

INTERNET-DRAFT
Intended Category: Standard Track
Expires: 4 May 2003
Obsoletes: RFC [2252](#), [RFC 2256](#)

K. Dally, Editor
The MITRE Corp.
S. Legg
ADACEL
[4](#) November 2002

LDAP: Syntaxes
<[draft-ietf-ldapbis-syntaxes-03](#)>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#).

This document is intended to be, after appropriate review and revision, submitted to the RFC Editor as a Standard Track document. Distribution of this memo is unlimited. Technical discussion of this document will take place on the IETF LDAP Revision Working Group (LDAPbis) mailing list <ietf-ldapbis@openldap.org>. Please send editorial comments directly to the editor <kdally@mitre.org>.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>. The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Copyright 2002, The Internet Society. All Rights Reserved.

Please see the Copyright section near the end of this document for more information.

Dally, Legg

Expires 4 May 2003

[Page 1]

Abstract

The Lightweight Directory Access Protocol (LDAP) [[Protocol](#)] provides for exchanging AttributeValue fields in protocol. This document defines a set of syntaxes for LDAP, rules by which attribute values of these syntaxes are represented in the LDAP protocol, and the matching rules which specify how values are compared. Also, this document indicates the syntax support requirements on LDAP servers. The syntaxes and matching rules defined in this document are used in schema definition documents to specify attribute types.

[Editor's note: This document is a modified version of parts of [RFC 2252](#) and [RFC 2256](#), in order to bring them up to date. This action is part of the maintenance activity that is needed in order to progress LDAP (v3) to Draft Standard. The changes are described in Annex C of this document. Open items are listed in Annex B. End of Editor's note]

Dally, Legg

Expires 4 May 2003

[Page 2]

Table of Contents

Status of this Memo.....	1
Abstract.....	2
1. Overview.....	5
2. General Issues.....	5
2.1 Notation.....	5
2.2 Syntaxes.....	5
2.2.1 LDAP-Specific Encodings.....	6
2.2.2 Syntaxes Implementation Status.....	6
2.2.3 Syntax Object Identifiers.....	6
2.2.4 Syntax Description.....	6
2.3 Matching Rules.....	7
2.3.1 Matching Rules Implementation Status.....	7
2.3.2 Matching Rule Description.....	7
3. Syntaxes.....	8
3.1 Attribute Type Description.....	8
3.2 Bit String.....	8
3.3 Boolean.....	8
3.4 Country String.....	9
3.5 Delivery Method.....	9
3.6 Directory String.....	9
3.7 DIT Content Rule Description.....	10
3.8 DIT Structure Rule Description.....	11
3.9 DN.....	11
3.10 Enhanced Guide.....	12
3.11 Facsimile Telephone Number.....	12
3.12 Fax.....	13
3.13 Generalized Time.....	13
3.14 Guide.....	13
3.15 IA5 String.....	14
3.16 Integer.....	14
3.17 JPEG.....	14
3.18 LDAP Syntax Description.....	15
3.19 Matching Rule Description.....	15
3.20 Matching Rule Use Description.....	15
3.21 MHS OR Address.....	15
3.22 Name and Optional UID.....	16
3.23 Name Form Description.....	16
3.24 Numeric String.....	16
3.25 Object Class Description.....	17
3.26 Octet String.....	17
3.27 OID.....	17
3.28 Other Mailbox.....	18

3.29	Postal Address.....	18
3.30	Presentation Address.....	19
3.31	Printable String.....	21
3.32	Substring Assertion	21
3.33	Telephone Number.....	21
3.34	Teletex Terminal Identifier.....	22

3.35	Telex Number.....	22
3.36	UTC Time.....	23
4.	Matching Rules.....	23
4.1	bitStringMatch.....	23
4.2	caseExactIA5Match.....	23
4.3	caseIgnoreIA5Match.....	24
4.4	caseIgnoreListMatch.....	24
4.5	caseIgnoreMatch.....	24
4.6	caseIgnoreOrderingMatch.....	24
4.7	caseIgnoreSubstringsMatch.....	25
4.8	distinguishedNameMatch.....	25
4.9	generalizedTimeMatch.....	25
4.10	generalizedTimeOrderingMatch.....	25
4.11	integerFirstComponentMatch.....	25
4.12	integerMatch.....	26
4.13	numericStringMatch.....	26
4.14	numericStringSubstringsMatch.....	26
4.15	objectIdentifierFirstComponentMatch.....	26
4.16	objectIdentifierMatch.....	26
4.17	octetStringMatch.....	27
4.18	presentationAddressMatch.....	27
4.19	protocolInformationMatch.....	27
4.20	telephoneNumberMatch.....	28
4.21	telephoneNumberSubstringsMatch.....	28
4.22	uniqueMemberMatch.....	28
5.	Security Considerations.....	28
5.1	Disclosure.....	28
5.2	Security Information Syntaxes.....	28
5.3	Securing the Directory.....	29
6.	Acknowledgements.....	29
7.	Authors' Addresses.....	29
8.	References.....	30
8.1	Normative.....	30
8.2	Informative.....	31
9.	Full Copyright Statement.....	31
Annex A	Object Identifiers for Syntaxes.....	32
Annex B	Topics to be Addressed in This Document.....	33
Annex C	Change Log.....	34

Dally, Legg

Expires 4 May 2003

[Page 4]

1. Overview

This document defines part of the framework for developing schemas for directories accessible via the Lightweight Directory Access Protocol (LDAP) [[Protocol](#)].

Schema is the collection of attribute type definitions, object class definitions and other information which specify the entries and their contents that a server holds. A server uses schema to determine how to match a filter or attribute value assertion (in a compare operation) against the attributes of an entry, and whether to permit add and modify operations. This document specifies syntaxes, which are used in defining attribute types, and matching rules.

Therefore, [Section 2](#) states the general requirements and notation for definition of syntaxes and matching rules.

[Section 3](#) lists syntaxes and [section 4](#) contains matching rules.

Additional documents define schemas for representing real-world objects as directory entries. See [Models], sections [2.4.1](#) and [2.6](#) and [Schema] for the definitions of user objects and attributes from [[X.501](#)], [[X.520](#)], and [[X.521](#)].

2. General Issues

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)].

This document describes the syntaxes of data conveyed in an Internet protocol.

Implementors are strongly advised to first read the description of how schema is represented in X.501 [[X.501](#)] before reading the rest of this document.

2.1 Notation

For the purposes of defining attribute syntaxes and matching rules, Augmented Backus-Naur Form (ABNF) is used. The ABNF productions used in this document are used by other documents in the LDAP set and are listed in [Models].

The schema definitions provided in this document are line-wrapped for readability.

In addition, the following ABNF productions are used in this

document:

[2.2](#) **Syntaxes**

This section defines general requirements for LDAP attribute

syntaxes. All documents defining attribute syntaxes for use with LDAP are expected to conform to these requirements. Syntaxes are also defined for matching rules whose assertion value syntax is different from the attribute value syntax.

[2.2.1](#) LDAP-Specific Encodings

In [[Protocol](#)], the encoding of the LDAP protocol is specified. The protocol encapsulates values of attributes in many places. In this specification, the encoding of the values is specified, as part of each syntax definition. These value encoding rules are termed "LDAP-specific encodings". The LDAP-specific encoding of a value is what is transmitted in the protocol.

The LDAP-specific encoding of a given attribute syntax always produces octet-aligned values. To the greatest extent possible, the LDAP-specific encoding of a value is supposed to be usable for display purposes. In particular, encoding rules for attribute syntaxes defining non-binary values are supposed to produce strings that can be displayed with little or no translation by clients implementing LDAP. There are a few cases (e.g., audio) however, when it is not sensible to produce a human-readable representation.

[2.2.2](#) Syntaxes Implementation Status

Clients and servers need not implement all the syntaxes listed in [section 3](#), and MAY implement other syntaxes.

Clients MUST NOT assume that the LDAP-specific encoding of a value of an unrecognized syntax is a human-readable character string.

Other documents define additional attribute syntaxes. However, the definition of additional arbitrary syntaxes is strongly deprecated since it will hinder interoperability. Today's client and server implementations generally do not have the ability to dynamically recognize new syntaxes.

[2.2.3](#) Syntax Object Identifiers

In an LDAP schema, a syntax is named by the Object Identifier (OID) assigned to it.

Syntaxes that are currently in use in the user schema specification [Schema] and the models specification [Models] are specified in this document in [section 3](#). The object identifiers assigned to these syntaxes are listed in Annex A.

[2.2.4](#) Syntax Description

The SyntaxDescription ABNF specified in [Models] is the method used in this document to define the values for each syntax.

For example, the syntax description of the INTEGER syntax for whole number values is:

```
( 1.3.6.1.4.1.1466.115.121.1.27 DESC 'INTEGER' )
```

[2.3](#) Matching Rules

The matching rules specified in this document are defined in [Section 4](#).

Matching rules are used by servers to compare attribute values against assertion values when performing Search and Compare operations. They are also used to identify the value to be added or deleted when modifying entries, and are used when comparing a purported distinguished name with the name of an entry.

Most of the attributes given in the user schema [Schema] have an equality matching rule defined.

...An OID is assigned to a matching rule when it is defined. A matching rule definition ought not be changed without having a new OID assigned to it.

[2.3.1](#) Matching Rules Implementation Status

Servers which support matching rules and the extensibleMatch SHOULD implement all the matching rules in [section 4](#).

Servers MUST publish in the matchingRules attribute, the definitions of matching rules referenced by values of the attributeTypes and matchingRuleUse attributes in the same subschema entry. Other unreferenced matching rules MAY be published in the matchingRules attribute.

If the server supports the extensibleMatch, then the server MAY use the matchingRuleUse attribute to indicate the applicability of selected matching rules to designated attribute types in an extensibleMatch.

[2.3.2](#) Matching Rule Description

The SyntaxDescription ABNF specified in [Models] is the method used. The Matching Rule descriptions are specified according to the MatchingRule ABNF specified in [Models].

Dally, Legg

Expires 4 May 2003

[Page 7]

3. Syntaxes

3.1 Attribute Type Description

A value in this syntax is a definition of an attribute type according to the ABNF given [Models]. The LDAP-specific encoding is the character codes in UTF-8 which correspond to the characters in the definition.

This syntax is the form in which schema attribute types are published in the directory in a subentry. The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.3 DESC 'Attribute Type Description' )
```

For example, this is the definition from [User] of the businessCategory attribute type:

```
( 2.5.4.15 NAME 'businessCategory'  
  EQUALITY caseIgnoreMatch  
  SUBSTR caseIgnoreSubstringsMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128} )
```

The syntax type for the businessCategory Attribute Type is Directory String.

This example definition is a value of the Attribute Type Description syntax. The LDAP-specific encoding of this value is the definition itself.

3.2 Bit String

A value in this syntax is a value of the BIT STRING data type from ASN.1 [X.680]. The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.6 DESC 'Bit String' )
```

The LDAP-specific encoding of a value is the following ABNF:

```
bitstring = SQUOTE *binary-digit SQUOTE "B"  
  
binary-digit = "0" / "1"
```

Example: '0101111101'B

3.3 Boolean

A value in this syntax is a value of the BOOLEAN data type from

ASN.1 [[X.680](#)]. That is, there are exactly two values: one value representing logically true, and the other representing logically

false. The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.7 DESC 'Boolean' )
```

The LDAP-specific encoding of a value is the following ABNF:

```
boolean = "TRUE" / "FALSE"
```

[3.4](#) Country String

A value in this syntax is two ASN.1 [\[X.680\]](#) printable string characters representing a country. The permitted values are as listed in ISO 3166 [\[ISO3166\]](#). The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.11 DESC 'Country String' )
```

The LDAP-specific encoding of a value is the following ABNF:

```
CountryString = ALPHA ALPHA
```

Example: US

[3.5](#) Delivery Method

A value in this syntax is a set of the ASN.1 [\[X.680\]](#) enumerated INTEGER values that indicates, in preference order, the service(s) by which the user, represented by the entry, is willing and/or capable of receiving messages.

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.14 DESC 'Delivery Method' )
```

The LDAP-specific encoding of a value is the following ABNF:

```
delivery-value = pdm / ( WSP pdm SP DOLLAR SP delivery-value )
```

```
pdm = "any" / "mhs" / "physical" / "telex" / "teletex" /  
      "g3fax" / "g4fax" / "ia5" / "videotex" / "telephone"
```

Example: telephone \$ videotex

[3.6](#) Directory String

A value in this syntax is a value of one of the TeletexString, PrintableString or UniversalString data types from ASN.1 [\[X.680\]](#). The minimum length of a Directory String value is one character, that

is, the string cannot be 'empty'. The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.15 DESC 'Directory String' )
```

The LDAP-specific encoding of a value is the character string itself.

Note: The form of DirectoryString is not indicated in protocol. Servers which convert to DAP MUST choose an appropriate form. Servers MUST NOT reject values merely because they contain legal Unicode characters outside of the range of printable ASCII.

Servers and clients MUST be prepared to receive arbitrary Unicode characters, including characters not presently assigned to any character set.

Example:

This is a string of DirectoryString containing #!%#@.

For characters in the PrintableString form, the value in the native LDAP encoding is the value itself.

If the string is in the TeletexString form, then the characters are transliterated to their equivalents in UniversalString, and encoded in UTF-8 [[RFC2044](#)].

If the string is in the UniversalString or BMPString forms [[ISO10646](#)], UTF-8 is the LDAP-specific encoding.

[3.7](#) DIT Content Rule Description

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.16 DESC 'DIT Content Rule
    Description' )
```

This syntax is the form in which schema content rules are published in the directory in a subentry.

A value in this syntax is a definition of a DIT content rule according to the ABNF in [Models].

The native LDAP encoding of a value is the character string (DirectoryString) itself.

Note: The form of DirectoryString is not indicated in protocol, unless the ;binary option is used (see [Prot]). Servers which convert to DAP MUST choose an appropriate form. Servers MUST NOT reject values merely because they contain legal Unicode characters outside of the range of printable ASCII.

Servers and clients MUST be prepared to receive arbitrary Unicode characters, including characters not presently assigned to any

character set.

Example:

This is a string of `DirectoryString` containing `#!%#@`.

For characters in the PrintableString form, the value in the native LDAP encoding is the value itself.

If it is in the TeletexString form, then the characters are transliterated to their equivalents in UniversalString, and encoded in UTF-8 [UTF-8].

If it is in the UniversalString or BMPString forms [[ISO10646](#)], UTF-8 is the native LDAP encoding.

3.8 DIT Structure Rule Description

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.17 DESC 'DIT Structure Rule
    Description' )
```

This syntax is the form in which schema structure rules are published in the directory in a subentry.

A value in the DIT Structure Rule Description syntax is a definition of a schema Structure Rule according to the ABNF in [Models].

The LDAP-specific encoding is the character codes in UTF-8 [UTF-8] which correspond to the characters in the structure rule definition.

3.9 DN

A value in the Distinguished Name syntax is a structured set of the ASN.1 [[X.680](#)] data types that are included in the DirectoryString syntax. The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.12 DESC 'DN' )
```

The LDAP-specific encoding of a value is defined in [[LDAPDN](#)]. Note that the LDAP-specific encoding is not reversible to the original BER encoding used in X.500 for Distinguished Names, as the CHOICE of any DirectoryString element in an RDN is not evident in the LDAP-specific encoding.. See the note in [section 3.7](#).

Examples (from [[LDAPDN](#)]):

CN=Steve Kille,O=Isode Limited,C=GB

OU=Sales+CN=J. Smith,O=Widget Inc.,C=US

CN=L. Eagle,O=Sue\, Grabbit and Runn,C=GB

CN=Before\0DAfter, O=Test, C=GB

Dally, Legg

Expires 4 May 2003

[Page 11]

1.1.3.6.1.4.1.1466.0=#04024869,0=Test,C=GB

SN=Lu\C4\8Di\C4\87

[3.10](#) Enhanced Guide

A value in the Enhanced Guide syntax is the matching criteria and scope of operation in an Enhanced Filter.

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.21 DESC 'Enhanced Guide' )
```

The LDAP-specific encoding of a value is defined by the following ABNF:

```
EnhancedGuide = SP oid WSP SHARP WSP criteria WSP SHARP  
                WSP subset
```

```
subset = "baseobject" / "oneLevel" / "wholeSubtree"
```

```
criteria = or-term / LPAREN or-term RPAREN
```

```
or-term = and-term *( "|" and-term )
```

```
and-term = not-term *( "&" not-term )
```

```
not-term = "!" not-term /  
            attributetype DOLLAR match-type /  
            LPAREN or-term RPAREN /  
            "?true" / ;  
            "?false"
```

```
match-type = "EQ" / "SUBSTR" / "GE" / "LE" / "APPROX"
```

The ?true term alternative represents an empty "and" in the Criteria.
The ?false alternative represents an empty "or" in the Criteria.

Example:

```
person#(sn)#oneLevel
```

[3.11](#) Facsimile Telephone Number

A value in the Facsimile Telephone Number syntax is a subscriber number on the (public) telephone network of a facsimile device. The telephone number is a character string based on E.123 [[E.123](#)]. The character string type is the PrintableString data type from ASN.1 [[X.680](#)]. The following syntax description gives the OID

assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.22 DESC 'Facsimile Telephone Number')
```

Dally, Legg

Expires 4 May 2003

[Page 12]

The LDAP-specific encoding of a value is defined by the following ABNF:

```
fax-number = printablestring [ "$" faxparameters ]
              ; telephone number, possibly followed by facsimile
              ; parameters

printablestring = 1*p

p = ALPHA / DIGIT / SQUOTE / LPAREN / RPAREN / PLUS / COMMA /
  HYPHEN / DOT / EQUALS / SLASH / COLON / QUESTION / SPACE

faxparameters = faxparm / ( faxparm "$" faxparameters )

faxparm = "twoDimensional" / "fineResolution" / "unlimitedLength"
  / "b4Length" / "a3Width" / "b4Width" / "uncompressed"
```

[3.12](#) Fax

A value in the Fax syntax is an image which is produced using the Group 3 facsimile process [Fax] to duplicate an object, such as a memo.

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.23 DESC 'Fax' )
```

Values in this syntax are expressed as octet strings containing Group 3 Fax images as defined in [Fax].

[3.13](#) Generalized Time

A value in the Generalized Time syntax is a date and time. The year is given as a four-digit number. The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.24 DESC 'Generalized Time' )
```

The LDAP-specific encoding is a value of the GeneralizedTime data type from ASN.1 [[X.680](#)]. Time zone MUST be present and SHOULD be GMT (Z).

Example:

199412161032Z means 10:32 a.m. Dec. 16, 1994 in the Greenwich Mean Time time zone.

[3.14](#) Guide

A value in the Guide syntax is the matching criteria in a Filter.

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.25 DESC 'Guide' )
```

The Guide syntax is not intended to be used for defining new attributes. It is important for backwards compatibility with LDAP systems that implement an earlier version of LDAP [[RFC1778](#)].

The LDAP-specific encoding of a value is defined by the following ABNF:

```
guide-value = [ object-class "#" ] criteria
```

```
object-class = SP oid
```

The criteria production is defined in the Enhanced Guide syntax in [section 3.11](#).

[3.15](#) IA5 String

A value in the IA5 String syntax is a value of the IA5String data type from ASN.1 [[X.680](#)]. International Alphabet 5 (IA5) [[IA5](#)] is the international version of the ASCII character set.

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.27 DESC 'IA5 String' )
```

The LDAP-specific encoding of a value in this syntax is the character string value itself.

[3.16](#) Integer

A value in the INTEGER syntax is a whole number as specified in the INTEGER data type from ASN.1 [[X.680](#)].

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.27 DESC 'INTEGER' )
```

The LDAP-specific encoding of a value is the decimal representation of the value, with each decimal digit represented by the its character equivalent. So, the number 1321 is represented by the character string "1321".

[3.17](#) JPEG

A value in the JPEG syntax is an image produced according to specific rules for light values. The following syntax description gives the OID assigned to this syntax:

(1.3.6.1.4.1.1466.115.121.1.28 DESC 'JPEG')

The LDAP-specific encoding of a value is an octet string of the light values representing the image.

[3.18](#) LDAP Syntax Description

A value in the LDAP Syntax Description syntax is a definition of a LDAP syntax description according to the ABNF given in [MODELS].

This syntax is the form in which schema syntax descriptions are published in the directory in a subentry. The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.54 DESC 'LDAP Syntax Description' )
```

Note that, in X.520 [Attr], syntaxes are not labeled distinctly with respect to attributes.

The LDAP-specific encoding is the character codes in UTF-8 [[ISO10646](#)] which correspond to the characters in the definition.

[3.19](#) Matching Rule Description

A value in the Matching Rule Description syntax is a definition of a matching rule according to the ABNF given in [MODELS]. This syntax is the form in which schema matching rules are published in the directory in a subentry. The following syntax definition gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.31 DESC 'Matching Rule Description' )
```

The LDAP-specific encoding is the character codes in UTF-8 [[ISO10646](#)] which correspond to the characters in the definition of a Matching Rule.

[3.20](#) Matching Rule Use Description

A value in the Matching Rule Use Description syntax is a definition of a matching Rule and the attribute types with which the rule could be used in an extensibleMatch search filter. The values are specified according to the ABNF given in [MODELS]. The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.31 DESC 'Matching Rule Use  
Description' )
```

This syntax is the form in which schema matching rule usage permissions are published in the directory in a subentry.

The LDAP-specific encoding is the character codes in UTF-8 [[ISO10646](#)] which correspond to the characters in the definition.

[3.21](#) MHS OR Address

A value in the MHS OR Address syntax is the addressing information of a user of an X.400 messaging service. The LDAP-specific encoding is defined in [RFC 1327](#) [[RFC1327](#)].

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.33 DESC 'MHS OR Address' )
```

[3.22](#) Name and Optional UID

A value of the Name and Optional UID (Unique Identifier) syntax is a Distinguished Name as defined in [section 3.9](#) plus a bit string that differentiates the value from otherwise identical names. The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.34 DESC 'Name And Optional UID' )
```

The LDAP-specific encoding of a value is the following ABNF:

```
NameAndOptionalUID = DistinguishedName [ "#" bitstring ]
```

Although the '#' character could occur in a string representation of a distinguished name, no additional special quoting is done.

Example:

```
1.3.6.1.4.1.1466.0=#04024869,0=Test,C=GB#'0101'B
```

[3.23](#) Name Form Description

A value in the Name Form Description syntax is a definition of a Name Form according to the ABNF given in [MODELS].

A value indicates the one or more attributes in an entry type (e.g., person, device) that are used as the Relative Distinguished Name of the entry.

This syntax is the form in which schema name forms are published in the directory. The LDAP-specific encoding of a value is the character codes in UTF-8 [[ISO10646](#)] which correspond to the characters in the definition.

The following syntax description gives the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.35 DESC 'Name Form Description' )
```

[3.24](#) Numeric String

A value in the Numeric String syntax is a series of numerals and spaces as specified in the NumericString data type from ASN.1 [[X.680](#)]. The following string states the OID assigned to

this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.36 DESC 'Numeric String' )
```


The representation of a string in this syntax is the string value itself.

Example: 1997

[3.25](#) Object Class Description

A value in this syntax is a character string which expresses the definition of an object class according to the ABNF given in [MODELS]. This syntax is the form in which schema object classes are published in the directory in a subentry. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.37 DESC 'Object Class Description' )
```

For example, the character string below specifies the country object class, which requires the c (country name) attribute and allows the searchGuide and description attributes. All of these schema elements are specified in [Schema].

```
( 2.5.6.2 NAME 'country' SUP top STRUCTURAL MUST c  
  MAY ( searchGuide $ description ) )
```

[3.26](#) Octet String

A value in the Octet String syntax is a value of the OCTET STRING data type from ASN.1 [[X.680](#)]. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.40 DESC 'Octet String' )
```

Values in this syntax are written as a series of 8-bit values, according to the octet string value notation specified in [[X.680](#)]. In the case of character strings, the characters themselves could be written.

Example:
secret

[3.27](#) OID

A value in the Object Identifier syntax is a series of integers, ordered as specified in the OBJECT IDENTIFIER data type from ASN.1 [[X.680](#)]. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.38 DESC 'OID' )
```

Values in this syntax are expressed according to the ABNF in

[MODELS], [section 1.3](#) for "oid".

Examples: 1.2.3.4
cn

[3.28](#) Other Mailbox

A value in the Other Mailbox syntax gives a mail system name with the name of a mailbox in the system. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.39 DESC 'Other Mailbox' )
```

Values in this syntax are written according to the following ABNF:

```
otherMailbox = mailbox-type DOLLAR mailbox
```

```
mailbox-type = printableString
```

```
mailbox = <an encoded IA5 String>
```

The printableString production is defined in [section 3.11](#).

In the above, mailbox-type represents the type of mail system in which the mailbox resides, for example "MCIMail"; and mailbox is the actual mailbox in the mail system defined by mailbox-type.

The representation of a string in this syntax is the string value itself.

[3.29](#) Postal Address

A value in the Postal Address syntax is a series of strings which form an address in a physical mail system. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.41 DESC 'Postal Address' )
```

Values in this syntax are written according to the following ABNF:

```
postal-address = dstring *( DOLLAR dstring )
```

In the above, each dstring component of a postal address value is written as a value of type Directory String syntax. Backslashes and dollar characters, if they occur in the component, are quoted as described in [MODELS]. Many servers limit the postal address to six lines of up to thirty characters.

Example:

```
1234 Main St.$Anytown, CA 12345$USA  
\241,000,000 Sweepstakes$PO Box 1000000$Anytown, CA 12345$USA
```

Dally, Legg

Expires 4 May 2003

[Page 18]

3.30 Presentation Address

A value in the Presentation Address syntax is an OSI Application Layer address of a remote application. Logically, a presentation address consists of:

- o A presentation selector
- o A session selector
- o A transport selector
- o A set of network addresses

The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.43 DESC 'Presentation Address' )
```

Values in this syntax are written according to the following ABNF:

```
presentation-address = [[[ psel "/" ] ssel "/" ] tsel "/" ]  
                        network-address-list
```

```
psel = selector
```

```
ssel = selector
```

```
tsel = selector
```

```
network-address-list = network-address USCORE  
                        network-address-list / network-address
```

```
network-address = "NS" PLUS dothexstring  
                  / afi PLUS idi [ PLUS dsp ]  
                  / idp PLUS hexstring
```

The first (NS) alternative is the Concrete Binary Representation. It is the compact encoding.

The afi alternative is a user-oriented representation of a network address.

The idp alternative is a form of network-address included for compatibility with ISO 8348 [[IS08348](#)].

```
selector = DQUOTE otherstring DQUOTE  
           / SHARP numericstring  
           / SQUOTE hexstring "'H"  
           / ""
```

The otherstring alternative for the selector is IA5 characters.

The "" alternative for the selector expresses the case where the selector is present, but Empty.

```
idp = numericstring
```

```
dsp = "d" numericstring
      / "x" dothexstring
      / "l" otherstring
      / "RFC-1006" PLUS prefix PLUS ip [ PLUS port [ PLUS tset ]]
      / "X.25(80)" PLUS prefix PLUS dte [ PLUS cudf-or-pid PLUS
        hexstring ]
      / "ECMA-117-Binary" PLUS hexstring PLUS hexstring PLUS
        hexstring / "ECMA-117-Decimal" PLUS numericstring PLUS
        numericstring PLUS numericstring
```

The d alternative is the Abstract Decimal form of the Domain Specific Part (dsp) in a network address.

The x alternative is the Abstract Binary form of the dsp in a network address.

The l alternative is IA5 characters and is only meaningful locally.

```
idi = numericstring

afi = "X121" / "DCC" / "TELEX" / "PSTN" / "ISDN" / "ICD" / "LOCAL"

prefix = DIGIT DIGIT

ip = numericstring
    ; dotted decimal form (e.g., 10.0.0.6) or
    domain (e.g., twg.com)

port = numericstring

tset = numericstring

dte = numericstring

cudf-or-pid = "CUDF" / "PID"

other = keychar / PLUS / DOT

domainchar = keychar / DOT

hexoctet = HEX HEX

decimal-octet = 1*3DIGIT

otherstring = other otherstring / other

domainstring = domainchar otherstring / domainchar

hexstring = hexoctet hexstring / hexoctet
```

```
dotstring = decimaloctet DOT dotstring /  
           decimaloctet DOT decimaloctet  
  
dothexstring = dotstring / hexstring
```


[3.31](#) Printable String

A value in the Printable String syntax is a series of alphabetic, numeric, and (limited) punctuation characters as specified in the PrintableString data type from ASN.1 [[X.680](#)] and in production p of [section 3.11](#). Values in this syntax are expressed as the string itself. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.44 DESC 'Printable String' )
```

Example: This is a PrintableString.

[3.32](#) Substring Assertion

The Substring Assertion syntax is used in rules which can be used in substrings and extensible matching rules. When using a substrings assertion, substrings components are provided in a SubstringFilter sequence. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.58 DESC 'Substring Assertion' )
```

When using a matching rule assertion, substring components are encoded according to the following ABNF and provided as the matchValue of the MatchingRuleAssertion:

```
substring = [initial] any [final]
```

```
initial = value
```

```
any = "*" *(value "*")
```

```
final = value
```

The <value> production is a UTF-8 [[ISO10646](#)] string. If a backslash or asterix character is present in a production of <value>, it is quoted as described in [MODELS].

[3.33](#) Telephone Number

A value in the telephone number syntax is the series of characters that express a number (address) assigned to a telephone system subscriber. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.50 DESC 'Telephone Number' )
```

Values in this syntax are written as if they were ASN.1 [[X.680](#)] Printable String types. Telephone numbers are defined in X.520

[[X.520](#)] to comply with the internationally agreed format for expressing international telephone numbers in Recommendation E.123 [[E.123](#)].

The representation of a string in this syntax is the string value itself.

Example: +1 512 315 0280

3.34 Teletex Terminal Identifier

A value in this syntax is a string of characters that express the identifier value assigned to a teletex service subscriber. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.51 DESC 'Teletex Terminal
  Identifier' )
```

Values in this syntax are written according to the following ABNF:

```
teletex-id = ttx-term 0*("$" ttx-param)

ttx-term   = printablestring

ttx-param  = ttx-key ":" ttx-value

ttx-key    = "graphic" / "control" / "misc" / "page" / "private"

ttx-value  = octetstring
```

In the above, the first printablestring is the encoding of the first portion of the teletex terminal identifier to be encoded, and the subsequent 0 or more octetstrings are subsequent portions of the teletex terminal identifier.

The representation of a string in this syntax is the string value itself.

3.35 Telex Number

A value in the Telex Number syntax is the number assigned to a telex system subscriber with the country and answerback values indicated.

The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.52 DESC 'Telex Number' )
```

Values in this syntax are written according to the following ABNF:

```
telex-number = actual-number "$" country "$" answerback

actual-number = printablestring

country       = printablestring
```

```
answerback    = printablestring
```

Dally, Legg

Expires 4 May 2003

[Page 22]

In the above, actual-number is the syntactic representation of the number portion of the TELEX number being written, country is the TELEX country code, and answerback is the answerback code of a TELEX terminal.

The representation of a string in this syntax is the string value itself.

[3.36](#) UTC Time

A value in the UTC Time syntax is a date and time indicating accuracy to minute or second. The year is given as a two-digit number. The following string states the OID assigned to this syntax:

```
( 1.3.6.1.4.1.1466.115.121.1.53 DESC 'UTC Time' )
```

Values in this syntax are written as if they were printable strings, formulated as specified for the UTCTime data type in ASN.1 [[X.680](#)]. It is strongly suggested that GMT time be used.

Note: This syntax is deprecated in favor of the Generalized Time syntax.

[4.](#) Matching Rules

When performing the caseExactMatch, caseIgnoreMatch, caseIgnoreListMatch, telephoneNumberMatch, caseExactIA5Match and caseIgnoreIA5Match, multiple adjoining whitespace characters are treated the same as an individual space, and leading and trailing whitespace is ignored.

[4.1](#) bitStringMatch

The following ABNF associates the bitStringMatch rule with the Bit String syntax:

```
( 2.5.13.16 NAME 'bitStringMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.6 ) ; Bit String
```

This matching rule is used to test equality.

[4.2](#) caseExactIA5Match

The following ABNF associates the caseExactIA5Match rule with the IA5 String syntax:

```
( 1.3.6.1.4.1.1466.109.114.1 NAME 'caseExactIA5Match'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 ) ; IA5 String
```

This matching rule is used to test equality.

Dally, Legg

Expires 4 May 2003

[Page 23]

[4.3](#) caseIgnoreIA5Match

The following ABNF associates the caseIgnoreIA5Match rule with the IA5 String syntax:

```
( 1.3.6.1.4.1.1466.109.114.2 NAME 'caseIgnoreIA5Match'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 ) ; IA5 String
```

This matching rule is used to test equality.

[4.4](#) caseIgnoreListMatch

The ABNF below associates the caseIgnoreListMatch rule with the Postal Address syntax. The X.520 [[X.520](#)] syntax for this matching rule is a SEQUENCE Of DirectoryString. Since the Postal Address syntax is such a sequence, it is used in defining the matching rule for LDAP, although the matching rule can be used with any SEQUENCE OF DirectoryString syntax/assertion.

```
( 2.5.13.11 NAME 'caseIgnoreListMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 ) ; Postal Address
```

This matching rule is used to test equality.

[4.5](#) caseIgnoreMatch

The following ABNF associates the caseIgnoreMatch rule with the Directory String syntax:

```
( 2.5.13.2 NAME 'caseIgnoreMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 ) ; Directory String
```

This matching rule is used to test equality.

[4.6](#) caseIgnoreOrderingMatch

The following ABNF associates the caseIgnoreOrderingMatch rule with the Directory String syntax:

```
( 2.5.13.3 NAME 'caseIgnoreOrderingMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 ) ; Directory String
```

This matching rule is used to test inequality, i.e., greaterOrEqual or lessOrEqual.

The sort ordering for a caseIgnoreOrderingMatch is implementation-dependent.

Dally, Legg

Expires 4 May 2003

[Page 24]

[4.7](#) **caseIgnoreSubstringsMatch**

The following ABNF associates the caseIgnoreSubstringsMatch rule with the Substring Assertion:

```
( 2.5.13.4 NAME 'caseIgnoreSubstringsMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 ) ; Substring Assertion
```

This matching rule is used to test substrings equality.

[4.8](#) **distinguishedNameMatch**

The following ABNF associates the distinguishedNameMatch rule with the DN syntax:

```
( 2.5.13.1 NAME 'distinguishedNameMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 ) ; DN
```

This matching rule is used to test equality.

[4.9](#) **generalizedTimeMatch**

The following ABNF associates the generalizedTimeMatch rule with the Generalized Time syntax:

```
( 2.5.13.27 NAME 'generalizedTimeMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 ) ; Generalized Time
```

This matching rule is used to test equality.

[4.10](#) **generalizedTimeOrderingMatch**

```
( 2.5.13.28 NAME 'generalizedTimeOrderingMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 ) ; Generalized Time
```

This matching rule is used to test inequality, i.e., greaterOrEqual or lessOrEqual.

[4.11](#) **integerFirstComponentMatch**

The following ABNF associates the integerFirstComponentMatch rule with the INTEGER syntax:

```
( 2.5.13.29 NAME 'integerFirstComponentMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 ) ; INTEGER
```

Implementors, note that the assertion syntax of this matching rule, an INTEGER, is different from the value syntax of attributes for which this is the equality matching rule.

This matching rule is used to test equality with the first component in a compound syntax.

[4.12](#) integerMatch

The following ABNF associates the integerMatch rule with the INTEGER syntax:

```
( 2.5.13.14 NAME 'integerMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 ) ; INTEGER
```

This matching rule is used to test equality.

[4.13](#) numericStringMatch

The following ABNF associates the numericStringMatch rule with the Numeric String syntax:

```
( 2.5.13.8 NAME 'numericStringMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.36 ) ; Numeric String
```

This matching rule is used to test equality.

[4.14](#) numericStringSubstringsMatch

```
( 2.5.13.10 NAME 'numericStringSubstringsMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 ) ; Substring Assertion
```

This matching rule is used to test substrings equality.

[4.15](#) objectIdentifierFirstComponentMatch

The following ABNF associates the objectIdentifierFirstComponentMatch rule with the OID syntax:

```
( 2.5.13.31 NAME 'objectIdentifierFirstComponentMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 ) ; OID
```

If the client supplies an extensible filter using an objectIdentifierFirstComponentMatch whose matchValue is in the "descr" form, and the OID is not recognized by the server, then the filter is Undefined.

This matching rule is used to test equality with the first component in a compound syntax.

[4.16](#) objectIdentifierMatch

The following ABNF associates the objectIdentifierMatch rule with the OID syntax:

```
( 2.5.13.0 NAME 'objectIdentifierMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 ) ; OID
```

This matching rule is used to test equality.

Dally, Legg

Expires 4 May 2003

[Page 26]

Implementors, note that the assertion syntax of this matching rule, an OID, is different from the value syntax of attributes for which this is the equality matching rule.

If the client supplies a filter using an `objectIdentifierMatch` whose `matchValue oid` is in the "descr" form, and the oid is not recognized by the server, then the filter is Undefined.

[4.17](#) **octetStringMatch**

Servers which implement the `extensibleMatch` filter SHOULD allow the matching rule listed in this section to be used in the `extensibleMatch`. In general these servers SHOULD allow matching rules to be used with all attribute types known to the server, when the assertion syntax of the matching rule is the same as the value syntax of the attribute.

The Octet String Match rule compares for equality an asserted octet string with an attribute value of type OCTET STRING.

The strings match if they are the same length and corresponding octets are identical.

The following ABNF associates the `octetStringMatch` rule with the OCTET STRING syntax:

```
( 2.5.13.17 NAME 'octetStringMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

[4.18](#) **presentationAddressMatch**

The following ABNF associates the `presentationAddressMatch` rule with the Presentation Address syntax:

```
( 2.5.13.22 NAME 'presentationAddressMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.43 ) ; Presentation Address
```

This matching rule is used to test equality.

[4.19](#) **protocolInformationMatch**

The following ABNF associates the `protocolInformationMatch` rule with the Protocol Information syntax:

```
( 2.5.13.24 NAME 'protocolInformationMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.42 ) ; Protocol Information
```

This matching rule is used to test equality.

Dally, Legg

Expires 4 May 2003

[Page 27]

[4.20](#) telephoneNumberMatch

The following ABNF associates the telephoneNumberMatch rule with the Telephone Number syntax:

```
( 2.5.13.20 NAME 'telephoneNumberMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 ) ; Telephone Number
```

This matching rule is used to test equality.

[4.21](#) telephoneNumberSubstringsMatch

The following ABNF associates the telephoneNumberSubstringsMatch rule with the Substring Assertion syntax:

```
( 2.5.13.21 NAME 'telephoneNumberSubstringsMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 ) ; Substring Assertion
```

This matching rule is used to test substrings equality.

[4.22](#) uniqueMemberMatch

The following ABNF associates the uniqueMemberMatch rule with the Name and Optional UID syntax:

```
( 2.5.13.23 NAME 'uniqueMemberMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.34 ) ; Name And Optional UID
```

This matching rule is used to test equality.

[5.](#) Security Considerations

[5.1](#) Disclosure

Attributes of directory entries are used to provide descriptive information about the real-world objects they represent, which can be people, organizations or devices. Most countries have privacy laws regarding the publication of information about people.

[5.2](#) Security Information Syntaxes

Several X.500 attributes, such as, the userCertificate attribute, are used to include key-based security information in directory entries. The attribute syntaxes for these attributes are:

```
Certificate  
CertificateList  
CertificatePair
```

SupportedAlgorithm

These syntaxes are specified for LDAP by the PKIX Working Group, and so, are not included in this document.

The ABNF specifications of "User Certificate", "Authority Revocation List", and "Certificate Pair" in [RFC 1778](#) [[RFC1778](#)] are not to be used.

[5.3](#) Securing the Directory

In order to protect the directory and its contents, strong authentication MUST have been used to identify the Client when an update operation is requested.

[6.](#) Acknowledgements

This document is an update of [RFC 2252](#) by M. Wahl, A. Coulbeck, T. Howes, and S. Kille. [RFC 2252](#) was a product of the IETF ASID Working Group.

This document is based upon input of the IETF LDAPBIS working group. The authors wish to thank J. Sermersheim and K. Zeilenga for their significant contribution to this update.

[7.](#) Authors' Addresses

Kathy Dally
The MITRE Corp.
7515 Colshire Dr., ms-W650
McLean VA 22102
USA

Phone: +1 703 883 6058
Fax: +1 703 883 7142
Email: kdally@mitre.org

Steven Legg
Adacel Technologies Ltd.
405-409 Ferntree Gully Road
Mount Waverley, Victoria 3149
AUSTRALIA

Phone: +61 3 9451 2107
Fax: +61 3 9541 2121
EMail: steven.legg@adacel.com.au

Dally, Legg

Expires 4 May 2003

[Page 29]

8. References

8.1 Normative

- [ABNF] Crocker, D., Overell, P., "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997
- [E.123] Notation for national and international telephone numbers, ITU-T Recommendation E.123, 1988.
- [IA5] International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) Information Technology - 7-Bit Coded Character Set for Information Interchange, ITU-T Recommendation T.50, 1992
- [ISO3166] ISO 3166, "Codes for the representation of names of countries".
- [ISO10646] Universal Multiple-Octet Coded Character Set (UCS) - Architecture and Basic Multilingual Plane, ISO/IEC 10646-1: 1993 (with amendments).
- [JPEG] JPEG File Interchange Format (Version 1.02). Eric Hamilton, C-Cube Microsystems, Milpitas, CA, September 1, 1992.
- [LDAPDN] K. Zeilenga (editor), "LDAP: String Representation of Distinguished Names", [draft-ietf-ldapbis-dn-xx.txt](#), (a work in progress).
- [Protocol] J. Sermersheim (editor), "LDAP: The Protocol", [draft-ietf-ldapbis-protocol-xx.txt](#), a work in progress.
- [RFC1327] Hardcastle-Kille, S., "Mapping between X.400(1988)/ISO 10021 and [RFC 822](#)", [RFC 1327](#), May 1992.
- [RFC2044] Yergeau, F., "UTF-8, a transformation format of Unicode and ISO 10646", [RFC 2044](#), October 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997.
- [RFC 2256] [draft-ietf-ldapbis-user-schema-xx](#), replacement for Wahl, M., "A Summary of the X.500(96) User Schema for use with LDAPv3", [RFC 2256](#), December 1997.
- [T.4] Standardization of Group 3 facsimile apparatus for document transmission - Terminal Equipment and Protocols for Telematic Services, ITU-T Recommendation T.4, 1993

[X.501] The Directory: Models, ITU-T Recommendation X.501, 1993.

[X.520] The Directory: Selected Attribute Types. ITU-T
Recommendation X.520, 1993.

[X.680] Information Technology - Abstract Syntax Notation One (ASN.1): Specification of Basic Notation, ITU-T Recommendation X.680, 1994

8.2 Informative References

[ISO8348] Information technology -- Open Systems Interconnection -- Network Service Definition, ISO/IEC 8348:1996

[RFC1777] Yeong, W., Howes, T., Kille, S., "Lightweight Directory Access Protocol", [RFC 1777](#), March 1995

[RFC1778] Howes, T., Kille, S., Yeong, W., Robbins, C., "The String Representation of Standard Attribute Syntaxes", [RFC 1778](#), March 1995.

[RFC2253] Wahl, M., Kille, S., and T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", [RFC 2253](#), December 1997.

9. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Dally, Legg

Expires 4 May 2003

[Page 31]

Annex A Object Identifiers of Syntaxes

This list contains the object identifiers for the syntaxes used in this specification and in the user schema specification [User].

Syntax of Value Represented	OBJECT IDENTIFIER
=====	
Attribute Type Description	1.3.6.1.4.1.1466.115.121.1.3
Bit String	1.3.6.1.4.1.1466.115.121.1.6
Boolean	1.3.6.1.4.1.1466.115.121.1.7
Country String	1.3.6.1.4.1.1466.115.121.1.11
Delivery Method	1.3.6.1.4.1.1466.115.121.1.14
Directory String	1.3.6.1.4.1.1466.115.121.1.15
DIT Content Rule Description	1.3.6.1.4.1.1466.115.121.1.16
DIT Structure Rule Description	1.3.6.1.4.1.1466.115.121.1.17
DN	1.3.6.1.4.1.1466.115.121.1.12
Enhanced Guide	1.3.6.1.4.1.1466.115.121.1.21
Facsimile Telephone Number	1.3.6.1.4.1.1466.115.121.1.22
Fax	1.3.6.1.4.1.1466.115.121.1.23
Generalized Time	1.3.6.1.4.1.1466.115.121.1.24
Guide	1.3.6.1.4.1.1466.115.121.1.25
IA5 String	1.3.6.1.4.1.1466.115.121.1.26
INTEGER	1.3.6.1.4.1.1466.115.121.1.27
JPEG	1.3.6.1.4.1.1466.115.121.1.28
LDAP Syntax Description	1.3.6.1.4.1.1466.115.121.1.54
Matching Rule Description	1.3.6.1.4.1.1466.115.121.1.31
Matching Rule Use Description	1.3.6.1.4.1.1466.115.121.1.31
MHS OR Address	1.3.6.1.4.1.1466.115.121.1.33
Name And Optional UID	1.3.6.1.4.1.1466.115.121.1.34
Name Form Description	1.3.6.1.4.1.1466.115.121.1.35
Numeric String	1.3.6.1.4.1.1466.115.121.1.36
Object Class Description	1.3.6.1.4.1.1466.115.121.1.37
Octet String	1.3.6.1.4.1.1466.115.121.1.40
OID	1.3.6.1.4.1.1466.115.121.1.38
Other Mailbox	1.3.6.1.4.1.1466.115.121.1.39
Postal Address	1.3.6.1.4.1.1466.115.121.1.41
Presentation Address	1.3.6.1.4.1.1466.115.121.1.43
Printable String	1.3.6.1.4.1.1466.115.121.1.44
Substring Assertion	1.3.6.1.4.1.1466.115.121.1.58
Telephone Number	1.3.6.1.4.1.1466.115.121.1.50
Teletex Terminal Identifier	1.3.6.1.4.1.1466.115.121.1.51
Telex Number	1.3.6.1.4.1.1466.115.121.1.52
UTC Time	1.3.6.1.4.1.1466.115.121.1.53

Dally, Legg

Expires 4 May 2003

[Page 32]

Annex B Topics Yet To Be Addressed In This Document

This appendix is provided for informational purposes only, it is not a normative part of this specification.

APPEARED: -00

Paragraph 2.2.3 - Should any syntaxes listed in the table be removed? Should any new syntaxes be added?

RESOLUTION: Cannot add syntaxes. Moving the table to an annex keeps a record of the OIDS that have been assigned. Deleted unspecified syntaxes from the list. APPLIED: -02

APPEARED: -00

Paragraph 2.2.4 - Should attribute syntaxes be allowed to be referenced by a common name, and if so, where should the name come from?

RESOLUTION: Rejected because of adding functionality. APPLIED: -01

APPEARED: -00

How does the data model draft <[draft-wahl-ladpv3-defns-01.txt](#)> affect this draft?

RESOLUTION: It does not. The draft was preliminary to the revised Schema and Protocol I-Ds. APPLIED: -01

APPEARED: -00

[Section 3](#) - Should all listed syntaxes from paragraph 2.2.3 be detailed in this section? Nearly half the listed syntaxes are not referenced in this section.

RESOLUTION: No, because many are not being used, currently.

APPLIED: -01

APPEARED: -01

[Section 4](#) - Should all of the X.520(1993) matching rules be included? In particular, how about caseExactMatch? Also, should octetStringMatch be moved from updated [RFC 2256](#)?

RESOLUTION: caseExactMatch not included. octetStringMatch moved to this document. APPLIED: -01

APPEARED: -00

[Section 6](#) - Recognized list of Object classes needs to be reconciled with updated [RFC 2256](#) and the data model draft.

RESOLUTION: Not necessary. APPLIED: -01

APPEARED: -00

[Section 7](#) - Proper security statement needs to be formulated.

RESOLUTION: Text has been expanded since [RFC 2252](#), but needs more work. APPLIED:

Dally, Legg

Expires 4 May 2003

[Page 33]

Annex C Change Log

This annex lists the changes that have been made from [RFC 2252](#) to this specification.

This annex is provided for informational purposes only. It is not a normative part of this specification. Items 32 - end are new in the -02 version of this document.

-00 changes

1. Removed the IESG Note.
2. Changed "types" to "syntaxes" in the last sentence of the Abstract. Also, added to the last sentence in order to indicate that syntaxes are not the only schema elements defined in this document.
3. Reorganized the sections so that:
 - * the schema element categories are specified in the order in which they build on one another: syntaxes, matching rules, attributes, object classes
 - * within each category the elements are specified in alphabetical order
4. Added an "Implementation Status" paragraph for each element, gathering the conformance statements.
5. Clarified schema description in the Overview.
6. Changed the "Common Encoding Aspects" section title to "Notation" and made corresponding changes throughout the document. The purpose being to relegate all encoding issues to the Protocol specification [[Protocol](#)].
7. Added a MUST statement regarding the syntaxes required of servers.
8. Expanded the discussion of each of the syntaxes in [section 3](#).
9. Added examples to some of the syntax descriptions.
10. Added NAME option to the syntax description ABNF in 2.2.4.

RESCINDED IN -01!!

11. Added a note deprecating the UTCTime attribute syntax description in 3.41

12. In the ABNF of the MatchingRuleDescription in paragraph 2.3.2, replaced "numericoid" with "oid".
13. In paragraph 2.4.1, replaced the conformance statement about attributes in 2256 with a reference.
14. Added caseIgnoreIA5Match as the EQUALITY matching rule for the altServer attribute type ABNF in paragraph 5.1. Note that this could be caseExactIA5Match instead. SHOULD IT BE??

RESCINDED IN -01

15. In paragraphs 5.10 and 5.11, changed "the MODIFY operation" to "LDAP update operations"
16. Added distinguishedNameMatch as the EQUALITY matching rule for the namingContexts attribute type ABNF in paragraph 5.13.

RESCINDED IN -01

17. Reworded paragraph 5.15.
18. Added distinguishedNameMatch as the EQUALITY matching rule for the namingContexts attribute type ABNF in paragraph 5.13.

RESCINDED IN -01

19. Added integerMatch as the EQUALITY and integerOrderingMatch as the Ordering matching rules for the supportedLDAPVersion attribute type ABNF in paragraph 5.18.

RESCINDED IN -01

20. Added caseIgnoreMatch as the EQUALITY matching rule for the supportedSASLMechanisms attribute type ABNF in paragraph 5.19. Note that this could be caseExactMatch instead. SHOULD IT BE??

RESCINDED IN -01

21. Made corrections to the ABNF in paragraph 3.12.
22. Added the seven syntax definitions from [RFC 2256](#) and ordered the definitions alphabetically.
23. Changed the "Bibliography" section title to "References".
24. Replaced the X.208 reference with one to X.680(1994), since X.680 is the ASN.1 referred to in the X.500(1993)-series.

Dally, Legg

Expires 4 May 2003

[Page 35]

-01 changes

25. Moved the table listing the syntaxes and their oids from paragraph 2.2.3 to a new Annex A.
- REMOVED SYNTAXES NOT DEFINED IN THIS I-D FROM THE LIST - 02
26. Moved the specification of the octetStringMatch matching rule from [RFC 2256](#) to [section 4](#) of this document.
27. Throughout this I-D, cleaned up whitespace in the ABNF definitions.
28. In [Section 2.1](#):
 - * Corrected the characters defined in the p rule to match the PrintableString syntax.
 - * Deleted the letterstring rule.
 - * Modified the utf8 and dstring rules according to a suggestion from K. Zeilenga.
 - * Deleted ";" from the keychar rule, which affects the anhstring, keystring, and descr rules.
 - * Removed the length option from the numericoid rule
29. In [section 2.2](#), deleted the sentence about needing a new OID when a syntax is modified.
30. In [section 2.2](#), replaced the editor's proposal and subject text with explanation of the LDAP-specific encoding of attribute values.
31. Removed [section 2.2.2](#) (and renumbered the remainder of [section 2.2](#)), leaving the description of binary encoding to the protocol I-D.

-02 changes

32. Revised specifications to use ABNF [[ABNF](#)] instead of BNF throughout the document.
33. Removed embedded comments from the ABNF productions throughout the document.
34. Removed the Binary syntax because it was not adequately specified, implementations with different interpretations exist, and it was confused with the ;binary transfer encoding.
35. Removed the syntaxes, which are not defined in this document, from the list in Annex A. Consult [RFC 2252](#) for the

assignments made previously for syntaxes that have not been defined to date.

36. Inserted the specification of the octetstring production, from [RFC 2234](#) [[ABNF](#)].j

- 37. Cleaned up the references; adopted word instead of number tags; split [Section 10](#) into normative and informative subsections.
- 38. Inserted ABNF from [RFC 1278](#) in place of a reference.
- 39. Deleted the certificate-related syntaxes and noted in the Security Considerations ([Section 7](#)) that they are covered in PKIX WG documents.

-03 changes

- 40. Removed all discussion of transfer options and the binary option.
- 41. Aligned the text to the [MODELS] document.

