

The String Representation of Standard Attribute Syntaxes

[<draft-ietf-asid-ldapv2-attributes-00.txt>](#)

1. Status of this Memo

This draft document will be submitted to the RFC Editor as a standards document. Distribution of this memo is unlimited. Please send comments to the authors, or the discussion group [<osi-ds@cs.ucl.ac.uk>](mailto:osi-ds@cs.ucl.ac.uk).

This document is an Internet-Draft. 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.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt' listing contained in the Internet-Drafts Shadow Directories on ds.internic.net (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

2. Abstract

The Lightweight Directory Access Protocol (LDAP) [9] requires that the contents of AttributeValue fields in protocol elements be octet strings. This document defines the requirements that must be satisfied by encoding rules used to render X.500 Directory attribute syntaxes into a form suitable for use in the LDAP, then goes on to define the encoding rules for the standard set of attribute syntaxes defined in [1,2] and [3].

3. Attribute Syntax Encoding Requirements.

This section defines general requirements for lightweight directory protocol attribute syntax encodings. All documents defining attribute syntax encodings for use by the lightweight directory protocols are expected to conform to these requirements.

The encoding rules defined for a given attribute syntax must produce octet strings. To the greatest extent possible, encoded octet strings should be usable in their native encoded form for display purposes. In particular, encoding rules for attribute syntaxes defining non-binary values should produce strings that can be displayed with little or no translation by clients implementing the lightweight directory protocols.

4. Table of LDAP Attributes

This section lists all Attribute Type names defined for this version of LDAP. Servers may support additional names and attributes not listed here by bilateral agreement.

4.1. Standard User Attributes

The attributes listed in this section are those defined in X.520(1988), likely to be present in user entries.

Attribute Type Name	OID	Syntax
=====	=====	=====
objectClass	2.5.4.0	OID
aliasedObjectName	2.5.4.1	DN
knowledgeInformation	2.5.4.2	caseIgnoreString
cn	2.5.4.3	caseIgnoreString
sn	2.5.4.4	caseIgnoreString
serialNumber	2.5.4.5	PrintableString
c	2.5.4.6	CountryString
l	2.5.4.7	caseIgnoreString
st	2.5.4.8	caseIgnoreString
street	2.5.4.9	caseIgnoreString
o	2.5.4.10	caseIgnoreString
ou	2.5.4.11	caseIgnoreString
title	2.5.4.12	caseIgnoreString
description	2.5.4.13	caseIgnoreString
searchGuide	2.5.4.14	Guide
businessCategory	2.5.4.15	caseIgnoreString
postalAddress	2.5.4.16	PostalAddress
postalCode	2.5.4.17	caseIgnoreString
postOfficeBox	2.5.4.18	caseIgnoreString
physicalDeliveryOfficeName	2.5.4.19	caseIgnoreString
telephoneNumber	2.5.4.20	TelephoneNumber

Expires 11/11/96

[Page 2]

telexNumber	2.5.4.21	TelexNumber
teletexTerminalIdentifier	2.5.4.22	TeletexTerminalIdentifier
facsimileTelephoneNumber	2.5.4.23	FacsimileTelephoneNumber
x121Address	2.5.4.24	NumericString
internationaliSDNNumber	2.5.4.25	NumericString
registeredAddress	2.5.4.26	PostalAddress
destinationIndicator	2.5.4.27	PrintableString
preferredDeliveryMethod	2.5.4.28	DeliveryMethod
presentationAddress	2.5.4.29	PresentationAddress
supportedApplicationContext	2.5.4.30	OID
member	2.5.4.31	DN
owner	2.5.4.32	DN
roleOccupant	2.5.4.33	DN
seeAlso	2.5.4.34	DN
userPassword	2.5.4.35	Password
userCertificate	2.5.4.36	Certificate
cACertificate	2.5.4.37	Certificate
authorityRevocationList	2.5.4.38	CertificateList
certificateRevocationList	2.5.4.39	CertificateList
crossCertificatePair	2.5.4.40	CertificatePair

4.2. Pilot User Attributes

These attributes are defined in [RFC 1274](#).

Attribute Type Name	OID	Syntax
=====	=====	=====
uid	0.9.2342.19200300.100.1.1	CaseIgnoreString
textEncodedORaddress	0.9.2342.19200300.100.1.2	CaseIgnoreString
mail	0.9.2342.19200300.100.1.3	CaseIgnoreIA5String
info	0.9.2342.19200300.100.1.4	CaseIgnoreString
drink	0.9.2342.19200300.100.1.5	CaseIgnoreString
roomNumber	0.9.2342.19200300.100.1.6	CaseIgnoreString
photo	0.9.2342.19200300.100.1.7	Fax
userClass	0.9.2342.19200300.100.1.8	CaseIgnoreString
host	0.9.2342.19200300.100.1.9	CaseIgnoreString
manager	0.9.2342.19200300.100.1.10	DN
documentIdentifier	0.9.2342.19200300.100.1.11	CaseIgnoreString
documentTitle	0.9.2342.19200300.100.1.12	CaseIgnoreString
documentVersion	0.9.2342.19200300.100.1.13	CaseIgnoreString
documentAuthor	0.9.2342.19200300.100.1.14	DN
documentLocation	0.9.2342.19200300.100.1.15	CaseIgnoreString
homePhone	0.9.2342.19200300.100.1.20	TelephoneNumber
secretary	0.9.2342.19200300.100.1.21	DN
otherMailbox	0.9.2342.19200300.100.1.22	OtherMailbox
lastModifiedTime	0.9.2342.19200300.100.1.23	UTCTime
lastModifiedBy	0.9.2342.19200300.100.1.24	DN

Expires 11/11/96

[Page 3]

dc	0.9.2342.19200300.100.1.25	CaseIgnoreIA5String
dNSRecord	0.9.2342.19200300.100.1.26	IA5String
mXRecord	0.9.2342.19200300.100.1.28	IA5String
nSRecord	0.9.2342.19200300.100.1.29	IA5String
sOARRecord	0.9.2342.19200300.100.1.30	IA5String
cNAMERRecord	0.9.2342.19200300.100.1.31	IA5String
associatedDomain	0.9.2342.19200300.100.1.37	CaseIgnoreIA5String
associatedName	0.9.2342.19200300.100.1.38	DN
homePostalAddress	0.9.2342.19200300.100.1.39	PostalAddress
personalTitle	0.9.2342.19200300.100.1.40	CaseIgnoreString
mobile	0.9.2342.19200300.100.1.41	TelephoneNumber
pager	0.9.2342.19200300.100.1.42	TelephoneNumber
co	0.9.2342.19200300.100.1.43	CaseIgnoreString
organizationalStatus	0.9.2342.19200300.100.1.45	CaseIgnoreString
janetMailbox	0.9.2342.19200300.100.1.46	CaseIgnoreIA5String
mailPreferenceOption	0.9.2342.19200300.100.1.47	MailPreference
buildingName	0.9.2342.19200300.100.1.48	CaseIgnoreString
personalSignature	0.9.2342.19200300.100.1.53	Fax
dITRedirect	0.9.2342.19200300.100.1.54	DN
audio	0.9.2342.19200300.100.1.55	Audio
documentPublisher	0.9.2342.19200300.100.1.56	CaseIgnoreString
jpegPhoto	0.9.2342.19200300.100.1.60	JPEG

5. Standard Attribute Syntax Encodings

For the purposes of defining the encoding rules for the standard attribute syntaxes, the following auxiliary BNF definitions will be used:

```

<a> ::= 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' |
        'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' |
        's' | 't' | 'u' | 'v' | 'w' | 'x' | 'y' | 'z' | 'A' |
        'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' |
        'K' | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' |
        'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z'

<d> ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

<hex-digit> ::= <d> | 'a' | 'b' | 'c' | 'd' | 'e' | 'f' |
                'A' | 'B' | 'C' | 'D' | 'E' | 'F'

<k> ::= <a> | <d> | '-'

<p> ::= <a> | <d> | ''' | '(' | ')' | '+' | ',' | '-' | '.' |
        '/' | ':' | '?' | ' '

<CRLF> ::= The ASCII newline character with hexadecimal value 0x0A

<letterstring> ::= <a> | <a> <letterstring>

```

Expires 11/11/96

[Page 4]

```
<numericstring> ::= <d> | <d> <numericstring>

<keystring> ::= <a> | <a> <anhstring>

<anhstring> ::= <k> | <k> <anhstring>

<printablestring> ::= <p> | <p> <printablestring>

<space> ::= ' ' | ' ' <space>
```

[5.1.](#) Undefined

This syntax is to be used for any values whose syntax is not defined by another section of this document. Values of type Undefined are encoded as if they were values of type Octet String, with the string value being the BER-encoded version of the value.

[5.2.](#) Case Ignore String

A string of type caseIgnoreStringSyntax is encoded as the string value itself.

[5.3.](#) Case Exact String

The encoding of a string of type caseExactStringSyntax is the string value itself.

[5.4.](#) Printable String

The encoding of a string of type printableStringSyntax is the string value itself.

[5.5.](#) Numeric String

The encoding of a string of type numericStringSyntax is the string value itself.

[5.6.](#) Octet String

The encoding of a string of type octetStringSyntax is the string value itself.

[5.7.](#) Case Ignore IA5 String

The encoding of a string of type caseIgnoreIA5String is the string value itself.

Expires 11/11/96

[Page 5]

[5.8.](#) IA5 String

The encoding of a string of type `ia5StringSyntax` is the string value itself.

[5.9.](#) T61 String

The encoding of a string of type `t61StringSyntax` is the string value itself.

[5.10.](#) Case Ignore List

Values of type `caseIgnoreListSyntax` are encoded according to the following BNF:

```
<caseignorelist> ::= <caseignorestring> |  
                    <caseignorestring> '$' <caseignorelist>  
  
<caseignorestring> ::= a string encoded according to the rules for Case  
                        Ignore String as above.
```

[5.11.](#) Case Exact List

Values of type `caseExactListSyntax` are encoded according to the following BNF:

```
<caseexactlist> ::= <caseexactstring> |  
                   <caseexactstring> '$' <caseexactlist>  
  
<caseexactstring> ::= a string encoded according to the rules for Case  
                      Exact String as above.
```

[5.12.](#) Distinguished Name

Values of type `distinguishedNameSyntax` are encoded to have the representation defined in [\[5\]](#).

[5.13.](#) Boolean

Values of type `booleanSyntax` are encoded according to the following BNF:

```
<boolean> ::= "TRUE" | "FALSE"
```

Boolean values have an encoding of "TRUE" if they are logically true, and have an encoding of "FALSE" otherwise.

Expires 11/11/96

[Page 6]

5.14. Integer

Values of type `integerSyntax` are encoded as the decimal representation of their values, with each decimal digit represented by the its character equivalent. So the digit 1 is represented by the character '1', the digit 2 is represented by the character '2' and so on.

5.15. Object Identifier

Values of type `objectIdentifierSyntax` are encoded according to the following BNF:

```
<oid> ::= <descr> | <descr> '.' <numericoid> | <numericoid>

<descr> ::= <keystring>

<numericoid> ::= <numericstring> | <numericstring> '.' <numericoid>
```

In the above BNF, `<descr>` is the syntactic representation of an object descriptor. When encoding values of type `objectIdentifierSyntax`, the first encoding option should be used in preference to the second, which should be used in preference to the third wherever possible. That is, in encoding object identifiers, object descriptors (where assigned and known by the implementation) should be used in preference to numeric oids to the greatest extent possible. For example, in encoding the object identifier representing an `organizationName`, the descriptor ```organizationName''` is preferable to ```ds.4.10''`, which is in turn preferable to the string ```2.5.4.10''`.

5.16. Telephone Number

Values of type `telephoneNumberSyntax` are encoded as if they were Printable String types.

5.17. Telex Number

Values of type `telexNumberSyntax` are encoded according to the following BNF:

```
<telex-number> ::= <actual-number> '$' <country> '$' <answerback>

<actual-number> ::= <printablestring>

<country> ::= <printablestring>

<answerback> ::= <printablestring>
```

In the above, `<actual-number>` is the syntactic representation of the number

Expires 11/11/96

[Page 7]

portion of the TELEX number being encoded, <country> is the TELEX country code, and <answerback> is the answerback code of a TELEX terminal.

5.18. Teletex Terminal Identifier

Values of type `teletexTerminalIdentifier` are encoded according to the following BNF:

```
<teletex-id> ::= <printablestring> 0*('$' <ttx-param>)  
  
<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 <printablestrings> are subsequent portions of the teletex terminal identifier.

5.19. Facsimile Telephone Number

Values of type `FacsimileTelephoneNumber` are encoded according to the following BNF:

```
<fax-number> ::= <printablestring> [ '$' <faxparameters> ]  
  
<faxparameters> ::= <faxparm> | <faxparm> '$' <faxparameters>  
  
<faxparm> ::= 'twoDimensional' | 'fineResolution' | 'unlimitedLength' |  
             'b4Length' | 'a3Width' | 'b4Width' | 'uncompressed'
```

In the above, the first <printablestring> is the actual fax number, and the <faxparm> tokens represent fax parameters.

5.20. Presentation Address

Values of type `PresentationAddress` are encoded to have the representation described in [6].

5.21. UTC Time

Values of type `uTCTimeSyntax` are encoded as if they were Printable Strings with the strings containing a UTCTime value.

Expires 11/11/96

[Page 8]

5.22. Guide (search guide)

Values of type Guide, such as values of the searchGuide attribute, are encoded according to the following BNF:

```

<guide-value> ::= [ <object-class> '#' ] <criteria>

<object-class> ::= an encoded value of type objectIdentifierSyntax

<criteria> ::= <criteria-item> | <criteria-set> | '!' <criteria>

<criteria-set> ::= [ '(' ] <criteria> '&' <criteria-set> [ ')' ] |
                  [ '(' ] <criteria> '|' <criteria-set> [ ')' ]

<criteria-item> ::= [ '(' ] <attributetype> '$' <match-type> [ ')' ]

<match-type> ::= "EQ" | "SUBSTR" | "GE" | "LE" | "APPROX"

```

5.23. Postal Address

Values of type PostalAddress are encoded according to the following BNF:

```

<postal-address> ::= <t61string> | <t61string> '$' <postal-address>

```

In the above, each <t61string> component of a postal address value is encoded as a value of type t61StringSyntax.

5.24. User Password

Values of type userPasswordSyntax are encoded as if they were of type octetStringSyntax.

5.25. User Certificate

Values of type userCertificate are encoded according to the following BNF:

```

<certificate> ::= <version> '#' <serial> '#' <signature-algorithm-id>
                  '#' <issuer> '#' <validity> '#' <subject>
                  '#' <public-key-info> '#' <encrypted-sign-value>

<version> ::= <integervalue>

<serial> ::= <integervalue>

<signature-algorithm-id> ::= <algorithm-id>

```


Expires 11/11/96

[Page 9]

```

<issuer> ::= an encoded Distinguished Name

<validity> ::= <not-before-time> '#' <not-after-time>

<not-before-time> ::= <utc-time>

<not-after-time> ::= <utc-time>

<algorithm-parameters> ::= <null> | <integervalue> |
                           '{ASN}' <hex-string>

<subject> ::= an encoded Distinguished Name

<public-key-info> ::= <algorithm-id> '#' <encrypted-sign-value>

<encrypted-sign-value> ::= <hex-string> | <hex-string> '-' <d>

<algorithm-id> ::= <oid> '#' <algorithm-parameters>

<utc-time> ::= an encoded UTCTime value

<hex-string> ::= <hex-digit> | <hex-digit> <hex-string>

```

Note that this certificate format is appropriate for reading, but cannot be guaranteed to be verifiable. This is because the string DN format used to encode the issuer and subject portions of the certificate does not produce a completely reversible encoding (i.e., one cannot always produce the original DER-encoded certificate from its string representation). By bilateral agreement, sites are free to exchange native DER-encoded certificates that can be verified, but via an attribute type name other than "userCertificate" or "caCertificate".

[5.26.](#) CA Certificate

Values of type caCertificate are encoded as if the values were of type userCertificate.

[5.27.](#) Authority Revocation List

Values of type authorityRevocationList are encoded according to the following BNF:

```

<certificate-list> ::= <signature-algorithm-id> '#' <issuer> '#' <utc-
time>
                        [ '#' <revoked-certificates> ]
                        '#' <signature-algorithm-id>
                        '#' <encrypted-sign-value>

<revoked-certificates> ::= 1*( '#' <revoked-certificate> )

```

Expires 11/11/96

[Page 10]

```
<signature-algorithm-id> '#' <encrypted-sign-  
value>
```

```
<revoked-certificate> ::= <signature-algorithm-id> '#' <issuer> '#'  
                           <serial> '#' <utc-time>
```

The syntactic components <signature-algorithm-id>, <issuer>, <encrypted-sign-value>, <utc-time>, <subject> and <serial> have the same definitions as in the BNF for the userCertificate attribute syntax.

Note that as with the "User Certificate" syntax above, values encoded in this syntax are not guaranteed to be verifiable. Also, servers which implement or gateway to Directory systems supporting the 1993 or later editions of the X.500 specifications may not be able to generate or parse LDAP authority or certificate revocation lists, as the format described in this section (based on the 1988 edition of X.509) is not compatible with the syntax of X.509(1993).

[5.28.](#) Certificate Revocation List

Values of type certificateRevocationList are encoded as if the values were of type authorityRevocationList.

[5.29.](#) Cross Certificate Pair

Values of type crossCertificatePair are encoded according to the following BNF:

```
<certificate-pair> ::= <forward> '#' <reverse>  
                        | <forward>  
                        | <reverse>  
  
<forward> ::= 'forward:' <certificate>  
  
<reverse> ::= 'reverse:' <certificate>
```

The syntactic component <certificate> has the same definition as in the BNF for the userCertificate attribute syntax.

Note that as with the "User Certificate" syntax above, values encoded in this syntax are not guaranteed to be verifiable. Also, servers which implement or gateway to Directory systems supporting the 1993 or later editions of the X.500 specifications may not be able to generate or parse LDAP authority or certificate revocation lists, as the format described in this section (based on the 1988 edition of X.509) is not compatible with the syntax of X.509(1993).

Expires 11/11/96

[Page 11]

5.30. Delivery Method

Values of type `deliveryMethod` are encoded according to the following BNF:

```
<delivery-value> ::= <pdm> | <pdm> '$' <delivery-value>

<pdm> ::= 'any' | 'mhs' | 'physical' | 'telex' | 'teletex' |
          'g3fax' | 'g4fax' | 'ia5' | 'videotex' | 'telephone'
```

5.31. Other Mailbox

Values of the type `otherMailboxSyntax` are encoded according to the following BNF:

```
<otherMailbox> ::= <mailbox-type> '$' <mailbox>

<mailbox-type> ::= an encoded Printable String

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

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

5.32. Mail Preference

Values of type `mailPreferenceOption` are encoded according to the following BNF:

```
<mail-preference> ::= "NO-LISTS" | "ANY-LIST" | "PROFESSIONAL-LISTS"
```

5.33. MHS OR Address

Values of type `MHS OR Address` are encoded as strings, according to the format defined in [\[10\]](#).

5.34. Distribution List Submit Permission

Values of type `DLSubmitPermission` are encoded as strings, according to the following BNF:

```
<dlsubmit-perm> ::= <dlgroup_label> ':' <dlgroup-value>
                  | <dl-label> ':' <dl-value>

<dlgroup-label> ::= 'group_member'
```

Expires 11/11/96

[Page 12]

```
<dlgroup-value> ::= <name>

<name> ::= an encoded Distinguished Name

<dl-label> ::= 'individual' | 'dl_member' | 'pattern'

<dl-value> ::= <orname>

<orname> ::= <address> '#' <dn>
           | <address>

<address> ::= <add-label> ':' <oraddress>

<dn> ::= <dn-label> ':' <name>

<add-label> = 'X400'

<dn-label> = 'X500'
```

where <oraddress> is as defined in [RFC 1327](#).

[5.35](#). Photo

Values of type Photo are encoded as if they were octet strings containing JPEG images in the JPEG File Interchange Format (JFIF), as described in [\[8\]](#).

[5.36](#). Fax

Values of type Fax are encoded as if they were octet strings containing Group 3 Fax images as defined in [\[7\]](#).

[6](#). Security Considerations

Security considerations are not discussed in this document.

[7](#). Acknowledgements

Many of the attribute syntax encodings defined in this document are adapted from those used in the QUIPU X.500 implementation. The contributions of the authors of the QUIPU implementation in the specification of the QUIPU syntaxes [\[4\]](#) are gratefully acknowledged.

[8](#). Bibliography

- [1] The Directory: Selected Attribute Syntaxes. CCITT, Recommendation X.520

Expires 11/11/96

[Page 13]

- [2] Information Processing Systems -- Open Systems Interconnection -- The Directory: Selected Attribute Syntaxes
- [3] The COSINE and Internet X.500 Schema. Paul Barker, Steve Kille; Request for Comment (RFC) 1274
- [4] The ISO Development Environment: User's Manual -- Volume 5: QUIPU. Colin Robbins, Stephen E. Kille
- [5] A String Representation of Distinguished Names. Steve Kille, [RFC 1779](#)
- [6] A String Representation for Presentation Addresses. Steve Kille; Request for Comment (RFC) 1278
- [7] Terminal Equipment and Protocols for Telematic Services - Standardization of Group 3 facsimile apparatus for document transmission. CCITT, Recommendation T.4
- [8] JPEG File Interchange Format (Version 1.02). Eric Hamilton, C-Cube Microsystems, Milpitas, CA, September 1, 1992
- [9] Lightweight Directory Access Protocol. Wengyik Yeong, Tim Howes, Steve Kille, Request for Comment (RFC) XXXX
- [10] Mapping between X.400 and [RFC-822](#) Message Bodies. H. Alvestrand, S. Kille, R. Miles, M. Rose, S. Thompson, Request for Comment (RFC) 1495

[9.](#) Author's Addresses

Tim Howes
University of Michigan
ITD Research Systems
535 W William St.
Ann Arbor, MI 48103-4943
USA
+1 313 747-4454
tim@umich.edu

Steve Kille
ISODE Consortium
The Dome, The Square
Richmond
TW9 1DT
UK
+44-181-332-9091
S.Kille@isode.com

Expires 11/11/96

[Page 14]

Wengyik Yeong
PSI Inc.
510 Huntmar Park Drive
Herndon, VA 22070
USA
+1 703-450-8001
yeongw@psilink.com

Colin Robbins
NeXor Ltd
University Park
Nottingham
NG7 2RD
UK

Mark Wahl
ISODE Consortium Inc.
3925 West Braker Lane, Suite 333
Austin, TX 78759
USA
+1 512-305-0280
M.Wahl@isode.com

Expires 11/11/96

[Page 15]