[MS-UPSLDAP]: User Profile Synchronization (UPS): Lightweight Directory Access Protocol Version 3 Extensions

---

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, email 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.

Revision Summary

Date	Revision History	Revision Class	Comments
08/14/2009	0.1	Major	First Release.
09/25/2009	0.2	Minor	Updated the technical content.
11/06/2009	0.2.1	Editorial	Revised and edited the technical content.
12/18/2009	0.2.2	Editorial	Revised and edited the technical content.
01/29/2010	0.2.3	Editorial	Revised and edited the technical content.
03/12/2010	0.2.4	Editorial	Revised and edited the technical content.
04/23/2010	0.2.5	Editorial	Revised and edited the technical content.
06/04/2010	0.2.6	Editorial	Revised and edited the technical content.
07/16/2010	0.2.6	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	0.3	Minor	Clarified the meaning of the technical content.
10/08/2010	0.3	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	0.3	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	0.3	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	0.3	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	0.3.1	Editorial	Changed language and formatting in the technical content.
05/06/2011	0.3.2	Editorial	Changed language and formatting in the technical content.
06/17/2011	0.4	Minor	Clarified the meaning of the technical content.
09/23/2011	0.4	No change	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	0.4	No change	No changes to the meaning, language, or formatting of the technical content.
03/30/2012	1.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
07/12/2012	2.0	Major	Significantly changed the technical content.
09/12/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2013	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/30/2013	2.1	Minor	Clarified the meaning of the technical content.
11/18/2013	2.1	No change	No changes to the meaning, language, or formatting of the technical content.
02/10/2014	2.1	No change	No changes to the meaning, language, or formatting of the technical content.
04/30/2014	2.1	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction 6
1.1 Glossary 6
1.2 References 6
1.2.1 Normative References 7
1.2.2 Informative References 8
1.3 Overview 9
1.4 Relationship to Other Protocols 10
1.5 Prerequisites/Preconditions 11
1.6 Applicability Statement 11
1.7 Versioning and Capability Negotiation 11
1.8 Vendor-Extensible Fields 11
1.9 Standards Assignments 11

2 Messages 13
2.1 Transport 13
2.2 Message Syntax 13
2.2.1 LDAP Digest Authentication 14
2.2.2 LDAP Negotiated Authentication 14
2.2.3 LDAP Paged Search Control 15
2.2.4 LDAP Sort Controls 15
2.2.5 LDAP Virtual List View Control 15
2.2.6 LDAP DirSync Control 15
2.2.7 LDAP Show Deleted Control 15
2.2.8 LDAP Extended DN Control 16
2.2.9 LDAP Lazy Commit Control 16
2.3 Directory Service Schema Elements 16
2.3.1 Operational Schema Elements Defined in LDAPv3 16
2.3.1.1 Attribute DITContentRules Syntax 17
2.3.2 Operational Schema Elements Defined as Active Directory Schema 17
2.3.2.1 Active Directory Directory Server AttributeTypes Syntax 18
2.3.2.2 Active Directory Directory Server ExtendedAttributeInfo Syntax 18
2.3.3 Operational Schema Elements Defined for Extension Bundle D 18
2.3.3.1 Novell eDirectory Directory Server AttributeTypes Syntax 18
2.3.4 Operational Schema Elements Defined for Extension Bundle B 19
2.3.5 Operational Schema Elements Defined for Extension Bundle C 19
2.3.5.1 Sun iPlanet Directory Server ObjectClasses Attribute 20
2.3.6 Operational Schema Elements Defined for Extension Bundles B and C 20
2.3.7 Operational Schema Elements Defined for Extension Bundles B, C, and D 20

3 Protocol Details 21
3.1 Common Details 21
3.1.1 Abstract Data Model 21
3.1.2 Timers 21
3.1.3 Initialization 21
3.1.4 Higher-Layer Triggered Events 21
3.1.5 Message Processing Events and Sequencing Rules 21
3.1.5.1 LDAP Client Implementing Extension Bundle A Connects to AD DS or AD LDS 22
3.1.5.2 LDAP Client Implementing Extension Bundle A Imports from AD DS or AD LDS 23
3.1.5.3 LDAP Client Implementing Extension Bundle A Exports to AD DS or AD LDS 23

3.1.5.4 LDAP Client Retrieves LDAP Schema from Directory Server 24
3.1.5.5 LDAP Client Disconnects from Directory Server 24
3.1.5.6 LDAP Client Connects to Directory Server Implementing Extension Bundle C 24
3.1.5.7 LDAP Client Imports from Directory Server Implementing Extension Bundle C 25
3.1.5.8 LDAP Client Exports to Directory Server Implementing Extension Bundle C 25
3.1.5.9 LDAP Client Connects to Directory Server Implementing Extension Bundle D 25
3.1.5.10 LDAP Client Imports from Directory Server Implementing Extension Bundle D 26
3.1.5.11 LDAP Client Exports to Directory Server Implementing Extension Bundle D 26
3.1.5.12 LDAP Client Connects to Directory Server Implementing Extension Bundle B 26
3.1.5.13 LDAP Client Imports from Directory Server Implementing Extension Bundle B 27
3.1.5.14 LDAP Client Exports to Directory Server Implementing Extension Bundle B 28
3.1.5.15 LDAP Client Connects to Directory Server Implementing Extension Bundle B for a Paging Connection 28
3.1.6 Timer Events 28
3.1.7 Other Local Events 28
3.2 Server Details 29
3.2.1 Abstract Data Model 29
3.2.2 Timers 29
3.2.3 Initialization 29
3.2.4 Higher-Layer Triggered Events 29
3.2.5 Message processing Events and Sequencing Rules 29
3.2.5.1 Receiving a Connection Request 29
3.2.5.2 Receiving a Bind Request Message 29
3.2.5.3 Receiving a Search Request Message 30
3.2.6 Timer Events 31
3.2.7 Other Local Events 31
3.3 Client Details 31
3.3.1 Abstract Data Model 31
3.3.2 Timers 31
3.3.3 Initialization 32
3.3.4 Higher-Layer Triggered Events 32
3.3.5 Message Processing Events and Sequencing Rules 32
3.3.6 Timer Events 32
3.3.7 Other Local Events 32

4 Protocol Examples 33

5 Security 34
5.1 Security Considerations for Implementers 34
5.2 Index of Security Parameters 34

6 Appendix A: Product Behavior 35

7 Change Tracking 36

8 Index 37

1 Introduction

The User Profile Synchronization (UPS) Lightweight Directory Access Protocol Version 3 Extensions specifies the extensions to the **Lightweight Directory Access Protocol (LDAP)** [RFC2251] which are used in the communication sequences between a Synchronization Service and directory servers. A Synchronization Service maintains the consistency of a database of directory entries by importing and exporting changes to one or more directory servers.

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 RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

Active Directory
Active Directory Domain Services (AD DS)
attribute
attribute syntax
Augmented Backus-Naur Form (ABNF)
cipher suite
distinguished name (DN)
Generic Security Services (GSS)
GUID
Kerberos
Lightweight Directory Access Protocol (LDAP)
object class
object identifier (OID)
operational attribute
protocol data units (PDUs)
SASL
schema object
Secure Sockets Layer (SSL)

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

Active Directory Lightweight Directory Services (AD LDS)
change log
digest

The following terms are specific to this document:

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

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

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.

[IBM-DS52SCHEMA] IBM, "IBM Tivoli Directory Server Schema", http://publib.boulder.ibm.com/tividd/td/IBMDS/IDSschema52/en_US/HTML/schema.html

[IBM-RootDSELogInfo] IBM, "Change log information in the root DSE entry", http://publib.boulder.ibm.com/infocenter/zvm/v5r4/index.jsp?topic=/com.ibm.zvm.v54.kldl0/clroot.htm

[ID-LDAPCHANGELOG] Good, G., and Poitou, L., "Definition of an Object Class to Hold LDAP Change Records", IETF Internet Draft, March 2003, http://tools.ietf.org/html/draft-good-ldap-changelog-04

[ID-LDAPINCREMENTAL] Kashi, A., and Randall, R., "Incremental Retrieval of Multi-valued Properties", November 2001, http://www.ietf.org/proceedings/02mar/I-D/draft-kashi-incremental-00.txt

[ID-LDAPVLV] Boreham, D., Sermersheim, N., and Kashi, A., "LDAP Extensions for Scrolling View Browsing of Search Results", November 2002, http://tools.ietf.org/html/draft-ietf-ldapext-ldapv3-vlv-09

[ID-NDSSCHEMA] Sermersheim, J., "LDAP Schema for NDS", May 2002, http://tools.ietf.org/html/draft-sermersheim-nds-ldap-schema-03

[MS-ADA1] Microsoft Corporation, "Active Directory Schema Attributes A-L".

[MS-ADA2] Microsoft Corporation, "Active Directory Schema Attributes M".

[MS-ADA3] Microsoft Corporation, "Active Directory Schema Attributes N-Z".

[MS-ADLS] Microsoft Corporation, "Active Directory Lightweight Directory Services Schema".

[MS-ADSC] Microsoft Corporation, "Active Directory Schema Classes".

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

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

[MS-KILE] Microsoft Corporation, "Kerberos Protocol Extensions".

[MS-NLMP] Microsoft Corporation, "NT LAN Manager (NTLM) Authentication Protocol".

[MS-UPSCDS] Microsoft Corporation, "User Profile Synchronization (UPS): Configuration Data Structure".

[NOVELL-SCHEMAREF] Novell Inc., "Novell eDirectory Schema Reference", June 2008, http://developer.novell.com/documentation/ndslib/pdfdoc/schm_enu/schm_enu.pdf

[RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", RFC 1964, June 1996, http://www.ietf.org/rfc/rfc1964.txt

[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

[RFC2222] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997, http://www.ietf.org/rfc/rfc2222.txt

[RFC2246] Dierks, T., and Allen, C., "The TLS Protocol Version 1.0", RFC 2246, January 1999, http://www.ietf.org/rfc/rfc2246.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

[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

[RFC2829] Wahl, M., Alvestrand, H., Hodges, J., and Morgan, R., "Authentication Methods for LDAP", RFC 2829, May 2000, http://www.ietf.org/rfc/rfc2829.txt

[RFC2831] Leach, P., and Newman, C., "Using Digest Authentication as a SASL Mechanism", RFC 2831, May 2000, http://www.ietf.org/rfc/rfc2831.txt

[RFC2891] Howes, T., Wahl, M., and Anantha, A., "LDAP Control Extension for Server Side Sorting of Search Results", RFC 2891, August 2000, http://www.ietf.org/rfc/rfc2891.txt

[RFC4178] Zhu, L., Leach, P., Jaganathan, K., and Ingersoll, W., "The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism", RFC 4178, October 2005, http://www.ietf.org/rfc/rfc4178.txt

[RFC4511] Sermersheim, J., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, June 2006, http://www.rfc-editor.org/rfc/rfc4511.txt

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981, http://www.ietf.org/rfc/rfc0793.txt

[SSL3] Netscape, "SSL 3.0 Specification", http://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00

[SUN-DS51Admin] Sun Microsystems, "Administrator's Guide, iPlanet Directory Server", version 5.1, February 2002, http://docs.oracle.com/cd/E19627-01/816-2670/816-2670.pdf

[SUN-DS51REF] Sun Microsystems, "Schema Reference iPlanet Directory Server Version 5.1", February 2002, http://docs.oracle.com/cd/E19627-01/816-2673/816-2673.pdf

[SUN-DS52RefMan] Sun Microsystems, "Reference Manual, Sun(TM) ONE Directory Server", version 5.2, June 2003, http://docs.oracle.com/cd/E19850-01/816-6699-10/816-6699-10.pdf

1.2.2 Informative References

[IANA-GSSAPI] IANA, "Generic Security Service Application Program Interface (GSSAPI)/Kerberos/Simple Authentication and Security Layer (SASL) Service Names", March 2009, http://www.iana.org/assignments/gssapi-service-names

[IANA-LDAP] IANA, "Lightweight Directory Access Protocol (LDAP) Parameters", April 2009, http://www.iana.org/assignments/ldap-parameters

[IANAPORT] IANA, "Port Numbers", November 2006, http://www.iana.org/assignments/port-numbers

[IANA-SASL] IANA, "Simple Authentication and Security Layer (SASL) Mechanisms", September 2012, http://www.iana.org/assignments/sasl-mechanisms

[MSDN-SCHANNELCIPHER] Microsoft Corporation, "Cipher Suites in Schannel", http://msdn.microsoft.com/en-us/library/aa374757(VS.85).aspx

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

[RFC4517] Legg, S., et al., "Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules", June 2006, http://www.rfc-editor.org/rfc/rfc4517.txt

1.3 Overview

This specification identifies the extensions to the Lightweight Directory Access Protocol (LDAP) which are used for synchronization of directory contents between a Synchronization Service and one or more directory servers. This specification groups the LDAP extensions into four bundles: extension bundle A, extension bundle B, extension bundle C and extension bundle D.

This specification covers the behavior of these extensions which the Synchronization Service implements the client role in the LDAP protocol. The server role in the LDAP protocol is performed by one or more of the following directory servers:

- **Active Directory Domain Services (AD DS)**
- **Active Directory Lightweight Directory Services (AD LDS)**
- A directory server

Within the Synchronization Service, the LDAP extensions are implemented by the following software components, each of which are LDAP clients:

- AD Management Agent (MA), which implements the LDAP extensions in extension bundle A
- IBM DS Management Agent (MA), which implements the LDAP extensions in extension bundle B
- iPlanet Management Agent (MA), which implements the LDAP extensions in extension bundle C
- eDirectory Management Agent (MA), which implements the LDAP extensions in extension bundle D

The AD MA establishes and manages LDAP connections to AD DS and AD LDS. The IBM DS MA, iPlanet MA, and eDirectory MA each establish and manage LDAP connections to directory servers that implement extension bundles B, C, or D, respectively.

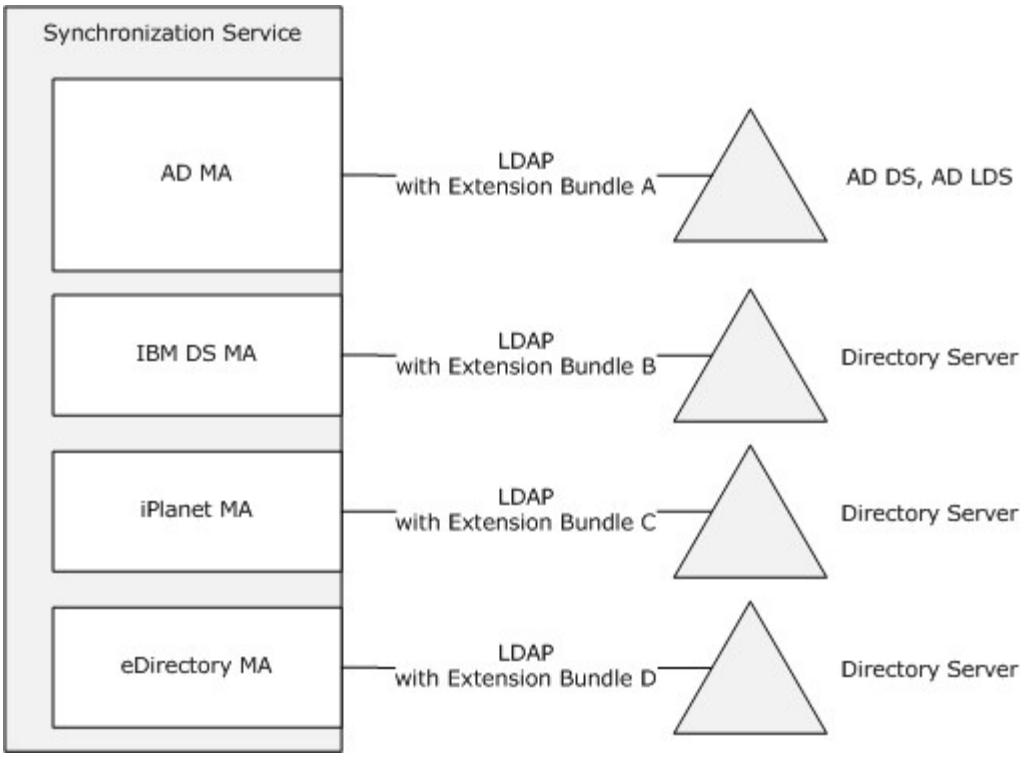

Figure 1: Architecture.

1.4 Relationship to Other Protocols

The LDAP extensions specified in this document are transported as schema, controls, and mechanisms within the LDAP protocol as defined in [RFC2251]. The LDAP protocol is transported atop one or more of TCP [RFC793], **Secure Sockets Layer (SSL)** [SSL3] atop TCP, a **SASL Generic Security Services (GSS)** wrapper mechanism [RFC2222] atop TCP, or a SASL wrapper mechanism inside of SSL.

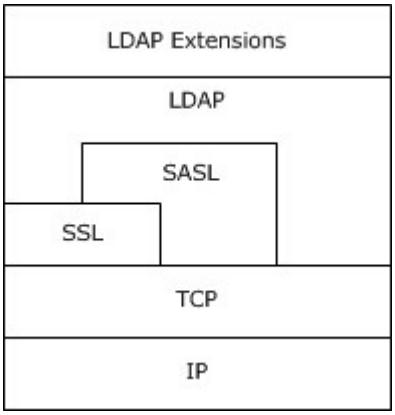

Figure 2: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

The following preconditions are expected to exist prior to protocol interactions between the Synchronization Service and connected directories.

- If SSL is to be used, then the operating system on which the Synchronization Service is installed is expected to have the SSL certificates available to enable the Synchronization Service to authenticate the connected directory.
- If **Kerberos** is to be used, then Kerberos trust relationships are expected to have been established between the operating system on which the Synchronization Service is installed and the system on which the connected directory is installed.
- An account is expected to be present in a directory server, the account representing the Synchronization Service, with the same password credential stored in the directory server and in the Synchronization Service configuration, for the Synchronization Service to be able to authenticate to that directory server.

1.6 Applicability Statement

This specification is applicable to a Synchronization Service in its communication to one or more directory servers. There are no higher-level protocols layered above these extensions.

1.7 Versioning and Capability Negotiation

There is no versioning in this specification.

The LDAP client performs capability negotiation by retrieving **attributes (2)** of one or more of the Directory-Server Specific Entries (DSEs), root DSE and monitor DSE, of the connected directory server in a baseObject Search request. The root DSE has a zero-length distinguished name (DN) (4); the monitor has **distinguished name (DN) (4)** of the literal string "cn=monitor".

The following attributes are retrieved by the LDAP client for version negotiation, which are discussed further in section 2:

- Retrieved in extension bundle A: root DSE attributes "forestFunctionality" and "supportedCapabilities", which are defined in [MS-ADTS].
- Retrieved in extension bundle B: root DSE attribute "vendorVersion", which is defined in [IBM-DS52SCHEMA].
- Retrieved in extension bundle C: monitor DSE (cn=monitor) attribute "version", which is defined in [SUN-DS52RefMan].
- Retrieved in extension bundle D: root DSE attribute "vendorVersion", which is defined in [NOVELL-SCHEMAREF].

1.8 Vendor-Extensible Fields

This specification does not provide any vendor-extensible fields.

1.9 Standards Assignments

The TCP port number assignments for LDAP (389) and for LDAP over SSL (636) have been registered at IANA by a third-party organization.

The LDAP Object Identifiers for controls have been registered by [RFC2696] and [RFC2891].

The LDAP Bind Authentication methods have been registered by [RFC4511].

The LDAP Syntaxes have been registered by [RFC4517].

The values of parameters in lower-layer protocols are set as defined in the following table.

Parameter	Value	Reference
Port Number (ldap)	389	[IANAPORT]
Port Number (ldaps)	636	[IANAPORT]
LDAP parameters	See [IANA-LDAP] for parameter values	[IANA-LDAP]
GSSAPI Service Name for LDAP	"ldap"	[IANA-GSSAPI]
SASL mechanisms	"GSSAPI", "GSS-SPNEGO", "DIGEST-MD5"	[IANA-SASL]

2 Messages

2.1 Transport

There are four transport options that are implemented by the LDAP clients as MAs in the Synchronization Service:

- LDAP protocol data units (PDUs) are sent directly atop TCP [RFC793],
- LDAP PDUs are encrypted according to SSL [SSL3] and the encrypted data is transported over TCP,
- LDAP PDUs are wrapped according to GSSAPI and SASL [RFC2222], and the wrapped data is transported over TCP, or
- LDAP PDUs are wrapped according to GSSAPI and SASL, the wrapped data is encrypted according to SSL, and the encrypted data is transported over TCP.

As specified in [MS-ADTS] section 3.1.1.3.1.10, AD LDS implements LDAP atop TCP. A directory server and a LDAP client implementing one or more of the extension bundles MUST implement LDAP atop TCP.

In extension bundle A, LDAP communication with **Active Directory** and AD LDS is performed via the LDAP PDUs either being encrypted within SSL or being sent directly atop TCP until the completion of SASL negotiation. If GSS-SPNEGO has been negotiated with signing and/or sealing, subsequent interactions of LDAP PDUs are wrapped according to GSSAPI and SASL. The SASL parameters are discussed further in sections 2.2.1 and 2.2.2 of this specification. As specified in [MS-ADTS] section 3.1.1.3.4.5, Active Directory Domain Services (AD DS) implements LDAP wrapped according to GSSAPI.

In extension bundles B, C and D, communication with a directory server is performed via either LDAP PDUs sent directly atop TCP, or via LDAP PDUs encrypted according to SSL, depending on the configuration of the LDAP client. An LDAP client that implements extension bundle B, C or D MUST implement LDAP over SSL with all of the **cipher suites** listed in the following table (for more information, see those cipher suites in the list of cipher suites for SSL [MSDN-SCHANNELCIPHER] where the Protocols category is "SSL 3.0"):

Cipher suite	Exchange	Encryption	Hash
TLS_RSA_WITH_RC4_128_SHA	RSA	RC4	SHA1
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES	SHA1
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DH	3DES	SHA1
TLS_RSA_WITH_RC4_128_MD5	RSA	RC4	MD5

A directory server that implements extension bundle B, C or D MUST implement LDAP over SSL 3.0 with one or more of the cipher suites listed in the preceding table.

2.2 Message Syntax

The LDAP message syntax which carries the extensions is documented in [RFC2251].

As specified in [MS-ADTS] section 3.1.1.3.1, AD DS implements the LDAP operations Bind, Search, Add, Delete, Modify, ModifyDN and Unbind. A directory server that implements extension bundle A, B, C or D MUST implement the LDAP operations Bind, Search, Add, Delete, Modify, ModifyDN and Unbind.

The SSL message syntax is documented in [SSL3].

The SASL message syntax, used following negotiation of a SASL mechanism in LDAP, is documented in [RFC2222] section 3.

The following table summarizes the LDAP extension elements (authentication mechanisms and controls) relevant for each extension bundle.

Section	Element	Extension Bundle A	Extension Bundle B	Extension Bundle C	Extension Bundle D
2.2.1	LDAP Digest Authentication		*	*	*
2.2.2	LDAP Negotiated Authentication	*			
2.2.3	LDAP Paged Search Control	*	*		
2.2.4	LDAP Sort Controls				*
2.2.5	LDAP Virtual List View Control				*
2.2.6	LDAP DirSync Control	*			
2.2.7	LDAP Show Deleted Control	*			
2.2.8	LDAP Extended DN Control	*			
2.2.9	LDAP Lazy Commit Control	*			

2.2.1 LDAP Digest Authentication

An LDAP client implementing extension bundle B, C or D will either use simple authentication in an LDAP Bind request, or will request the DIGEST-MD5 SASL security mechanism in an LDAP Bind request, depending on the configuration of the LDAP client.

The usage of **digest** authentication with LDAP and the SASL mechanism "DIGEST-MD5" are documented in [RFC2829] section 6.1, and in [RFC2831].

A directory server implementing extension bundle B, C or D MAY implement SASL and the SASL mechanism "DIGEST-MD5".

2.2.2 LDAP Negotiated Authentication

When communicating with AD DS or AD LDS directory servers, an LDAP client implementing extension bundle A will request the GSS-SPNEGO SASL security mechanism in an LDAP Bind request, with requests for signing and encryption of subsequent communications on this connection.

This mechanism is documented in [RFC4178]. AD DS supports Kerberos (see [MS-KILE] and RFC 1964 [RFC1964]) and NTLM (see [MS-NLMP]) when using GSS-SPNEGO.

2.2.3 LDAP Paged Search Control

The LDAP paged search control, with **object identifier (OID)** (3) 1.2.840.113556.1.4.319, is used with an LDAP Search operation to permit clients to perform searches that return more objects than a server-defined limit, by splitting the search into multiple searches.

This extension is defined in the Active Directory Technical Specification [MS-ADTS] section 3.1.1.3.4.1.1 ("LDAP_PAGED_RESULT_OID_STRING") for AD DS, and is documented in [RFC2696].

A directory server implementing extension bundle B MUST implement the LDAP paged search control.

[RFC2696] section 3 permits a directory server to set the value of the "size" protocol element in a search result done message to "0" if the total size cannot be provided. A directory server MAY set the value of the "size" protocol element to "0".

2.2.4 LDAP Sort Controls

The LDAP sort request control, with OID 1.2.840.113556.1.4.473, and its corresponding sort response control, with OID 1.2.840.113556.1.4.474, are documented in [RFC2891].

A directory server implementing extension bundle D MUST implement receiving the LDAP sort request control and returning the LDAP sort response control.

2.2.5 LDAP Virtual List View Control

The LDAP virtual list view (VLV) request control, with OID 2.16.840.1.113730.3.4.9, is used with an LDAP Search operation to retrieve a subset of the objects that satisfy the search request.

A directory server implementing extension bundle D MUST implement receiving the LDAP virtual list view request control and returning the LDAP virtual list view response control.

2.2.6 LDAP DirSync Control

The presence of this LDAP control, with object identifier (OID) (3) 1.2.840.113556.1.4.841, in an LDAP Search request instructs AD DS to retrieve the changes made to objects since a previous search with this control OID was performed.

This extension is defined for AD DS in the Active Directory Technical Specification [MS-ADTS] section 3.1.1.3.4.1.3 ("LDAP_SERVER_DIRSYNC_OID").

2.2.7 LDAP Show Deleted Control

The presence of this LDAP control, with OID 1.2.840.113556.1.4.417, in an LDAP Search request instructs AD DS to specify that tombstones and deleted objects are to be visible to the client.

This extension is defined for AD DS in the Active Directory Technical Specification [MS-ADTS] section 3.1.1.3.4.1.14 ("LDAP_SERVER_SHOW_DELETED_OID").

2.2.8 LDAP Extended DN Control

The presence of this LDAP control, with OID 1.2.840.113556.1.4.529, in an LDAP Search request instructs AD DS to return each distinguished name (DN) (4) in an extended format containing the values of the "objectGUID" and "objectSid" attributes.

This extension is defined for AD DS in the Active Directory Technical Specification [MS-ADTS] section 3.1.1.3.4.1.5 ("LDAP_SERVER_EXTENDED_DN_OID").

2.2.9 LDAP Lazy Commit Control

The presence of this LDAP control, with OID 1.2.840.113556.1.4.619, instructs AD DS that it MAY sacrifice durability guarantees on updates to improve performance.

This extension is defined for AD DS in the Active Directory Technical Specification [MS-ADTS] section 3.1.1.3.4.1.7 ("LDAP_SERVER_LAZY_COMMIT_OID").

2.3 Directory Service Schema Elements

The Synchronization Service has an extensible schema model to support the administrator-defined schema which is present in each connected directory server.

An LDAP client implementing one or more of the extension bundles defined in this document accesses the following Directory Service schema classes and attributes listed in the following table(s) which control the protocol interactions described in this document.

2.3.1 Operational Schema Elements Defined in LDAPv3

As specified in [MS-ADTS] section 3.1.1.3.1.1, AD DS implements the **operational attributes** **namingContexts**, **subschemaSubentry**, **attributeTypes**, **objectClasses**, **objectClass**, **ditContentRules**.

A directory server implementing extension bundle B, C or D MUST implement the operational attributes **namingContexts**, **subschemaSubentry**, **attributeTypes**, **objectClasses**, **objectClass**.

The following table lists operational attributes defined in [RFC2251] and the locations in the directory information tree in which they can occur. For the syntactic specifications of the attributes in this table for a directory server implementing extension bundle B, C or D, refer to [RFC2252].

Location	Attributes
root DSE	namingContexts, supportedControl, supportedSASLMechanisms, subschemaSubentry
subschema entry	attributeTypes, objectClasses, matchingRules, matchingRuleUse, dITContentRules
any entry with object class top	objectClass, createTimestamp, modifyTimestamp, creatorsName, modifiersName

2.3.1.1 Attribute DITContentRules Syntax

The Synchronization Service LDAP client parses the attribute **dITContentRules** according to the following **Augmented Backus-Naur Form (ABNF)** definition for the **attribute syntax**, defined in [RFC2252] section 6.11.

```
DITContentRuleDescription =
     "(" whsp
     numericoid whsp     ; Structural ObjectClass identifier
     [ "NAME" qdescrs ]
     [ "DESC" qdstring ]
     [ "OBSOLETE" whsp ]
     [ "AUX" oids ]      ; Auxiliary ObjectClasses
     [ "MUST" oids ]     ; AttributeType identifiers
     [ "MAY" oids ]      ; AttributeType identifiers
     [ "NOT" oids ]      ; AttributeType identifiers
     ")"
```

The productions "numericoid", "qdescrs", "qdstring", "whsp" and "oids" are defined in [RFC2252].

2.3.2 Operational Schema Elements Defined as Active Directory Schema

AD DS and AD LDS implements the schema elements described in this section.

The following table lists operational attributes implemented by AD DS or AD LDS and the locations in the directory information tree in which they can occur. For the syntactic specifications of the attributes in this table, refer either to:

- Active Directory Domain Services (AD DS) ([MS-ADA1], [MS-ADA2], [MS-ADA3], and [MS-ADSC]).

or to:

- Active Directory Lightweight Directory Services (AD LDS) [MS-ADLS].

Location	Attributes
root DSE	forestFunctionality, supportedCapabilities
subschema entry	attributeTypes, extendedAttributeInfo
the "CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,<Domain>" object	dsHeuristics
any entry with object class top	USNChanged, whenCreated, name, objectGUID, parentGUID, isDeleted, proxiedObjectName

2.3.2.1 Active Directory Directory Server AttributeTypes Syntax

The Synchronization Service permits the identifiers of the attribute and of the syntax in a value of the **attributeTypes** attribute to be an alphanumeric identifier, and does not require these identifiers to be numericoid format identifiers.

2.3.2.2 Active Directory Directory Server ExtendedAttributeInfo Syntax

The Synchronization Service LDAP client parses the values of the attribute **extendedAttributeInfo**, defined in [MS-ADA1] section 2.224, according to the following BNF:

```
        extendedAttributeInfo =
            "(" whsp
            numericoid whsp                 ; AttributeType identifier
            [ "NAME" qdescrs ]              ; name used in AttributeType
            [ "RANGE-LOWER" qdstring ]      ; Numeric lower bound
            [ "RANGE-UPPER" qdstring ]      ; Numeric upper bound
            [ others ]
            ")"
    others = 1*utf8
```

The productions "whsp", "numericoid", "qdescrs", "qdstring" and "utf8" are defined in [RFC2252].

2.3.3 Operational Schema Elements Defined for Extension Bundle D

A directory server implementing extension bundle D MUST implement the schema elements specified in this section.

For the syntactic specifications of the following <Class> or <Class><Attribute> pairs, refer to the documents [ID-NDSSCHEMA] and [NOVELL-SCHEMAREF]:

Class	Attribute
N/A (root DSE)	vendorVersion, directoryTreeName
N/A (subschema entry)	attributeTypes
N/A	GUID, subordinateCount, equivalentToMe, groupMembership, securityEquals, passwordExpirationInterval, passwordExpirationTime, loginGraceLimit, loginGraceRemaining
sASSecurity, treeRoot	All

2.3.3.1 Novell eDirectory Directory Server AttributeTypes Syntax

In contrast to the syntax of the **AttributeTypes** attribute defined in [RFC2252] section 4.2, the syntax of this attribute as implemented by a directory server or LDAP client implementing extension

bundle D is defined in section 4.4 of the document [ID-NDSSCHEMA]. This BNF includes the extension "X-NDS_LOWER_BOUND".

```
AttributeTypeDescription =
    "(" whsp
    numericoid whsp                  ; AttributeType identifier
    [ "NAME" qdescrs ]               ; name used in AttributeType
    [ "DESC" qdstring ]              ; description
    [ "OBSOLETE" whsp ]
    [ "SUP" woid ]                   ; derived from other Attribute
    [ "EQUALITY" woid ]              ; Matching Rule name
    [ "ORDERING" woid ]              ; Matching Rule name
    [ "SUBSTR" woid ]                ; Matching Rule name
    [ "SYNTAX" whsp noidlen whsp ]   ;
    [ "SINGLE-VALUE" whsp ]          ; default multi-valued
    [ "COLLECTIVE" whsp ]            ; default not collective
    [ "NO-USER-MODIFICATION" whsp ]  ; default user modifiable
    [ "USAGE" whsp AttributeUsage ]  ; default userApplications
    [ "X-NDS_LOWER_BOUND" qdstrings ] ; lower bound. default
                                     ; ('0')(upper is specified in
                                     ; SYNTAX)
    whsp ")"
```

The productions "whsp", "numericoid", "qdescrs", "qdstring", "woid", "noidlen", "AttributeUsage" and "qdstrings" are defined in [RFC2252].

2.3.4 Operational Schema Elements Defined for Extension Bundle B

A directory server implementing extension bundle B MUST implement the schema elements specified in this section.

For the syntactic specifications of the following <Class> or <Class><Attribute> pairs, refer to [IBM-DS52SCHEMA].

Class	Attribute
N/A (root DSE)	lastChangeNumber, firstChangeNumber, vendorVersion, ibm-sasldigestrealmname
N/A	ibm-entryUuid
ibm-realm	All

2.3.5 Operational Schema Elements Defined for Extension Bundle C

A directory server implementing extension bundle C MUST implement the schema elements specified in this section.

The following table lists operational attributes and the locations in the directory information tree in which they can occur. For the syntactic specifications of the attributes in this table, see the associated references.

Location	Attributes	References
root DSE	lastChangeNumber, firstChangeNumber	[IBM-RootDSELogInfo]
cn=config	nsslapd-sizelimit	[SUN-DS52RefMan]
cn=monitor	version	[SUN-DS52RefMan]
subschema entry	attributeTypes, objectClasses	[SUN-DS51REF]
any entry of object class top	nsUniqueId	[SUN-DS51Admin]

2.3.5.1 Sun iPlanet Directory Server ObjectClasses Attribute

In contrast to the definition of the ObjectClass identifier in the ABNF production of ObjectClassDescription in [RFC2252] section 4.4, the Synchronization Service LDAP client permits the **object class (2)** identifier in a value of the **objectClasses** attribute to be an alphanumeric string identifier, and does not require it to be a numeric identifier.

2.3.6 Operational Schema Elements Defined for Extension Bundles B and C

A directory server implementing extension bundle B or extension bundle C MUST implement the **change log** schema specified in this section.

For the syntactic specifications of the following `<Class>` or `<Class><Attribute>` pairs, refer to [ID-LDAPCHANGELOG].

Class	Attribute
(N/A, root DSE)	changelog
changeLogEntry	All

2.3.7 Operational Schema Elements Defined for Extension Bundles B, C, and D

A directory server implementing extension bundle B, extension bundle C, or extension bundle D MUST recognize the attributes **isDefunct**, defined in [MS-ADA1] section 2.335, and **ldapDisplayName**, defined in [MS-ADA1] section 2.356, in searches of the subschema entry.

Class	Attribute
(N/A, subschema entry)	isDefunct, ldapDisplayName

3 Protocol Details

3.1 Common Details

This section specifies details that are common to both client and server behavior.

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.

The abstract data model of LDAP is defined in [RFC2251] and [RFC2252].

The document [ID-LDAPCHANGELOG] defines an extension to the LDAP data model, to represent the change log.

Additional schema requirements are specified in section 2.1.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

A run profile directs an MA of the Synchronization Service to connect to a directory server and synchronize its copy of directory information with the connected directory server.

3.1.5 Message Processing Events and Sequencing Rules

The message sequencing rules for the LDAP messages themselves in a single request-response interaction are as defined in [RFC2251].

A run profile import step or export step specifies one of the following patterns of tasks, the selection of which is made in the configuration of the Synchronization Service. Each task specifies a sequence of message exchanges between an LDAP client, such as an MA of the Synchronization Service, and a directory server.

For extension bundle A, the patterns are:

- The sequence of: LDAP client implementing extension bundle A connects to AD DS (section 3.1.5.1), LDAP client implementing extension bundle A imports from AD DS (section 3.1.5.2), LDAP client disconnects from directory server (section 3.1.5.5)
- The sequence of: LDAP client implementing extension bundle A connects to AD DS (section 3.1.5.1), LDAP client implementing extension bundle A exports to AD DS (section 3.1.5.3), LDAP client disconnects from directory server (section 3.1.5.5)

- The sequence of: LDAP client implementing extension bundle A connects to AD DS (section 3.1.5.1), LDAP client retrieves LDAP schema from directory server (section 3.1.5.4), LDAP client disconnects from directory server (section 3.1.5.5)

For extension bundle B, the patterns are:

- The sequence of: LDAP client connects to directory server implementing extension bundle B (section 3.1.5.12), LDAP client imports from directory server implementing extension bundle B (section 3.1.5.13), LDAP client disconnects from directory server (section 3.1.5.5)
- The sequence of: LDAP client connects to directory server implementing extension bundle B (section 3.1.5.12), LDAP client exports to directory server implementing extension bundle B (section 3.1.5.14), LDAP client disconnects from directory server (section 3.1.5.5)
- The sequence of: LDAP client connects to directory server implementing extension bundle B (section 3.1.5.12), LDAP client retrieves LDAP schema from directory server (section 3.1.5.4), LDAP client disconnects from directory server (section 3.1.5.5)

For extension bundle C, the patterns are:

- The sequence of: LDAP client connects to directory server implementing extension bundle C (section 3.1.5.6), LDAP client imports from directory server implementing extension bundle C (section 3.1.5.7), LDAP client disconnects from directory server (section 3.1.5.5)
- The sequence of: LDAP client connects to directory server implementing extension bundle C (section 3.1.5.6), LDAP client exports to directory server implementing extension bundle C (section 3.1.5.8), LDAP client disconnects from directory server (section 3.1.5.5)
- The sequence of: LDAP client connects to directory server implementing extension bundle C (section 3.1.5.6), LDAP client retrieves LDAP schema from directory server (section 3.1.5.4), LDAP client disconnects from directory server (section 3.1.5.5)

For extension bundle D, the patterns are:

- The sequence of: LDAP client connects to directory server implementing extension bundle D (section 3.1.5.9), LDAP client imports from directory server implementing extension bundle D (section 3.1.5.10), LDAP client disconnects from directory server (section 3.1.5.5)
- The sequence of: LDAP client connects to directory server implementing extension bundle D (section 3.1.5.9), LDAP client exports to directory server implementing extension bundle D (section 3.1.5.11), LDAP client disconnects from directory server (section 3.1.5.5)
- The sequence of: LDAP client connects to directory server implementing extension bundle D (section 3.1.5.9), LDAP client retrieves LDAP schema from directory server (section 3.1.5.4), LDAP client disconnects from directory server (section 3.1.5.5)

3.1.5.1 LDAP Client Implementing Extension Bundle A Connects to AD DS or AD LDS

The task for an LDAP client implementing extension bundle A to connect to AD DS or AD LDS comprises the following sequence of protocol exchanges:

1. The LDAP client establishes a TCP connection to the directory server (AD DS or AD LDS).
2. If specified by configuration, the LDAP client negotiates SSL, as described in section 2.1 of this document.

3. If the LDAP client has not communicated with the directory server in a previous interaction within a predetermined time interval, the LDAP client requests a baseObject Search of the root DSE, requesting that the **supportedSASLMechanisms** attribute be returned.<1>

4. The LDAP client binds using a SASL mechanism, as described in section 2.2.2.

5. The LDAP client requests a baseObject Search of the root DSE, requesting that the attribute **forestFunctionality** be returned.

6. The LDAP client validates that AD DS returns a value of the root DSE attribute **forestFunctionality**, defined in [MS-ADTS] section 3.1.1.3.2.27.

7. The LDAP client requests a baseObject Search of the root DSE, requesting that the attribute **supportedCapabilities** be returned.

8. The LDAP client validates that one of the values of the **supportedCapabilities** attribute returned by the directory server (AD DS or AD LDS) is a string containing either the OID "1.2.840.113556.1.4.800" or the OID "1.2.840.113556.1.4.1851" as described in [MS-ADTS] section 3.1.1.3.4.3. Specifically, if the LDAP client was configured to communicate with AD LDS, and the value contains the OID "1.2.840.113556.1.4.800", or if the LDAP client was configured to communicate with AD DS, and the value contains the OID "1.2.840.113556.1.4.1851", then the LDAP client will close the connection and terminate the run-profile step.

3.1.5.2 LDAP Client Implementing Extension Bundle A Imports from AD DS or AD LDS

The task for an LDAP client implementing extension bundle A to import directory entries from AD DS or AD LDS comprises the following sequence of protocol exchanges:

- The LDAP client submits one or more Search requests to the directory server (AD DS or AD LDS). Each search request can be either a search request to retrieve results in bulk, or a search request to retrieve change history.

The Search requests have the following characteristics:

- The Search requests to retrieve results in bulk contain the paged search control, described in section 2.2.3 of this document.
- The Search requests to retrieve change history contain the DirSync control described in section 2.2.6 of this document.
- The Search requests contain the show deleted control, described in section 2.2.7 of this document, and the extended distinguished name (DN) (4) control, described in section 2.2.8 of this document.
- Search requests specify the following operational attributes to be returned, if present: **objectGUID** (specified in [MS-ADTS] section 3.1.1.1.3), **whenCreated**, and **proxiedObjectName** (specified in [MS-ADTS] section 3.1.1.5.4.2.3).

3.1.5.3 LDAP Client Implementing Extension Bundle A Exports to AD DS or AD LDS

The task for an LDAP client implementing extension bundle A to export directory entries to the directory server (AD DS or AD LDS) comprises the following sequence of protocol exchanges:

- The LDAP client submits one or more Add, Delete, Modify and ModifyDN requests, each with the lazy export control, defined in the Active Directory Technical Specification [MS-ADTS] section 3.1.1.3.4.1.7 ("LDAP_SERVER_LAZY_COMMIT_OID"), present on each request.

3.1.5.4 LDAP Client Retrieves LDAP Schema from Directory Server

The task for the LDAP client to retrieve LDAP schema from the directory servers comprises the following sequence of protocol exchanges:

1. The LDAP client requests the **subschemaSubentry** attribute of the root DSE.
2. The LDAP client requests one or more baseObject searches of the subschema entry, a **schema object**.
   - To retrieve extended attribute information, the LDAP client requests a search which specifies that the attribute **extendedAttributeInfo** be returned.
   - To retrieve DIT content rules, the LDAP client requests a search which specifies that the attribute **dITContentRules** be returned.
   - To retrieve the list of attribute types, the LDAP client requests a search which specifies that the attribute **attributeTypes** be returned.
   - To retrieve the list of object classes (2), the LDAP client requests a search which specifies that the attribute **objectClasses** be returned.
3. The LDAP client requests a subtree search of the subschema entry with filter "(**isDefunct**=TRUE)" requesting that the attribute **ldapDisplayName** be returned.

The behavior of AD DS and of other directory servers implementing extension bundle A, B, C or D can be different from that described in [RFC2252], as follows:

- AD DS and other directory servers can specify an alphanumeric identifier as the object class (2) identifier in a value of the attribute **objectClasses** of the subschema entry,
- AD DS and other directory servers can specify an alphanumeric identifier as the attribute type identifier in a value of the attribute **attributeTypes** of the subschema entry,
- AD DS and other directory servers can specify an alphanumeric identifier as the syntax identifier in a value of the attribute **attributeTypes** of the subschema entry, and
- A directory server implementing extension bundle D can include an "X-NDS_LOWER_BOUND" token extension in a value of the attribute "attributeTypes" of the subschema entry.

3.1.5.5 LDAP Client Disconnects from Directory Server

On each connection that the LDAP client established to the directory server, the LDAP client sends an LDAP Unbind request to the directory server and closes the TCP connection.

3.1.5.6 LDAP Client Connects to Directory Server Implementing Extension Bundle C

The task for an LDAP client to connect to a directory server implementing extension bundle C comprises the following sequence of protocol exchanges:

1. The LDAP client establishes a TCP connection to the directory server.

2. If configured, the LDAP client negotiates SSL, as described in section 2.1 of this document.

3. The LDAP client requests to bind to the directory server using either the simple authentication method or using the DIGEST-MD5 authentication method as described in section 2.2.1 of this document.

4. The LDAP client requests a **baseObject** Search of the monitor DSE "cn=monitor" requesting the attribute **version**.

5. The LDAP client requests a **baseObject** Search of the root DSE requesting the attribute **changelog**.

6. The LDAP client requests a **baseObject** Search of the **configuration DSE** "cn=config" requesting the attribute **nsslapd-sizelimit**.

3.1.5.7 LDAP Client Imports from Directory Server Implementing Extension Bundle C

The task for an LDAP client to import directory entries from a directory server implementing extension bundle C comprises the following protocol exchanges:

- The LDAP client submits one or more **Search** requests. Each **Search** request is either a request to retrieve the state of the change log, a request to retrieve entries from the change log, or a request to locate an entry in a naming context other than the change log by the entry's globally unique identifier.
  - **Search** requests to retrieve the state of the change log are searches of the root DSE which have scope **baseObject**, and request the attributes **changelog**, **lastchangenumber** or **firstchangenumber**.
  - Search requests to retrieve entries from the change log are searches of the change log container which have scope singleLevel.
  - Search requests to locate an entry by its **GUID** (as specified in [MS-DTYP] section 2.3.2) have either scope **baseObject** or scope **wholeSubtree** and a filter that is an equality match of the **nsUniqueId** attribute.

3.1.5.8 LDAP Client Exports to Directory Server Implementing Extension Bundle C

The task for an LDAP client to export directory entries to a directory server implementing extension bundle C comprises the following protocol exchanges:

- The LDAP client submits one or more **Add**, **Delete**, **Modify**, **ModifyDN** and **Search** requests.
  - Search requests to retrieve the GUID attribute value of particular referenced entries have scope **baseObject** and request the attribute **nsUniqueId** be returned.
  - Search requests to locate an entry by its GUID (as specified in [MS-DTYP] section 2.3.2) have scope **baseObject** or scope **wholeSubtree** and a filter that is an equality match of the **nsUniqueId** attribute.

3.1.5.9 LDAP Client Connects to Directory Server Implementing Extension Bundle D

The task for an LDAP client to connect to a directory server implementing extension bundle D comprises the following sequence of protocol exchanges:

1. The LDAP client establishes a TCP connection to the directory server.

2. If configured, the LDAP client negotiates SSL, as described in section 2.1 of this document.

3. The LDAP client requests to bind to the directory server using either the simple authentication method or the DIGEST-MD5 authentication method as described in section 2.2.1 of this document.

4. The LDAP client requests a **baseObject** Search of the root DSE requesting the attribute **vendorVersion**.

3.1.5.10 LDAP Client Imports from Directory Server Implementing Extension Bundle D

The task for an LDAP client to import directory entries from a directory server implementing extension bundle D comprises the following protocol exchanges:

- The LDAP client submits one or more Search requests. Each search request can be either a request to retrieve entries in the subtree, or a request to retrieve the GUID attribute value of entries.
  - Search requests to retrieve the entries in the subtrees of naming contexts have scope **singleLevel** or **wholeSubtree**, the filter "(GUID=*)", and if configured, the LDAP client includes in these requests the virtual list view control described in section 2.2.5 of this document and the sort control described in section 2.2.4 of this document.
  - **Search** requests to retrieve the **GUID** attribute value of particular referenced entries have scope **baseObject** and request the attribute **GUID** be returned.

3.1.5.11 LDAP Client Exports to Directory Server Implementing Extension Bundle D

The task for an LDAP client to export directory entries to a directory server implementing extension bundle D comprises the following protocol exchanges:

- The LDAP client submits one or more **Add**, **Delete**, **Modify**, **ModifyDN** and **Search** requests.
  - Search requests to retrieve the GUID attribute value of particular referenced entries have scope **baseObject** and request the attribute **GUID** be returned.
  - Search requests to locate an entry by its GUID have scope **baseObject** or scope **wholeSubtree** and a filter that is an equality match of the **GUID** attribute.
  - Search requests to retrieve the password management parameters of a directory entry have scope **baseObject** and request the attribute **passwordExpirationInterval**, the attribute **loginGraceLimit** or the attribute **loginGraceRemaining**, as defined in [NOVELL-SCHEMAREF], be returned.

3.1.5.12 LDAP Client Connects to Directory Server Implementing Extension Bundle B

The task for an LDAP client to connect to a directory server implementing extension bundle B comprises the following sequence of protocol exchanges:

1. The LDAP client establishes a first TCP connection to the directory server.

2. If configured, the LDAP client negotiates SSL, as described in section 2.1 of this document.

3. If configured to use DIGEST-MD5, the LDAP client issues two requests. First it requests a **baseObject** Search of the root DSE requesting the attribute **vendorVersion**. It then requests a **baseObject** Search of the root DSE requesting the attribute ibm-**saslldigestrealmname**.

4. The LDAP client binds using either the simple authentication method or the DIGEST-MD5 authentication method as described in section 2.2.1 of this document.

5. If the LDAP client did not request retrieving the vendor version in step 3, the LDAP client requests a **baseObject** Search of the root DSE requesting the attribute **vendorVersion**.

6. If configured to not use SSL and not use DIGEST-MD5:

    1. The LDAP client establishes a second TCP connection to the directory server.

    2. The LDAP client binds using the simple authentication method on the second TCP connection, as described in section 2.2.1 of this document.

    3. The LDAP client requests on the second connection a baseObject Search of the root DSE requesting the attribute **vendorVersion**. This second connection is no longer used by the LDAP client until it is unbound and closed as described in section 3.1.5.5 of this document.

3.1.5.13 LDAP Client Imports from Directory Server Implementing Extension Bundle B

The task for an LDAP client to import directory entries from a directory server implementing extension bundle B comprises the following protocol exchanges:

1. The LDAP client submits one or more Search requests. Each Search request is a request to retrieve the location and status of the change log.

    - Search requests to retrieve the location and status of the change log are **baseObject** Searches of the root DSE requesting that one of the attributes **lastchangenumber**, **firstchangenumber**, or **changelog** be returned.

2. If configured to retrieve search results in bulk or from the change log, the LDAP client obtains an additional paging connection, as described in section 3.1.5.15.

3. The LDAP client submits one or more Search requests. Each Search request can either request to retrieve attributes of a particular entry, request to retrieve results in bulk, or request to retrieve changes from the change log.

    - Search requests to retrieve attributes of a particular entry are **baseObject** Searches on the first connection requesting that the attribute named **objectClass**, the attribute named **ibm-entryUuid** as defined in [IBM-DS52SCHEMA], or all attributes be returned.

    - Search requests to retrieve results in bulk are wholeSubtree Searches requested on the paging connection with filter "(**objectClass**=*)" and which contain the paged search control, described in section 2.2.3 of this document.

    - Search requests to retrieve changes from the change log are **singleLevel** Searches requested on the paging connection, which contain the paged search control, based at the change log container with either equality or range filters on the **changenumber** attribute as defined in [ID-LDAPCHANGELOG].

3.1.5.14 LDAP Client Exports to Directory Server Implementing Extension Bundle B

The task for an LDAP client to export directory entries to a directory server implementing extension bundle B comprises the following protocol exchanges:

- The LDAP client submits one or more **Add**, **Delete**, **Modify**, **ModifyDN** and **Search** requests.
  - Search requests to retrieve the GUID attribute value of particular referenced entries have scope **baseObject** and request the attribute **ibm-entryUuid** be returned.
  - Search requests to locate an entry by its GUID have scope **baseObject** or scope wholeSubtree and a filter that is an equality match of the "ibm-entryUuid" attribute.

3.1.5.15 LDAP Client Connects to Directory Server Implementing Extension Bundle B for a Paging Connection

The task for an LDAP client to connect to a directory server that implements extension bundle B to obtain a paging connection contains the following sequence of protocol exchanges:

1. The LDAP client establishes an additional TCP connection, the paging connection, to the directory server.
2. If configured to do so, the LDAP client negotiates SSL on this connection, as described in section 2.1.
3. If configured to use DIGEST-MD5, the LDAP client issues two requests on this connection. First it requests a **baseObject** Search of the root DSE requesting the attribute **vendorVersion**. Then it requests a **baseObject** Search of the root DSE requesting the attribute **ibm-slapdigestrealmname**.
4. The LDAP client binds using either the simple authentication method or the DIGEST-MD5 authentication method, as described in section 2.2.1.
5. If the LDAP client did not request the retrieval of the vendor version in step 3, the LDAP client requests a **baseObject** Search of the root DSE requesting the attribute **vendorVersion**.
6. If configured to not use SSL and not use DIGEST-MD5:
   1. The LDAP client establishes a fourth TCP connection to the directory server.
   2. The LDAP client binds using the simple authentication method on the fourth TCP connection, as described in section 2.2.1.
   3. The LDAP client requests on the fourth connection a **baseObject** Search of the root DSE requesting the attribute **vendorVersion**. This fourth connection is no longer used by the LDAP client until it is unbound and closed as described in section 3.1.5.5.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

This section specifies the behavior of a directory server implementing extension bundle B, C or D. The behavior of AD DS is specified in [MS-ADTS].

3.2.1 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.

The abstract data model of LDAP is defined in [RFC2251] sections 3.2 through 3.4 and [RFC2252] sections 3 through 8.

The document [ID-LDAPCHANGELOG] defines an extension to the LDAP data model, to represent the change log.

Additional schema requirements are specified in section 2.1.

3.2.2 Timers

None.

3.2.3 Initialization

A directory server will listen for incoming connections on the assigned TCP ports identified in section 1.9.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message processing Events and Sequencing Rules

A directory server MUST implement the protocol model of [RFC2251].

The sequencing rules are defined in the following subsections.

3.2.5.1 Receiving a Connection Request

The following requirements are placed on a directory server implementing extension bundle B, C or D:

- The directory server MUST respond to a TCP connection establishment request, and
- The directory server MUST respond to SSL negotiation, as described in section 2.1 of this document.

3.2.5.2 Receiving a Bind Request Message

The following requirements are placed on a directory server implementing extension bundle B, C or D:

- The directory server MUST support binding with simple authentication, and

- The directory server MAY support binding with the SASL DIGEST-MD5 mechanism, as described in section 2.2.1 of this document.

3.2.5.3 Receiving a Search Request Message

The following requirements are placed on a directory server implementing extension bundle C:

- The directory server MUST support a baseObject search of the root DSE returning the attribute **namingContexts**,
- The directory server MUST support a baseObject search of the monitor DSE "cn=monitor",
- The directory server MUST support an attribute **version** of the monitor DSE in which there is a single string value, in which the value starts with one of the strings "iPlanet-Directory/5", "Netscape-Directory/5", "Netscape-Directory/4", "Netscape-Directory/6", "Sun-ONE-Directory/5", or "Sun Java(TM) System Directory Server/5",
- The directory server MUST support a baseObject search of the config DSE "cn=config" returning one entry,
- The directory server MUST support returning the attribute **nsslapd-sizelimit** of the config DSE, as described in [SUN-DS52RefMan],
- The directory server MUST support a change log, as described in section 2.3.6 of this document,
- The directory server MUST support the root DSE attributes **firstchangenumber**, **lastchangenumber** and **changelog** as defined in [IBM-RootDSELogInfo],
- The directory server MUST support searching a naming context in the directory information tree with a filter of an equalityMatch of the attribute **nsUniqueId**, and
- The directory server MUST provide a single-valued attribute **nsUniqueId** in each search result entry if that attribute was requested.

The following requirements are placed on a directory server implementing extension bundle D:

- The directory server MUST support a **baseObject** search of the root DSE,
- The directory server MUST support the root DSE attribute **vendorVersion** in which the directory server MUST return a single value, one of either "eDirectory v8.6.2", "eDirectory v8.7" or "LDAP Agent for Novell eDirectory 8.7",
- The directory server MUST support **singleLevel** and **wholeSubtree** searches with the filter "(GUID=*)", the search request including the virtual list view control described in section 2.2.5 of this document, and the sort control described in section 2.2.4 of this document,
- The directory server MUST provide the single-valued attributes **GUID** and **subordinateCount**, as defined in [NOVELL-SCHEMAREF], in each search result entry if those attributes were requested,
- The directory server MUST support searching a naming context in the directory information tree with a filter of an **equalityMatch** of the attribute **GUID**,
- The directory server MUST provide a single-valued attribute **GUID** in each search result entry if that attribute was requested, and
- When responding to a Search request specifying the attribute **passwordExpirationInterval**, **loginGraceLimit** or **loginGraceRemaining** be returned, the directory server MUST either

provide a single-valued attribute in the response, return a response omitting the attribute, or in contrast to [RFC4511] section 4.5.1.8, return the LDAP error code **noSuchAttribute**.

- The following requirements are placed on a directory server implementing extension bundle B:
- The directory server MUST support a baseObject search of the root DSE,
- If the directory server supports binding with a SASL DIGEST-MD5 mechanism, the directory server MUST support the root DSE attribute **ibm-sasldigestrealmname**,
- The directory server MUST support the root DSE attribute **vendorVersion** in which the directory server MUST return a single value of a string that starts with either of the literals "4.", "5.", "5.1" or "6.",
- The directory server MUST implement the change log as defined in [ID-LDAPCHANGELOG], and provide in the root DSE the operational attributes **lastchangenumber**, **firstchangenumber** and **changelog** defined by that document,
- The directory server MUST provide a single-valued attribute **ibm-entryUuid** in each search result entry if that attribute was requested,
- The directory server MUST support searching the directory information tree with a filter of an **equalityMatch** of the attribute ibm-**entryUuid**, and
- The directory server MUST provide a single-valued attribute **ibm-entryUuid** in each search result entry if that attribute was requested.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

3.3 Client Details

This section specifies the behavior of the client for the LDAP Extensions.

3.3.1 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.

The abstract data model of LDAP is defined in [RFC2251] and [RFC2252].

3.3.2 Timers

The LDAP client implements a timer to detect a non-responding directory server. This timer is discussed in section 3.3.6.

3.3.3 Initialization

The Synchronization Service contains an LDAP client which initiates an LDAP connection to a directory server when such action is specified by an import or export step in a Synchronization Service run profile.

3.3.4 Higher-Layer Triggered Events

A run profile import step or export step performs one of the patterns of tasks specified in section 3.1.4, the choice of which depends on the configuration of the Synchronization Service.

3.3.5 Message Processing Events and Sequencing Rules

The LDAP client does not return error indications to the connected directory server. The receipt of an unexpected response message will result in the Synchronization Service terminating the run profile step.

3.3.6 Timer Events

The LDAP client implements a countdown timer to detect a non-responding directory server. The timer is set to a predetermined time period and started when a request is sent to the directory server, and is stopped when the final response to the request is received. If the timer reaches 0 before the final response is received, the Synchronization Service terminates the run profile step, which closes the connection to the directory server.

The predetermined time period is configurable using the time-limit element of the step data defined in [MS-UPSCDS] section 2.2.32.6.4.1, [MS-UPSCDS] section 2.2.32.6.4.2, [MS-UPSCDS] section 2.2.32.6.4.3 and [MS-UPSCDS] section 2.2.32.6.4.4.

3.3.7 Other Local Events

None.

4 Protocol Examples

An example of a paged search interaction can be found in [RFC2696] section 4.

An example of the LDAP change log can be found in section 7 of the document "Definition of an Object Class to Hold LDAP Change Records" [ID-LDAPCHANGELOG].

An example of the LDAP virtual list view extension interaction can be found in section 7 of the document "LDAP Extensions for Scrolling View Browsing of Search Results" [ID-LDAPVLV].

An example of range retrieval can be found in section 5.2 of the document "Incremental Retrieval of Multi-valued Properties" [ID-LDAPINCREMENTAL].

5 Security

5.1 Security Considerations for Implementers

Additional discussion of security considerations can be found in [ID-LDAPCHANGELOG] section 10, [ID-LDAPVLV] section 10, [RFC2246] appendix F, [RFC2251] section 7, [RFC2696] section 6, [RFC2829] section 12, [RFC2831] section 3, [RFC2891] section 4, [RFC4178] section 7, and [RFC4511] section 6.

5.2 Index of Security Parameters

Security Parameter	Sections
Transport Level Security Mechanisms	2.1
Authentication Mechanisms	2.2.1 and 2.2.2

6 Appendix A: Product Behavior

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

- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013

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 3.1.5.1: The Windows LDAP client implementation might cache the value of root DSE attributes that are retrieved from AD DS or AD LDS for up to 15 minutes. Therefore, this request might not always be sent by the Synchronization Service.

7 Change Tracking

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

8 Index

A

Abstract data model
  client 31
  server 29
Applicability 11

C

Capability negotiation 11
Change tracking 36
Client
  abstract data model 31
  higher-layer triggered events 32
  initialization 32
  message processing 32
  other local events 32
  overview (section 3.1 21, section 3.3 31)
  sequencing rules 32
  timer events 32
  timers 31

D

Data model - abstract
  client 31
  server 29
Directory service schema elements 16

E

Elements - directory service schema 16

F

Fields - vendor-extensible 11

G

Glossary 6

H

Higher-layer triggered events
  client 32
  server 29

I

Implementer - security considerations 34
Index of security parameters 34
Informative references 8
Initialization
  client 32
  server 29
Introduction 6

L

LDAP Digest Authentication message 14
LDAP DirSync Control message 15
LDAP Extended DN Control message 16
LDAP Lazy Commit Control message 16
LDAP Negotiated Authentication message 14
LDAP Paged Search Control message 15
LDAP Show Deleted Control message 15
LDAP Sort Controls message 15
LDAP Virtual List View Control message 15

M

Message processing
  client 32
Messages
  LDAP Digest Authentication 14
  LDAP DirSync Control 15
  LDAP Extended DN Control 16
  LDAP Lazy Commit Control 16
  LDAP Negotiated Authentication 14
  LDAP Paged Search Control 15
  LDAP Show Deleted Control 15
  LDAP Sort Controls 15
  LDAP Virtual List View Control 15
  transport 13

N

Normative references 7

O

Other local events
  client 32
  server 31
Overview (synopsis) 9

P

Parameters - security index 34
Preconditions 11
Prerequisites 11
Product behavior 35

R

References 6
  informative 8
  normative 7
Relationship to other protocols 10

S

Schema elements - directory service 16
Security
  implementer considerations 34
  parameter index 34
Sequencing rules
  client 32

Server
abstract data model 29
higher-layer triggered events 29
initialization 29
other local events 31
overview (section 3.1 21, section 3.2 29)
timer events 31
timers 29
Standards assignments 11

T

Timer events
client 32
server 31
Timers
client 31
server 29
Tracking changes 36
Transport 13
Triggered events - higher-layer
client 32
server 29

V

Vendor-extensible fields 11
Versioning 11