[MS-ADDM-Diff]:

Active Directory Web Services: Data Model and Common Elements

---

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
12/5/2008	0.1	Major	Initial Availability
1/16/2009	1.0	Major	Updated and revised the technical content.
2/27/2009	2.0	Major	Updated and revised the technical content.
4/10/2009	3.0	Major	Updated and revised the technical content.
5/22/2009	4.0	Major	Updated and revised the technical content.
7/2/2009	5.0	Major	Updated and revised the technical content.
8/14/2009	5.1	Minor	Clarified the meaning of the technical content.
9/25/2009	6.0	Major	Updated and revised the technical content.
11/6/2009	7.0	Major	Updated and revised the technical content.
12/18/2009	8.0	Major	Updated and revised the technical content.
1/29/2010	8.0.1	Editorial	Changed language and formatting in the technical content.
3/12/2010	8.0.2	Editorial	Changed language and formatting in the technical content.
4/23/2010	8.1	Minor	Clarified the meaning of the technical content.
6/4/2010	8.1.1	Editorial	Changed language and formatting in the technical content.
7/16/2010	9.0	Major	Updated and revised the technical content.
8/27/2010	9.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	9.0	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	9.0	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	9.0	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	9.0	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	9.0	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	9.0	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	9.1	Minor	Clarified the meaning of the technical content.
9/23/2011	9.1	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	10.0	Major	Updated and revised the technical content.
3/30/2012	10.0	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
7/12/2012	10.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	10.0	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	10.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	11.0	Major	Updated and revised the technical content.
11/14/2013	11.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	11.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	11.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	12.0	Major	Significantly changed the technical content.
10/16/2015	12.0	None	No changes to the meaning, language, or formatting of the technical content.
7/14/2016	13.0	Major	Significantly changed the technical content.
6/1/2017	13.0	None	No changes to the meaning, language, or formatting of the technical content.
9/15/2017	14.0	Major	Significantly changed the technical content.
9/12/2018	15.0	Major	Significantly changed the technical content.
3/15/2019	16.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

Active Directory Web Services: Data Model and Common Elements contains an XML data model and other protocol components (such as the definition of an XPath 1.0-derived selection language) that are used in various protocols that belong to the set of Active Directory Web Services protocols. The documentation for individual protocols contains references to this document, as needed.

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

1.1 Glossary

This document uses the following terms:

Active Directory: The Windows implementation of a general-purpose directory service, which uses LDAP as its primary access protocol. Active Directory 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]. Active Directory is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS), which are both described in [MS-ADOD]: 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 [MS-ADTS].

Active Directory Lightweight Directory Services (AD LDS): A directory service (DS) implemented by a domain controller (DC). AD LDS is a deployment of Active Directory [MS-ADTS]. The most significant difference between AD LDS and Active Directory Domain Services (AD DS) is that AD LDS does not host domain naming contexts (domain NCs). A server can host multiple AD LDS DCs. Each DC is an independent AD LDS instance, with its own independent state. AD LDS can be run as an operating system DS or as a directory service provided by a standalone application (Active Directory Application Mode (ADAM)).

attribute syntax: Specifies the format and range of permissible values of an attribute. The syntax of an attribute is defined by several attributes on the attributeSchema object, as specified in [MS-ADTS] section 3.1.1.2. Attribute syntaxes supported by Active Directory include Boolean, Enumeration, Integer, LargeInteger, String(UTC-Time), Object(DS-DN), and String(Unicode).

directory object: A Lightweight Directory Access Protocol (LDAP) object, as specified in [RFC2251], that is a specialization of an object.

directory service (DS): A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also Active Directory.

directory tree: An LDAP directory service is organized into a hierarchical tree structure in which each directory object has exactly one parent directory object (except for one object that serves as the root of the tree) and zero or more child directory objects.

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

endpoint: In the context of a web service, a network target to which a SOAP message can be addressed. See [WSADDR].

global catalog (GC): A unified partial view of multiple naming contexts (NCs) in a distributed partitioned directory. The Active Directory directory service GC is implemented by GC servers. The definition of global catalog is specified in [MS-ADTS] section 3.1.1.1.8.

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

Lightweight Directory Access Protocol (LDAP): The primary access protocol for Active Directory. Lightweight Directory Access Protocol (LDAP) is an industry-standard protocol, established by the Internet Engineering Task Force (IETF), which allows users to query and update information in a directory service (DS), as described in [MS-ADTS]. The Lightweight Directory Access Protocol can be either version 2 [RFC1777] or version 3 [RFC3377].

naming context (NC): An NC is a set of objects organized as a tree. It is referenced by a DSName. The DN of the DSName is the distinguishedName attribute of the tree root. The GUID of the DSName is the objectGUID attribute of the tree root. The security identifier (SID) of the DSName, if present, is the objectSid attribute of the tree root; for Active Directory Domain Services (AD DS), the SID is present if and only if the NC is a domain naming context (domain NC). Active Directory supports organizing several NCs into a tree structure.

object reference property: In Active Directory Web Services, this is the property that uniquely identifies a directory object. It can be expressed as either a GUID or as a distinguished name.

object reference syntax: An attribute syntax that supports object references. The five object reference syntaxes are specified in [MS-ADTS] section 3.1.1.1.6, and the referential integrity constraints around attributes with these syntaxes are specified in [MS-ADTS] section 3.1.1.2.2.3.

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

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

SOAP fault: A container for error and status information within a SOAP message. See [SOAP1.2-1/2007] section 5.4 for more information.

SOAP header: A mechanism for implementing extensions to a SOAP message in a decentralized manner without prior agreement between the communicating parties. See [SOAP1.2-1/2007] section 5.2 for more information.

SOAP message: An XML document consisting of a mandatory SOAP envelope, an optional SOAP header, and a mandatory SOAP body. See [SOAP1.2-1/2007] section 5 for more information.

synthetic attribute: In Active Directory Web Services, an attribute that is part of the XML view of a directory object but which is not part of the directory object as stored in the directory service.

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of messages in client and server applications communicating over open networks. TLS supports server and, optionally, client authentication by using X.509 certificates (as specified in [X509]). TLS is standardized in the IETF TLS working group.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

universally unique identifier (UUID): A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and RPC objects. UUIDs are highly likely to be unique. UUIDs are also known as globally unique identifiers (GUIDs) and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

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

1.2 References

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

1.2.1 (Updated Section) 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.

[MS-ADCAP] Microsoft Corporation, "Active Directory Web Services: Custom Action Protocol".

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

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

[MS-ERREF] Microsoft Corporation, "Windows Error Codes".

[MS-NMFTB] Microsoft Corporation, ".NET Message Framing TCP Binding Protocol".

[MS-NNS] Microsoft Corporation, ".NET NegotiateStream Protocol".

[MS-WSDS] Microsoft Corporation, "WS-Enumeration: Directory Services Protocol Extensions".

[MS-WSPELD] Microsoft Corporation, "WS-Transfer and WS-Enumeration Protocol Extension for Lightweight Directory Access Protocol v3 Controls".

[MS-WSTIM] Microsoft Corporation, "WS-Transfer: Identity Management Operations for Directory Access Extensions".

[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

[RFC2251] Wahl, M., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997, http://www.ietf.org/rfc/rfc2251.txt

[RFC2252] Wahl, M., Coulbeck, A., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997, http://www.ietf.org/rfc/rfc2252.txt

[RFC4122] Leach, P., Mealling, M., and Salz, R., "A Universally Unique Identifier (UUID) URN Namespace", RFC 4122, July 2005, http://www.rfc-editor.org/rfc/rfc4122.txt

[RFC4346] Dierks, T., and Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006, http://www.ietf.org/rfc/rfc4346.txt

[SOAP1.2-1/2003] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part1-20030624

[WSADDR] Gudgin, M., Hadley, M., and Rogers, T., "Web Services Addressing (WS-Addressing) 1.0", W3C Recommendation, May 2006, http://www.w3.org/2005/08/addressing

[WSASB] Gudgin, M., Hadley, M., and Rogers, T., Eds., "Web Services Addressing 1.0 - SOAP Binding", W3C Recommendation, May 2006, http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/

[WSENUM] Alexander, J., Box, D., Cabrera, L.F., et al., "Web Services Enumeration (WS-Enumeration)", March 2006, http://www.w3.org/Submission/2006/SUBM-WS-Enumeration-20060315/

[WSMETA] Ballinger, K., Bissett, B., Box, D., et al., "Web Services Metadata Exchange (WS-MetadataExchange)", Version 1.1, August 2006, http://specs.xmlsoap.org/ws/2004/09/mex/WS-MetadataExchange.pdf

[WSSUTP1.1] OASIS Standard, "Web Services Security UsernameToken Profile 1.1", February 2006, http://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf

[WSS] OASIS, "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)", February 2006, http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

[WXFR] Alexander, J., Box, D., Cabrera, L.F., et al., "Web Services Transfer (WS-Transfer)", September 2006, http://www.w3.org/Submission/2006/SUBM-WS-Transfer-20060927/

[XML10] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Third Edition)", February 2004, http://www.w3.org/TR/2004/REC-xml-20040204/

[XMLNS-2ED] Bray, T., Hollander, D., Layman, A., and Tobin, R., Eds., "Namespaces in XML 1.0 (Second Edition)", W3C Recommendation, August 2006, http://www.w3.org/TR/2006/REC-xml-names-20060816/

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XPATH] Clark, J. and DeRose, S., "XML Path Language (XPath), Version 1.0", W3C Recommendation, November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116/

1.2.2 Informative References

[MSFT-RSAT] Microsoft Corporation, "Remote Server Administration Tools (RSAT) for Windows operating systems", https://support.microsoft.com/en-us/kb/2693643

1.3 Overview

Active Directory Web Services (ADWS) permits access to Active Directory [MS-ADTS] via the use of common SOAP-based Web Service protocols such as WS-Transfer [WXFR] and WS-Enumeration

[WSENUM]. These protocols operate on an XML [XML10] view of the data stored in the Active Directory directory service. The same XML view is shared by all the protocols in the ADWS protocol set. This document specifies that shared XML view.

Additionally, the protocols share a selection language, derived from XPath 1.0 [XPATH], that is used to specify which aspect of the XML view is operated on. That shared selection language is also specified in this document.

This document also specifies other shared cross-protocol aspects of ADWS, such as the endpoints used and shared SOAP headers and SOAP fault details [SOAP1.2-1/2003].

Finally, this document provides a mechanism for performing a range retrieval operation through some Web Service protocols in the ADWS protocol set. Range retrieval, as specified in section 2.7, allows for returning only a portion of the complete set of values of a multivalued attribute, or specifying that only a certain portion of the set of values of a multivalued attribute be retrieved. For the same purpose, it defines an extension to the shared XML view of data that incorporates this range retrieval extension.

Note that this document does not define a protocol. Rather, it serves as a common repository for information used across the entire ADWS protocol set. For operations such as range retrieval, it provides common extensions to [WXFR] and [WSENUM], which are used by certain protocols within the ADWS protocol set.<1>

1.4 Relationship to Protocols and Other Structures

The information in this document is used by protocols in the set of Active Directory Web Services protocols. The ADWS protocol documentation set comprises this document and the following documents: [MS-WSDS], [MS-WSPELD], [MS-WSTIM], and [MS-ADCAP].

1.5 Applicability Statement

The XML data model and XPath 1.0-derived selection language is suitable for use when the implementer desires to retrieve and manipulate data stored in a directory service via an XML-based model. It can be particularly useful with protocols, such as many SOAP-based Web Service protocols, that are intended to operate over data that is represented as an XML document.

There is an implicit assumption in the design of the data model that the directory service exposes semantics similar to that of a Lightweight Directory Access Protocol (LDAP) version 3 directory service [RFC2251]. For example, it assumes that objects in the directory consist of attribute-value pairs in which each attribute can have one or more values. It also assumes that the directory objects can be arranged in a single hierarchical tree structure. The XML data model described in this document might not be suitable for use with directories that do not expose such semantics.

1.6 Versioning and Localization

None.

1.7 Vendor-Extensible Fields

None.

2 Data Model and Common Elements

This section discusses the shared protocol elements that are used by various protocols in the set of Active Directory Web Service protocols. In this document, the convention from [MS-ADTS] section 3.1.1.1.2 is adopted such that, if variable *O* refers to a directory object and a is the LDAP display name of an attribute, then *O*!a denotes the value or values of attribute a on object *O*.

2.1 (Updated Section) Endpoints

This section specifies the Web Service endpoints that are used by protocols in the ADWS protocol set. ADWS exposes protocols that can be accessed via an endpoint. Each endpoint can be uniquely identified by a Uniform Resource Identifier (URI). The URIs for the ADWS protocols are shown in the following table. All endpoints use the "net.tcp" URI binding type. For semantics of this binding type, see [MS-NMFTB].

Endpoint URI	Protocol exposed by endpoint	Authentication mechanism (see below)
net.tcp://localhost:9389/ActiveDirectoryWebServices/Windows/Resource	[WXFR], [MS-WSTIM]	Windows Integrated
net.tcp://localhost:9389/ActiveDirectoryWebServices/Windows/ResourceFactory	[MS-WSTIM]	Windows Integrated
net.tcp://localhost:9389/ActiveDirectoryWebServices/Windows/Enumeration	[WSENUM], [MS-WSDS]	Windows Integrated
net.tcp://localhost:9389/ActiveDirectoryWebServices/Windows/AccountManagement	[MS-ADCAP]	Windows Integrated
net.tcp://localhost:9389/ActiveDirectoryWebServices/Windows/TopologyManagement	[MS-ADCAP]	Windows Integrated
net.tcp://localhost:9389/ActiveDirectoryWebServices/UserName/Resource	[WXFR], [MS-WSTIM]	Username/password
net.tcp://localhost:9389/ActiveDirectoryWebServices/UserName/ResourceFactory	[MS-WSTIM]	Username/password
net.tcp://localhost:9389/ActiveDirectoryWebServices/UserName/Enumeration	[WSENUM], [MS-WSDS]	Username/password
net.tcp://localhost:9389/ActiveDirectoryWebServices/UserName/AccountManagement	[MS-ADCAP]	Username/password
net.tcp://localhost:9389/ActiveDirectoryWebServices/UserName/TopologyManagement	[MS-ADCAP]	Username/password
net.tcp://localhost:9389/ActiveDirectoryWebServices/mex	[WSMETA]	None

In the preceding table, "localhost" represents the DNS hostname of the server hosting the endpoint. All endpoints listen on TCP port 9389.

The ADWS protocol set uses two types of authentication. Each endpoint (except for the "mex" endpoint) supports one or the other. The forms of authentication are:

- Windows Integrated: These endpoints use ~~Transport Layer Security (TLS) [RFC4346] to protect the TCP transport. Integrated~~integrated Windows authentication ~~using~~with the .Net Negotiate Stream protocol [MS-NNS] ~~is used~~ to authenticate the client ~~to the server~~and provide message security at the transport layer~~ and to negotiate the session key used for TLS~~.

- Username/password: These endpoints use TLS to protect the TCP transport. TLS is used to negotiate a session key to protect the TCP transport. The client authenticates (at the message layer) to the server by providing a plaintext username and password, as documented in WS-Security [WSS] and the WS-Security UserNameToken profile [WSSUTP1.1].

The "mex" endpoint neither requires nor supports authentication.

2.2 XML Namespaces and URIs

The following XML namespaces are defined and referenced by the ADWS protocol set, using the XML namespace mechanisms defined in [XMLNS-2ED]. A brief informative summary of each namespace is included in the table below. The detailed usage and semantics of each namespace are explained in the portion of the document that makes use of it. Some namespaces are used by multiple ADWS protocols or protocol components. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and is not significant for interoperability.

Prefix	Namespace URI	Informative summary
ad:	http://schemas.microsoft.com/2008/1/ActiveDirectory	The core ADWS namespace. Most ADWS protocol elements are located in this namespace.
addata:	http://schemas.microsoft.com/2008/1/ActiveDirectory/Data	The namespace for ADWS protocol elements that correspond to the LDAP display names of Active Directory classes and attributes.
adlq:	http://schemas.microsoft.com/2008/1/ActiveDirectory/Dialect/LdapQuery	The LdapQuery language, defined in [MS-WSDS].
da:	http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess	The namespace for the [MS-WSTIM] protocol.
ca:	http://schemas.microsoft.com/2008/1/ActiveDirectory/CustomActions	The namespace for the [MS-ADCAP] protocol.

Additionally, ADWS defines the following three URIs which do not correspond to XML namespaces.

URI	Informative summary
http://schemas.microsoft.com/2008/1/ActiveDirectory/Data/fault	The fault action URI [SOAP1.2-1/2003] for ADWS-defined SOAP faults, excluding those defined by [MS-WSTIM] (used for the "[Action]" property of [WSASB]).

URI	Informative summary
http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess/fault	The fault action URI [SOAP1.2-1/2003] for SOAP faults defined by [MS-WSTIM] protocol (used for the "[Action]" property of [WSASB]).
http://schemas.microsoft.com/2008/1/ActiveDirectory/Dialect/XPath-Level-1	The name of the XPath 1.0-derived selection language defined in section 2.4.

Although not defined by ADWS, the following XML namespaces are referenced elsewhere in this document.

Prefix	Namespace URI	Reference
soapenv:	http://www.w3.org/2003/05/soap-envelope	[SOAP1.2-1/2003]
wsa:	http://www.w3.org/2005/08/addressing	[WSADDR]
wsen:	http://schemas.xmlsoap.org/ws/2004/09/enumeration	[WSENUM]
wxf:	http://schemas.xmlsoap.org/ws/2004/09/transfer	[WXFR]
xsd:	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
xsi:	http://www.w3.org/2001/XMLSchema-instance	[XMLSCHEMA1]

2.3 XML Data Model

This section documents how directory objects, each of which is a collection of LDAP attributes (with one or more values stored in each attribute) [MS-ADTS], are represented in XML. This XML view of directory objects is shared by the protocols in the ADWS protocol set.

2.3.1 Object Naming

In the ADWS data model, directory objects are identified by their object reference property. The object reference property can be either a GUID or the object's LDAP distinguished name.

Note Unless otherwise specified, GUID values are represented using the following forms in this document:

- In the descriptive text, GUID values are represented by Curly Braced String form defined in [MS-DTYP] section 2.3.4.3.
- In the XML examples and definitions, GUID values are represented by the string form of a universally unique identifier (UUID), as specified in [RFC4122] section 3.

For a directory object *O*, to specify the object reference property of *O* as a GUID, the value of the GUID MUST equal the value of *O*!objectGUID. Alternatively, the object reference property of *O* can be specified as *O*'s LDAP distinguished name (*O*!distinguishedName) instead.

The object reference property (in either GUID or distinguished name form) in a SOAP message request identifies the directory object that should be operated on by the operation specified in that message (see section 2.5.2). The object reference property in a SOAP response message indicates the identity of a directory object that is returned in that response message.

The object reference property value in the GUID form of {11111111-1111-1111-1111-111111111111} exclusively refers to the LDAP rootDSE [RFC2251].

The following SOAP message requests use the object reference property as either the GUID or the distinguished name:

- In adlq:BaseObject in LdapQuery [MS-WSDS]
- In the ad:objectReferenceProperty SOAP header for a WS-Transfer [WXFR] Get, Put, or Delete operation (section 2.5.2)
- In the ad:objectReferenceProperty SOAP header for a [MS-WSTIM] BaseObjectSearchRequest or ModifyRequest operation (section 2.5.2)
- As the value of a directory attribute which has an object reference syntax (see [MS-ADTS], section 3.1.1.1.6)
- In the ad:container-hierarchy-parent (see section 2.3.3.2) synthetic attribute for a WS-Transfer Put or Create operation
- In the ad:container-hierarchy-parent (see section 2.3.3.2) synthetic attribute for a [MS-WSTIM] ModifyRequest or AddRequest operation

The object reference property in a protocol response can be in either GUID or distinguished name form.

2.3.2 XML View of Directory Objects

In the XML view of the directory objects presented by ADWS, the XML elements are named for the LDAP classes and attributes used in the directory object. Additionally, XML elements are used to represent the ADWS synthetic attributes, described in the next section.

Begin by defining how a single LDAP attribute and its value(s) are represented in the XML view. Let *A* be the LDAP display name of an attribute that has values *V1(A)*...*Vn(A)*. Let *S1(A)*...*Sn(A)* be the XML representation of values *V1*...*Vn* as described in section 2.3.4. Let *LDAPSYN(A)* be the LDAP attribute syntax of attribute *A*, and let *XMLSYN(A)* be the corresponding XML syntax, as described in section 2.3.4. The XML representation for this attribute is the following.

```
<addata:A LdapSyntax="LDAPSYN(A)">
    <ad:value xsi:type="XMLSYN(A)">
        S1(A)
    </ad:value>
    ...
    ...
    <ad:value xsi:type="XMLSYN(A)">
        Sn(A)
    </ad:value>
</addata:A>
```

Now extend this view to an entire directory object. Let *O* be an object in the directory. Let *C* be the LDAP display name of the most specific structural object class ([MS-ADTS] section 3.1.1.1.4) of *O*. Let *A1*...*An* be the LDAP display names of all the LDAP attributes of *O*. Then, the representation of *O* as the XML view in the data model is the following.

```
<addata:C>
    <addata:A1 LdapSyntax="LDAPSYN(A1)">
        <ad:value xsi:type="XMLSYN(A1)">
            S1(A1)
        </ad:value>
        ...
```

```
            ...
            <ad:value xsi:type="XMLSYN(A1)">
                Sn(A1)
            </ad:value>
        </addata:A1>
        ...
        ...
        <addata:An LdapSyntax="LDAPSYN(An)">
            <ad:value xsi:type="XMLSYN(An)">
                S1(An)
            </ad:value>
            ...
            ...
            <ad:value xsi:type="XMLSYN(An)">
                Sn(An)
            </ad:value>
        </addata:An>
    </addata:C>
```

Not shown in the above example are the ADWS synthetic attributes. These are shown in the next section.

The root element is named for the LDAP display name of the most specific structural object class of *O*, and is in the http://schemas.microsoft.com/2008/1/ActiveDirectory/Data XML namespace. When representing an LDAP display name where the most specific structural object class of *O* is not available, "top" is used for the name of the root element. Additionally, when representing the LDAP rootDse, "top" is used for the name of the root element.

Each child element represents a single LDAP attribute stored on that object and is named for that attribute's LDAP display name (and is also located in the http://schemas.microsoft.com/2008/1/ActiveDirectory/Data XML namespace). This element can have an XML attribute named LdapSyntax that represents the LDAP attribute syntax of that LDAP attribute. Each child element under an attribute represents a single value stored in that attribute. The actual value is represented as a text node under this ad:value element.

The LdapSyntax XML attribute is present for each LDAP attribute specified in a SOAP response, including the above XML representation of a directory object.

The LdapSyntax XML attribute is optional in a SOAP request.

Multiple directory objects are represented as sibling XML elements, regardless of the hierarchical relationship between the objects in the LDAP directory tree.

2.3.3 Synthetic Attributes

In addition to containing the LDAP attributes of a directory object, the XML view of that object contains up to four additional attributes that are not part of that object's representation stored in the directory service (that is, the four attributes are constructed by the server implementing the ADWS protocol set). These are referred to as the synthetic attributes of ADWS. They can be distinguished from LDAP attributes because the elements that represent the synthetic attributes have names that are in the http://schemas.microsoft.com/2008/1/ActiveDirectory XML namespace rather than in the http://schemas.microsoft.com/2008/1/ActiveDirectory/Data XML namespace that is used for LDAP attributes and classes. Additionally, the LdapSyntax XML attribute is never included in the XML representation of a synthetic attribute.

The four synthetic attributes are specified in the following subsections.

2.3.3.1 ad:objectReferenceProperty

The synthetic attribute ad:objectReferenceProperty contains the object reference property of the directory object, as described in section 2.3.1. Values of this attribute have xsi:type equal to "xsd:string".

This attribute is read only.

This attribute is optional.<2>

The following is an example of the ad:objectReferenceProperty synthetic attribute as it would be found in the XML view of a directory object. In this example, the object reference property is in the GUID form.

```
<ad:objectReferenceProperty>
    <ad:value xsi:type="xsd:string">
        e4f8a504-d7df-4b63-a636-5642d3bf1cf6
    </ad:value>
</ad:objectReferenceProperty>
```

2.3.3.2 ad:container-hierarchy-parent

The synthetic attribute ad:container-hierarchy-parent contains the object reference property (as described in section 2.3.1) of the directory object that is the object's parent in the directory tree. If the directory object has no parent (that is, if it is the root of its naming context), this attribute is omitted from the object's XML view.<3> Values of this attribute have xsi:type equal to "xsd:string".

This attribute can be modified. When this attribute is modified, the object's location in the directory is made consistent with the value of this attribute.

The following is an example of the ad:container-hierarchy-parent synthetic attribute as it would be found in the XML view of a directory object. In this example, the object reference property is in the GUID form.

```
<ad:container-hierarchy-parent>
    <ad:value xsi:type="xsd:string">
        d8f7a25a-26f5-4463-bbe3-aa01e4002afd
    </ad:value>
</ad:container-hierarchy-parent>
```

2.3.3.3 ad:distinguishedName

The synthetic attribute ad:distinguishedName contains the LDAP distinguished name of the directory object; that is, the value of *O*!distinguishedName where *O* is the directory object being represented as a XML view. Values of this attribute have xsi:type equal to "xsd:string".

This attribute is read only.

The following is an example of the ad:distinguishedName synthetic attribute as it would be found in the XML view of a directory object.

```
<ad:distinguishedName>
    <ad:value xsi:type="xsd:string">CN=Test,DC=fabrikam,DC=com</ad:value>
</ad:distinguishedName>
```

2.3.3.4 ad:relativeDistinguishedName

The synthetic attribute ad:relativeDistinguishedName contains the relative distinguished name of the directory object. Values of this attribute have xsi:type equal to "xsd:string".

This attribute can be modified. When this attribute is modified, the object's relative distinguished name is made consistent with the value of this attribute.

The following is an example of the ad:relativeDistinguishedName synthetic attribute as it would be found in the XML view of a directory object.

```
<ad:relativeDistinguishedName>
    <ad:value xsi:type="xsd:string">CN=Test</ad:value>
</ad:relativeDistinguishedName>
```

2.3.4 Syntax Mapping

As mentioned in section 2.3.2, the content of the <ad:value> element is the value of the directory attribute (or synthetic attribute) represented as an XML value. For LDAP directory attributes, the choice of the XML syntax for this value (and thus, the corresponding textual representation of that XML value) is dependent on the attribute syntax of the LDAP directory attribute. This mapping is specified in the following table. *LDAPSYN* and *XMLSYN* refer to the variables of the same names used in section 2.3.2. The attribute syntaxes are as specified in [MS-ADTS] section 3.1.1.2.2.2.

LDAP attribute syntax	LDAPSYN	XML syntax (XMLSYN)
Boolean	Boolean	xsd:string
Enumeration	Enumeration	xsd:string
Integer	Integer	xsd:string
LargeInteger	LargeInteger	xsd:string
Object(Access-Point)	AccessPoint	xsd:string
Object(DN-String)	DNString	xsd:string
Object(OR-Name)	ORName	xsd:string
Object(DN-Binary)	DNBinary	xsd:string
Object(DS-DN)	DSDNString	xsd:string
Object(Presentation-Address)	PresentationAddress	xsd:string
Object(Replica-Link)	ReplicaLink	xsd:base64Binary
String(Case)	CaseString	xsd:string
String(IA5)	IA5String	xsd:string
String(NT-Sec-Desc)	NTSecurityDescriptor	xsd:base64Binary
String(Numeric)	NumericString	xsd:string
String(Object-Identifier)	ObjectIdentifier	xsd:string
String(Octet)	OctetString	xsd:base64Binary
String(Printable)	PrintableString	xsd:string

LDAP attribute syntax	LDAPSYN	XML syntax (XMLSYN)
String(Sid)	SidString	xsd:base64Binary
String(Teletex)	TeletexString	xsd:string
String(Unicode)	UnicodeString	xsd:string
String(UTC-Time)	UTCTimeString	xsd:string
String(Generalized-Time)	GeneralizedTimeString	xsd:string

The LDAP directory attributes located on the LDAP rootDse do not have attribute syntaxes defined for them. Mappings between an implementation's **rootDse** attributes and XML syntaxes is implementation specific. <4>

For the synthetic attributes, the choice of XML syntax is as specified in the following table.

Synthetic attribute	XML syntax (XMLSYN)
ad:objectReferenceProperty	xsd:string
ad:container-hierarchy-parent	xsd:string
ad:distinguishedName	xsd:string
ad:relativeDistinguishedName	xsd:string

2.4 (Updated Section) XPath 1.0-Derived Selection Language

Some Web Service protocols in the ADWS protocol set require the use of a selection language to specify which portion of the directory object to operate on. In other words, the selection language permits the requestor to specify that only certain attributes are to be retrieved from the directory object (rather than every attribute) or to specify that a particular attribute or attribute value is to be added, replaced, or removed from a directory object.

The ADWS protocol set uses a selection language that is derived from XPath 1.0 [XPATH] for this purpose. This selection language is applied to the XML view (described in section 2.3.2) of the directory object. A compliant implementation need only implement the subset of the language described in this section. This derived language is identified by the following URI:

http://schemas.microsoft.com/2008/1/ActiveDirectory/Dialect/XPath-Level-1

For simplicity, this language will be referred to as "XpathSelection" in the remainder of this section.

The grammar for XpathSelection is shown below in ABNF notation.

```
XpathSelection = (root elements)
root = "/"
elements = (element [additional-element] [selection-predicate])
additional-element = ("/" element)
element = QName ~~(qualified name) as defined in grammar rule [7] of [XMLNS-2ED].~~
selection-predicate = ("[" value-element "=" value "]")
```

value-element = ~~see below~~

```
(Prefix ":value ~~= dstring as~~") / "value"
```

```
value = qdstring
```

Where qdstring is defined in [RFC2252] ~~*surrounded by quotation marks*~~section 4.1 and QName and Prefix are defined in [XMLNS-2ED] section 4.

The value element is the string literal "value" qualified with an XML namespace prefix that corresponds to the XML namespace URI "http://schemas.microsoft.com/2008/1/ActiveDirectory" in the scope of the XML node in which the XpathSelection expression appears. This is illustrated in the following example.

```
<node1 xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory">
    /element1/element2[ad:value="abc"]
</node1>
```

Without the selection-predicate, an XpathSelection expression is analogous to an XPath 1.0 absolute location path with one or two location steps along the child axis. The expression "/X" selects the XML element named "X" whose parent is the root node of the XML document. The expression "/X/Y" selects the XML element named "Y" whose parent is the XML element named "X" whose parent, in turn, is the root node of the document. For example, given the following XML document:

```
<addata:user>
    <addata:description LdapType="UnicodeString">
        <ad:value xsi:type="xsd:string">
            First sample description
        </ad:value>
        <ad:value xsi:type="xsd:string">
            Second sample description
        </ad:value>
    </addata:description>
</addata:user>
```

The XpathSelection expression "/addata:user" selects the entire <addata:user> element (including child elements), while the XpathSelection expression "/addata:user/addata:description" selects the following portion.

```
<addata:description LdapType="UnicodeString">
    <ad:value xsi:type="xsd:string">
        First sample description
    </ad:value>
    <ad:value xsi:type="xsd:string">
        Second sample description
    </ad:value>
</addata:description>
```

However, unlike an XPath 1.0 expression, the comparison of the LocalPart of the QName is done in a case-insensitive manner. For example, the following XpathSelection expressions are equivalent:

/addata:user/addata:description

/addata:USER/addata:DESCRIPTION

/addata:User/addata:Description

The inclusion of a selection predicate allows an individual <ad:value> element to be specified. The predicate "[ad:value="X"]" matches the <ad:value> element whose child text node is equal to "X". The equality comparison is done using a comparison operation appropriate to the attribute syntax of the directory attribute, as specified in [MS-ADTS] section 3.1.1.2.2.4 (for example, values for

attributes that are of type String(case) in the directory are compared using a case-insensitive string comparison, while values of type Integer are compared as integers).

Using the previous XML document, the XpathSelection expression "/addata:user/addata:description[ad:value="First sample description"]" selects the following portion.

```
<ad:value xsi:type="xsd:string">
    First sample description
</ad:value>
```

2.5 Common SOAP Headers

The following sections describe SOAP headers that are defined by the ADWS protocol set. These headers, and the ADWS protocols that use them, are summarized in the following table.

SOAP 1.2 header (with namespace prefix)	Informative summary	Protocols in which header is used
ad:instance	Specifies the directory service against which the operation is to be performed.	[WXFR] [MS-WSTIM] [WSENUM]/[MS-WSDS]
ad:objectReferenceProperty	Specifies the object reference property of the directory object against which the operation is to be performed.	[WXFR] [MS-WSTIM]

2.5.1 ad:instance Header

An implementation can allow multiple directory services to be accessed via a single endpoint.<5> The ad:instance SOAP header, which is located in the http://schemas.microsoft.com/2008/1/ActiveDirectory XML namespace, is included in a SOAP request message to specify which directory service the request is intended for.

The content of the ad:instance header is the string literal "ldap:" followed by an integer (expressed as a string in base 10) that specifies the TCP port number of the desired directory service's LDAP interface.

In the following example, the requestor is asking that the operation (a WS-Transfer Get [WXFR]) that is specified in the SOAP message be performed against the directory service that listens on port 3268.

```
<soapenv:Envelope>
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
        http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </wsa:Action>
    <ad:objectReferenceProperty>
        a492d5f2-18c3-4f93-87d8-09a8c66bb5e4
    </ad:objectReferenceProperty>
    <ad:instance>ldap:3268</ad:instance>
    <wsa:MessageID>
        urn:uuid:d3cf5d97-3e9d-4c1c-b7b7-f2893685ddea
    </wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>
          http://www.w3.org/2005/08/addressing/anonymous
      </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soapenv:mustUnderstand="1">
```

```
        net.tcp://server01.fabrikam.com:9389/ActiveDirectoryWebServices/Windows/Resource
    </wsa:To>
  </soapenv:Header>
  <soapenv:Body />
</soapenv:Envelope>
```

2.5.2 ad:objectReferenceProperty Header

The ad:objectReferenceProperty SOAP header, which is located in the http://schemas.microsoft.com/2008/1/ActiveDirectory XML namespace, is attached to a SOAP request message to specify the object reference property of the directory object against which the operation specified in the SOAP message is to be performed. For example, if the SOAP message specifies a WS-Transfer Get operation [WXFR], the ad:objectReferenceProperty header specifies the directory object that is to be returned.

The content of the ad:objectReferenceProperty header is the directory object's object reference property in either GUID or distinguished name form, as specified in section 2.3.1. For example, in the following request, the requestor is asking that the operation (a WS-Transfer Get) specified in the SOAP message be performed against the object whose object reference property (specified as a GUID) is {a492d5f2-18c3-4f93-87d8-09a8c66bb5e4}. In conjunction with the ad:instance SOAP header, this uniquely identifies a single directory object located in a single directory service.

```
<soapenv:Envelope>
  <soapenv:Header>
    <wsa:Action s:mustUnderstand="1">
        http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </wsa:Action>
    <ad:objectReferenceProperty>
         a492d5f2-18c3-4f93-87d8-09a8c66bb5e4
    </ad:objectReferenceProperty>
    <ad:instance>ldap:3268</ad:instance>
    <wsa:MessageID>
        urn:uuid:d3cf5d97-3e9d-4c1c-b7b7-f2893685ddea
    </wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>
          http://www.w3.org/2005/08/addressing/anonymous
      </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soapenv:mustUnderstand="1">
        net.tcp://server01.fabrikam.com:9389/ActiveDirectoryWebServices/Windows/Resource
    </wsa:To>
  </soapenv:Header>
  <soapenv:Body />
</soapenv:Envelope>
```

The ad:objectReferenceProperty is relative to the ad:instance header specified in the request. If the ad:instance header is not specified, the directory object cannot be uniquely identified, because directory objects on different directory services could share the same GUID or distinguished name.

The ad:instance and ad:objectReferenceProperty header elements are included in the wxf:resourceCreated/wsa:ReferenceParameters element of the response to a WS-Transfer Create operation, as shown in the following example.

```
<soapenv:Envelope>
  <soapenv:Header>
  ...
  </soapenv:Header>
  <soapenv:Body>
```

```
    <wxf:ResourceCreated>
      <wsa:Address>...</wsa:Address>
      <wsa:ReferenceParameters>
        <ad:objectReferenceProperty>...</ad:objectReferenceProperty>
        <ad:instance>...</ad:Instance>
      </wsa:ReferenceParameters>
    </wxf:ResourceCreated>
  </soapenv:Body>
</soapenv:Envelope>
```

2.6 Common SOAP Fault Detail

This section defines a SOAP fault Detail element [SOAP1.2-1/2003] that is used by the ADWS protocol set. This element is used for the "[Detail]" property of [WSASB]. The SOAP fault detail is specified via the following XML schema [XMLSCHEMA1] definition.

```
<xsd:schema
    targetNamespace="http://schemas.microsoft.com/2008/1/ActiveDirectory"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"
    xmlns:da="http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess"
    elementFormDefault="qualified">
  <xsd:complexType name="ArgumentErrorType">
    <xsd:sequence>
      <xsd:element name="Message" type="xsd:string" minOccurs="0"/>
      <xsd:element name="ParameterName" type="xsd:string"
                   minOccurs="0"/>
      <xsd:element name="ShortMessage" type="xsd:string"
                   minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="DirectoryErrorType">
    <xsd:sequence>
      <xsd:element name="Message" type="xsd:string" minOccurs="0"/>
      <xsd:element name="ErrorCode" type="xsd:string" minOccurs="0"/>
      <xsd:element name="ExtendedErrorMessage" type="xsd:string"
                   minOccurs="0"/>
      <xsd:element name="MatchedDN" type="xsd:string" minOccurs="0"/>
      <xsd:element name="Referral" type="xsd:string" minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="Win32ErrorCode" type="xsd:string" minOccurs="0"/>
      <xsd:element name="ShortMessage" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="ChangeType">
    <xsd:sequence>
      <xsd:element name="AttributeType" type="da:AttributeType"/>
      <xsd:element name="AttributeValue" type="da:AttributeValue"/>
    </xsd:sequence>
    <xsd:attribute name="Operation" type="xsd:string"/>
  </xsd:complexType>

  <xsd:complexType name="InvalidAttributeTypeOrValueType">
    <xsd:sequence>
      <xsd:element name="AttributeType" type="da:AttributeType"/>
      <xsd:element name="AttributeValue" type="da:AttributeValue"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="FaultDetailType">
    <xsd:sequence>
      <xsd:element name="Error" type="xsd:string" minOccurs="0"/>
      <xsd:choice>
        <xsd:element name="ArgumentError" type="ad:ArgumentErrorType"/>
```

```
          <xsd:element name="DirectoryError" type="ad:DirectoryErrorType"/>
          <xsd:element name="InvalidAttributeType" type="xsd:string"/>
          <xsd:element name="InvalidOperation" type="xsd:string"/>
          <xsd:element name="InvalidChange" type="ad:ChangeType"/>
          <xsd:element name="InvalidAttributeTypeOrValue"
                       type="ad:InvalidAttributeTypeOrValueType"/>
        </xsd:choice>
        <xsd:element name="ShortError" type="xsd:string" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:element name="FaultDetail" type="ad:FaultDetailType"/>

  </xsd:schema>
```

In the following descriptions, XPath 1.0 [XPATH] notation is used to indicate the XML element or attribute that is being referred to.

A single SOAP fault can specify ad:FaultDetail/ad:Error, ad:FaultDetail/ad:ShortError, or both in addition to exactly one of the following:<6>

- ad:FaultDetail/ad:ArgumentError
- ad:FaultDetail/ad:DirectoryError
- ad:FaultDetail/ad:InvalidAttributeType
- ad:FaultDetail/ad:InvalidOperation
- ad:FaultDetail/ad:InvalidChange
- ad:FaultDetail/ad:InvalidAttributeTypeOrValue

The use of ad:FaultDetail/ad:ArgumentError is implementation-defined.<7>

The presence of ad:FaultDetail/ad:DirectoryError in a SOAP fault indicates that an error was returned by the directory service.

Element	Contents
ad:FaultDetail/ad:DirectoryError/ad:Message	A human-readable error message string explaining the nature of the directory error that occurred.
ad:FaultDetail/ad:DirectoryError/ad:ErrorCode	An LDAP resultCode as specified in [RFC2251].
ad:FaultDetail/ad:DirectoryError/ad:ExtendedErrorMessage	An LDAP errorMessage as specified in [RFC2251].
ad:FaultDetail/ad:DirectoryError/ad:MatchedDN	An LDAP matchedDN as specified in [RFC2251].
ad:FaultDetail/ad:DirectoryError/ad:Referral	An LDAP referral URL as specified in [RFC2251].
ad:FaultDetail/ad:DirectoryError/ad:Win32ErrorCode	An error code generated from ad:ErrorCode(*).
ad:FaultDetail/ad:DirectoryError/ad:ShortMessage	A non-localized error message string representing the nature of the directory error that occurred in ad:Message(**).

(*) The information in the following product behavior note applies to this element.<8>

(**) The information in the following product behavior note applies to this element.<9>

The ad:FaultDetail/ad:InvalidAttributeType element indicates that a [MS-WSTIM] ModifyRequest operation specified a da:ModifyRequest/da:Change/da:AttributeValue when a value was not permitted

to be specified by the setting of the da:ModifyRequest/da:Change/@Operation attribute, or did not specify a value when one was required by the setting of that attribute.

Element	Contents
ad:FaultDetail/ad:InvalidAttributeType	The value of the da:ModifyRequest/da:Change/da:AttributeType for the da:ModifyRequest/da:Change with the incorrectly specified value.

The ad:FaultDetail/ad:InvalidOperation element indicates that a [MS-WSTIM] ModifyRequest operation specified an invalid value for the da:ModifyRequest/da:Change/@Operation attribute.

Element	Contents
ad:FaultDetail/ad:InvalidOperation	The invalid value specified for the da:ModifyRequest/da:Change/@Operation attribute.

The ad:FaultDetail/ad:InvalidChange element indicates that a [MS-WSTIM] ModifyRequest specified an invalid value for the contents of a da:ModifyRequest/da:Change/da:AttributeValue.

Element	Contents
ad:FaultDetail/ad:InvalidChange/@Operation	The value of the da:ModifyRequest/da:Change/@Operation attribute for the da:ModifyRequest/da:Change with the invalid value.
ad:FaultDetail/ad:InvalidChange/da:AttributeType	The value of the da:ModifyRequest/da:Change/da:AttributeType for the da:ModifyRequest/da:Change with the invalid value.
ad:FaultDetail/ad:InvalidChange/da:AttributeValue	The value of the da:ModifyRequest/da:Change/da:AttributeValue for the da:ModifyRequest/da:Change with the invalid value.

The ad:FaultDetail/ad:InvalidAttributeTypeOrValue element indicates that a [MS-WSTIM] AddRequest specified an invalid da:AddRequest/da:AttributeTypeAndValue.

Element	Contents
ad:FaultDetail/ad:InvalidAttributeTypeOrValue/da:AttributeType	The value of the da:AddRequest/da:AttributeTypeAndValue/da:AttributeType for the invalid da:AddRequest/da:AttributeTypeAndValue.
ad:FaultDetail/ad:InvalidAttributeTypeOrValue/da:AttributeValue	The value of the da:AddRequest/da:AttributeTypeAndValue/da:AttributeValue for the invalid da:AddRequest/da:AttributeTypeAndValue.

The ad:FaultDetail/ad:Error element provides a human-readable error explaining the error. This option is used when none of the other options apply and can be used in addition to the other options. Unlike ad:FaultDetail/ad:DirectoryError/ad:Message, the contents of ad:FaultDetail/ad:Error/ad:Message need not be an error related to the directory service.

Element	Contents
ad:FaultDetail/ad:Error	A human-readable error message string explaining the nature of the error that occurred.

For example, the following demonstrates the SOAP fault detail that could be returned when the directory service returns an LDAP referral error code.

```
<soapenv:Envelope>
  <soapenv:Header>
  ....
  </soapenv:Header>
  <soapenv:Body>
    <soapenv:Fault>
      ....
      <soapenv:Detail>
        <FaultDetail
            xmlns="http://schemas.microsoft.com/2008/1/ActiveDirectory">
          <DirectoryError>
            <Message>An operation error occurred.</Message>
            <ErrorCode>10</ErrorCode>
            <ExtendedErrorMessage>
              0000202B: RefErr: DSID-03100768, data 0, 1 access points ref 1:
'server01.fabrikam.com'
            </ExtendedErrorMessage>
            <MatchedDN>
            </MatchedDN>
            <Referral>
              ldap://server01.fabrikam.com/CN=Test,DC=fabrikam,DC=com
            </Referral>
            <ShortMessage>ELdap</ShortMessage>
            <Win32ErrorCode>8235</Win32ErrorCode>
          </DirectoryError>
        </FaultDetail>
      </soapenv:Detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

2.7 Range Retrieval

Retrieving the contents of a multivalued attribute from a group such as a distribution list can often result in a large number of returned values. A directory service can place limits on the maximum number of attribute values that can be retrieved in a single query.<10> If an attribute has more values than can be returned by the server in a single call, the only way to enumerate all of the attribute values is through the use of the range option.

Range retrieval involves requesting a limited number of attribute values in a single query. The number of values requested must be less than or equal to the maximum number of values supported by the server. To reduce the number of times the query must contact the server, the number of values requested should be as close to this maximum as possible.

To support range retrieval, the WS-Transfer and WS-Enumeration Web Service protocols in the ADWS protocol set require defining an XML representation to return portions of a multivalued attribute or to specify which portion of the attribute to retrieve. The following sections provide an extension to the XML serialization of the data model defined in section 2.3.2 that specifies an XML representation of an attribute with only a portion of its values. They also define extensions to the WS-Transfer [WXFR] and WS-Enumeration [WSENUM] protocols that indicate how a requester is to specify the portion of the attribute values to be returned.

2.7.1 XML View of Multivalued Attribute with Range Option

Section 2.3.2 describes the XML view of a directory object and its attributes as presented by ADWS. This section defines extensions to such an XML view for a multivalued attribute in which only a subset of the values are represented in the XML. This subset is referred to as a range of values. This range of

values is represented by XML attributes RangeLow and RangeHigh. For example, suppose that an attribute contains 5,000 values. The XML view might contain only the first 1,000 values, in which case RangeLow and RangeHigh would be 0 and 999, respectively.

The following description defines how a multivalued LDAP attribute and a portion of its value(s) limited by a range are represented in the XML view. Let *B* be the LDAP display name of the multivalued attribute that contains the complete set of values *V1(B)…Vn(B)*. Let *RANGELOW(B)* and *RANGEHIGH(B)* be the respective lower and higher range of values returned by the server for the multivalued attribute. Let $V_{RANGELOW(B)}$ and $V_{RANGEHIGH(B)}$ be the returned values lying between *RANGELOW(B)* and *RANGEHIGH(B)* of values *V1(B)* and *Vn(B)*. Let $S_{RANGELOW(B)}(B)…S_{RANGEHIGH(B)}(B)$ be the XML representation of values $V_{RANGELOW(B)}…V_{RANGEHIGH(B)}$ as described in section 2.3.4. Let *LDAPSYN(B)* be the LDAP attribute syntax of attribute *B* and let *XMLSYN(B)* be the corresponding XML syntax, as described in section 2.3.4. The XML representation for this multivalued attribute with range option is the following.

```
<addata:B RangeLow="RANGELOW(B)" RangeHigh="RANGEHIGH(B)" LdapSyntax="LDAPSYN(B)">
    <ad:value xsi:type="XMLSYN(B)">
        S(RANGELOW(B))(B)
    </ad:value>
    ...
    ...
    <ad:value xsi:type="XMLSYN(B)">
        S(RANGEHIGH(B))(B)
    </ad:value>
</addata:B>
```

If *O* is the directory object, *C* being the LDAP display name of the most specific structural object class ([MS-ADTS] section 3.1.1.1.4) and *B* being one its multivalued attributes, then the following representation of *O* as the XML view in the data model described in section 2.3.2 remains unchanged except for the multivalued attribute XML representation comprised of range attributes.

```
<addata:C>
    ...
    ...
    <addata:B RangeLow="RANGELOW(B)" RangeHigh="RANGEHIGH(B)" LdapSyntax="LDAPSYN(B)">
        <ad:value xsi:type="XMLSYN(B)">
            S(RANGELOW(B))(B)
        </ad:value>
        ...
        ...
        <ad:value xsi:type="XMLSYN(B)">
            S(RANGEHIGH(B))(B)
        </ad:value>
    </addata:B>
    ...
    ...
</addata:C>
```

For each multivalued LDAP attribute for which the server is including only a portion of the values contained in that attribute, both the RangeLow and RangeHigh XML attributes are returned.

ADWS specification of the possible values of these XML attributes, which are returned in the response as part of the XML view of the object for a request with range specification, is illustrated in section 2.7.2.

2.7.2 Range Specifiers for Requests

The range option for an attribute query is represented using the following XML attributes in the request:

RangeLow="*RANGELOW*" RangeHigh="*RANGEHIGH*"

where *RANGELOW* is the zero-based index of the first attribute value to retrieve, and *RANGEHIGH* is the zero-based index of the last attribute value to retrieve.

When querying for an attribute, a request can specify a RangeLow XML attribute in addition to a RangeHigh XML attribute to retrieve values between the lower and higher range, inclusively. A SOAP request to retrieve multivalued attributes containing a RangeHigh XML attribute must also contain a RangeLow XML attribute. A SOAP request to retrieve multivalued attributes not containing a RangeHigh attribute specifies a request to retrieve all the values beyond *RANGELOW* (this is subject to the limit imposed by the server on the maximum number of values that can be returned).

In search queries and results, zero is used for *RANGELOW* to specify the first entry and the wildcard character (*) is used for *RANGEHIGH* to specify all remaining entries. If specified, *RANGELOW* MUST be of type positive integer. If specified, *RANGEHIGH* MUST be either a positive integer or the wildcard character (*).

Both RangeLow and RangeHigh can be absent if the range retrieval extensions as illustrated in 2.7.2.1 and 2.7.2.2 are not used. Both RangeLow and RangeHigh attributes being absent from a request specifies a request for all values to be returned in that single call. For example, if a distribution list contains 1,000 member values, and if this number is less than the directory service-imposed limit on the maximum values that can be retrieved in a single query (1,500 for instance), all 1,000 values must be returned.

If the list of values is larger than the maximum limit of values the server can return, for example 2,000 member values, the first response contains the member attribute with RangeLow and RangeHigh XML attributes specifying the lower and higher range, respectively, and containing all the member values in this range.

To retrieve the next group of member values in the previous example, the search query can be repeated with a range specification that begins at the attribute number that is one past the RANGEHIGH value that was returned in the previous call. This process can be repeated until the last group of values is retrieved. In the above example, the first call would return member values in the range RangeLow as "0" and RangeHigh as "1,499". To retrieve the remaining values, the search query would request member values with RangeLow = 1,500 and RangeHigh = *, and would be given member values in the range RangeLow as "1,500" and RangeHigh as "2000".

The ADWS protocol set shares these range specifiers as common XML attributes that are used to extend WS-Transfer [WXFR] Get and WS-Enumeration [WSENUM] Enumerate requests, which are described in subsequent sections.

The following table lists examples of how to implement range specifiers.

Example	Meaning
RangeLow="0" RangeHigh="*"	Retrieve all attribute values.*
RangeLow="0" RangeHigh="500"	Retrieve the 1st to 501st values, inclusive.
RangeLow="2" RangeHigh="3"	Retrieve the 3rd and 4th values.
RangeLow="501" RangeHigh="*"	Retrieve the 502nd and all remaining values.*

(*) This is subject to the limits imposed by the server.<11>

The following sections illustrate extensions to the WS-Transfer [WXFR] and WS-Enumeration [WSENUM] protocols by specifying how requestors could retrieve only a portion of the attribute values through an Enumerate or Get request using the range specifiers defined previously.

2.7.2.1 WS-Transfer Range Retrieval Extensions

This section illustrates a range retrieval extension to the Get operation of the WS-Transfer [WXFR] protocol, which, when used with [MS-WSTIM] extensions, provides a way to retrieve portions of a multivalued attribute of a specific directory object.

In this example, the following is the XML representation of the da:AttributeType XML element defined in [MS-WSTIM] section 2.2.3.1, with range specifiers for retrieving only portions of the multivalued attribute.

```
<da:AttributeType RangeLow="0" RangeHigh="*">
  addata:member
</da:AttributeType>
```

The updated XML schema definition for the da:AttributeType element relative to the schema definition defined in [MS-WSTIM] would be similar to the following.

```
<xsd:element name="AttributeType">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="ExtensibleType">
        <xsd:attribute name="RangeLow" use="required" type="xsd:nonNegativeInteger"/>
        <xsd:attribute name="RangeHigh" use="optional" type="xsd:string"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="ExtensibleType">
  <xsd:complexContent mixed="true">
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:any processContents="lax"
            minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

2.7.2.2 WS-Enumeration Range Retrieval Extensions

This section illustrates a range retrieval extension to the Enumerate operation of the WS-Enumerate [WSENUM] protocol, which, when used with [MS-WSDS] extensions, provides a way to retrieve portions of a multivalued attribute of selected directory objects during a pull operation.

In this example, the following is the XML representation of the ad:SelectionProperty XML element defined in [MS-WSDS] section 3.1.4.1.1.2.1, with range specifiers for retrieving only portions of the multivalued attribute.

```
<ad:SelectionProperty RangeLow="0" RangeHigh="*">
  addata:member
</ad:SelectionProperty>
```

The updated XML schema definition for the ad:SelectionProperty element relative to the schema defined in [MS-WSDS] would be similar to the following.

```
<xsd:element name="SelectionProperty">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="RangeLow" use="required" type="xsd:nonNegativeInteger"/>
        <xsd:attribute name="RangeHigh" use="optional" type="xsd:string"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

3 Structure Examples

This section contains examples of the XML view of sample directory objects, including the ADWS synthetic attributes. For illustrative purposes, these examples are shown in the context of protocols in the ADWS protocol set.

3.1 WS-Transfer 'Get' Example

The following example shows a WS-Transfer Get [WXFR] operation. Both the SOAP request message and the SOAP response message are shown. This example retrieves the complete XML view of a directory object. In this example, the most specific structural object class of the directory object is user. The object has an LDAP distinguished name of "CN=TestUser1,DC=fabrikam,DC=com". The GUID for its object reference property is {1e0f3427-bbcb-474d-a532-a2ba6168c4dc}, and its parent object has a object reference property whose GUID is {e4f8a504-d7df-4b63-a636-5642d3bf1cf6}.

SOAP request message:

```
<soapenv:Envelope
       xmlns:wsa="http://www.w3.org/2005/08/addressing"
       xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
        http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </wsa:Action>
    <objectReferenceProperty
       xmlns="http://schemas.microsoft.com/2008/1/ActiveDirectory">
        1e0f3427-bbcb-474d-a532-a2ba6168c4dc
    </objectReferenceProperty>
    <instance xmlns="http://schemas.microsoft.com/2008/1/ActiveDirectory">
        ldap:389
    </instance>
    <wsa:MessageID>
        urn:uuid:720f1d9c-5181-42c8-91ab-3deef105d0ff
    </wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>
          http://www.w3.org/2005/08/addressing/anonymous
      </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soapenv:mustUnderstand="1">
        net.tcp://server01.fabrikam.com:9389/ActiveDirectoryWebServices/Windows/Resource
    </wsa:To>
  </soapenv:Header>
  <soapenv:Body/>
</soapenv:Envelope>
```

SOAP response message:

```
<soapenv:Envelope
       xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
       xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
      http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
    </wsa:Action>
    <wsa:RelatesTo>
      urn:uuid:720f1d9c-5181-42c8-91ab-3deef105d0ff
    </wsa:RelatesTo>
    <wsa:To soapenv:mustUnderstand="1">
      http://www.w3.org/2005/08/addressing/anonymous
    </wsa:To>
  </soapenv:Header>
  <soapenv:Body>
```

```
<addata:user
    xmlns:addata="http://schemas.microsoft.com/2008/1/ActiveDirectory/Data"
    xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ad:objectReferenceProperty>
    <ad:value xsi:type="xsd:string">
        1e0f3427-bbcb-474d-a532-a2ba6168c4dc
    </ad:value>
  </ad:objectReferenceProperty>
  <addata:lastLogon LdapSyntax="LargeInteger">
    <ad:value xsi:type="xsd:string">0</ad:value>
  </addata:lastLogon>
  <addata:dSCorePropagationData LdapSyntax="GeneralizedTimeString">
    <ad:value xsi:type="xsd:string">16010101000000.0Z</ad:value>
  </addata:dSCorePropagationData>
  <addata:objectSid LdapSyntax="SidString">
    <ad:value xsi:type="xsd:base64Binary">
        AQUAAAAAAAUVAAAAbTIi8R3L2V3ypAE4plMAAA==
    </ad:value>
  </addata:objectSid>
  <addata:whenCreated LdapSyntax="GeneralizedTimeString">
    <ad:value xsi:type="xsd:string">20080722202149.0Z</ad:value>
  </addata:whenCreated>
  <addata:badPasswordTime LdapSyntax="LargeInteger">
    <ad:value xsi:type="xsd:string">0</ad:value>
  </addata:badPasswordTime>
  <addata:accountExpires LdapSyntax="LargeInteger">
    <ad:value xsi:type="xsd:string">9223372036854775807</ad:value>
  </addata:accountExpires>
  <addata:name LdapSyntax="UnicodeString">
    <ad:value xsi:type="xsd:string">TestUser1</ad:value>
  </addata:name>
  <addata:uSNChanged LdapSyntax="LargeInteger">
    <ad:value xsi:type="xsd:string">166235</ad:value>
  </addata:uSNChanged>
  <addata:objectCategory LdapSyntax="DSDNString">
    <ad:value xsi:type="xsd:string">
        CN=Person,CN=Schema,CN=Configuration,DC=Fabrikam,DC=com
    </ad:value>
  </addata:objectCategory>
  <addata:sAMAccountType LdapSyntax="Integer">
    <ad:value xsi:type="xsd:string">805306368</ad:value>
  </addata:sAMAccountType>
  <addata:codePage LdapSyntax="Integer">
    <ad:value xsi:type="xsd:string">0</ad:value>
  </addata:codePage>
  <addata:instanceType LdapSyntax="Integer">
    <ad:value xsi:type="xsd:string">4</ad:value>
  </addata:instanceType>
  <addata:countryCode LdapSyntax="Integer">
    <ad:value xsi:type="xsd:string">0</ad:value>
  </addata:countryCode>
  <addata:distinguishedName LdapSyntax="DSDNString">
    <ad:value xsi:type="xsd:string">
        CN=TestUser1,DC=Fabrikam,DC=com
    </ad:value>
  </addata:distinguishedName>
  <addata:cn LdapSyntax="UnicodeString">
    <ad:value xsi:type="xsd:string">TestUser1</ad:value>
  </addata:cn>
  <addata:objectClass LdapSyntax="ObjectIdentifier">
    <ad:value xsi:type="xsd:string">top</ad:value>
    <ad:value xsi:type="xsd:string">person</ad:value>
    <ad:value xsi:type="xsd:string">organizationalPerson</ad:value>
    <ad:value xsi:type="xsd:string">user</ad:value>
  </addata:objectClass>
  <addata:logonCount LdapSyntax="Integer">
    <ad:value xsi:type="xsd:string">0</ad:value>
  </addata:logonCount>
```

```
      <addata:uSNCreated LdapSyntax="LargeInteger">
        <ad:value xsi:type="xsd:string">166234</ad:value>
      </addata:uSNCreated>
      <addata:userAccountControl LdapSyntax="Integer">
        <ad:value xsi:type="xsd:string">546</ad:value>
      </addata:userAccountControl>
      <addata:objectGUID LdapSyntax="OctetString">
        <ad:value xsi:type="xsd:base64Binary">
          JzQPHsu7TUelMqK6YWjE3A==
        </ad:value>
      </addata:objectGUID>
      <addata:primaryGroupID LdapSyntax="Integer">
        <ad:value xsi:type="xsd:string">513</ad:value>
      </addata:primaryGroupID>
      <addata:lastLogoff LdapSyntax="LargeInteger">
        <ad:value xsi:type="xsd:string">0</ad:value>
      </addata:lastLogoff>
      <addata:sAMAccountName LdapSyntax="UnicodeString">
        <ad:value xsi:type="xsd:string">testusr1</ad:value>
      </addata:sAMAccountName>
      <addata:badPwdCount LdapSyntax="Integer">
        <ad:value xsi:type="xsd:string">0</ad:value>
      </addata:badPwdCount>
      <addata:whenChanged LdapSyntax="GeneralizedTimeString">
        <ad:value xsi:type="xsd:string">20080722202149.0Z</ad:value>
      </addata:whenChanged>
      <addata:pwdLastSet LdapSyntax="LargeInteger">
        <ad:value xsi:type="xsd:string">0</ad:value>
      </addata:pwdLastSet>
      <ad:container-hierarchy-parent>
        <ad:value xsi:type="xsd:string">
            e4f8a504-d7df-4b63-a636-5642d3bf1cf6
        </ad:value>
      </ad:container-hierarchy-parent>
      <ad:relativeDistinguishedName>
        <ad:value xsi:type="xsd:string">CN=TestUser1</ad:value>
      </ad:relativeDistinguishedName>
      <ad:distinguishedName>
        <ad:value xsi:type="xsd:string">
            CN=TestUser1,DC=Fabrikam,DC=com
        </ad:value>
      </ad:distinguishedName>
    </addata:user>
  </soapenv:Body>
</soapenv:Envelope>
```

3.2 WS-Transfer Identity Management Extension 'ModifyRequest' Example

This example demonstrates a [MS-WSTIM] ModifyRequest operation. Both the SOAP request message and the SOAP response message are shown. In the SOAP request message, the requestor is specifying that the LDAP directory attribute whose LDAP display name is "description" is to have its value replaced with the new value "Modified description attribute". The requestor is also asking that the values "(212) 555-0100" and "(516) 555-0100" be appended to the set of existing values (if any) in the LDAP directory attribute whose LDAP display name is "otherTelephone". The directory object on which this operation is being performed is identified by its GUID-valued object reference property {cf041608-84b9-4fd0-a83c-46d40a964b88}.

SOAP request message:

```
<soapenv:Envelope
    xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
        http://schemas.xmlsoap.org/ws/2004/09/transfer/Put
```

```
    </wsa:Action>
    <IdentityManagementOperation
        xmlns="http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess"/>
    <objectReferenceProperty
       xmlns="http://schemas.microsoft.com/2008/1/ActiveDirectory">
        cf041608-84b9-4fd0-a83c-46d40a964b88
    </objectReferenceProperty>
    <instance xmlns="http://schemas.microsoft.com/2008/1/ActiveDirectory">
        ldap:389
    </instance>
    <wsa:MessageID>
        urn:uuid:e36457ff-d0f1-4c85-abe6-6cdf4bd511e9
    </wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>
          http://www.w3.org/2005/08/addressing/anonymous
      </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soapenv:mustUnderstand="1">
        net.tcp://server01.fabrikam.com:9389/ActiveDirectoryWebServices/Windows/Resource
    </wsa:To>
  </soapenv:Header>
  <soapenv:Body>
    <da:ModifyRequest
        Dialect="http://schemas.microsoft.com/2008/1/ActiveDirectory/Dialect/XPath-Level-1"
        xmlns:da="http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess"
        xmlns:addata="http://schemas.microsoft.com/2008/1/ActiveDirectory/Data"
        xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <da:Change Operation="replace">
        <da:AttributeType>addata:description</da:AttributeType>
        <da:AttributeValue>
          <ad:value xsi:type="xsd:string">
              Modified description attribute
          </ad:value>
        </da:AttributeValue>
      </da:Change>
      <da:Change Operation="add">
        <da:AttributeType>addata:otherTelephone</da:AttributeType>
        <da:AttributeValue>
          <ad:value xsi:type="xsd:string">(212) 555-0100</ad:value>
          <ad:value xsi:type="xsd:string">(516) 555-0100</ad:value>
        </da:AttributeValue>
      </da:Change>
    </da:ModifyRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP response message:

```
<soapenv:Envelope
    mlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <soapenv:Header>
        <wsa:Action soapenv:mustUnderstand="1">
            http://schemas.xmlsoap.org/ws/2004/09/transfer/PutResponse
        </wsa:Action>
        <wsa:RelatesTo>
            urn:uuid:e36457ff-d0f1-4c85-abe6-6cdf4bd511e9
        </wsa:RelatesTo>
        <wsa:To soapenv:mustUnderstand="1">
            http://www.w3.org/2005/08/addressing/anonymous
        </wsa:To>
    </soapenv:Header>
    <soapenv:Body/>
</soapenv:Envelope>
```

3.3 WS-Enumeration 'Pull' Example

This example demonstrates a WS-Enumeration Pull operation [WSENUM] using a previously obtained enumeration context. Both the SOAP request message and the SOAP response message are shown. In the response message, two directory objects are returned. Both objects have the same parent directory object, as evidenced by the fact that both have the same value for their ad:container-hierarchy-parent synthetic attribute. In this example, the WS-Enumeration Enumerate operation that began the search requested three attributes to be returned: the LDAP directory attribute addata:givenName and the synthetic attributes ad:container-hierarchy-parent and ad:relativeDistinguishedName. The ad:objectReferenceProperty synthetic attribute is automatically included in the response by the server [MS-WSDS].

SOAP request message:

```
<soapenv:Envelope
    xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull
    </wsa:Action>
    <wsa:MessageID>
      urn:uuid:b22747a9-ca15-41de-8c91-5a51bd88669c
    </wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>
        http://www.w3.org/2005/08/addressing/anonymous
      </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soapenv:mustUnderstand="1">
        net.tcp://server01.fabrikam.com:9389/ActiveDirectoryWebServices/Windows/Enumeration
    </wsa:To>
  </soapenv:Header>
  <soapenv:Body>
    <wsen:Pull
        xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory">
      <wsen:EnumerationContext>
        f52c7e9d-80c2-40cd-b8c9-55bc94fc3e47
      </wsen:EnumerationContext>
      <wsen:MaxTime>PT10S</wsen:MaxTime>
      <wsen:MaxElements>2</wsen:MaxElements>
    </wsen:Pull>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP response message:

```
<soapenv:Envelope
    xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
      http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse
    </wsa:Action>
    <wsa:RelatesTo>
      urn:uuid:b22747a9-ca15-41de-8c91-5a51bd88669c
    </wsa:RelatesTo>
    <wsa:To soapenv:mustUnderstand="1">
      http://www.w3.org/2005/08/addressing/anonymous
    </wsa:To>
  </soapenv:Header>
```

```
  <soapenv:Body>
    <wsen:PullResponse
        xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"
        xmlns:addata="http://schemas.microsoft.com/2008/1/ActiveDirectory/Data">
      <wsen:EnumerationContext>
        d22e957c-8278-4eb9-a57f-41574c55305d
      </wsen:EnumerationContext>
      <wsen:Items>
        <addata:user>
          <ad:objectReferenceProperty>
            <ad:value xsi:type="xsd:string">
              373e1409-cf88-41dc-b8ea-bdd27d54e073
            </ad:value>
          </ad:objectReferenceProperty>
          <ad:container-hierarchy-parent>
            <ad:value xsi:type="xsd:string">
              41816238-95ca-48d9-9a99-3bd9ae9e0e42
            </ad:value>
          </ad:container-hierarchy-parent>
          <ad:relativeDistinguishedName>
            <ad:value xsi:type="xsd:string">CN=TestUser1</ad:value>
          </ad:relativeDistinguishedName>
          <addata:givenName LdapSyntax="UnicodeString">
            <ad:value xsi:type="xsd:string">John</ad:value>
          </addata:givenName>
        </addata:user>
        <addata:user>
          <ad:objectReferenceProperty>
            <ad:value xsi:type="xsd:string">
              51d67624-d52d-421d-a0d6-1dc350abd009
            </ad:value>
          </ad:objectReferenceProperty>
          <ad:container-hierarchy-parent>
            <ad:value xsi:type="xsd:string">
              41816238-95ca-48d9-9a99-3bd9ae9e0e42
            </ad:value>
          </ad:container-hierarchy-parent>
          <ad:relativeDistinguishedName>
            <ad:value xsi:type="xsd:string">CN=TestUser2</ad:value>
          </ad:relativeDistinguishedName>
          <addata:givenName LdapSyntax="UnicodeString">
            <ad:value xsi:type="xsd:string">Robert</ad:value>
          </addata:givenName>
        </addata:user>
      </wsen:Items>
    </wsen:PullResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

4 Security

4.1 Security Considerations for Implementers

None.

4.2 Index of Security Fields

None.

5 Appendix A: Product Behavior

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

The terms "earlier" and "later", when used with a product version, refer to either all preceding versions or all subsequent versions, respectively. The term "through" refers to the inclusive range of versions. Applicable Microsoft products are listed chronologically in this section.

- Windows Server 2008 R2 operating system
- Windows Server 2012 operating system
- Windows Server 2012 R2 operating system
- Windows Server 2016 operating system
- Windows Server operating system
- Windows Server 2019 operating system

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: The following products are applicable to Active Directory Web Services: Data Model and Common Elements:

- Active Directory Management Gateway Service contains the server implementation of the ADWS set of protocols that use Active Directory Web Services: Data Model and Common Elements.
- Remote Server Administration Tools (excluding Remote Server Administration Tools for Windows Vista operating system) contains the client implementation. For more information about Remote Server Administration Tools, see [MSFT-RSAT].
- Windows Server 2008 R2 and later contain both the server and the client implementations.

Active Directory Management Gateway Service is available for Windows Server 2003 operating system with Service Pack 2 (SP2), Windows Server 2003 R2 operating system with Service Pack 2 (SP2), and Windows Server 2008 operating system.

<2> Section 2.3.3.1: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements do not return the ad:objectReferenceProperty synthetic attribute if the requestor does not have permission to read *O*!objectGUID, where *O* is the directory object being represented as an XML view.

<3> Section 2.3.3.2: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements in Active Directory Management Gateway Service for Windows Server 2003 operating system omits this attribute from the XML view of all directory objects, regardless of whether the directory object has a parent.

<4> Section 2.3.4: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements use the following mapping between **rootDse** attributes (specified by their LDAP display names) and XML syntaxes.

rootDse attribute name	LDAPSYN	XML syntax (XMLSYN)
configurationnamingcontext	DSDNString	xsd:string
Currenttime	GeneralizedTimeString	xsd:string
defaultnamingcontext	DSDNString	xsd:string
Dnshostname	UnicodeString	xsd:string
Dsschemaattrcount	Integer	xsd:string
Dsschemaclasscount	Integer	xsd:string
dsschemaprefixcount	Integer	xsd:string
Dsservicename	DSDNString	xsd:string
highestcommittedusn	LargeInteger	xsd:string
Isglobalcatalogready	Boolean	xsd:string
Issynchronized	Boolean	xsd:string
Ldapservicename	UnicodeString	xsd:string
Namingcontexts	DSDNString	xsd:string
pendingpropagations	DSDNString	xsd:string
rootdomainnamingcontext	DSDNString	xsd:string
schemanamingcontext	DSDNString	xsd:string
Servername	DSDNString	xsd:string
Subschemasubentry	DSDNString	xsd:string
supportedcapabilities	ObjectIdentifier	xsd:string
Supportedcontrol	ObjectIdentifier	xsd:string
supportedldappolicies	UnicodeString	xsd:string
supportedldapversion	Integer	xsd:string
supportedsaslmechanisms	UnicodeString	xsd:string
domaincontrollerfunctionality	Integer	xsd:string
Domainfunctionality	Integer	xsd:string
Forestfunctionality	Integer	xsd:string
msds-replallinboundneighbors	UnicodeString	xsd:string
msds-replalloutboundneighbors	UnicodeString	xsd:string
msds-replconnectionfailures	UnicodeString	xsd:string
msds-repllinkfailures	UnicodeString	xsd:string
msds-replpendingops	UnicodeString	xsd:string
msds-replqueuestatistics	UnicodeString	xsd:string

rootDse attribute name	LDAPSYN	XML syntax (XMLSYN)
msds-topquotausage	UnicodeString	xsd:string
supportedconfigurablesettings	UnicodeString	xsd:string
Supportedextension	ObjectIdentifier	xsd:string
Validfsmos	DSDNString	xsd:string
Dsaversionstring	UnicodeString	xsd:string
msds-portldap	Integer	xsd:string
msds-portssl	Integer	xsd:string
msds-principalname	UnicodeString	xsd:string
Serviceaccountinfo	UnicodeString	xsd:string
Spnregistrationresult	Integer	xsd:string
Tokengroups	SidString	xsd:base64Binary
Usnatrifm	LargeInteger	xsd:string
becomePdcWithCheckPoint	SidString	xsd:base64Binary
checkPhantoms	UnicodeString	xsd:string
doGarbageCollection	Integer	xsd:string
dumpDatabase	UnicodeString	xsd:string
fixupInheritance	UnicodeString	xsd:string
invalidateRidPool	SidString	xsd:base64Binary
recalcHierarchy	UnicodeString	xsd:string
schemaUpdateNow	UnicodeString	xsd:string
removeLingeringObject	UnicodeString	xsd:string
doLinkCleanup	UnicodeString	xsd:string
doOnlineDefrag	Integer	xsd:string
replicateSingleObject	UnicodeString	xsd:string
updateCachedMemberships	UnicodeString	xsd:string
doGarbageCollectionPhantomsNow	Integer	xsd:string
invalidateGCConnection	UnicodeString	xsd:string
renewServerCertificate	UnicodeString	xsd:string
rODCPurgeAccount	UnicodeString	xsd:string
sqmRunOnce	UnicodeString	xsd:string
runProtectAdminGroupsTask	UnicodeString	xsd:string

<5> Section 2.5.1: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements provide access to any Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS) directory service that is running on the same computer as ADWS. AD DS can be accessed via "ldap:389". If the machine is also an AD DS global catalog, then the global catalog can be accessed as "ldap:3268". An AD LDS instance can be accessed as "ldap:N", where N is the LDAP port number that the AD LDS instance has been configured to use.

<6> Section 2.6: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements include both ad:FaultDetail/ad:Error and ad:FaultDetail/ad:ShortError elements.

<7> Section 2.6: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements use ad:FaultDetail/ad:ArgumentError to indicate that an invalid argument was passed from one internal function to another.

Element	Contents
ad:FaultDetail/ad:ArgumentError/ad:Message	A human-readable error message string explaining the nature of the argument error that occurred.
ad:FaultDetail/ad:ArgumentError/ad:ParameterName	The name of the function parameter whose argument was invalid.

<8> Section 2.6: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements translate LDAP error codes to Win32 error codes ([MS-ERREF] section 2.2) using the following table.

LDAP Error		Win32 Error	
Code (Hex)	Name	Code (Dec)	Name
0x00	LDAP_SUCCESS	0	NO_ERROR
0x01	LDAP_OPERATIONS_ERROR	8224	ERROR_DS_OPERATIONS_ERROR
0x02	LDAP_PROTOCOL_ERROR	8225	ERROR_DS_PROTOCOL_ERROR
0x03	LDAP_TIMELIMIT_EXCEEDED	8226	ERROR_DS_TIMELIMIT_EXCEEDED
0x04	LDAP_SIZELIMIT_EXCEEDED	8227	ERROR_DS_SIZELIMIT_EXCEEDED
0x05	LDAP_COMPARE_FALSE	8229	ERROR_DS_COMPARE_FALSE
0x06	LDAP_COMPARE_TRUE	8230	ERROR_DS_COMPARE_TRUE
0x07	LDAP_AUTH_METHOD_NOT_SUPPORTED	8231	ERROR_DS_AUTH_METHOD_NOT_SUPPORTED
0x08	LDAP_STRONG_AUTH_REQUIRED	8232	ERROR_DS_STRONG_AUTH_REQUIRED
0x09	LDAP_PARTIAL_RESULTS	299	ERROR_PARTIAL_COPY
0x0a	LDAP_REFERRAL	8235	ERROR_DS_REFERRAL
0x0b	LDAP_ADMIN_LIMIT_EXCEEDED	8228	ERROR_DS_ADMIN_LIMIT_EXCEEDED
0x0c	LDAP_UNAVAILABLE_CRIT_EXTENSION	8236	ERROR_DS_UNAVAILABLE_CRIT_EXTENSION
0x0d	LDAP_CONFIDENTIALITY_REQUIRED	8237	ERROR_DS_CONFIDENTIALITY_REQUIRED
0x0e	LDAP_SASL_BIND_IN_PROGRESS	590610	SEC_I_CONTINUE_NEEDED

LDAP Error		Win32 Error	
0x10	LDAP_NO_SUCH_ATTRIBUTE	8202	ERROR_DS_NO_ATTRIBUTE_OR_VALUE
0x11	LDAP_UNDEFINED_TYPE	8204	ERROR_DS_ATTRIBUTE_TYPE_UNDEFINED
0x12	LDAP_INAPPROPRIATE_MATCHING	8238	ERROR_DS_INAPPROPRIATE_MATCHING
0x13	LDAP_CONSTRAINT_VIOLATION	8239	ERROR_DS_CONSTRAINT_VIOLATION
0x14	LDAP_ATTRIBUTE_OR_VALUE_EXISTS	8205	ERROR_DS_ATTRIBUTE_OR_VALUE_EXISTS
0x15	LDAP_INVALID_SYNTAX	8203	ERROR_DS_INVALID_ATTRIBUTE_SYNTAX
0x20	LDAP_NO_SUCH_OBJECT	8240	ERROR_DS_NO_SUCH_OBJECT
0x21	LDAP_ALIAS_PROBLEM	8241	ERROR_DS_ALIAS_PROBLEM
0x22	LDAP_INVALID_DN_SYNTAX	8242	ERROR_DS_INVALID_DN_SYNTAX
0x23	LDAP_IS_LEAF	8243	ERROR_DS_IS_LEAF
0x24	LDAP_ALIAS_DEREF_PROBLEM	8244	ERROR_DS_ALIAS_DEREF_PROBLEM
0x30	LDAP_INAPPROPRIATE_AUTH	8233	ERROR_DS_INAPPROPRIATE_AUTH
0x31	LDAP_INVALID_CREDENTIALS	1326	ERROR_LOGON_FAILURE
0x32	LDAP_INSUFFICIENT_RIGHTS	5	ERROR_ACCESS_DENIED
0x33	LDAP_BUSY	8206	ERROR_DS_BUSY
0x34	LDAP_UNAVAILABLE	8207	ERROR_DS_UNAVAILABLE
0x35	LDAP_UNWILLING_TO_PERFORM	8245	ERROR_DS_UNWILLING_TO_PERFORM
0x36	LDAP_LOOP_DETECT	8246	ERROR_DS_LOOP_DETECT
0x3C	LDAP_SORT_CONTROL_MISSING	8261	ERROR_DS_SORT_CONTROL_MISSING
0x3D	LDAP_OFFSET_RANGE_ERROR	8262	ERROR_DS_OFFSET_RANGE_ERROR
0x40	LDAP_NAMING_VIOLATION	8247	ERROR_DS_NAMING_VIOLATION
0x41	LDAP_OBJECT_CLASS_VIOLATION	8212	ERROR_DS_OBJ_CLASS_VIOLATION
0x42	LDAP_NOT_ALLOWED_ON_NONLEAF	8213	ERROR_DS_CANT_ON_NON_LEAF
0x43	LDAP_NOT_ALLOWED_ON_RDN	8214	ERROR_DS_CANT_ON_RDN
0x44	LDAP_ALREADY_EXISTS	5010	ERROR_OBJECT_ALREADY_EXISTS
0x45	LDAP_NO_OBJECT_CLASS_MODS	8215	ERROR_DS_CANT_MOD_OBJ_CLASS
0x46	LDAP_RESULTS_TOO_LARGE	8248	ERROR_DS_OBJECT_RESULTS_TOO_LARGE
0x47	LDAP_AFFECTS_MULTIPLE_DSAS	8249	ERROR_DS_AFFECTS_MULTIPLE_DSAS
0x4c	LDAP_VIRTUAL_LIST_VIEW_ERROR	8341	ERROR_DS_GENERIC_ERROR
0x50	LDAP_OTHER	31	ERROR_GEN_FAILURE
0x51	LDAP_SERVER_DOWN	8250	ERROR_DS_SERVER_DOWN
0x52	LDAP_LOCAL_ERROR	8251	ERROR_DS_LOCAL_ERROR

LDAP Error		Win32 Error	
0x53	LDAP_ENCODING_ERROR	8252	ERROR_DS_ENCODING_ERROR
0x54	LDAP_DECODING_ERROR	8253	ERROR_DS_DECODING_ERROR
0x55	LDAP_TIMEOUT	1460	ERROR_TIMEOUT
0x56	LDAP_AUTH_UNKNOWN	8234	ERROR_DS_AUTH_UNKNOWN
0x57	LDAP_FILTER_ERROR	8254	ERROR_DS_FILTER_UNKNOWN
0x58	LDAP_USER_CANCELLED	1223	ERROR_CANCELLED
0x59	LDAP_PARAM_ERROR	8255	ERROR_DS_PARAM_ERROR
0x5a	LDAP_NO_MEMORY	8	ERROR_NOT_ENOUGH_MEMORY
0x5b	LDAP_CONNECT_ERROR	1225	ERROR_CONNECTION_REFUSED
0x5c	LDAP_NOT_SUPPORTED	8256	ERROR_DS_NOT_SUPPORTED
0x5e	LDAP_NO_RESULTS_RETURNED	8257	ERROR_DS_NO_RESULTS_RETURNED
0x5d	LDAP_CONTROL_NOT_FOUND	8258	ERROR_DS_CONTROL_NOT_FOUND
0x5f	LDAP_MORE_RESULTS_TO_RETURN	234	ERROR_MORE_DATA
0x60	LDAP_CLIENT_LOOP	8259	ERROR_DS_CLIENT_LOOP
0x61	LDAP_REFERRAL_LIMIT_EXCEEDED	8260	ERROR_DS_REFERRAL_LIMIT_EXCEEDED

<9> Section 2.6: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements attempt to find the ad:Message value in column B of the first table shown below.

- If the value is found in column B of the first table, the ad:ShortMessage is populated with text from column A.
- If no match is found in column B of the first table, ad:ShortMessage is populated from column A of the second table based on the error encountered, as described in column B.

First table:

A	B
AnonymousNotAllowed	Anonymous access to the directory is not permitted.
AttributeValueNotaObjRef	The attribute found is not a valid object reference (neither a GUID nor a string DN).
AttributeValueNotaString	The attribute found is not a String.
AttributeValueNotByteOrStringOrGuid	The attribute found is not a String, byte[] or GUID.
BadPutOrCreateValue	A Create or Put operation is being attempted with a bad value or values.
BadValue	An update is being attempted with an bad value.
BadValueForRangeHigh	Bad value has been specified for RangeHigh attribute.

A	B
BadValueForRangeLow	Bad value has been specified for RangeLow attribute.
CanOnlyReplaceParentObjectRefForUpdate	The parent object identity can only be replaced, not removed or added.
CanOnlyReplaceRdnForUpdate	The relative distinguished name (RDN) can only be replaced, not removed or added.
CantSetDistinguishedNameForCreate	The distinguished name attribute cannot be set during object creation. It is automatically set based on the relative distinguished name (RDN) and the parent object.
CantSetDistinguishedNameForUpdate	The distinguished name attribute cannot be updated. It is generated from the object's relative distinguished name (RDN) and the parent object.
CantSetObjectRefPropertyForCreate	The object reference property attribute cannot be set during object creation. It is automatically assigned by the directory.
CantSetObjectRefPropertyForUpdate	The object reference property attribute cannot be changed. It is automatically assigned by the directory at object creation.
CouldntFindObjectForMove	The object could not be found in the directory.
CouldntFindParentObjectForCreation	The parent object under which the new object is to be created could not be found in the directory.
CouldntRetrieveRootDSEForFilter	The RootDSE could not be retrieved from the directory. Please specify filter for the enumerate request under such circumstances.
CreateMissingValues	An AttributeTypeAndValue element in the Create operation did not contain any AttributeValue elements.
DuplicateAttributeWithValues	The attribute was found more than once, and has values.
DuplicateEnumerationCacheEntry	The EnumerationCacheEntry is a duplicate entry.
EmptyAttribute	The attribute has no value.
EmptyCreate	The Create operation did not contain any AttributeTypeAndValue elements.
EmptyPut	The Put operation did not contain any Change elements.
EnumContextAbsentInTheRequest	Request must specify the enumeration context.
ErrorWhileFetchingAttributeValues	Error while retrieving values for attribute {0} from the directory.
ImpersonationLevelNotSetToImpersonate	Impersonation level not set to Impersonate or higher by caller
InvalidBase64Binary	The base64Binary value len is not 4, or a multiple of 4.
InvalidDnWithStringBinaryAccessPointValue	The DN-with-binary, DN-with-string, or access-point value is in an invalid format.
InvalidEnumerationCacheEntry	The EnumerationCacheEntry is invalid.
InvalidObjectReferenceProperty	The supplied object reference property is not valid.
InvalidParentObjectRefForCreateAndUpdate	A single value must be specified for container-hierarchy-parent attribute.
InvalidPredicate	An update was specified with an invalid predicate.

A	B
InvalidPutSyntax	There is a mismatch between Put 'Operation' and the presence of an AttributeValue element
MaxEnumCtxsTotalReached	The maximum allowed number of enumeration contexts has been reached.
MissingDialect	Dialect not specified in the request.
MissingExpression	A create or update is missing an attribute.
MissingLowerRange	RangeLow attribute must be specified on the element with range qualifier.
MissingOrMultipleBaseObjectNodes	LdapQuery filter has a missing or multiple baseobject nodes
MissingOrMultipleFilterNodes	LdapQuery filter has a missing or multiple filter nodes
MissingOrMultipleScopeNodes	LdapQuery filter has a missing or multiple scope nodes
MissingScopeOrBaseObjectOrFilterNode	LdapQuery is missing the Scope, BaseObject or Filter node
MissingSelectionDialect	Selection dialect not specified in the request.
MissingSortingDialect	Sorting dialect not specified in the request.
MissingTypeAttribute	The attribute type is missing.
MissingValue	A create is missing the value for an attribute.
MustSpecifyBaseDnForQuery	Distinguished name search base must be supplied in the LdapQuery element.
MustSpecifyContainerForMove	Must specify the destination container to which the object is to be moved.
MustSpecifyDnForIdentifierLookup	Must specify the distinguished name to retrieve the object reference property.
MustSpecifyInstanceInfoInTheHeader	Instance Information is not provided in the Request Header.
MustSpecifyNamespace	An object has been found with no qualifying namespace.
MustSpecifyNonnullAttrValue	An attribute value cannot be null.
MustSpecifyObjectClassForCreation	Must specify the object class of the new object that is to be created.
MustSpecifyObjectRefPropInTheHeader	No object reference property element is present in the request header.
MustSpecifyParentForCreation	Must specify the parent object under which the new object is to be created.
MustSpecifyRdnForCreation	Must specify a relative distinguished name (RDN) for the new object during object creation.
MustSpecifyRdnForRename	Must specify the relative distinguished name (RDN) to which the object is to be renamed.
NewExpirationTimeNotSpecified	New expiration time/duration for the enumeration context is not specified in the renew request.
NoAttrTypeAndValsPresentInTheBody	There are no AttributeTypeAndValues present in the body to

A	B
	operate on.
NoChangesPresentInTheBody	There are no Changes present in the body to operate on.
NoConnectionAvailable	No connection is currently available to process the requested operation. This is typically a transient condition.
NoDCInstanceForGCSchemaLookup	No Domain Controller instance was found running on the system to look up schema for the Global Catalog instance.
NoDefaultNamingContextFoundForFilter	Default Naming Context could not be retrieved from the directory. Please specify filter for the enumerate request under such circumstances.
NoSuchEnumCtxGuidExists	Unknown or expired enumeration context.
NotCorrectFilterType	The supplied filter is of the wrong type.
NotificationSearchControlNotAllowed	Unsupported LDAP control. The Notification Search (1.2.840.113556.1.4.528) and Shutdown Notify (1.2.840.113556.1.4.1907) controls are not supported.
ObjectCreatedButIdentityUnknown	The object was created but its object reference property could not be retrieved from the directory.
ObjectInWrongNamespace	The object is in the wrong namespace.
OperationTimeout	The operation timed-out.
PageSizeMustBeGreaterThanZero	The number of items to retrieve must be greater than zero.
PutOperationUnsupported	The Put 'Operation' is invalid for this operation, or is unrecognized.
ReservedConnectionInvalidated	The connection for processing this request is unavailable. It may have been closed for being open or idle too long.
ScopeNodeNotOnelevelNorSubtreeNorBase	LdapQuery filter scope is not onelevel nor subtree nor base
ServerTimeMustBeNonNegative	The maximum duration for the Pull operation cannot be negative.
SessionsMismatch	Enumeration context belongs to a different principal.
SortKeyIsSpecialAttribute	Sort key on the specified directory attribute is not supported.
TooManySortKeySpecified	Too many sort keys were specified. Only one sort key is supported.
UnknownAttribute	The specified attribute {0} is unknown.
UnknownAttributeType	Unrecognized attribute found.
UnknownXmlNode	An unknown XML node was encountered and cannot be processed.
UnrecognizedDateAndTime	Expiration time does not correspond to any of the recognized datetime or duration format patterns.
UnrecognizedMaxElements	MaxElements does not correspond to valid xs:positiveInteger data type.

Second table:

A	B
EArgument	An ArgumentException was returned.
ECreate	A CreateException was returned.
EDirectoryOperation	A DirectoryOperationException was returned.
EEnumContextLimitExceeded	An EnumerationContextLimitExceededException was returned.
EInvalidExpression	An InvalidExpressionException was returned.
EInvalidModifyRequestSyntax	An InvalidModifyRequestSyntaxException was returned.
EInvalidOperation	An InvalidOperationException was returned.
EInvalidXml	An XmlException was returned.
ELdap	An LdapException was returned.
EModifyOperationUnsupported	An InvalidOperationException was returned.
ENoConnection	A NoConnectionAvailableException was returned.
EPut	A PutException was returned.
ESerialization	A SerializationException was returned.
EUnknownAttribute	An UnknownAttributeException was returned.

<10> Section 2.7: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements accessing any AD DS or AD LDS directory service impose the same limit on the maximum number of attribute values returned as the server version to which it is connected. Active Directory behavior for range retrieval and its imposed limits on values returned are defined in [MS-ADTS] section 3.1.1.3.1.3.3.

<11> Section 2.7.2: Microsoft implementations of Active Directory Web Services: Data Model and Common Elements, when accessing any AD DS or AD LDS directory service, impose the same limit on the maximum number of attribute values returned as the server version to which it is connected. Active Directory behavior for range retrieval and its imposed limits on values returned are defined in [MS-ADTS] section 3.1.1.3.1.3.3.

6 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
2.1 Endpoints	9224 : Clarified that the .Net Negotiate Stream protocol does not operate over Transport Layer Secutity (TLS).	Major

7 Index

F

G

I

L

N

O

P

R

S

T

U

V

W

X