[MS-WSPELD]:

WS-Transfer and WS-Enumeration Protocol Extension for Lightweight Directory Access Protocol v3 Controls

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

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

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

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

- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology,

there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
12/5/2008	0.1	Major	Initial Availability
1/16/2009	0.1.1	Editorial	Changed language and formatting in the technical content.
2/27/2009	0.1.2	Editorial	Changed language and formatting in the technical content.
4/10/2009	0.1.3	Editorial	Changed language and formatting in the technical content.
5/22/2009	0.1.4	Editorial	Changed language and formatting in the technical content.
7/2/2009	1.0	Major	Updated and revised the technical content.
8/14/2009	1.1	Minor	Clarified the meaning of the technical content.
9/25/2009	2.0	Major	Updated and revised the technical content.
11/6/2009	2.1	Minor	Clarified the meaning of the technical content.
12/18/2009	2.2	Minor	Clarified the meaning of the technical content.
1/29/2010	2.3	Minor	Clarified the meaning of the technical content.
3/12/2010	3.0	Major	Updated and revised the technical content.
4/23/2010	3.0.1	Editorial	Changed language and formatting in the technical content.
6/4/2010	3.0.2	Editorial	Changed language and formatting in the technical content.
7/16/2010	3.0.2	None	No changes to the meaning, language, or formatting of the technical content.
8/27/2010	4.0	Major	Updated and revised the technical content.
10/8/2010	4.0	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	4.0	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	5.0	Major	Updated and revised the technical content.
2/11/2011	5.0	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	5.0	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	5.0	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	5.1	Minor	Clarified the meaning of the technical content.
9/23/2011	5.1	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	6.0	Major	Updated and revised the technical content.
3/30/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
7/12/2012	6.1	Minor	Clarified the meaning of the technical content.
10/25/2012	6.1	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	6.1	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	7.0	Major	Updated and revised the technical content.
11/14/2013	7.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	7.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	7.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	8.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

This document specifies the WS-Transfer and WS-Enumeration Protocol Extension for Lightweight Directory Access Protocol v3 Controls, also known as WSPELD. WSPELD extends the Web Services Enumeration (WS-Enumeration) [WSENUM] and Web Services Transfer (WS-Transfer) [WXFR] Protocols. The WSPELD specification specifically targets directory servers exposing semantics similar to **LDAP** v3 [RFC2251].

WSPELD extensions to the SOAP-based Enumeration and Transfer protocol include the support for sending and receiving LDAP v3 request and response controls, respectively, while retrieving, querying, and modifying objects or object fragments stored in the **directory server** through **SOAP** messages. These extensions act as a controlling factor and influence the operation's behavior implemented using the preceding protocols in a manner specific to the control [RFC2251].

This document also describes how WSPELD extensions need to be used in conjunction with the WS-Transfer: Identity Management Operations for Directory Access Extensions [MS-WSTIM] of Web Services Transfer (WS-Transfer).

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

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

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

directory server: A persistent storage for DNS zones and records. A DNS server can access DNS data stored in a **directory server** using the **LDAP** protocol or a similar directory access mechanism.

domain controller (DC): The service, running on a server, that implements **Active Directory**, or the server hosting this service. The service hosts the data store for objects and interoperates with other **DCs** to ensure that a local change to an object replicates correctly across all **DCs**. When **Active Directory** is operating as Active Directory Domain Services (AD DS), the **DC** contains full NC replicas of the configuration naming context (config NC), schema naming context (schema NC), and one of the domain NCs in its forest. If the AD DS **DC** is a global catalog server (GC server), it contains partial NC replicas of the remaining domain NCs in its forest. For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS]. When **Active Directory** is operating as Active Directory Lightweight Directory Services (AD LDS), several AD LDS **DCs** can run on one server. When **Active Directory** is operating as AD DS, only one AD DS **DC** can run on one server. However, several AD LDS **DCs** can coexist with one AD DS **DC** on one server. The AD LDS **DC** contains full NC replicas of the config NC and the schema NC in its forest.

file/directory attributes: The attributes of a file or a directory, as specified in [MS-FSCC].

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].

object identifier (OID): In the context of an object server, a 64-bit number that uniquely identifies an object.

schema: The set of attributes and object classes that govern the creation and update of objects.

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 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.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL port type: A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

XML: The Extensible Markup Language, as described in [XML1.0].

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by **XML** itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2 References

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

1.2.1 Normative References

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

[MS-ADDM] Microsoft Corporation, "Active Directory Web Services: Data Model and Common Elements".

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

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

[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

[RFC2696] Weider, C., Herron, A., Anantha, A., and Howes, T., "LDAP Control Extension for Simple Paged Results Manipulation", RFC 2696, September 1999, http://www.ietf.org/rfc/rfc2696.txt

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[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

[SOAP1.2-2/2003] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part2-20030624

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[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/

[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/

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

[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/

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

None.

1.3 Overview

The WS-Enumeration [WSENUM] and WS-Transfer [WXFR] specifications are SOAP-based protocols that define various operations to manage data on a server. The WSPELD Protocol specifies LDAP v3

control extensions to operations defined by these protocols when the service implementation is targeting a LDAP v3 directory server [RFC2251].

Section 4.1.12 of [RFC2251] specifies that a client may send a control to the directory server by attaching a control structure to an LDAP operation.

A control consists of the following:

- An **object identifier (OID)** assigned to the control.

- The criticality value (true or false).

- The controlValue contents.

A control sent by the client to a server is known as a **request control**. In some cases, the server includes a corresponding control structure attached to the response for the LDAP operation. These controls are known as **response controls**.

The control structures are used for extending existing LDAP operations. For instance, a directory server based on LDAP v3 [RFC2251] may enforce the limits to the maximum number of objects (maxLimit) that can be returned in a single search operation performed. If the server enforces the maxLimit, then the pagedResultsControl control documented in [RFC2696] can be used to perform searches that return more objects than maxLimit by splitting the search into multiple searches, each of which returns no more objects than maxLimit. The pagedResultsControl control can be used as both a request control and response control.

WSPELD provides a way of sending and receiving the LDAP v3 controls explained here to control the operations defined by the WS-Enumeration and WS-Transfer protocols in a specific manner depending on the nature of the control [RFC2251]. These LDAP v3 control extensions apply to both request and response **SOAP messages**.

1.4   Relationship to Other Protocols

WSPELD is a common extension to both WS-Enumeration [WSENUM] and WS-Transfer [WXFR] protocols built on top of SOAP binding ([SOAP1.2-1/2003]) as shown in the following layering diagram.
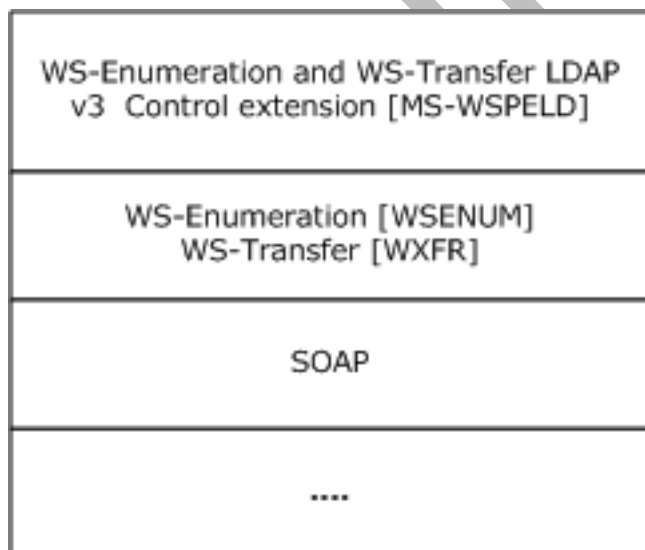
Figure 1: WSPELD protocol stack

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

There is an implicit assumption that the directory service in consideration supports an LDAP v3 directory service [RFC2251]. That is, it supports including LDAP v3 controls specifying extensions to existing directory operations by way of request and response messages.<1>

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** The WSPELD protocol extension can be implemented using transports that support sending SOAP messages as described in section Transport (2.1).

- **Protocol Versions:** The WSPELD protocol extension is not versioned.

- **Capability Negotiation:** The WSPELD protocol does not support capability negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2   Messages

2.1   Transport

The WSPELD protocol imposes no transport requirements or behaviors beyond those of the underlying WS-Enumeration [WSENUM] or WS-Transfer [WXFR] protocols. WS-Enumeration and WS-Transfer provide SOAP bindings for both SOAP 1.1 [SOAP1.1] and SOAP 1.2 [SOAP1.2-1/2003] [SOAP1.2-2/2003]. All messages MUST be formatted as specified by either SOAP 1.1 or SOAP 1.2.

2.2   Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses **XML schema**, as defined in [XMLSCHEMA1] and [XMLSCHEMA2], and **Web Services Description Language (WSDL)** as defined in [WSDL].

2.2.1   Namespaces

This specification references various **XML** namespaces using the mechanisms specified in [XMLNS]. 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 not significant for interoperability.

Prefixes and XML namespaces used in this specification include the following.

Prefix	Namespace URI	Reference
soapenv:	http://www.w3.org/2003/05/soap-envelope	[SOAP1.2-1/2003]
soapenv11:	http://schemas.xmlsoap.org/soap/envelope	[SOAP1.1]
wsa:	http://www.w3.org/2005/08/addressing	[WSENUM]
wsen:	http://schemas.xmlsoap.org/ws/2004/09/enumeration	[WSENUM]
ad:	http://schemas.microsoft.com/2008/1/ActiveDirectory	[MS-ADDM]
addata:	http://schemas.microsoft.com/2008/1/ActiveDirectory/Data	[MS-ADDM]
adlq:	http://schemas.microsoft.com/2008/1/ActiveDirectory/Dialect/LdapQuery	[MS-WSDS]
da:	http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess	[MS-WSTIM]

2.2.2   Messages

None.

2.2.3   Elements

The following table summarizes the set of common XML schema element definitions defined by this specification. XML schema element definitions that are specific to a particular operation are described with the operation.

Element	Description
ad:controls	Specify LDAP v3 request controls sent by a client and response controls returned back by the directory server.

2.2.3.1 ad:controls

ad:controls is an optional element to be specified by the client. In cases where a response control is required to be returned when the client sends an LDAP v3 request control, the server MUST return an LDAP v3 response control.<2>

The optional ad:controls element, if present:

- MUST be a child element to the Body element of the SOAP request and response message in the case of Get and Delete [WXFR] operations.

- MUST be a child element to the **da:BaseObjectSearchRequest**, **da:AddRequest**, and **da:ModifyRequest** element in the SOAP request message (and **da:BaseObjectSearchResponse**, **soapenv:Body**, and **soapenv:Body** element in the SOAP response message) in cases of Get, Create, and Put [MS-WSTIM] operations, respectively.

- MUST be a child element to the **wsen:Pull** element of the SOAP request message and the **wsen:PullResponse** element of the SOAP response message in the case of a Pull [WSENUM] operation.

Additionally, the **ad:controls** element MUST be the last child element in the SOAP response. WSPELD through this element specifies the support for including LDAP request controls sent through SOAP request messages. The messages pertain to various operations and LDAP response controls that are returned by the directory server to the caller in the SOAP response messages of respective operations.

The XML schema [XMLSCHEMA1] definition of the controls element is as follows.

```
<xsd:element name="controls">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="ad:control"
                        minOccurs="1" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

For example, the following XML shows the contents of the controls element.

```
<ad:controls>
    <ad:control type="OID as xsi:string" criticality="true|false">
    <ad:controlValue xsi:type="xsd:base64Binary">
        [base-64 encoded value]
    </ad:controlValue>
</ad:control>
</ad:controls>
```

2.2.3.1.1 ad:control

The required element **ad:control** MUST be the child element of the ad:controls (section 2.2.3.1) element. The **ad:control** element may have a value (see section 2.2.3.1.2 for the ad:controlValue element).

The XML schema [XMLSCHEMA1] definition of the control element is as follows.

```
<xsd:element name="control">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="ad:controlValue"
                        minOccurs="0" maxOccurs="1" />
```

```
            </xsd:sequence>
      <xsd:attribute name="type" use="required" type= "xsd:string" />
      <xsd:attribute name="criticality" use="required" type= "xsd:boolean" />
       </xsd:complexType>
   </xsd:element>
```

For example, the following XML shows the contents of the control element.

```
<ad:control  type="OID as xsi:string" criticality=(true|false)>
<ad:controlValue xsi:type="xsd:base64Binary">
       [base-64 encoded value]
</ad:controlValue>
</ad:control>
```

For the value of attributes "type" and "criticality", see sections 2.2.6.2 and 2.2.6.1, respectively.

2.2.3.1.2 ad:controlValue

The optional element **ad:controlValue**, if present, MUST be the child element of the ad:control element defined in section 2.2.3.1.1. This element specifies the value for the respective LDAP v3 request or response control.

The XML schema [XMLSCHEMA1] definition of the controlValue element is as follows.

```
<xsd:element name="controlValue" type="xsd:base64Binary" />
```

For example, the following XML shows the contents of the controlValue element.

```
<ad:controlValue xsi:type="xsd:base64Binary">
           [base-64 encoded value]
</ad:controlValue>
```

The controlValue MUST be a base64-encoded value. Attribute "xsi:type" SHOULD be included with its value equal to "xsd:base64Binary".

2.2.4 Complex Types

This specification does not define any common XML schema complex type definitions.

2.2.5 Simple Types

This specification does not define any common XML schema simple type definitions.

2.2.6 Attributes

The following table summarizes the set of common XML schema attribute definitions defined by this specification. XML schema attributes that are specific to a particular operation are described with the operation.

Attribute	Description
criticality	Specifies whether the LDAP v3 request or response control is critical.
type	Contains the Object Identifier (OID) of the LDAP v3 request or response control.

2.2.6.1 ad:controls/ad:control/@criticality

The value for required attribute ad:controls/ad:control/@criticality MUST be **Boolean** (true or false) as **String**.

2.2.6.2 ad:controls/ad:control/@type

The value for required attribute ad:controls/ad:control/@type MUST be a UTF-8 encoded, dotted-decimal representation of an OID as **String**. This attribute uniquely identifies the LDAP v3 controls as specified in section 4.1.12 of [RFC2251].

2.2.7 Groups

This specification does not define any common XML schema group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML schema attribute group definitions.

2.2.9 Common Data Structures

None.

2.3 Directory Service Schema Elements

This specification does not make use of any directory service **schema** elements.

3   Protocol Details

The following sections describe the behavior of the WSPELD protocol. As an extension to WS-Enumeration and WS-Transfer, the WSPELD extensions follow the same client-server model, in which a client sends a SOAP message containing a request (either a Get, Put, Create, Delete or Pull operation) to the server and the server responds with a SOAP message containing the response, or a SOAP fault, if an error occurred during the server processing.

The WSPELD extensions apply to both WS-Enumeration WSDL and WS-Transfer **WSDL port types**: the Enumeration port type (on which Pull operations are processed), the Resource WSDL port type (on which Get, Put, and Delete operations are processed), and the ResourceFactory WSDL port type (on which Create operations are processed). Since the WSPELD extensions are common to all these operations on all WSDL port types, a common server processing section is included that contains protocol details.

The client side of this protocol is a pass-through. No additional timers or state information is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

3.1   Common Server Details

This section documents the abstract data model and initialization procedure common to all WSDL port types.

3.1.1   Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

WSPELD extensions apply to both the WS-Enumeration [WSENUM] and WS-Transfer [WXFR] protocols. When added, these extensions participate as a controlling factor, influencing the existing operations performed by implementing those protocols, in a specific manner. WSPELD does not define any abstract data model of its own beyond the underlying data model description provided by the WS-Enumeration and WS-Transfer Protocols. Implementers should refer to the abstract model defined in section 3.1.1 of [MS-WSTIM], when these extensions are being used with WSTIM extensions.

3.1.2   Timers

None.

3.1.3   Initialization

WSPELD imposes no initialization requirements beyond those of the underlying WS-Enumeration or WS-Transfer Protocol. When initialized, WS-Enumeration and WS-Transfer begin listening on endpoints for the Enumeration, Resource, and ResourceFactory WSDL port types. The URIs for the endpoints, as well as the transport and security mechanisms to use, are implementation-dependent.<3>

3.2   Resource Server Details

In WS-Transfer [WXFR], this WSDL port type is used to process three WSDL operations:

▪   Get

- Put

- Delete

3.2.1 Abstract Data Model

See the abstract data model in the common processing section 3.1.1.

3.2.2 Timers

None.

3.2.3 Initialization

See the initialization in the common server details, section 3.1.3.

3.2.4 Message Processing Events and Sequencing Rules

The Resource WSDL port type includes three operations.

Operation	Description
Get	Retrieves one or more **directory attributes** from a **directory object**.
Put	Modifies the contents of one or more directory attributes on a directory object.
Delete	Removes an existing directory object.

The LDAP request and response controls element specified in section 2.2.3.1 could be a part of both SOAP request and response messages, respectively, for the preceding operations.

3.2.4.1 Get

This section specifies extensions to the Get operation defined by WS-Transfer [WXFR]. It also specifies how these extensions must be used in conjunction with the Get operation extension, defined by the WSTIM extensions [MS-WSTIM].

The WS-Transfer specification of Get operates at the level of granularity of the complete representation of an entire directory object. That is, when the object is retrieved, the complete contents of the object MUST be retrieved.

There are no body blocks defined by default for a Get request in the WS-Transfer specification. Additionally, WS-Transfer mandates that the directory object requested MUST be the initial child element of the **soapenv:Body** element of the response message.

The presence of subsequent child elements is service-specific according to [WXFR].

WSPELD specifies that the **ad:controls** element defined in section 2.2.3.1, if present in the Get request message, MUST be a child element of the **soapenv:Body** element and may be specified in any order in the presence of other child elements. See the example in section 4.3 for the WS-Transfer Get request message with controls extension. However, for the Get Response message, the protocol mandates that the **ad:controls** element, if present, MUST be the last child element of the **soapenv:Body** element.

The WSTIM extensions of Get extensions operates at the level of granularity of portions of the representation of a directory object. That is, its extensions allow retrieving a fragment of a directory object. To request such granularity, it defines a **BaseObjectSearchRequest** element, wrapping various child elements representing the requested fragment inside the **soapenv:Body** element of the request message (the response message of the preceding request contains the

BaseObjectSearchResponse element, containing one or more **PartialAttribute** elements representing the fragment of the directory object. WSPELD specifies that **ad:controls** element defined in section 2.2.3.1, if present, MUST be the child element of the **BaseObjectSearchRequest** element in the request message and may be specified in any order). However, for the Response message, the protocol mandates that the **ad:controls** element, if present, MUST be the last child element of the **BaseObjectSearchResponse** element.

3.2.4.2 Delete

This section specifies extensions to the delete operation defined by WS-Transfer [WXFR].

There are no body elements defined for a Delete request and Response message by WS-Transfer. WSPELD specifies that the **ad:controls** element defined in section 2.2.3.1, if present in the Delete request message, MUST be a child element of the **soapenv:Body** element and may be specified in any order. However, for the Response message, the protocol mandates that the **ad:controls** element, if present, MUST be the last child element of the **soapenv:Body** element.

See the example in section 4.2 for the WS-Transfer Delete request message with controls extension.

3.2.4.3 Put

This section specifies extensions to the Put operation extensions defined by the WSTIM extensions [MS-WSTIM].

The WSTIM extensions specification of Put extensions operates at the level of granularity of portions of the representation of a directory object. That is, its extensions allow modifying a fragment of a directory object.

To request modifications at such granularity, the change elements are wrapped inside the **ModifyRequest** element (defined in [MS-WSTIM] section 3.2.4.2) of the request message. WSPELD specifies that the **ad:controls** element defined in section 2.2.3.1, if present, MUST be the child element of the **ModifyRequest** element in the request message and may be specified in any order in the presence of other child elements. The response to this Put operation, **ModifyResponseMessage** ([MS-WSTIM] section 3.2.4.2.1.2), has an empty message body. However, for a response message the **ad:controls** element, if present, MUST be a child element of the **soapenv:Body** element of this **ModifyResponseMessage**.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

3.3 ResourceFactory Server Details

In WS-Transfer [WXFR], this WSDL port type is used to process the Create operation.

Note that unlike operations performed on the Resource WSDL port type, the Create operation does not target a specific directory object but rather causes the creation of a directory object, which can subsequently be targeted by operations on the Resource WSDL port type.

3.3.1 Abstract Data Model

See the abstract data model in the common processing section 3.1.1.

3.3.2 Timers

None.

3.3.3 Initialization

See the initialization in the common processing section 3.1.3.

3.3.4 Message Processing Events and Sequencing Rules

The ResourceFactory WSDL port type includes one operation.

Operation	Description
Create	Creates a new directory object

The LDAP request and response controls element specified in section 2.2.3.1 could be a part of both Create SOAP request and response messages, respectively.

3.3.4.1 Create

This section specifies extensions to the Create operation extensions defined by the WSTIM extensions [MS-WSTIM].

The WSTIM extensions specification of create extensions permits the client to specify only a portion of the directory object that is to be created, with the server supplying any remaining portions of the directory object when it creates the object.

WSPELD specifies that the **ad:controls** element defined in section 2.2.3.1, if present, MUST be the child element of the **AddRequest** element in the request message and may be specified in any order in the presence of other child elements. The contents of SOAP response message (AddResponseMessage [MS-WSTIM] (section 3.3.4.1.1.2)) are a resource reference, <wxf:ResourceCreated>, for the newly created object. WSPELD specifies that the **ad:controls** element, if present, MUST be the last child element of the soapenv:Body element of this **AddResponseMessage**.

See the example in section 4.1 for the WSTIM extensions Create request and response message with controls extension.

3.3.5 Timer Events

None.

3.3.6 Other Local Events

None.

3.4 Enumeration Server Details

In WS-Enumeration [WSENUM], this WSDL port type is used to process the following five WSDL operations:

- Enumeration

- Pull

- Renew

- GetStatus

- Release

However, WSPELD extensions apply only to the Pull WSDL operation among the five of them.

3.4.1 Abstract Data Model

See the abstract data model in the common processing section 3.1.1.

3.4.2 Timers

None.

3.4.3 Initialization

See the initialization in the common processing section 3.1.3.

3.4.4 Message Processing Events and Sequencing Rules

The Enumeration WSDL port type includes one operation to which WSPELD applies.

Operation	Description
Pull	Pulls the resultant directory objects in the context of a specific enumeration

The LDAP request and response controls element specified in section 2.2.3.1 could be a part of both SOAP Pull request and response messages, respectively.

3.4.4.1 Pull

This section specifies extensions to the Pull operation defined by WS-Enumeration [WSENUM].

WS-Enumeration specifies that the Pull operation, which involves pulling directory object fragments requested during Enumeration, is initiated by sending a Pull request message containing the Pull element inside a **soapenv:Body** element to the directory server.

On successful processing of a Pull request message, the directory server is expected to return a Pull response message containing the resultant object fragments wrapped by a **pullResponse** element inside a **soapenv:Body** element.

WSPELD specifies that the **ad:controls** element defined in section 2.2.3.1, if present, MUST be the child element of the Pull element in the request message and may be specified in any order in the presence of other child elements. However, for the response message, the protocol mandates that the **ad:controls** element, if present, MUST be the last child element of the **PullResponse** element.

See the example in section 4.4 for the WS-Enumeration [WSENUM] Pull request and response message with controls extension.

3.4.5 Timer Events

None.

3.4.6 Other Local Events

None.

4 Protocol Examples

In this section, several sample WS-Transfer and WS-Enumeration operations with controls are presented.

1. Section 4.1 : A directory object is created using the Create operation.

2. Section 4.2: The directory object is deleted from the directory server using the Delete operation.

3. Section 4.3: Information from that deleted directory object is then retrieved using the Get operation including ShowDeleted control.

4. Section 4.4: The object is enumerated and pulled using the WS-Enumeration Pull operation.

For illustrative purposes, these examples have been shown in the context of **Active Directory** Web Services.

4.1 WS-Transfer LDAP v3 Controls Extensions Create Example

In this example, a new directory object is created using a Create operation with the [MS-WSTIM] protocol extensions. During the creation, an LDAP control (LDAP_SERVER_LAZY_COMMIT_OID) ([MS-ADTS] section 3.1.1.3.4.1.7) is added to the request, which instructs the **DC** that it may sacrifice durability guarantees on updates to improve performance. Sending this control to the DC does not cause the server to include any controls in its response.

SOAP Request Message

```
<soapenv:Envelope
   xmlns: soapenv="http://www.w3.org/2003/05/soap-envelope"
   xmlns:wsa="http://www.w3.org/2005/08/addressing"
   xmlns:wxf="http://schemas.xmlsoap.org/ws/2004/09/transfer"
   xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"
   xmlns:addata=
"http://schemas.microsoft.com/2008/1/ActiveDirectory/Data"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <soapenv:Header>
      <wsa:Action soapenv:mustUnderstand="1">
        http://schemas.xmlsoap.org/ws/2004/09/transfer/Create
      </wsa:Action>
      <IdentityManagementOperation soapenv:mustUnderstand="1" xmlns=
"http://schemas.microsoft.com/2006/11/IdentityManagement/
DirectoryAccess" />
      <instance xmlns="http://schemas.microsoft.com/2008/1/
ActiveDirectory">
        ldap:389
      </instance>
      <wsa:MessageID>
        urn:uuid:e0a0abb3-0c4f-4679-ad0c-366078c47c32
      </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/ResourceFactory</wsa:To>
    </soapenv:Header>
    <soapenv:Body>
      <AddRequest Dialect="http://schemas.microsoft.com/2008/1/
ActiveDirectory/Dialect/XPath-Level-1"
xmlns="http://schemas.microsoft.com/2006/11/IdentityManagement/
DirectoryAccess>        <AttributeTypeAndValue>
        <AttributeType>addata:objectClass</AttributeType>
```

```
              <AttributeValue>
                <ad:value xsi:type="xsd:string">user</ad:value>
              </AttributeValue>
            </AttributeTypeAndValue>
            <AttributeTypeAndValue>
              <AttributeType>addata:description</AttributeType>
              <AttributeValue>
                <ad:value xsi:type="xsd:string">Sample description.
</ad:value>
              </AttributeValue>
            </AttributeTypeAndValue>
            <AttributeTypeAndValue>
              <AttributeType>addata:otherTelephone</AttributeType>
              <AttributeValue>
                <ad:value xsi:type="xsd:string">(425) 555-0100
</ad:value>
                <ad:value xsi:type="xsd:string">(206) 555-0100
</ad:value>
              </AttributeValue>
            </AttributeTypeAndValue>
            <AttributeTypeAndValue>
              <AttributeType>ad:container-hierarchy-parent
</AttributeType>
              <AttributeValue>
                <ad:value xsi:type="xsd:string">
                  CN=Users,DC=fabrikam,DC=com
                </ad:value>
              </AttributeValue>
            </AttributeTypeAndValue>
            <AttributeTypeAndValue>
              <AttributeType>ad:relativeDistinguishedName
</AttributeType>
              <AttributeValue>
                <ad:value xsi:type="xsd:string">CN=Sample User
</ad:value>
              </AttributeValue>
            </AttributeTypeAndValue>
            <ad:controls>
              <ad:control type='1.2.840.113556.1.4.619'
criticality='true' />
            </ad:controls>
            </AddRequest>
        </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"
      xmlns:wxf="http://schemas.xmlsoap.org/ws/2004/09/transfer"
      xmlns:ad="http://schemas.microsoft.com/2008/1/
ActiveDirectory">
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
      http://schemas.xmlsoap.org/ws/2004/09/transfer/
CreateResponse
    </wsa:Action>
    <wsa:RelatesTo>
      urn:uuid:e0a0abb3-0c4f-4679-ad0c-366078c47c32
    </wsa:RelatesTo>
    <wsa:To soapenv:mustUnderstand="1">
      http://www.w3.org/2005/08/addressing/anonymous
    </wsa:To>
  </soapenv:Header>
  <soapenv:Body>
```

```
        <wxf:ResourceCreated>        <wsa:Address>net.tcp://
server01.fabrikam.com:9389/ActiveDirectoryWebServices/Windows/
Resource</wsa:Address>
          <wsa:ReferenceParameters>
            <ad:objectReferenceProperty>
              afc7af4d-951b-4ae7-b80e-6dacae67ceb
            </ad:objectReferenceProperty>
            <ad:instance>ldap:389</ad:instance>
          </wsa:ReferenceParameters>
        </wxf:ResourceCreated>
  </soapenv:Body>
  </soapenv:Envelope>
```

4.2   WS-Transfer LDAP v3 Controls Extensions Delete Example

In this example, the directory object created previously is deleted using a [WXFR] delete operation. During the deletion, the *LDAP_SERVER_TREE_DELETE_OID* control ([MS-ADTS] section 3.1.1.3.4.1.15) is added to the request, which causes the server to recursively delete the entire subtree of objects located under the object specified in the request (including the specified object). The response to this operation could be seen in the response message.

SOAP Request Message

```
<soapenv:Envelope
      xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
      xmlns:wsa="http://www.w3.org/2005/08/addressing"
      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">
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete</wsa:Action>
      <ad:objectReferenceProperty>
        51419d0c-cadd-4b70-9c8d-8a73fdd2d32a
      </ad:objectReferenceProperty>
      <ad:instance xmlns="http://schemas.microsoft.com/2008/1/
ActiveDirectory">ldap:389</ad:instance>
      <wsa:MessageID>
         urn:uuid:c694c005-75a2-4cb1-84f9-8d6890da9958
      </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>
    <ad:controls xmlns:ad='http://schemas.microsoft.com/2008/1/
ActiveDirectory'>
      <ad:control type="1.2.840.113556.1.4.805"
criticality="true" />
    </ad:controls>
  </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/DeleteResponse
</wsa:Action>
    <wsa:RelatesTo>urn:uuid:c694c005-75a2-4cb1-84f9-8d6890da9958
</wsa:RelatesTo>
    <wsa:To soapenv:mustUnderstand="1">http://www.w3.org/2005/08/
addressing/anonymous</wsa:To>
  </soapenv:Header>
  <soapenv:Body />
</soapenv:Envelope>
```

4.3   WS-Transfer LDAP v3 Controls Extensions Get Example

In this example, the directory object deleted previously is retrieved using a [WXFR] Get operation. During the extraction, the LDAP show deleted control (*LDAP_SERVER_SHOW_DELETED_OID*) ([MS-ADTS] section 3.1.1.3.4.1.14) is added to the request, which specifies that the search results performed by the Get operation for extraction covers any deleted objects. The response to this operation could be seen in the response message.

SOAP Request Message

```
<soapenv:Envelope
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory">
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</wsa:Action>
    <ad:objectReferenceProperty>
      51419d0c-cadd-4b70-9c8d-8a73fdd2d32a
    </ad:objectReferenceProperty>
    <ad:instance>ldap:389</ad:instance>
    <wsa:MessageID>
         urn:uuid:928f9abe-875d-4bc1-bb5f-d1be16c035c3
    </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>
    <ad:controls>
      <ad:control type="1.2.840.113556.1.4.417"
criticality="false" />
    </ad:controls>
  </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"
     xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"
     xmlns:addata="http://schemas.microsoft.com/2008/1/
```

```
ActiveDirectory/Data"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soapenv:Header>
    <wsa:Action soapenv:mustUnderstand="1">
http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
</wsa:Action>
    <wsa:RelatesTo>urn:uuid:928f9abe-875d-4bc1-bb5f-d1be16c035c3
</wsa:RelatesTo>
    <wsa:To soapenv:mustUnderstand="1">http://www.w3.org/2005/08/
addressing/anonymous</wsa:To>
  </soapenv:Header>
  <soapenv:Body>
    <addata:organizationalUnit>
      <ad:objectReferenceProperty>

51419d0c-cadd-4b70-9c8d-8a73fdd2d32a
      </ad:objectReferenceProperty>
      <addata:objectClass LdapSyntax="ObjectIdentifier">
        <ad:value xsi:type="xsd:string">top</ad:value>
        <ad:value xsi:type="xsd:string">organizationalUnit
</ad:value>
      </addata:objectClass>
      <addata:uSNCreated LdapSyntax="LargeInteger">
        <ad:value xsi:type="xsd:string">29823</ad:value>
      </addata:uSNCreated>
      <addata:name LdapSyntax="UnicodeString">
        <ad:value xsi:type="xsd:string">GetADOU</ad:value>
      </addata:name>
      <addata:objectGUID LdapSyntax="OctetString">
        <ad:value xsi:type="xsd:base64Binary">
90xlHcX0nUyOp2io+Q397Q==</ad:value>
      </addata:objectGUID>
      <addata:whenCreated LdapSyntax="GeneralizedTimeString">
        <ad:value xsi:type="xsd:string">20080811233505.0Z
</ad:value>
      </addata:whenCreated>
      <addata:whenChanged LdapSyntax="GeneralizedTimeString">
        <ad:value xsi:type="xsd:string">20080811233505.0Z
</ad:value>
      </addata:whenChanged>
      <addata:dSCorePropagationData
LdapSyntax="GeneralizedTimeString">
        <ad:value xsi:type="xsd:string">16010101000000.0Z
</ad:value>
      </addata:dSCorePropagationData>
      …….
      …….
      …….
      …….
      …….
      …….
      …….
      …….
      …….
    </addata:organizationalUnit>
  </soapenv:Body>
</soapenv:Envelope>
```

4.4   WS-Transfer LDAP v3 Controls Extensions Pull Example

In this example, the directory object is retrieved using a [WSENUM] Pull operation in the context of an Enumeration specifying the search of the directory object in previous examples. During the extraction, both the LDAP show deleted control (*LDAP_SERVER_SHOW_DELETED_OID*) ([MS-ADTS] section 3.1.1.3.4.1.14) and the LDAP statistical control (*LDAP_SERVER_GET_STATS_OID*) ([MS-ADTS] section

3.1.1.3.4.1.6) are added to the request. The response to this control could be seen in the response message.

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:ad="http://schemas.microsoft.com/2008/1/
ActiveDirectory">
      <wsen:EnumerationContext>
cda3e08b-cec1-42bb-8245-7cb6235a24b8</wsen:EnumerationContext>
      <wsen:MaxTime>PT10S</wsen:MaxTime>
      <wsen:MaxElements>1</wsen:MaxElements>
      <ad:controls>
        <ad:control type="1.2.840.113556.1.4.970"
criticality="true" />
        <ad:control type="1.2.840.113556.1.4.417"
criticality="false" />
      </ad:controls>
    </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>
      cda3e08b-cec1-42bb-8245-7cb6235a24b8
    </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>
    <ad:controls>
      <ad:control type="1.2.840.113556.1.4.970"
criticality="false">
        <ad:controlValue xsi:type="xsd:base64Binary">
MIQAAAB2AgEBAgEBAgEDAgEgAgEFAgECAgEGAgEFAgEHBBUgKG9iamVjdENsYXN
zPXVzZXIpIAACAQgEFVRFTVBPUkFSWV9TT1JUX0lOREVYAAIBCQIBXQIBCgIBBQ
IBCwIBAAIBDAIBAgIBDQIBBQIBDgIBAAIBDwIBAA==</ad:controlValue>
      </ad:control>
    </ad:controls>
  </wsen:PullResponse>
</soapenv:Body>
</soapenv:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

There are no known additional security considerations for these protocol extensions, but server implementers are encouraged to use transport mechanisms that support encryption and integrity-verification of the messages.

5.2 Index of Security Parameters

This protocol extension has no security parameters.

6   Appendix A: WSDL (Contains Extension Elements Only)

The WSPELD protocol extension does not define a WSDL of its own. It does not extend the [WSENUM] WSDL and extends only [WSENUM] XML schema. The schema of extension elements defined by this protocol for [WSENUM] is specified in section 7.1 of Appendix B. For a server to implement WSPELD protocol extensions of [WSENUM] it should use the full WSDL definition of the WS-Enumeration [WSENUM] protocol, except replacing [WSENUM]'s definitions of XML elements, attributes, and complex types that are being extended by this protocol with those documented in section 7 Appendix B.

 The WSDL extension elements of [WXFR] and [MS-WSTIM] defined by this protocol are shown below. For a server to implement WSPELD protocol extensions of [WXFR] it should use the full WSDL definition of the [WXFR] protocol, except replacing the [WXFR] definitions of WSDL elements that are being extended by this protocol with those documented below. For a server to implement WSPELD protocol extensions to [MS-WSTIM] it should use the full WSDL definition of the [MS-WSTIM] protocol, except replacing the [MS-WSTIM] definitions of WSDL elements that are being extended by this protocol with those documented below. It should also replace XML elements, attributes, and complex types that are being extended by this protocol with those documented in section 7 Appendix B.

Section 7.2 of Appendix B contains the schema of extension elements defined by this protocol, which are used together with the extension elements defined by [MS-WSTIM] to extend the [WXFR] protocol. For a server to implement WSPELD and WSTIM protocol extensions of [WXFR] together, it should use the full WSDL definition of the [MS-WSTIM] protocol, except replacing the [MS-WSTIM] definitions of XML elements, attributes, and complex types that are being extended by this protocol with those documented in section 7 Appendix B.

```xml
<?xml version="1.0" encoding="utf-8"?>
<!-- extended [WXFR] WSDL-->

<wsdl:definitions
    targetNamespace="http://schemas.xmlsoap.org/ws/2004/09/transfer"
    xmlns:tns="http://schemas.xmlsoap.org/ws/2004/09/transfer"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"


  <!--import schema for elements defined by [MS-WSPELD]-->
  <wsdl:types>
    <xsd:schema>
      <xsd:import
        namespace="http://schemas.xmlsoap.org/ws/2004/09/transfer"
        schemaLocation="http://schemas.xmlsoap.org/ws/2004/09/transfer/transfer.xsd"/>
      <xsd:import
        namespace="http://schemas.microsoft.com/2008/1/ActiveDirectory" />
    </xsd:schema>
  </wsdl:types>

  <!--...other WSDL elements defined by [WXFR]
  ...
  ...
  ...
  ...-->

  <!-- extended [WXFR] messages-->

  <wsdl:message name="DeleteRequestMessage">
```

```
      <wsdl:part name="DeleteRequestMessageBody" element="ad:controls" minOccurs ="0"
maxOccurs="1"/>
   </wsdl:message>

   <wsdl:message name="DeleteResponseMessage">
      <wsdl:part name="DeleteResponseMessageBody" element="ad:controls" minOccurs ="0"
maxOccurs="1"/>
   </wsdl:message>

   <wsdl:message name="GetRequestMessage">
      <wsdl:part name="GetRequestMessageBody" element="ad:controls" minOccurs ="0"
maxOccurs="1"/>
   </wsdl:message>

   <wsdl:message name="GetResponseMessage">
      <wsdl:part name="GetResponseMessageBody" element="ad:controls" minOccurs ="0"
maxOccurs="1"/>
   </wsdl:message>
</wsdl:definitions>
```

The following is the extended WSDL for [WSPELD] extensions to [MS-WSTIM].

```
<?xml version="1.0" encoding="utf-8"?>
<!-- extended [MS-WSTIM] WSDL-->
<wsdl:definitions
     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess"
xmlns:da="http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wxf="http://schemas.xmlsoap.org/ws/2004/09/transfer"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12"
  xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"

  >
<!--import schema for elements defined by [WSPELD]-->
<wsdl:types>
   <xsd:schema>
     <xsd:import
       namespace="http://schemas.xmlsoap.org/ws/2004/09/transfer"
       schemaLocation="http://schemas.xmlsoap.org/ws/2004/09/transfer/transfer.xsd"/>
     <xsd:import
       namespace="http://schemas.microsoft.com/2008/1/ActiveDirectory" />
     <xsd:import
      namespace="http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess" />
   </xsd:schema>
</wsdl:types>

<!--...other WSDL elements defined by [MS-WSTIM]
...
...
...
...-->

<!-- extended [MS-WSTIM] messages-->

<wsdl:message name="ModifyResponseMessage">
   <wsdl:part name="ModifyResponseMessageBody" element="ad:controls" />
</wsdl:message>

<!--...other WSDL elements defined by [MS-WSTIM]
...
...
...
```

```
    ...-->

<wsdl:message name="AddResponseMessage">
    <wsdl:part name="AddResponseMessageBody" type="da:AddResponseType"/>
  </wsdl:message>

</wsdl:definitions>
```

7 Appendix B: Schema

This section provides the additional schema elements for the extensions defined by this protocol. Subsequent subsections show extended schema of [WSENUM], [WXFR] and [MS-WSTIM] importing these extensions elements.

```xml
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema targetNamespace="http://schemas.microsoft.com/2008/1/ActiveDirectory"
   xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >

  <!--[MS-WSPELD] extension elements]-->

  <xsd:element name="controlValue" type="xsd:base64Binary" />

  <xsd:element name="control">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ad:controlValue"
                   minOccurs="0" maxOccurs="1" />
      </xsd:sequence>
      <xsd:attribute name="type" use="required" type= "xsd:string" />
      <xsd:attribute name="criticality" use="required" type= "xsd:boolean" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="controls">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ad:control" minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

7.1 WS-Enumeration [WSENUM] Extended XML Schema

For clarity, elements of the WS-Enumeration [WSENUM] schema, which are untouched and have not been extended by this protocol are not shown in the extended schema. To obtain the full extended [WSENUM] schema, non-extended XML elements, attributes, complex types, and so on from the [WSENUM] schema MUST be included.

```xml
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
    targetNamespace="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
    xmlns:tns="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    blockDefault="#all"
    xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory">

  <!--import schema for [MS-WSPELD] extension elements-->
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace" />
  <xsd:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    schemaLocation= "http://schemas.xmlsoap.org/ws/2004/08/addressing/addressing.xsd" />
  <xsd:import namespace="http://schemas.microsoft.com/2008/1/ActiveDirectory"/>

  <!--other elements defined by [WSENUM]
  ...
  ...
  ...
  ...
```

```
      ...
      ...-->

    <!-- Extended Pull request -->
    <xsd:element name="Pull">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="EnumerationContext"
                       type="tns:EnumerationContextType" />
          <xsd:element name="MaxTime" type="tns:PositiveDurationType"
                       minOccurs="0" />
          <xsd:element name="MaxElements" type="xsd:positiveInteger"
                       minOccurs="0" />
          <xsd:element name="MaxCharacters" type="xsd:positiveInteger"
                       minOccurs="0" />
          <xsd:element ref="ad:controls"
                       minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="Dialect" type="xsd:anyURI" />
      </xsd:complexType>
    </xsd:element>

    <!-- Extended Pull response -->
    <xsd:element name="PullResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="EnumerationContext"
                       type="tns:EnumerationContextType"
                       minOccurs="0" />
          <xsd:element name="Items" type="tns:ItemListType"
                       minOccurs="0" />
          <xsd:element name="EndOfSequence" minOccurs="0" />
          <xsd:element ref="ad:controls"
                       minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="Dialect" type="xsd:anyURI" />
      </xsd:complexType>
    </xsd:element>

    <!--other elements defined by [WSENUM]
    ...
    ...
    ...
    ...
    ...
    ...-->
  </xsd:schema>
```

7.2   Identity Management Operations for Directory Access [WSTIM] Extended XML Schema

This section illustrates the application of WSPELD extensions to [MS-WSTIM] protocol i.e. it shows how WSPELD extensions must be used in conjunction with [MS-WSTIM] extensions. For clarity, elements of [MS-WSTIM] schema which are untouched and not been extended by this protocol are not shown in the extended schema. To obtain the full extended [MS-WSTIM] schema, non extended XML elements, attributes, complex types etc from [MS-WSTIM] schema MUST be included.

```
    <?xml version="1.0" encoding="utf-8"?>
    <xsd:schema
      targetNamespace="http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess"
      xmlns:da="http://schemas.microsoft.com/2006/11/IdentityManagement/DirectoryAccess"
      xmlns:wsa="http://www.w3.org/2005/08/addressing"
      xmlns:wxf="http://schemas.xmlsoap.org/ws/2004/09/transfer"
      xmlns:ad="http://schemas.microsoft.com/2008/1/ActiveDirectory"
```

```
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12"
        >

    <!--import schema for [MS-WSPELD] extension elements-->
    <xsd:import namespace="http://schemas.microsoft.com/2008/1/ActiveDirectory" />
    <xsd:import namespace="http://schemas.xmlsoap.org/ws/2004/09/transfer" />
    <!--other elements defined by [MS-WSTIM]
    ...
    ...
    ...
    ...
    ...
    ...-->

    <xsd:element name="BaseObjectSearchRequest">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="da:AttributeType" minOccurs="0"
                       maxOccurs="unbounded" />
          <xsd:element ref="ad:controls"
                             minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="Dialect" type="xsd:anyURI" />
      </xsd:complexType>
    </xsd:element>

    <xsd:element name="BaseObjectSearchResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="PartialAttribute"
                       type="da:PartialAttributeXmlType"
                       minOccurs="1" maxOccurs="unbounded" />
          <xsd:element ref="ad:controls" minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>

    <xsd:element name="AddRequest">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="da:AttributeTypeAndValue"
                       minOccurs="0" maxOccurs="unbounded" />
          <xsd:element ref="ad:controls" minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="Dialect" type="xsd:anyURI" />
      </xsd:complexType>
    </xsd:element>

    <!--other elements defined by [MS-WSTIM]
    ...
    ...
    ...
    ...
    ...
    ...-->

    <xsd:element name="ModifyRequest">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="da:Change"
                       minOccurs="0" maxOccurs="unbounded" />
          <xsd:element ref="ad:controls" minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
        <xsd:attribute name="Dialect" type="xsd:anyURI" />
      </xsd:complexType>
    </xsd:element>
```

```
        <!--other elements defined by [MS-WSTIM]
      ...
      ...
      ...
      ...
      ...
      ...-->
      <xsd:complexType name="AddResponseType" >
        <xsd:sequence>
          <xsd:element ref="wxf:ResourceCreated" />
          <xsd:element ref="ad:controls" minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
      </xsd:complexType>

      <!--other elements defined by [MS-WSTIM]
      ...
      ...
      ...
      ...
      ...
      ...-->
    </xsd:schema>
```

8   Appendix C: Product Behavior

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

Note: Some of the information in this section is subject to change because it applies to an unreleased, preliminary version of the Windows Server operating system, and thus may differ from the final version of the server software when released. All behavior notes that pertain to the unreleased, preliminary version of the Windows Server operating system contain specific references to Windows Server 2016 Technical Preview as an aid to the reader.

- Active Directory Management Gateway Service

- Remote Server Administration Tools for Windows 7

- Windows Server 2008 R2 operating system

- Remote Server Administration Tools for Windows 8 operating system

- Windows Server 2012 operating system

- Remote Server Administration Tools for Windows 8.1 operating system

- Windows Server 2012 R2 operating system

- Remote Server Administration Tools for Windows 10 operating system

- Windows Server 2016 Technical Preview operating system

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

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

<1> Section 1.6: Active Directory Management Gateway Service contains the server implementation of the WS-Transfer and WS-Enumeration Protocol Extension for Lightweight Directory Access Protocol v3 Controls; all versions of the Remote Server Administration Tools found in the applicability list at the beginning of this section, contain the client implementation; all versions of Windows Server operating system, according to the applicability list at the beginning of this section, have both the server and the client implementations.

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

<2> Section 2.2.3.1: All controls listed in [MS-ADTS] are supported by Microsoft implementations of WS-Transfer and WS-Enumeration Protocol Extension for Lightweight Directory Access Protocol v3 Controls. However, Active Directory Web Services clients make use of following LDAP v3 request controls:

- 1.2.840.113556.1.4.417 (LDAP_SERVER_SHOW_DELETED_OID)

- 1.2.840.113556.1.4.2065 (LDAP_SERVER_SHOW_DEACTIVATED_LINK)

- 1.2.840.113556.1.4.521 (LDAP_SERVER_CROSSDOM_MOVE_TARGET_OID)

- 1.2.840.113556.1.4.801 (LDAP_SERVER_SD_FLAGS_OID)

- 1.2.840.113556.1.4.805 (LDAP_SERVER_TREE_DELETE_OID)

- 1.2.840.113556.1.4.1413 (LDAP_SERVER_PERMISSIVE_MODIFY_OID)

- 1.2.840.113556.1.4.1504 (LDAP_SERVER_ASQ_OID)

- 1.2.840.113556.1.4.2026 (LDAP_SERVER_INPUT_DN_OID)

and the following LDAP v3 response controls:

- 1.2.840.113556.1.4.1504 (LDAP_SERVER_ASQ_OID)

The previous list can change in the future versions.

<3> Section 3.1.3: Microsoft implementations of WS-Transfer and WS-Enumeration Protocol Extension for Lightweight Directory Access Protocol v3 Controls use SOAP 1.2 [SOAP1.2-1/2003]. The transports used, as well as the authentication mechanisms supported and the endpoints exposed, are specified in section 2.1 of [MS-ADDM].

9 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.

- The removal of a document from the documentation set.

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

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

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- Obsolete document removed.

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

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

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
8 Appendix C: Product Behavior	Updated product applicability list to include Remote Server Administration Tools for Windows 10.	Y	Content update.

10 Index

A

Abstract data model
  enumeration server (section 3.1.1 16, section 3.4.1 20)
  resource server (section 3.1.1 16, section 3.2.1 17)
  ResourceFactory server (section 3.1.1 16, section 3.3.1 18)
  server (section 3.1.1 16, section 3.2.1 17, section 3.3.1 18, section 3.4.1 20)
ad:Controls
  ad:Control 13
  ad:ControlValue 14
  overview 13
ad:controls element 13
ad:Controls/ad:Control/@Criticality 15
ad:controls/ad:control/@criticality attribute 15
ad:Controls/ad:Control/@Type 15
ad:controls/ad:control/@type attribute 15
Applicability 11
Attribute groups 15
Attributes 14
  ad:controls/ad:control/@criticality 15
  ad:controls/ad:control/@type 15
  overview 14

C

Capability negotiation 11
Change tracking 38
Common data structures 15
Complex types 14
Create example 21
Create operation - ResourceFactory server 19

D

Data model - abstract
  enumeration server (section 3.1.1 16, section 3.4.1 20)
  resource server (section 3.1.1 16, section 3.2.1 17)
  ResourceFactory server (section 3.1.1 16, section 3.3.1 18)
  server (section 3.1.1 16, section 3.2.1 17, section 3.3.1 18, section 3.4.1 20)
Data structures - common 15
Delete example 23
Delete operation - resource server 18
Directory service schema elements 15

E

Elements
  ad:controls 13
    ad:Control 13
    ad:ControlValue 14
    overview 13
  overview 12
Elements - directory service schema 15
Enumeration server
  abstract data model (section 3.1.1 16, section 3.4.1 20)
  initialization (section 3.1.3 16, section 3.4.3 20)
  local events 20
  message processing
    overview 20
    Pull operation 20
  overview (section 3.1 16, section 3.4 19)
  sequencing rules
    overview 20
    Pull operation 20
  timer events 20
  timers (section 3.1.2 16, section 3.4.2 20)
Events
  local - server (section 3.2.6 18, section 3.3.6 19, section 3.4.6 20)
  timer - server (section 3.2.5 18, section 3.3.5 19, section 3.4.5 20)
Examples
  "Create" 21
  "Delete" 23
  "Get" 24
  "Pull" 25
  overview 21

F

Fields - vendor-extensible 11

G

Get example 24
Get operation - resource server 17
Glossary 7
Groups 15

I

Implementer - security considerations 28
Index of security parameters 28
Informative references 9
Initialization
  enumeration server (section 3.1.3 16, section 3.4.3 20)
  resource server (section 3.1.3 16, section 3.2.3 17)
  ResourceFactory server (section 3.1.3 16, section 3.3.3 19)

server ([section 3.1.2](#) 16, [section 3.2.2](#) 17, [section 3.3.2](#) 19, [section 3.4.2](#) 20)

V

W