuffer [outputIndex + 1] := length
49. outputBuffer [outputIndex + 2] := length / 256
50. outputIndex := outputIndex + 3
51. End
52. Else
53. Begin
54. outputBuffer [outputIndex] := length
55. outputIndex := outputIndex + 1
56. End
57. End
58. return outputIndex
59. End

## Pseudo Code to Decompress an Address Book File

Decompressing a compressed file consists of the following steps:

1. Looping over the input file.
2. Reading a **COMPRESSED\_BLOCK\_HEADER**.
3. Reading from that header the **Data** portion of the **COMPRESSED\_BLOCK** based on the **CompressedSize** field in the **COMPRESSED\_BLOCK\_HEADER** structure.
4. Allocating an output buffer based on the **DecompressedSize** field in the **COMPRESSED\_BLOCK\_HEADER** structure and decompressing the input buffer into the output buffer according to the algorithm described in earlier in this section.

The following pseudo code implements this procedure. The pseudo code assumes that file handles of some sort have been opened to the input file to be decompressed and to the decompressed file that is being created. It also assumes the existences of **ReadFile** and **WriteFile** primitives to read and write bytes to and from those files handles. In the interest of brevity, there is no error checking logic in the following pseudo code, and the pseudo code assumes a well-formed input file.

### Function to Decompress a File

1. Define DecompressFile as Function With Parameters (
2. inputFile as FileHandle,
3. outputFile as FileHandle
4. )
5. Begin
6. Define blockHeader as structure
7. Begin
8. Define crc32 as unsigned integer
9. Define compressedSize as unsigned integer
10. Define decompressedSize as unsigned integer
11. End
12. Define bufferSize as unsigned integer := 65536
13. Define inputBuffer as unsigned char [bufferSize]
14. Define outputBuffer as unsigned char [bufferSize]
15. Define bytesRead as unsigned integer
16. While (ReadFile (inputFile, blockHeader, sizeof (blockHeader)) not equal 0)
17. Begin
18. ReadFile (inputFile, inputBuffer, blockHeader.compressedSize);
19. If (blockHeader.compressedSize < blockHeader.decompressedSize)
20. Begin
21. DecompressBuffer (inputBuffer, outputBuffer, blockHeader.decompressedSize);
22. WriteFile (outputFile, outputBuffer, blockHeader.decompressedSize);
23. End
24. Else
25. Begin
26. // Block is not compressed, just write to output file
27. WriteFile (outputFile, inputBuffer, blockHeader.compressedSize);
28. End
29. End
30. End

### Function to Decompress the Bytes in a Block

The following pseudo code shows how to decompress the bytes in the **Data** field of a **COMPRESSED\_BLOCK** structure. The Print statements are for debugging purposes and are not necessary, nor are the tokenGroup and tokenIndex variables, which are there to support the Print statements.

1. Define DecompressBuffer as Function With Parameters (
2. inputBuffer as unsigned char []
3. outputBuffer as unsigned char []
4. outputSize as unsigned integer
5. )
6. Begin
7. Define inputIndex as unsigned integer
8. Define outputIndex as unsigned integer
9. Define tokenType as unsigned integer
10. Define tokenTypeMask as unsigned integer
11. Define nibbleIndex as integer
12. Define offset as integer
13. Define length as unsigned integer
14. Define tokenGroup as unsigned integer
15. Define tokenIndex as unsigned integer
16. inputIndex := 0
17. outputIndex := 0
18. tokenTypeMask := 0
19. nibbleIndex := -1
20. tokenGroup := 1
21. While (outputIndex < outputSize)
22. Begin
23. If (tokenTypeMask < 2)
24. Begin
25. tokenType := inputBuffer [inputIndex] |
26. inputBuffer [inputIndex + 1] times 256 |
27. inputBuffer [inputIndex + 2] times 256 times 256 |
28. inputBuffer [inputIndex + 3] times 256 times 256 times 256
29. inputIndex := inputIndex + 4
30. tokenTypeMask := 0x80000000
31. Print "Token Group [" + tokenGroup + "]: tokenType: " + tokenType + NewLine
32. tokenGroup := tokenGroup + 1
33. End
34. Else
35. Begin
36. tokenTypeMask := tokenTypeMask / 2
37. End
38. Print " Token [" + tokenIndex + "]: "
39. tokenIndex := tokenIndex + 1
40. If ((tokenType & tokenTypeMask) equal 0)
41. Begin
42. Print " LITERAL: " + inputBuffer [inputIndex] +
43. " copied to " + outputIndex + NewLine
44. outputBuffer [outputIndex] := inputBuffer [inputIndex]
45. inputIndex := inputIndex + 1
46. outputIndex := outputIndex + 1
47. End
48. Else
49. Begin
50. offset := inputBuffer [inputIndex] | (inputBuffer [inputIndex+1] times 256)
51. Print " RUN:" + offset
52. length := offset & 0x7
53. offset := offset / 8
54. inputIndex := inputIndex + 2
55. If (length equals 7)
56. Begin
57. If (nibbleIndex equals -1)
58. Begin
59. Print " " + inputBuffer [inputIndex] +" (low nibble)"
60. nibbleIndex := inputIndex
61. length := inputBuffer [nibbleIndex] & 0xF
62. inputIndex := inputIndex + 1
63. End
64. Else
65. Begin
66. Print " " + inputBuffer [nibbleIndex] +" (high nibble)"
67. length := inputBuffer [nibbleIndex] / 16
68. nibbleIndex := -1
69. End
70. If (length equals 15)
71. Begin
72. Print " " + inputBuffer [inputIndex]
73. length := inputBuffer [inputIndex]
74. inputIndex := inputIndex + 1
75. If (length equals 255)
76. Begin
77. length := inputBuffer [inputIndex] |
78. inputBuffer [inputIndex + 1] times 256
79. Print " " + length
80. inputIndex := inputIndex + 2
81. length := length - 15 - 7
82. End
83. length := length + 15
84. End
85. length := length + 7
86. End
87. length := length + 3
88. Print " - length: " + length + " offset: " + offset +
89. " Data at " + (outputIndex - offset - 1) +
90. ": copied to " + outputIndex + NewLine
91. While (length not equals 0)
92. Begin
93. outputBuffer [outputIndex] := outputBuffer [outputIndex - offset - 1]
94. Print " " + outputBuffer [outputIndex]
95. outputIndex := outputIndex + 1
96. length := length - 1
97. End
98. Print NewLine
99. End
100. End
101. End

# Appendix B: Hash Function

The ABF\_FULL\_TRAILER and ABF\_DELTA\_TRAILER structures contain [**hash**](#gt_b7e2b611-0af5-4fec-8af2-3f9ce7bad205) fields. The [**cryptographic hash function**](#gt_a70d1a05-9add-4870-ab52-ec5d939e93e2) used to compute these fields is implementation-specific, with the requirement that the same function be used in computing each [**hash code**](#gt_6823dc32-b762-4aa0-823b-e7daa32848b1) for each Address Book Server file in a deployment and that the hash value have a high probability of being unique for each file[<13>](#Appendix_A_13" \o "Product behavior note 13). The following pseudo code is an example of a cryptographic hash function.

1. Define HashContacts as Function With Parameters (
2. contacts as ABF\_CONTACT []
3. numberOfContacts as unsigned integer
4. )
5. Begin
6. Define contactIndex as unsigned integer
7. Define hashValue as unsigned integer
8. hashValue = 0
9. For (contactIndex=0; contactIndex<numberOfContacts; contactIndex++)
10. Begin
11. hashValue = hashValue + HashContact (contacts [contactIndex])
12. End
13. //
14. // Return 16 bit hash value
15. //
16. Return (hashValue & 0xFFFF) XOR (hashValue >> 16)
17. End
18. Define HashContact as Function With Parameters (
19. contact as ABF\_CONTACT
20. Begin
21. Define hashValue as unsigned integer
22. Define attributeIndex as unsigned integer
23. hashValue = HashBytes (contact.Id, SizeOf (contact.Id))
24. For (attributeIndex=0; attributeIndex<contact.NumberOfAttributes; attributeIndex++)
25. Begin
26. hashValue = hashValue + HashContactAttribute (contact.Attributes [attributeIndex])
27. End
28. Return hashValue
29. End
30. Define HashContactAttribute as Function With Parameters (
31. contactAttribute as ABF\_CONTACT\_ATTRIBUTE
32. )
33. Begin
34. // contactAttribute.Length may or may not be a computed field
35. // depending upon the type of attribute.
36. Return HashBytes (contactAttribute.Data, contactAttribute.Length)
37. End
38. Define HashBytes as Function With Parameters (
39. buffer as unsigned char []
40. bufferLength as unsigned integer
41. )
42. Begin
43. Define hashValue as unsigned integer
44. Define bufferIndex as unsigned integer
45. Define c as unsigned char
46. hashValue = 0
47. For (bufferIndex=0; bufferIndex<bufferLength; bufferIndex++)
48. Begin
49. c = buffer [bufferIndex]
50. hashValue = hashValue + (c << 1) + (c >> 1) + c
51. End
52. Return hashValue
53. End

# Appendix C: Active Directory Scanning Algorithm

The primary function of the Address Book Server is to scan Active Directory for users, contacts, and groups and determine which objects to include in the Address Book Server files and, for each object included, which attributes to include. The following pseudo-code example illustrates how a scanning process works.

1. // This is a representation of the [ABF\_ATTRIBUTES Structure](#Section_63c0af9fa14d4987b8d2f1c06e91da65)
2. //
3. extern Dictionary<String, ABF\_ATTRIBUTE> attributeTable;
4. adObjects = select from Active Directory where ObjectType == Group Or
5. ObjectType == User Or
6. ObjectType == Contact
7. foreach (adObject in adObjects)
8. {
9. hideObject = false;
10. includeAttributeSeen = false;
11. requiredAttributeSeen = false;
12. foreach (adAttribute in adObject.Atttributes)
13. {
14. if (attributeTable [adAttribute.Name].Flags &
15. ABF\_ATTRIBUTE\_FLAGS\_EXCLUDE)
16. {
17. // Exclude attribute seen, so bail now.
18. // Not going to take this adObject.
19. hideObject = true;
20. break;
21. }
22. if (attributeTable [adAttribute.Name].Flags &
23. ABF\_ATTRIBUTE\_FLAGS\_INCLUDE)
24. {
25. // Include attribute seen, so assume
26. // this adObject will be taken
27. includeAttributeSeen = true;
28. }
29. if (attributeTable [adAttribute.Name].Flags &
30. ABF\_ATTRIBUTE\_FLAGS\_REQUIRED)
31. {
32. // Required attribute seen, so assume
33. // this adObject will not be taken
34. requiredAttributeSeen = true;
35. }
36. }
37. If (attributeTable.HasRequiredAttributes && !requiredAttributeSeen)
38. {
39. // There was at least one ABF\_ATTRIBUTE\_FLAGS\_REQUIRED attribute
40. // defined in the attributeTable, but no required
41. // attributes on this adObject, so hide it.
42. hideObject = true;
43. }
44. else
45. If (attributeTable.HasIncludeAttributes && !includeAttributeSeen)
46. {
47. // There was at least one ABF\_ATTRIBUTE\_FLAGS\_INCLUDE attribute
48. // defined in the attributeTable, but no include
49. // attributes on this adObject, so hide it.
50. hideObject = true;
51. }
52. if (hideObject)
53. {
54. // Not interested in this adObject so on to the next one.
55. continue;
56. }
57. //
58. // This adObject should be in address book files so process it
59. //
60. }

# Appendix D: Product Behavior

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

* Microsoft Office Communications Server 2007
* Microsoft Office Communications Server 2007 R2
* Microsoft Office Communicator 2007
* Microsoft Office Communicator 2007 R2
* Microsoft Lync Server 2010
* Microsoft Lync 2010
* Microsoft Lync Server 2013
* Microsoft Lync Client 2013/Skype for Business
* Microsoft Skype for Business 2016
* Microsoft Skype for Business Server 2015
* Microsoft Skype for Business 2019
* Microsoft Skype for Business Server 2019
* Microsoft Skype for Business 2021

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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 1.3](#Appendix_A_Target_1): Office Communications Server 2007, Office Communicator 2007: Product behavior not supported. Product behavior added in relation to Knowledge Base Article 972403, July 2009. This hotfix applies to Office Communications Server 2007 R2 and Office Communicator 2007 R2.

[<2> Section 1.3](#Appendix_A_Target_2): Office Communications Server 2007, Office Communicator 2007: Product behavior not supported. Product behavior added in relation to Knowledge Base Article 972403, July 2009. This hotfix applies to Office Communications Server 2007 R2 and Office Communicator 2007 R2.

[<3> Section 1.3.3](#Appendix_A_Target_3): Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. This behavior added in relation to Knowledge Base Article 972403, July 2009. This hotfix applies to Office Communications Server 2007 R2 and Office Communicator 2007 R2.

[<4> Section 2.1.4](#Appendix_A_Target_4): This value is not used by Communicator clients.

[<5> Section 2.2.3](#Appendix_A_Target_5): Office Communicator 2007 and Office Communicator 2007 R2 uses MaximumAttributeId+1 as the value of NumberOfAttributes if MaximumAttributeId is less than 256. Otherwise, it uses the NumberOfAttributes field. To work with the clients in Appendix B, values in the Id field of ABF\_ATTRIBUTE structure must start at 1 and be contiguous.

[<6> Section 2.2.4](#Appendix_A_Target_6): Office Communications Server 2007, Office Communicator 2007: Product behavior not supported. Product behavior added in relation to Knowledge Base Article 972403, July 2009. This hotfix applies to Office Communications Server 2007 R2, Office Communicator 2007 R2.

[<7> Section 2.2.4](#Appendix_A_Target_7): Office Communicator 2007 and Office Communicator 2007 R2 uses MaximumAttributeId+1 as the value of NumberOfAttributes if MaximumAttributeId is less than 256. Otherwise, it uses the NumberOfAttributes field. To work with the clients in Appendix B, values in the Id field of ABF\_ATTRIBUTE structure must start at 1 and be contiguous.

[<8> Section 2.2.5](#Appendix_A_Target_8): Office Communicator 2007, Office Communications Server 2007: The Microsoft .NET Framework 2.0 Regular Expression library is used. For all other products, Microsoft .NET Framework 3.5 Regular Expression library is used.

[<9> Section 2.2.15](#Appendix_A_Target_9): Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. This behavior added in relation to Knowledge Base Article 972403, July 2009. This hotfix applies to Office Communications Server 2007 R2 and Office Communicator 2007 R2.

[<10> Section 2.2.16](#Appendix_A_Target_10): Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. This behavior added in relation to Knowledge Base Article 972403, July 2009. This hotfix applies to Office Communications Server 2007 R2 and Office Communicator 2007 R2.

[<11> Section 3.1](#Appendix_A_Target_11): Office Communications Server 2007, Office Communicator 2007: Product behavior not supported. Product behavior added in relation to Knowledge Base Article 972403, July 2009; starting with the phrase "and a compact delta book file". This hotfix applies to Office Communications Server 2007 R2, Office Communicator 2007 R2.

[<12> Section 3.1](#Appendix_A_Target_12): Office Communications Server 2007, Office Communicator 2007: Product behavior not supported. Product behavior added in relation to Knowledge Base Article 972403, July 2009. This hotfix release applies to Office Communications Server 2007 R2, Office Communicator 2007 R2.

[<13> Section 6](#Appendix_A_Target_13): Office Communications Server 2007 calls the Microsoft .NET Framework GetHashCode method on the individual objects to compute the hash code. The other supported products call .NET Framework 3.5 GetHashCode method on the individual objects to compute the hash code.

# Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

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

* A document revision that incorporates changes to interoperability requirements.
* A document revision that captures changes to protocol functionality.

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

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

| Section | Description | Revision class |
| --- | --- | --- |
| [8](#Section_23412d36c1b8490bb1c1d700e0422bd4) Appendix D: Product Behavior | Updated list of supported products. | Major |

# Index

3

[32-bit CRC algorithm](#section_06966aa270da4bf984483355f277cd77) 63

A

ABF\_ATTRIBUTE

 example

 [company](#section_c7e52e25406f40b095a72cebae3aa2f0) 43

 [displayName](#section_b8a003933fe64f288bf17c5f0629fa3b) 44

 [givenName](#section_7674554ce3574ad0b08fbaa3b03df149) 45

 [groupType](#section_7f5da36d634443bc9f49e86e1b047f67) 36

 [homePhone](#section_2377061dc32f43fbaa81920cab16601d) 40

 [ipPhone](#section_8695acf654cb42af80e29b84e85cd82d) 38

 [mail](#section_3317c8572d5842f0acfff998197850b4) 37

 [mailNickname](#section_3030849e9d2f4d23add2ba324f4593a5) 43

 [manager](#section_f7a06cb7a6444ee8a06273c9836f0db1) 35

 [mobile](#section_911ea0baa44d44118ed4535f6fd2a375) 39

 [msExchHideFromAddressLists](#section_169278bd7fe5458c99b213c48d19b4b2) 46

 [msRTCSIP-PrimaryUserAddress](#section_5386bf26c02341838160d280ce5bfec4) 41

 [otherHomePhone](#section_f8c4eb88b6c8401fb4096c93f56fa659) 40

 [otherMobile](#section_96487134d99c41fa976706258fdb3c5a) 39

 [otherTelephone](#section_15869da76f4248648a90cf5a8eee1a89) 38

 [physicalDeliveryOfficeName](#section_04d82f4e362c4ab7b72d95f4d1db85fc) 42

 [proxyAddresses](#section_e5b0106f99f243989b4d3aeee3303f49) 37

 [sn](#section_33967222ecf44be6b4bc184f5a72592e) 45

 [telephoneNumber](#section_f4c59d9f8481447e9c7e48e472b6eeca) 41

 [title](#section_1817c34d68364e6ab661172b78f6cc55) 44

 [structure](#section_b96276233d6e48ccb343cece7ee82faa) 21

[ABF\_ATTRIBUTE\_CLIENT\_FIELD enumeration](#section_4aa3492cd1ed4ba1b03f7eb9b58680f2) 16

[ABF\_ATTRIBUTE\_FLAGS enumeration](#section_1d7b790d2ced45ea9b87c13862f1965d) 14

[ABF\_ATTRIBUTE\_NORMALIZATION\_CONTROL enumeration](#section_e6728755205c4bc294f60f09c9f094b1) 16

[ABF\_ATTRIBUTE\_TYPE enumeration](#section_c98381f94f8647ec8e2ad063fae70690) 15

[ABF\_ATTRIBUTES structure](#section_63c0af9fa14d4987b8d2f1c06e91da65) 21

ABF\_CONTACT

 [example](#section_e2aba34997ca44f5bcf654e8215fcb08) 57

 [ABSUser1](#section_47ab6714ae35416590523f8e0242a301) 46

 [ABSUser2](#section_9811faa1cab547e39a71837095c1f9e0) 55

 [ABSUser5](#section_60f16f32c6ff4bbdb7f873eccfd4044f) 52

 [structure](#section_0f063341df544eebaabeaf2cb67ab572) 22

ABF\_CONTACT\_ATTRIBUTE

 example

 ABSUser1

 [company](#section_d5c840bd61d642b99453a74e3c65c7ec) 51

 [displayName](#section_c6be3c3fe2524d55a96ce785dfed943a) 51

 [givenName](#section_b5658346eda348beadddbfd2273be613) 52

 [homePhone](#section_b72656d18c2b43af856820982e69b9d5) 49

 [homePhone, normalized](#section_1ddc006dd4a746f8b16ddb2222c4810b) 49

 [mail](#section_27088e7a1c7d4fbd884d639f4cb5c816) 47

 [mobile](#section_79f3ac8bd9c543a39dc062cf72ba376f) 48

 [mobile, normalized](#section_0fafb92655ab4176983693a6e4202f20) 48

 [msRTCSIP-PrimaryUserAddress](#section_654b6ebbf0fb4a6689574fd00046a73e) 50

 [otherHomePhone](#section_0f687cb64c9e40f0b83dce3e914eee0d) 48

 [otherHomePhone, normalized](#section_82f4d00d54fe4d8cb2ffdd59fe722f49) 49

 [otherTelephone](#section_e9a9433af1ee4342bd2fde39b0d539c9) 47

 [otherTelephone, normalized](#section_b8bb3dd2bec54ca0af3d6e0bdb49956d) 47

 [physicalDeliveryOfficeName](#section_dcdc1f50d09d465e9d03defc28f50b53) 51

 [sn](#section_96b363ada6fa4ca483664778e9cd0f0d) 52

 [telephoneNumber](#section_aaa47d0983514b1da846d017afd0f3e1) 50

 [telephoneNumber, normalized](#section_edf69b95e21642378efe160b776ecdca) 50

 [title](#section_69d84acfd8484c0789f12b6b16b39a2d) 51

 ABSUser2

 [displayName](#section_f1fa1e3abbcb40d29987fd21fab64cbe) 57

 [givenName](#section_d5c7bca87de54f08a4dd80b023646a39) 57

 [msRTCSIP-PrimaryUserAddress](#section_055aca03f4804673b51c77690365f076) 56

 [sn](#section_790cf8d3209a4aae9a2399bb87dc9d8a) 57

 [telephoneNumber](#section_484c265166574ef09c66dd46e585d3a4) 56

 [telephoneNumber, normalized](#section_ff9e3ccd36d5455daf6010b2f2a660ee) 56

 ABSUser5

 [displayName](#section_1dfdf80af994468e82b8a69beab57ac4) 54

 [givenName](#section_f3fcf749e0634c4e96a699e27fab4478) 55

 [msRTCSIP-PrimaryUserAddress](#section_50630c1b6ffe4ebc9c0e68b01fc306f5) 54

 [sn](#section_73265ea065554087b489da1496634418) 54

 [telephoneNumber](#section_816cc4244a504afca780d0239ff70ef7) 53

 [telephoneNumber, normalized](#section_ad08de3c2d7743f3a5c5b0bb45c7d368) 53

 [structure](#section_157ffb6cd5494f64aa62e065b73b7892) 22

[ABF\_CONTACT\_CHANGES structure](#section_bce1c3bad9ad447195ad08a43865d736) 24

[ABF\_CONTACTS structure](#section_f4ce14fb9f80465a99ffbf1dd2241675) 21

[ABF\_CONTACTS\_CHANGES structure](#section_59cfb6528dc8468f9c099c09ba182650) 24

[ABF\_DELETED\_CONTACTS structure](#section_b7b5bc53dba9491cab974462e024f20d) 22

ABF\_DELTA\_HEADER

 [example](#section_14b5c87c3be345bd81efb0d467992800) 32

 [structure](#section_ec9e5b68d82a46b5adb8242ef10c005a) 19

ABF\_DELTA\_TRAILER

 [structure](#section_04c0e6760fa7437186360781b89c9a55) 23

[ABF\_FULL\_HEADER structure](#section_b8ca13d5612047ccb4ab9bd0cf4b27c1) 18

[ABF\_FULL\_TRAILER structure](#section_b745e4fbba9f44c9885cf066d2005440) 23

ABF\_NORMALIZATION\_RULES

 [example](#section_9d0e8d06305743929b848bbed8b32315) 33

 [structure](#section_1699d600cec8415cb59d05db90be0fb2) 20

ABF\_TRAILER\_LENGTH

 [example](#section_929419d2c90b475aacdf5978aa814dac) 59

 [structure](#section_63d7ef3d524040bdb169ab4bd097c70e) 24

[Active Directory scanning algorithm](#section_79750d26128e45649969aa94f4632b47) 72

Address Book

 example

 ABF\_ATTRIBUTE

 [company](#section_c7e52e25406f40b095a72cebae3aa2f0) 43

 [displayName](#section_b8a003933fe64f288bf17c5f0629fa3b) 44

 [givenName](#section_7674554ce3574ad0b08fbaa3b03df149) 45

 [groupType](#section_7f5da36d634443bc9f49e86e1b047f67) 36

 [homePhone](#section_2377061dc32f43fbaa81920cab16601d) 40

 [ipPhone](#section_8695acf654cb42af80e29b84e85cd82d) 38

 [mail](#section_3317c8572d5842f0acfff998197850b4) 37

 [mailNickname](#section_3030849e9d2f4d23add2ba324f4593a5) 43

 [manager](#section_f7a06cb7a6444ee8a06273c9836f0db1) 35

 [mobile](#section_911ea0baa44d44118ed4535f6fd2a375) 39

 [msExchHideFromAddressLists](#section_169278bd7fe5458c99b213c48d19b4b2) 46

 [msRTCSIP-PrimaryUserAddress](#section_5386bf26c02341838160d280ce5bfec4) 41

 [otherHomePhone](#section_f8c4eb88b6c8401fb4096c93f56fa659) 40

 [otherMobile](#section_96487134d99c41fa976706258fdb3c5a) 39

 [otherTelephone](#section_15869da76f4248648a90cf5a8eee1a89) 38

 [physicalDeliveryOfficeName](#section_04d82f4e362c4ab7b72d95f4d1db85fc) 42

 [proxyAddresses](#section_e5b0106f99f243989b4d3aeee3303f49) 37

 [sn](#section_33967222ecf44be6b4bc184f5a72592e) 45

 [telephoneNumber](#section_f4c59d9f8481447e9c7e48e472b6eeca) 41

 [title](#section_1817c34d68364e6ab661172b78f6cc55) 44

 [ABF\_CONTACT](#section_e2aba34997ca44f5bcf654e8215fcb08) 57

 [ABSUser1](#section_47ab6714ae35416590523f8e0242a301) 46

 ABF\_CONTACT\_ATTRIBUTE

 [company](#section_d5c840bd61d642b99453a74e3c65c7ec) 51

 [displayName](#section_c6be3c3fe2524d55a96ce785dfed943a) 51

 [givenName](#section_b5658346eda348beadddbfd2273be613) 52

 [homePhone](#section_b72656d18c2b43af856820982e69b9d5) 49

 [homePhone, normalized](#section_1ddc006dd4a746f8b16ddb2222c4810b) 49

 [mail](#section_27088e7a1c7d4fbd884d639f4cb5c816) 47

 [mobile](#section_79f3ac8bd9c543a39dc062cf72ba376f) 48

 [mobile, normalized](#section_0fafb92655ab4176983693a6e4202f20) 48

 [msRTCSIP-PrimaryUserAddress](#section_654b6ebbf0fb4a6689574fd00046a73e) 50

 [otherHomePhone](#section_0f687cb64c9e40f0b83dce3e914eee0d) 48

 [otherHomePhone, normalized](#section_82f4d00d54fe4d8cb2ffdd59fe722f49) 49

 [otherTelephone](#section_e9a9433af1ee4342bd2fde39b0d539c9) 47

 [otherTelephone, normalized](#section_b8bb3dd2bec54ca0af3d6e0bdb49956d) 47

 [physicalDeliveryOfficeName](#section_dcdc1f50d09d465e9d03defc28f50b53) 51

 [sn](#section_96b363ada6fa4ca483664778e9cd0f0d) 52

 [telephoneNumber](#section_aaa47d0983514b1da846d017afd0f3e1) 50

 [telephoneNumber, normalized](#section_edf69b95e21642378efe160b776ecdca) 50

 [title](#section_69d84acfd8484c0789f12b6b16b39a2d) 51

 [ABSUser2](#section_9811faa1cab547e39a71837095c1f9e0) 55

 ABF\_CONTACT\_ATTRIBUTE

 [displayName](#section_f1fa1e3abbcb40d29987fd21fab64cbe) 57

 [givenName](#section_d5c7bca87de54f08a4dd80b023646a39) 57

 [msRTCSIP-PrimaryUserAddress](#section_055aca03f4804673b51c77690365f076) 56

 [sn](#section_790cf8d3209a4aae9a2399bb87dc9d8a) 57

 [telephoneNumber](#section_484c265166574ef09c66dd46e585d3a4) 56

 [telephoneNumber, normalized](#section_ff9e3ccd36d5455daf6010b2f2a660ee) 56

 [ABSUser5](#section_60f16f32c6ff4bbdb7f873eccfd4044f) 52

 ABF\_CONTACT\_ATTRIBUTE

 [displayName](#section_1dfdf80af994468e82b8a69beab57ac4) 54

 [givenName](#section_f3fcf749e0634c4e96a699e27fab4478) 55

 [msRTCSIP-PrimaryUserAddress](#section_50630c1b6ffe4ebc9c0e68b01fc306f5) 54

 [sn](#section_73265ea065554087b489da1496634418) 54

 [telephoneNumber](#section_816cc4244a504afca780d0239ff70ef7) 53

 [telephoneNumber, normalized](#section_ad08de3c2d7743f3a5c5b0bb45c7d368) 53

 [ABF\_DELTA\_HEADER](#section_14b5c87c3be345bd81efb0d467992800) 32

 [ABF\_NORMALIZATION\_RULES](#section_9d0e8d06305743929b848bbed8b32315) 33

 [ABF\_TRAILER\_LENGTH](#section_929419d2c90b475aacdf5978aa814dac) 59

 [DELTA\_TRAILER](#section_f55f3d548f194528b7f5bfc12c459fbd) 58

Address Book File

 [decompress](#section_85b0a5ae15584690ba1d6708488fe5ee) 66

 [function to decompress bytes in block](#section_04d787e643ba4c58acccd72dd3973315) 67

 [function to decompress file](#section_7e434c5ebd6649118dc1659b3ddfcf58) 66

 enumerations

 [ABF\_ATTRIBUTE\_CLIENT\_FIELD](#section_4aa3492cd1ed4ba1b03f7eb9b58680f2) 16

 [ABF\_ATTRIBUTE\_FLAGS](#section_1d7b790d2ced45ea9b87c13862f1965d) 14

 [ABF\_ATTRIBUTE\_NORMALIZATION\_CONTROL](#section_e6728755205c4bc294f60f09c9f094b1) 16

 [ABF\_ATTRIBUTE\_TYPE](#section_c98381f94f8647ec8e2ad063fae70690) 15

 structures

 [ABF\_ATTRIBUTE](#section_b96276233d6e48ccb343cece7ee82faa) 21

 [ABF\_ATTRIBUTES](#section_63c0af9fa14d4987b8d2f1c06e91da65) 21

 [ABF\_CONTACT](#section_0f063341df544eebaabeaf2cb67ab572) 22

 [ABF\_CONTACT\_ATTRIBUTE](#section_157ffb6cd5494f64aa62e065b73b7892) 22

 [ABF\_CONTACT\_CHANGES](#section_bce1c3bad9ad447195ad08a43865d736) 24

 [ABF\_CONTACTS](#section_f4ce14fb9f80465a99ffbf1dd2241675) 21

 [ABF\_CONTACTS\_CHANGES](#section_59cfb6528dc8468f9c099c09ba182650) 24

 [ABF\_DELETED\_CONTACTS](#section_b7b5bc53dba9491cab974462e024f20d) 22

 [ABF\_DELTA\_HEADER](#section_ec9e5b68d82a46b5adb8242ef10c005a) 19

 [ABF\_DELTA\_TRAILER](#section_04c0e6760fa7437186360781b89c9a55) 23

 [ABF\_FULL\_HEADER](#section_b8ca13d5612047ccb4ab9bd0cf4b27c1) 18

 [ABF\_FULL\_TRAILER](#section_b745e4fbba9f44c9885cf066d2005440) 23

 [ABF\_NORMALIZATION\_RULES](#section_1699d600cec8415cb59d05db90be0fb2) 20

 [ABF\_TRAILER\_LENGTH](#section_63d7ef3d524040bdb169ab4bd097c70e) 24

 [COMPRESSED\_BLOCK](#section_cd3364d57ec349b7af2132428b59c6b1) 17

 [COMPRESSED\_BLOCK\_HEADER](#section_b2232f9c391c4e308624f5fb792f1203) 18

[Address Book File example](#section_d714352cfb9d4c9d93cabd40f8fd6ad0) 26

Address Book Server

 [configuration](#section_8c89adff4e304c5e832ab7ae8abf62ed) 10

[Applicability](#section_2d7c6de79d5c4bb095a7433b7ebfb4a6) 13

B

[Byte ordering](#section_330f65526be94024a6b38514b2651c47) 12

C

[Change tracking](#section_6651a5c3fd4a4bcb93a8b37e76465641) 76

[Common data types and fields](#section_53a2aff1da2a4c518384aeb900437a3b) 14

[Compressed Address Book File example](#section_868ed3a406ac44c09d732cc32774198b) 59

[COMPRESSED\_BLOCK structure](#section_cd3364d57ec349b7af2132428b59c6b1) 17

[COMPRESSED\_BLOCK\_HEADER example](#section_c86688c07ae64afa944d900ad3abf2aa) 60

[COMPRESSED\_BLOCK\_HEADER structure](#section_b2232f9c391c4e308624f5fb792f1203) 18

[Compression format](#section_5c0d93f605b1421fb971ab547414a6d2) 63

 [32-bit CRC algorithm](#section_06966aa270da4bf984483355f277cd77) 63

 [compressed data format](#section_fa1b4d95a5624322948311aaeb06293a) 64

 [decompress Address Book File](#section_85b0a5ae15584690ba1d6708488fe5ee) 66

 [function to decompress bytes in block](#section_04d787e643ba4c58acccd72dd3973315) 67

 [function to decompress file](#section_7e434c5ebd6649118dc1659b3ddfcf58) 66

 [encode offset and length in token](#section_37627fd8daa94598a892afa47b7eea81) 65

 [run encoding](#section_0ffecae3c30945d3bbd0b0b6d57d9a1a) 64

D

[Data types and fields - common](#section_53a2aff1da2a4c518384aeb900437a3b) 14

[Decompress Address Book File](#section_85b0a5ae15584690ba1d6708488fe5ee) 66

 [function to decompress bytes in block](#section_04d787e643ba4c58acccd72dd3973315) 67

 [function to decompress file](#section_7e434c5ebd6649118dc1659b3ddfcf58) 66

DELTA\_TRAILER

 [example](#section_f55f3d548f194528b7f5bfc12c459fbd) 58

Details

 [common data types and fields](#section_53a2aff1da2a4c518384aeb900437a3b) 14

E

Enumerations

 [ABF\_ATTRIBUTE\_CLIENT\_FIELD](#section_4aa3492cd1ed4ba1b03f7eb9b58680f2) 16

 [ABF\_ATTRIBUTE\_FLAGS](#section_1d7b790d2ced45ea9b87c13862f1965d) 14

 [ABF\_ATTRIBUTE\_NORMALIZATION\_CONTROL](#section_e6728755205c4bc294f60f09c9f094b1) 16

 [ABF\_ATTRIBUTE\_TYPE](#section_c98381f94f8647ec8e2ad063fae70690) 15

Examples

 [Address Book File](#section_d714352cfb9d4c9d93cabd40f8fd6ad0) 26

 Address Book File Example

 ABF\_ATTRIBUTE

 [company](#section_c7e52e25406f40b095a72cebae3aa2f0) 43

 [displayName](#section_b8a003933fe64f288bf17c5f0629fa3b) 44

 [givenName](#section_7674554ce3574ad0b08fbaa3b03df149) 45

 [groupType](#section_7f5da36d634443bc9f49e86e1b047f67) 36

 [homePhone](#section_2377061dc32f43fbaa81920cab16601d) 40

 [ipPhone](#section_8695acf654cb42af80e29b84e85cd82d) 38

 [mail](#section_3317c8572d5842f0acfff998197850b4) 37

 [mailNickname](#section_3030849e9d2f4d23add2ba324f4593a5) 43

 [manager](#section_f7a06cb7a6444ee8a06273c9836f0db1) 35

 [mobile](#section_911ea0baa44d44118ed4535f6fd2a375) 39

 [msExchHideFromAddressLists](#section_169278bd7fe5458c99b213c48d19b4b2) 46

 [msRTCSIP-PrimaryUserAddress](#section_5386bf26c02341838160d280ce5bfec4) 41

 [otherHomePhone](#section_f8c4eb88b6c8401fb4096c93f56fa659) 40

 [otherMobile](#section_96487134d99c41fa976706258fdb3c5a) 39

 [otherTelephone](#section_15869da76f4248648a90cf5a8eee1a89) 38

 [physicalDeliveryOfficeName](#section_04d82f4e362c4ab7b72d95f4d1db85fc) 42

 [proxyAddresses](#section_e5b0106f99f243989b4d3aeee3303f49) 37

 [sn](#section_33967222ecf44be6b4bc184f5a72592e) 45

 [telephoneNumber](#section_f4c59d9f8481447e9c7e48e472b6eeca) 41

 [title](#section_1817c34d68364e6ab661172b78f6cc55) 44

 [ABF\_CONTACT](#section_e2aba34997ca44f5bcf654e8215fcb08) 57

 [ABSUser1](#section_47ab6714ae35416590523f8e0242a301) 46

 ABF\_CONTACT\_ATTRIBUTE

 [company](#section_d5c840bd61d642b99453a74e3c65c7ec) 51

 [displayName](#section_c6be3c3fe2524d55a96ce785dfed943a) 51

 [givenName](#section_b5658346eda348beadddbfd2273be613) 52

 [homePhone](#section_b72656d18c2b43af856820982e69b9d5) 49

 [homePhone, normalized](#section_1ddc006dd4a746f8b16ddb2222c4810b) 49

 [mail](#section_27088e7a1c7d4fbd884d639f4cb5c816) 47

 [mobile](#section_79f3ac8bd9c543a39dc062cf72ba376f) 48

 [mobile, normalized](#section_0fafb92655ab4176983693a6e4202f20) 48

 [msRTCSIP-PrimaryUserAddress](#section_654b6ebbf0fb4a6689574fd00046a73e) 50

 [otherHomePhone](#section_0f687cb64c9e40f0b83dce3e914eee0d) 48

 [otherHomePhone, normalized](#section_82f4d00d54fe4d8cb2ffdd59fe722f49) 49

 [otherTelephone](#section_e9a9433af1ee4342bd2fde39b0d539c9) 47

 [otherTelephone, normalized](#section_b8bb3dd2bec54ca0af3d6e0bdb49956d) 47

 [physicalDeliveryOfficeName](#section_dcdc1f50d09d465e9d03defc28f50b53) 51

 [sn](#section_96b363ada6fa4ca483664778e9cd0f0d) 52

 [telephoneNumber](#section_aaa47d0983514b1da846d017afd0f3e1) 50

 [telephoneNumber, normalized](#section_edf69b95e21642378efe160b776ecdca) 50

 [title](#section_69d84acfd8484c0789f12b6b16b39a2d) 51

 [ABSUser2](#section_9811faa1cab547e39a71837095c1f9e0) 55

 ABF\_CONTACT\_ATTRIBUTE

 [displayName](#section_f1fa1e3abbcb40d29987fd21fab64cbe) 57

 [givenName](#section_d5c7bca87de54f08a4dd80b023646a39) 57

 [msRTCSIP-PrimaryUserAddress](#section_055aca03f4804673b51c77690365f076) 56

 [sn](#section_790cf8d3209a4aae9a2399bb87dc9d8a) 57

 [telephoneNumber](#section_484c265166574ef09c66dd46e585d3a4) 56

 [telephoneNumber, normalized](#section_ff9e3ccd36d5455daf6010b2f2a660ee) 56

 [ABSUser5](#section_60f16f32c6ff4bbdb7f873eccfd4044f) 52

 ABF\_CONTACT\_ATTRIBUTE

 [displayName](#section_1dfdf80af994468e82b8a69beab57ac4) 54

 [givenName](#section_f3fcf749e0634c4e96a699e27fab4478) 55

 [msRTCSIP-PrimaryUserAddress](#section_50630c1b6ffe4ebc9c0e68b01fc306f5) 54

 [sn](#section_73265ea065554087b489da1496634418) 54

 [telephoneNumber](#section_816cc4244a504afca780d0239ff70ef7) 53

 [telephoneNumber, normalized](#section_ad08de3c2d7743f3a5c5b0bb45c7d368) 53

 [ABF\_DELTA\_HEADER](#section_14b5c87c3be345bd81efb0d467992800) 32

 [ABF\_NORMALIZATION\_RULES](#section_9d0e8d06305743929b848bbed8b32315) 33

 [ABF\_TRAILER\_LENGTH](#section_929419d2c90b475aacdf5978aa814dac) 59

 [DELTA\_TRAILER](#section_f55f3d548f194528b7f5bfc12c459fbd) 58

 [Compressed Address Book File](#section_868ed3a406ac44c09d732cc32774198b) 59

 [COMPRESSED\_BLOCK\_HEADER](#section_c86688c07ae64afa944d900ad3abf2aa) 60

F

[Fields - vendor-extensible](#section_e2c9229a5a414b569484a0a1bb32299f) 13

File format

 [compressed](#section_8ec5de01be254a75875e68b194ce6d3a) 11

 [decompressed](#section_8b67c8d09df24218abef7bd5b9a58366) 11

G

[Glossary](#section_0c7c72f3b6214f80b2eac5be149646ec) 7

H

[Hash function](#section_5af7bd1aa87f4f2a8485bd10d85534fa) 70

I

[Implementer - security considerations](#section_82b4e5f6c78a48518801a85533d198e3) 62

[Informative references](#section_c01f67af4259456aa091eb4bf745500c) 9

[Introduction](#section_482029443bc44bdb970d11eeb56d3dbe) 7

L

[Localization](#section_ee3f9619f0634828b52e53593bc88562) 13

N

[Normative references](#section_2e6f2cd1b0a54aea8b53048e7b217c1d) 9

O

[Overview (synopsis)](#section_a87e47f66f1c44b3b7c31eded5f117d9) 9

 [Address Book Server configuration](#section_8c89adff4e304c5e832ab7ae8abf62ed) 10

 [byte ordering](#section_330f65526be94024a6b38514b2651c47) 12

 file format

 [compressed](#section_8ec5de01be254a75875e68b194ce6d3a) 11

 [decompressed](#section_8b67c8d09df24218abef7bd5b9a58366) 11

P

[Product behavior](#section_23412d36c1b8490bb1c1d700e0422bd4) 74

R

[References](#section_3b305d0f1a0e498482fab00cbaac5a21) 9

 [informative](#section_c01f67af4259456aa091eb4bf745500c) 9

 [normative](#section_2e6f2cd1b0a54aea8b53048e7b217c1d) 9

[Relationship to protocols and other structures](#section_91c484f431f34e8e8cdc845b28188e24) 13

S

[Scanning algorithm](#section_79750d26128e45649969aa94f4632b47) 72

Security

 [implementer considerations](#section_82b4e5f6c78a48518801a85533d198e3) 62

 [index of security fields](#section_bf1737881deb403c885911e77fb3b4c3) 62

Structures

 [ABF\_ATTRIBUTE](#section_b96276233d6e48ccb343cece7ee82faa) 21

 [ABF\_ATTRIBUTES](#section_63c0af9fa14d4987b8d2f1c06e91da65) 21

 [ABF\_CONTACT](#section_0f063341df544eebaabeaf2cb67ab572) 22

 [ABF\_CONTACT\_ATTRIBUTE](#section_157ffb6cd5494f64aa62e065b73b7892) 22

 [ABF\_CONTACT\_CHANGES](#section_bce1c3bad9ad447195ad08a43865d736) 24

 [ABF\_CONTACTS](#section_f4ce14fb9f80465a99ffbf1dd2241675) 21

 [ABF\_CONTACTS\_CHANGES](#section_59cfb6528dc8468f9c099c09ba182650) 24

 [ABF\_DELETED\_CONTACTS](#section_b7b5bc53dba9491cab974462e024f20d) 22

 [ABF\_DELTA\_HEADER](#section_ec9e5b68d82a46b5adb8242ef10c005a) 19

 [ABF\_DELTA\_TRAILER](#section_04c0e6760fa7437186360781b89c9a55) 23

 [ABF\_FULL\_HEADER](#section_b8ca13d5612047ccb4ab9bd0cf4b27c1) 18

 [ABF\_FULL\_TRAILER](#section_b745e4fbba9f44c9885cf066d2005440) 23

 [ABF\_NORMALIZATION\_RULES](#section_1699d600cec8415cb59d05db90be0fb2) 20

 [ABF\_TRAILER\_LENGTH](#section_63d7ef3d524040bdb169ab4bd097c70e) 24

 [COMPRESSED\_BLOCK](#section_cd3364d57ec349b7af2132428b59c6b1) 17

 [COMPRESSED\_BLOCK\_HEADER](#section_b2232f9c391c4e308624f5fb792f1203) 18

 [overview](#section_53a2aff1da2a4c518384aeb900437a3b) 14

T

[Tracking changes](#section_6651a5c3fd4a4bcb93a8b37e76465641) 76

V

[Vendor-extensible fields](#section_e2c9229a5a414b569484a0a1bb32299f) 13

[Versioning](#section_ee3f9619f0634828b52e53593bc88562) 13