LDAP: Schema for User Applications
draft-ietf-ldapbis-user-schema-10.txt

Status of this Memo

This document is intended to be, after appropriate review and
revision, submitted to the RFC Editor as a Standard Track document.
Distribution of this document is unlimited. Technical discussion of
this document should take place on the IETF LDAP Revision Working
Group (LDAPbis) mailing list <ietf-ldapbis@openldap.org>. Please
send editorial comments directly to the editor
<andrew.sciberras@eb2bcom.com>.

This Internet-Draft expires on 11 January 2006.

Abstract

   This document is an integral part of the Lightweight Directory Access
   Protocol (LDAP) technical specification [Roadmap]. It provides a
   technical specification of attribute types and object classes
   intended for use by LDAP directory clients for many directory
   services, such as, White Pages. These objects are widely used as a
   basis for the schema in many LDAP directories. This document does
   not cover attributes used for the administration of directory
   servers, nor does it include directory objects defined for specific
   uses in other documents.

Table of Contents

1. Introduction

   This document provides an overview of attribute types and object
   classes intended for use by Lightweight Directory Access Protocol
   (LDAP) directory clients for many directory services, such as, White
   Pages. Originally specified in the X.500 [X.500] documents, these
   objects are widely used as a basis for the schema in many LDAP
   directories. This document does not cover attributes used for the
   administration of directory servers, nor does it include directory
   objects defined for specific uses in other documents.

1.1 Relationship with other specifications

   This document is an integral part of the LDAP technical specification
   [Roadmap] which obsoletes the previously defined LDAP technical
   specification, RFC 3377, in its entirety. In terms of RFC 2256,
   Sections 6 and 8 of RFC 2256 are obsoleted by [Syntaxes]. Sections
   5.1, 5.2, 7.1 and 7.2 of RFC 2256 are obsoleted by [Models]. The
   remainder of RFC 2256 is obsoleted by this document. Section 2.4 of
   this document supersedes the technical specification for the 'dc'
   attribute type and 'dcObject' object class found in RFC 2247. The
   remainder of RFC 2247 remains in force.

   This document updates RFC 2798 by replacing the informative
   description of the 'uid' attribute type, with the definitive
   description provided in Section 2.39 of this document.

   A number of schema elements which were included in the previous
   revision of the LDAP Technical Specification are not included in this
   revision of LDAP. PKI-related schema elements are now specified in
   [LDAP-PKI]. Unless reintroduced in future technical specifications,
   the remainder are to be considered Historic.

   The descriptions in this document SHALL be considered definitive for
   use in LDAP.

1.2 Conventions

   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 RFC 2119 [RFC2119].

1.3 General Issues

   This document references Syntaxes defined in Section 3 of [Syntaxes]
   and Matching Rules defined in Section 4 of [Syntaxes].

   The definitions of Attribute Types and Object Classes are written

using the Augmented Backus-Naur Form (ABNF) [RFC2234] of
AttributeTypeDescription and ObjectClassDescription given in
[Models]. Lines have been folded for readability. When such values
are transferred as attribute values in the LDAP Protocol the values
will not contain line breaks.

2. Attribute Types

The Attribute Types contained in this section hold user information.

There is no requirement that servers implement the 'searchGuide' and
'teletexTerminalIdentifier' attribute types. In fact, their use is
greatly discouraged.

An LDAP server implementation SHOULD recognize the rest of the
attribute types described in this section.

2.1 'businessCategory'

The 'businessCategory' attribute type describes the kinds of business
performed by an organization. Each kind is one value of this
multi-valued attribute.
(Source: X.520 [X.520])

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

1.3.6.1.4.1.1466.115.121.1.15 refers to the Directory String syntax
[Syntaxes].

Examples: "banking", "transportation" and "real estate".

2.2 'c'

The 'c' ('countryName' in X.500) attribute type contains a two-letter
ISO 3166 [ISO3166] country code.
(Source: X.520 [X.520])

```
( 2.5.4.6 NAME 'c'
    SUP name
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.11
    SINGLE-VALUE )
```

1.3.6.1.4.1.1466.115.121.1.11 refers to the Country String syntax
[Syntaxes].

Examples: "DE", "AU" and "FR".

2.3 'cn'

The 'cn' ('commonName' in X.500) attribute type contains names of an
object. Each name is one value of this multi-valued attribute. If
the object corresponds to a person, it is typically the person's full
name.
(Source: X.520 [X.520])

       (2.5.4.3 NAME 'cn'
          SUP name)

Examples: "Martin K Smith", "Marty Smith" and "printer12".

2.4 'dc'

The 'dc' ('domainComponent' in RFC 2247) attribute type is a string
holding one component, a <label> [RFC1034], of a DNS domain name.
The encoding of IA5String for use in LDAP is simply the characters of
the string itself. The equality matching rule is case insensitive,
as is today's DNS.
(Source: RFC 2247 [RFC2247])

       (0.9.2342.19200300.100.1.25 NAME 'dc'
          EQUALITY caseIgnoreIA5Match
          SUBSTR caseIgnoreIA5SubstringsMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
          SINGLE-VALUE)

1.3.6.1.4.1.1466.115.121.1.26 refers to the IA5 String syntax
[Syntaxes].

Examples: Valid values include "example" and "com". The value
          "example.com" is invalid, because it contains two <label>
          components.

It is noted that the directory will not ensure that values of this
attribute conform to the label production [RFC1034]. It is the
application's responsibility to ensure domains it stores in this
attribute are appropriately represented.

It is also noted that applications supporting Internationalized
Domain Names SHALL use the ToASCII method [RFC3490] to produce
components of the <domain> [RFC1034] production. The special
considerations discussed in section 4 of RFC 3490 [RFC3490] should be
taken, depending on whether the domain component is used for "stored"
or "query" purposes.

2.5 'description'

   The 'description' attribute type contains human-readable descriptive
   phrases about the object. Each description is one value of this
   multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.13 NAME 'description'
         EQUALITY caseIgnoreMatch
         SUBSTR caseIgnoreSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

   1.3.6.1.4.1.1466.115.121.1.15 refers to the Directory String syntax
   [Syntaxes].

   Examples: "a color printer", "Maintenance is done every Monday, at
             1pm." and "distribution list for all technical staff".

2.6 'destinationIndicator'

   The 'destinationIndicator' attribute type contains country and city
   strings, associated with the object (the addressee), needed to
   provide the Public Telegram Service. The strings are composed in
   accordance with CCITT Recommendations F.1 [F.1] and F.31 [F.31].
   Each string is one value of this multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.27 NAME 'destinationIndicator'
         EQUALITY caseIgnoreMatch
         SUBSTR caseIgnoreSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.44)

   1.3.6.1.4.1.1466.115.121.1.44 refers to the Printable String syntax
   [Syntaxes].

   Examples: "AASD" as a destination indicator for Sydney, Australia.
             "GBLD" as a destination indicator for London, United
             Kingdom.

   It is noted that the directory will not ensure that values of this
   attribute conform to the F.1 and F.30 CCITT Recommendations. It is
   the application's responsibility to ensure destination indicators
   that it stores in this attribute are appropriately constructed.

2.7 'distinguishedName'

   The 'distinguishedName' attribute type is not used as the name of the
   object itself, but it is instead a base type from which some user

   attribute types with a DN syntax can inherit.

   It is unlikely that values of this type itself will occur in an
   entry. LDAP server implementations which do not support attribute
   subtyping need not recognize this attribute in requests. Client
   implementations MUST NOT assume that LDAP servers are capable of
   performing attribute subtyping.
   (Source: X.520 [X.520])

      (2.5.4.49 NAME 'distinguishedName'
         EQUALITY distinguishedNameMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)

   1.3.6.1.4.1.1466.115.121.1.12 refers to the DN syntax [Syntaxes].

2.8 'dnQualifier'

   The 'dnQualifier' attribute type contains disambiguating information
   strings to add to the relative distinguished name of an entry. The
   information is intended for use when merging data from multiple
   sources in order to prevent conflicts between entries which would
   otherwise have the same name. Each string is one value of this
   multi-valued attribute. It is recommended that a value of the
   'dnQualifier' attribute be the same for all entries from a particular
   source.
   (Source: X.520 [X.520])

      (2.5.4.46 NAME 'dnQualifier'
         EQUALITY caseIgnoreMatch
         ORDERING caseIgnoreOrderingMatch
         SUBSTR caseIgnoreSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.44)

   1.3.6.1.4.1.1466.115.121.1.44 refers to the Printable String syntax
   [Syntaxes].

   Examples: "20050322123345Z" - timestamps can be used to disambiguate
            information.
            "123456A" - serial numbers can be used to disambiguate
            information.

2.9 'enhancedSearchGuide'

   The 'enhancedSearchGuide' attribute type contains sets of information
   for use by directory clients in constructing search filters. Each
   set is one value of this multi-valued attribute.
   (Source: X.520 [X.520])

         (2.5.4.47 NAME 'enhancedSearchGuide'
             SYNTAX 1.3.6.1.4.1.1466.115.121.1.21)

   1.3.6.1.4.1.1466.115.121.1.21 refers to the Enhanced Guide syntax
   [Syntaxes].

   Examples: "person#(sn$APPROX)#wholeSubtree" and
             "organizationalUnit#(ou$SUBSTR)#oneLevel".

2.10 'facsimileTelephoneNumber'

   The 'facsimileTelephoneNumber' attribute type contains telephone
   numbers (and, optionally, the parameters) for facsimile terminals.
   Each telephone number is one value of this multi-valued attribute.
   (Source: X.520 [X.520])

         (2.5.4.23 NAME 'facsimileTelephoneNumber'
             SYNTAX 1.3.6.1.4.1.1466.115.121.1.22)

   1.3.6.1.4.1.1466.115.121.1.22 refers to the Facsimile Telephone
   Number syntax [Syntaxes].

   Examples: "+61 3 9896 7801" and "+81 3 347 7418$fineResolution".

2.11 'generationQualifier'

   The 'generationQualifier' attribute type contains name strings that
   are the part of a person's name which typically is the suffix. Each
   string is one value of this multi-valued attribute.
   (Source: X.520 [X.520])

         (2.5.4.44 NAME 'generationQualifier'
             SUP name)

   Examples: "III", "3rd" and "Jr.".

2.12 'givenName'

   The 'givenName' attribute type contains name strings that are the
   part of a person's name which is not their surname. Each string is
   one value of this multi-valued attribute.
   (Source: X.520 [X.520])

         (2.5.4.42 NAME 'givenName'
             SUP name)

   Examples: "Andrew", "Charles" and "Joanne".

2.13 'houseIdentifier'

The 'houseIdentifier' attribute type contains identifiers for a
building within a location. Each identifier is one value of this
multi-valued attribute.
(Source: X.520 [X.520])

      (2.5.4.51 NAME 'houseIdentifier'
         EQUALITY caseIgnoreMatch
         SUBSTR caseIgnoreSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

1.3.6.1.4.1.1466.115.121.1.15 refers to the Directory String syntax
[Syntaxes].

Examples: "20" to represent the house number 20.

2.14 'initials'

The 'initials' attribute type contains strings of initials of some or
all of an individual's names, except the surname(s). Each string is
one value of this multi-valued attribute.
(Source: X.520 [X.520])

      (2.5.4.43 NAME 'initials'
         SUP name)

Examples: "K. A." and "K".

2.15 'internationalISDNNumber'

The 'internationalISDNNumber' attribute type contains Integrated
Services Digital Network (ISDN) addresses, as defined in the
International Telecommunication Union (ITU) Recommendation E.164
[E.164]. Each address is one value of this multi-valued attribute.
(Source: X.520 [X.520])

      (2.5.4.25 NAME 'internationalISDNNumber'
         EQUALITY numericStringMatch
         SUBSTR numericStringSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.36)

1.3.6.1.4.1.1466.115.121.1.36 refers to the Numeric String syntax
[Syntaxes].

Example: "0198 333 333".

2.16 **'l'**

   The 'l' ('localityName' in X.500) attribute type contains names of a
   locality or place, such as a city, county or other geographic region.
   Each name is one value of this multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.7 NAME 'l'
         SUP name)

   Examples: "Geneva", "Paris" and "Edinburgh".

2.17 **'member'**

   The 'member' attribute type contains the Distinguished Names of
   objects that are on a list or in a group. Each name is one value of
   this multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.31 NAME 'member'
         SUP distinguishedName)

   Examples: "cn=James Clarke,ou=Finance,o=Widget\, Inc." and
             "cn=John Xerri,ou=Finance,o=Widget\, Inc." may
             be two members of the financial team (group) at Widget,
             Inc. In which case, both of these distinguished names would
             be present as individual values of the member attribute.

2.18 **'name'**

   The 'name' attribute type is the attribute supertype from which user
   attribute types with the name syntax inherit. Such attribute types
   are typically used for naming. The attribute type is multi-valued.

   It is unlikely that values of this type itself will occur in an
   entry. LDAP server implementations which do not support attribute
   subtyping need not recognize this attribute in requests. Client
   implementations MUST NOT assume that LDAP servers are capable of
   performing attribute subtyping.
   (Source: X.520 [X.520])

      (2.5.4.41 NAME 'name'
         EQUALITY caseIgnoreMatch
         SUBSTR caseIgnoreSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

   1.3.6.1.4.1.1466.115.121.1.15 refers to the Directory String syntax
   [Syntaxes].

2.19 'o'

The 'o' ('organizationName' in X.500) attribute type contains the
names of an organization. Each name is one value of this
multi-valued attribute.
(Source: X.520 [X.520])

        (2.5.4.10 NAME 'o'
           SUP name)

Examples: "Widget", "Widget, Inc." and "Widget, Incorporated.".

2.20 'ou'

The 'ou' ('organizationalUnitName' in X.500) attribute type contains
the names of an organizational unit. Each name is one value of this
multi-valued attribute.
(Source: X.520 [X.520])

        (2.5.4.11 NAME 'ou'
           SUP name)

Examples: "Finance", "Human Resources" and "Research and
            Development".

2.21 'owner'

The 'owner' attribute type contains the Distinguished Names of
objects that have an ownership responsibility for the object that is
owned. Each owner's name is one value of this multi-valued
attribute.
(Source: X.520 [X.520])

        (2.5.4.32 NAME 'owner'
           SUP distinguishedName)

Example: The mailing list object, whose DN is "cn=All Employees,
            ou=Mailing List,o=Widget\, Inc.", is owned by the Human
            Resources Director.
            Therefore, the value of the 'owner' attribute within the
            mailing list object, would be the DN of the director (role):
            "cn=Human Resources Director,ou=employee,o=Widget\, Inc.".

2.22 'physicalDeliveryOfficeName'

The 'physicalDeliveryOfficeName' attribute type contains names that a
Postal Service uses to identify a post office.
(Source: X.520 [X.520])

```
    ( 2.5.4.19 NAME 'physicalDeliveryOfficeName'
       EQUALITY caseIgnoreMatch
       SUBSTR caseIgnoreSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

1.3.6.1.4.1.1466.115.121.1.15 refers to the Directory String syntax
[Syntaxes].

Examples: "Bremerhaven, Main" and "Bremerhaven, Bonnstrasse".

2.23 'postalAddress'

The 'postalAddress' attribute type contains addresses used by a
Postal Service to perform services for the object. Each address is
one value of this multi-valued attribute.
(Source: X.520 [X.520])

```
    ( 2.5.4.16 NAME 'postalAddress'
       EQUALITY caseIgnoreListMatch
       SUBSTR caseIgnoreListSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 )
```

1.3.6.1.4.1.1466.115.121.1.41 refers to the Postal Address syntax
[Syntaxes].

Example: "15 Main St.$Ottawa$Canada".

2.24 'postalCode'

The 'postalCode' attribute type contains codes used by a Postal
Service to identify postal service zones. Each code is one value of
this multi-valued attribute.
(Source: X.520 [X.520])

```
    ( 2.5.4.17 NAME 'postalCode'
       EQUALITY caseIgnoreMatch
       SUBSTR caseIgnoreSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

1.3.6.1.4.1.1466.115.121.1.15 refers to the Directory String syntax
[Syntaxes].

Example: "22180", to identify Vienna, VA in the USA.

2.25 'postOfficeBox'

The 'postOfficeBox' attribute type contains postal box identifiers
that a Postal Service uses when a customer arranges to receive mail

   at a box on premises of the Postal Service. Each postal box
   identifier is a single value of this multi-valued attribute.
   (Source: X.520 [X.520])

```
( 2.5.4.18 NAME 'postOfficeBox'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

   1.3.6.1.4.1.1466.115.121.1.15 refers to the Directory String syntax
   [Syntaxes].

   Example: "Box 45".

2.26 'preferredDeliveryMethod'

   The 'preferredDeliveryMethod' attribute type contains an indication
   of the preferred method of getting a message to the object.
   (Source: X.520 [X.520])

```
( 2.5.4.28 NAME 'preferredDeliveryMethod'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.14
    SINGLE-VALUE )
```

   1.3.6.1.4.1.1466.115.121.1.14 refers to the Delivery Method syntax
   [Syntaxes].

   Example: If the mhs-delivery Delivery Method is preferred over
            telephone-delivery, which is preferred over all other
            methods, the value would be: "mhs $ telephone".

2.27 'registeredAddress'

   The 'registeredAddress' attribute type contains postal addresses
   suitable for reception of telegrams or expedited documents, where it
   is necessary to have the recipient accept delivery. Each address is
   one value of this multi-valued attribute.
   (Source: X.520 [X.520])

```
( 2.5.4.26 NAME 'registeredAddress'
    SUP postalAddress
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 )
```

   1.3.6.1.4.1.1466.115.121.1.41 refers to the Postal Address syntax
   [Syntaxes].

   Example: "Receptionist$Widget, Inc.$15 Main St.$Ottawa$Canada".

2.28 **'roleOccupant'**

   The 'roleOccupant' attribute type contains the Distinguished Names of
   objects (normally people) that fulfill the responsibilities of a role
   object. Each distinguished name is one value of this multi-valued
   attribute.
   (Source: X.520 [X.520])

      (2.5.4.33 NAME 'roleOccupant'
         SUP distinguishedName)

   Example: The role object, "cn=Human Resources
            Director,ou=Position,o=Widget\, Inc.", is fulfilled by two
            people whose object names are "cn=Mary
            Smith,ou=employee,o=Widget\, Inc." and "cn=James
            Brown,ou=employee,o=Widget\, Inc.". The 'roleOccupant'
            attribute will contain both of these distinguished names,
            since they are the occupants of this role.

2.29 **'searchGuide'**

   The 'searchGuide' attribute type contains sets of information for use
   by clients in constructing search filters. It is superseded by
   'enhancedSearchGuide', described above in section 2.9. Each set is
   one value of this multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.14 NAME 'searchGuide'
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.25)

   1.3.6.1.4.1.1466.115.121.1.25 refers to the Guide syntax [Syntaxes].

   Example: "person#sn$EQ".

2.30 **'seeAlso'**

   The 'seeAlso' attribute type contains Distinguished Names of objects
   that are related to the subject object. Each related object name is
   one value of this multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.34 NAME 'seeAlso'
         SUP distinguishedName)

   Example: The person object, "cn=James Brown,ou=employee,o=Widget\,
            Inc." is related to the role objects, "cn=Football Team
            Captain,ou=sponsored activities,o=Widget\, Inc." and
            "cn=Chess Team,ou=sponsored activities,o=Widget\, Inc.".

Since the role objects are related to the person object, the
'seeAlso' attribute will contain the distinguished name of
each role object as separate values.

2.31 'serialNumber'

The 'serialNumber' attribute type contains the serial numbers of
devices. Each serial number is one value of this multi-valued
attribute.
(Source: X.520 [X.520])

```
( 2.5.4.5 NAME 'serialNumber'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.44 )
```

1.3.6.1.4.1.1466.115.121.1.44 refers to the Printable String syntax
[Syntaxes].

Examples: "WI-3005" and "XF551426".

2.32 'sn'

The 'sn' ('surname' in X.500) attribute type contains name strings
for the family names of a person. Each string is one value of this
multi-valued attribute.
(Source: X.520 [X.520])

```
( 2.5.4.4 NAME 'sn'
    SUP name )
```

Example: "Smith".

2.33 'st'

The 'st' ('stateOrProvinceName' in X.500) attribute type contains the
full names of states or provinces. Each name is one value of this
multi-valued attribute.
(Source: X.520 [X.520])

```
( 2.5.4.8 NAME 'st'
    SUP name )
```

Example: "California".

2.34 **'street'**

   The 'street' ('streetAddress' in X.500) attribute type contains site
   information from a postal address (i.e., the street name, place,
   avenue, and the house number.). Each street is one value of this
   multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.9 NAME 'street'
         EQUALITY caseIgnoreMatch
         SUBSTR caseIgnoreSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

   1.3.6.1.4.1.1466.115.121.1.15 refers to the Directory String syntax
   [Syntaxes].

   Example: "15 Main St.".

2.35 **'telephoneNumber'**

   The 'telephoneNumber' attribute type contains telephone numbers that
   comply with the ITU Recommendation E.123 [E.123]. Each number is one
   value of this multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.20 NAME 'telephoneNumber'
         EQUALITY telephoneNumberMatch
         SUBSTR telephoneNumberSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.50)

   1.3.6.1.4.1.1466.115.121.1.50 refers to the Telephone Number syntax
   [Syntaxes].

   Example: "+1 234 567 8901".

2.36 **'teletexTerminalIdentifier'**

   The withdrawal of Rec. F.200 has resulted in the withdrawal of this
   attribute.
   (Source: X.520 [X.520])

      (2.5.4.22 NAME 'teletexTerminalIdentifier'
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.51)

   1.3.6.1.4.1.1466.115.121.1.51 refers to the Teletex Terminal
   Identifier syntax [Syntaxes].

2.37 **'telexNumber'**

   The 'telexNumber' attribute type contains sets of strings which are a
   telex number, country code, and answerback code of a telex terminal.
   Each set is one value of this multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.21 NAME 'telexNumber'
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.52)

   1.3.6.1.4.1.1466.115.121.1.52 refers to the Telex Number syntax
   [Syntaxes].

   Example: "12345$023$ABCDE".

2.38 **'title'**

   The 'title' attribute type contains the title of a person in their
   organizational context. Each title is one value of this multi-valued
   attribute.
   (Source: X.520 [X.520])

      (2.5.4.12 NAME 'title'
         SUP name)
   Examples: "Vice President", "Software Engineer" and "CEO".

2.39 **'uid'**

   The 'uid' ('userid' in RFC 1274) attribute type contains computer
   system login names associated with the object. Each name is one
   value of this multi-valued attribute.
   (Source: RFC 2798 [RFC2798] and RFC 1274 [RFC1274])

      (0.9.2342.19200300.100.1.1 NAME 'uid'
         EQUALITY caseIgnoreMatch
         SUBSTR caseIgnoreSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

   1.3.6.1.4.1.1466.115.121.1.15 refers to the Directory String syntax
   [Syntaxes].

   Examples: "s9709015", "admin" and "Administrator".

2.40 **'uniqueMember'**

   The 'uniqueMember' attribute type contains the Distinguished Names of
   an object that is on a list or in a group, where the Relative
   Distinguished Names of the object include a value that distinguishes

   between objects when a distinguished name has been reused. Each
   distinguished name is one value of this multi-valued attribute.
   (Source: X.520 [X.520])

      (2.5.4.50 NAME 'uniqueMember'
         EQUALITY uniqueMemberMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.34)

   1.3.6.1.4.1.1466.115.121.1.34 refers to the Name and Optional UID
   syntax [Syntaxes].

   Example: If "ou=1st Battalion,o=Defense,c=US" is a battalion that was
            disbanded, establishing a new battalion with the "same" name
            would have a unique identifier value added, resulting in
            "ou=1st Battalion, o=Defense,c=US#'010101'B".

2.41 'userPassword'

   The 'userPassword' attribute contains octet strings that are known
   only to the user and the system to which the user has access. Each
   string is one value of this multi-valued attribute.

   The application SHOULD prepare textual strings used as passwords by
   transcoding them to Unicode, applying SASLprep [RFC4013], and
   encoding as UTF-8. The determination of whether a password is
   textual is a local client matter.
   (Source: X.509 [X.509])

      (2.5.4.35 NAME 'userPassword'
         EQUALITY octetStringMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)

   1.3.6.1.4.1.1466.115.121.1.40 refers to the Octet String syntax
   [Syntaxes].

   Passwords are stored using an Octet String syntax and are not
   encrypted. Transfer of cleartext passwords is strongly discouraged
   where the underlying transport service cannot guarantee
   confidentiality and may result in disclosure of the password to
   unauthorized parties.

   An example of a need for multiple values in the 'userPassword'
   attribute is an environment where every month the user was expected
   to use a different password generated by some automated system.
   During transitional periods, like the last and first day of the
   periods, it may be necessary to allow two passwords for the two
   consecutive periods to be valid in the system.

2.42 'x121Address'

The 'x121Address' attribute type contains data network addresses as
defined by ITU Recommendation X.121 [X.121]. Each address is one
value of this multi-valued attribute.
(Source: X.520 [X.520])

      (2.5.4.24 NAME 'x121Address'
         EQUALITY numericStringMatch
         SUBSTR numericStringSubstringsMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.36)

1.3.6.1.4.1.1466.115.121.1.36 refers to the Numeric String syntax
[Syntaxes].

Example: "36111222333444555".

2.43 'x500UniqueIdentifier'

The 'x500UniqueIdentifier' attribute type contains binary strings
that are used to distinguish between objects when a distinguished
name has been reused. Each string is one value of this multi-valued
attribute.
In X.520 [X.520], this attribute type is called 'uniqueIdentifier'.
This is a different attribute type from both the 'uid' and
'uniqueIdentifier' LDAP attribute types. The 'uniqueIdentifier'
attribute type is defined in [RFC1274].
(Source: X.520 [X.520])

      (2.5.4.45 NAME 'x500UniqueIdentifier'
         EQUALITY bitStringMatch
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.6)

1.3.6.1.4.1.1466.115.121.1.6 refers to the Bit String syntax
[Syntaxes].

3. **Object Classes**

   LDAP servers SHOULD recognize all the Object Classes listed here as
   values of the 'objectClass' attribute (see [Models]).

3.1 **'applicationProcess'**

   The 'applicationProcess' object class definition is the basis of an
   entry which represents an application executing in a computer system.
   (Source: X.521 [X.521])

```
   ( 2.5.6.11 NAME 'applicationProcess'
      SUP top
      STRUCTURAL
      MUST cn
      MAY ( seeAlso $
            ou $
            l $
            description ) )
```

3.2 **'country'**

   The 'country' object class definition is the basis of an entry which
   represents a country.
   (Source: X.521 [X.521])

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

3.3 **'dcObject'**

   The 'dcObject' object class permits an entry to contains domain
   component information. This object class is defined as auxiliary,
   because it will be used in conjunction with an existing structural
   object class.
   (Source: RFC 2247 [RFC2247])

```
   ( 1.3.6.1.4.1.1466.344 NAME 'dcObject'
      SUP top
      AUXILIARY
      MUST dc )
```

3.4 'device'

   The 'device' object class is the basis of an entry which represents
   an appliance, computer or network element.
   (Source: X.521 [X.521])

```
   ( 2.5.6.14 NAME 'device'
      SUP top
      STRUCTURAL
      MUST cn
      MAY ( serialNumber $
            seeAlso $
            owner $
            ou $
            o $
            l $
            description ) )
```

3.5 'groupOfNames'

   The 'groupOfNames' object class is the basis of an entry which
   represents a set of named objects including information related to
   the purpose or maintenance of the set.
   (Source: X.521 [X.521])

```
   ( 2.5.6.9 NAME 'groupOfNames'
      SUP top
      STRUCTURAL
      MUST ( member $
            cn )
      MAY ( businessCategory $
            seeAlso $
            owner $
            ou $
            o $
            description ) )
```

3.6 'groupOfUniqueNames'

   The 'groupOfUniqueNames' object class is the same as the
   'groupOfNames' object class except that the object names are not
   repeated or reassigned within a set scope.
   (Source: X.521 [X.521])

```
   ( 2.5.6.17 NAME 'groupOfUniqueNames'
      SUP top
      STRUCTURAL
      MUST ( uniqueMember $
```

```
                cn )
        MAY ( businessCategory $
                seeAlso $
                owner $
                ou $
                o $
                description ) )
```

3.7 'locality'

The 'locality' object class is the basis of an entry which represents
a place in the physical world.
(Source: X.521 [X.521])

```
    ( 2.5.6.3 NAME 'locality'
        SUP top
        STRUCTURAL
        MAY ( street $
                seeAlso $
                searchGuide $
                st $
                l $
                description ) )
```

3.8 'organization'

The 'organization' object class is the basis of an entry which
represents a structured group of people.
(Source: X.521 [X.521])

```
    ( 2.5.6.4 NAME 'organization'
        SUP top
        STRUCTURAL
        MUST o
        MAY ( userPassword $ searchGuide $ seeAlso $
                businessCategory $ x121Address $ registeredAddress $
                destinationIndicator $ preferredDeliveryMethod $
                telexNumber $ teletexTerminalIdentifier $
                telephoneNumber $ internationaliSDNNumber $
                facsimileTelephoneNumber $ street $ postOfficeBox $
                postalCode $ postalAddress $ physicalDeliveryOfficeName $
                st $ l $ description ) )
```

3.9 'organizationalPerson'

The 'organizationalPerson' object class is the basis of an entry
which represents a person in relation to an organization.
(Source: X.521 [X.521])

```
( 2.5.6.7 NAME 'organizationalPerson'
    SUP person
    STRUCTURAL
    MAY ( title $ x121Address $ registeredAddress $
          destinationIndicator $ preferredDeliveryMethod $
          telexNumber $ teletexTerminalIdentifier $
          telephoneNumber $ internationaliSDNNumber $
          facsimileTelephoneNumber $ street $ postOfficeBox $
          postalCode $ postalAddress $ physicalDeliveryOfficeName $
          ou $ st $ l ) )
```

3.10 'organizationalRole'

The 'organizationalRole' object class is the basis of an entry which
represents a job, function or position in an organization.
(Source: X.521 [X.521])

```
( 2.5.6.8 NAME 'organizationalRole'
    SUP top
    STRUCTURAL
    MUST cn
    MAY ( x121Address $ registeredAddress $ destinationIndicator $
          preferredDeliveryMethod $ telexNumber $
          teletexTerminalIdentifier $ telephoneNumber $
          internationaliSDNNumber $ facsimileTelephoneNumber $
          seeAlso $ roleOccupant $ preferredDeliveryMethod $
          street $ postOfficeBox $ postalCode $ postalAddress $
          physicalDeliveryOfficeName $ ou $ st $ l $
          description ) )
```

3.11 'organizationalUnit'

The 'organizationalUnit' object class is the basis of an entry which
represents a piece of an organization.
(Source: X.521 [X.521])

```
( 2.5.6.5 NAME 'organizationalUnit'
    SUP top
    STRUCTURAL
    MUST ou
    MAY ( businessCategory $ description $ destinationIndicator $
          facsimileTelephoneNumber $ internationaliSDNNumber $ l $
          physicalDeliveryOfficeName $ postalAddress $ postalCode $
          postOfficeBox $ preferredDeliveryMethod $
          registeredAddress $ searchGuide $ seeAlso $ st $ street $
          telephoneNumber $ teletexTerminalIdentifier $
          telexNumber $ userPassword $ x121Address ) )
```

3.12 'person'

The 'person' object class is the basis of an entry which represents a
human being.
(Source: X.521 [X.521])

```
( 2.5.6.6 NAME 'person'
    SUP top
    STRUCTURAL
    MUST ( sn $
           cn )
    MAY ( userPassword $
          telephoneNumber $
          seeAlso $ description ) )
```

3.13 'residentialPerson'

The 'residentialPerson' object class is the basis of an entry which
includes a person's residence in the representation of the person.
(Source: X.521 [X.521])

```
( 2.5.6.10 NAME 'residentialPerson'
    SUP person
    STRUCTURAL
    MUST l
    MAY ( businessCategory $ x121Address $ registeredAddress $
          destinationIndicator $ preferredDeliveryMethod $
          telexNumber $ teletexTerminalIdentifier $
          telephoneNumber $ internationaliSDNNumber $
          facsimileTelephoneNumber $ preferredDeliveryMethod $
          street $ postOfficeBox $ postalCode $ postalAddress $
          physicalDeliveryOfficeName $ st $ l ) )
```

3.14 'uidObject'

The 'uidObject' object class permits an entry to contains user
identification information. This object class is defined as
auxiliary, because it will be used in conjunction with an existing
structural object class.
(Source: RFC 2377 [RFC2377])

```
( 1.3.6.1.1.3.1 NAME 'uidObject'
    SUP top
    AUXILIARY
    MUST uid )
```

[4](#). **IANA Considerations**

It is requested that the Internet Assigned Numbers Authority (IANA)
update the LDAP descriptors registry as indicated in the following
template:

    Subject: Request for LDAP Descriptor Registration Update
    Descriptor (short name): see comment
    Object Identifier: see comment
    Person & email address to contact for further information:
        Andrew Sciberras <andrew.sciberras@eb2bcom.com>
    Usage: (A = attribute type, O = Object Class) see comment
    Specification: RFC XXXX [editor's note: The RFC number will be
        the one assigned to this document.]
    Author/Change Controller: IESG
Comments
In the LDAP descriptors registry, the following descriptors (short
names) should be updated to refer to RFC XXXX [editor's note: This
document]. Names that need to be reserved, rather than assigned to
an Object Identifier, will contain an Object Identifier value of
RESERVED.

    NAME   Type OID
    ----------------------   ---- ----------------------------
    applicationProcess   O   2.5.6.11
    businessCategory   A   2.5.4.15
    c   A   2.5.4.6
    cn   A   2.5.4.3
    commonName   A   2.5.4.3
    country   O   2.5.6.2
    countryName   A   2.5.4.6
    DC   A   0.9.2342.19200300.100.1.25
    dcObject   O   1.3.6.1.4.1.1466.344
    description   A   2.5.4.13
    destinationIndicator   A   2.5.4.27
    device   O   2.5.6.14
    distinguishedName   A   2.5.4.49
    dnQualifier   A   2.5.4.46
    domainComponent   A   0.9.2342.19200300.100.1.25
    enhancedSearchGuide   A   2.5.4.47
    facsimileTelephoneNumber   A   2.5.4.23
    generationQualifier   A   2.5.4.44
    givenName   A   2.5.4.42
    GN   A   RESERVED
    groupOfNames   O   2.5.6.9
    groupOfUniqueNames   O   2.5.6.17
    houseIdentifier   A   2.5.4.51
    initials   A   2.5.4.43

```
        internationalISDNNumber     A     2.5.4.25
        L                           A     2.5.4.7
        locality                    O     2.5.6.3
        localityName                A     2.5.4.7
        member                      A     2.5.4.31
        name                        A     2.5.4.41
        o                           A     2.5.4.10
        organization                O     2.5.6.4
        organizationName            A     2.5.4.10
        organizationalPerson        O     2.5.6.7
        organizationalRole          O     2.5.6.8
        organizationalUnit          O     2.5.6.5
        organizationalUnitName      A     2.5.4.11
        ou                          A     2.5.4.11
        owner                       A     2.5.4.32
        person                      O     2.5.6.6
        physicalDeliveryOfficeName  A     2.5.4.19
        postalAddress               A     2.5.4.16
        postalCode                  A     2.5.4.17
        postOfficeBox               A     2.5.4.18
        preferredDeliveryMethod     A     2.5.4.28
        registeredAddress           A     2.5.4.26
        residentialPerson           O     2.5.6.10
        roleOccupant                A     2.5.4.33
        searchGuide                 A     2.5.4.14
        seeAlso                     A     2.5.4.34
        serialNumber                A     2.5.4.5
        sn                          A     2.5.4.4
        st                          A     2.5.4.8
        street                      A     2.5.4.9
        surname                     A     2.5.4.4
        telephoneNumber             A     2.5.4.20
        teletexTerminalIdentifier   A     2.5.4.22
        telexNumber                 A     2.5.4.21
        title                       A     2.5.4.12
        uid                         A     0.9.2342.19200300.100.1.1
        uidObject                   O     1.3.6.1.1.3.1
        uniqueMember                A     2.5.4.50
        userId                      A     0.9.2342.19200300.100.1.1
        userPassword                A     2.5.4.35
        x121Address                 A     2.5.4.24
        x500UniqueIdentifier        A     2.5.4.45
```

5. Security Considerations

   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.

Transfer of cleartext passwords is strongly discouraged where the underlying transport service cannot guarantee confidentiality and may result in disclosure of the password to unauthorized parties.

Multiple attribute values for the 'userPassword' attribute need to be used with care. Especially reset/deletion of a password by an admin without knowing the old user password gets tricky or impossible if multiple values for different applications are present.

Certainly, applications which intend to replace the 'userPassword' value(s) with new value(s) should use modify/replaceValues (or modify/deleteAttribute+addAttribute). Additionally, server implementations are encouraged to provide administrative controls which, if enabled, restrict the 'userPassword' attribute to one value.

Note that when used for authentication purposes [AuthMeth], the user need only prove knowledge of one of the values, not all of the values.

6. Acknowledgements

The definitions, on which this document is based, have been developed by committees for telecommunications and international standards.

This document is an update of RFC 2256 by Mark Wahl. RFC 2256 was a product of the IETF ASID Working Group.

The 'dc' attribute type definition and the 'dcObject' object class definition in this document supersede the specification in RFC 2247 by S. Kille, M. Wahl, A. Grimstad, R. Huber, and S. Sataluri.

The 'uid' attribute type definition in this document supersedes the specification of the 'userid' in RFC 1274 by P. Barker and S. Kille and of the uid in RFC 2798 by M. Smith.

The 'uidObject' object class definition in this document supersedes the specification of the 'uidObject' in RFC 2377 by A. Grimstad, R. Huber, S. Sataluri and M. Smith.

This document is based upon input of the IETF LDAPBIS working group. The author wishes to thank S. Legg and K. Zeilenga for their significant contribution to this update. The author would also like to thank Kathy Dally who edited early drafts of this document.

7. References

7.1 Normative

[E.123]   Notation for national and international telephone
               numbers, ITU-T Recommendation E.123, 1988

[E.164]   The international public telecommunication numbering
               plan, ITU-T Recommendation E.164, 1997

[F.1]   Operational Provisions For The International Public
               Telegram Service Transmission System, CCITT
               Recommendation F.1, 1992

[F.31]   Telegram Retransmission System, CCITT Recommendation
               F.31, 1988

[ISO3166]   ISO 3166, "Codes for the representation of names of
               countries".

[Models]   K. Zeilenga, "LDAP: The Models", draft-ietf-ldapbis-
               models-xx (a work in progress)

[RFC1034]   P. Mockapetris, " DOMAIN NAMES - CONCEPTS AND
               FACILITIES", RFC 1034, January 1987

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
               Requirement Levels", RFC 2119, March 1997

[RFC2234]   Crocker, D., Overell P., "Augmented BNF for Syntax
               Specifications: ABNF", RFC 2234, November 1997

[RFC3490]   Faltstrom P., Hoffman P., Costello A.,
               "Internationalizing Domain Names in Applications
               (IDNA)", RFC 3490, March 2003

[RFC4013]   Zeilenga K., "SASLprep: Stringprep profile for User
               Names and Passwords", RFC 4013, February 2005.

[Roadmap]   Zeilenga, K., "LDAP: Technical Specification Road
               Map", draft-ietf-ldapbis-roadmap-xx (a work in
               progress)

[Syntaxes]   S. Legg (editor), "LDAP: Syntaxes", draft-ietf-ldapbis-
               syntaxes-xx (a work in progress)

[X.121]   International numbering plan for public data networks,
               ITU-T Recommendation X.121, 1996

    [X.509]   The Directory: Authentication Framework, ITU-T
                   Recommendation X.509, 1993

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

    [X.521]   The Directory: Selected Object Classes. ITU-T
                   Recommendation X.521, 1993

7.2 Informative

    [AuthMeth]   Harrison R., "LDAP: Authentication Methods and
                   Connection Level Security Mechanisms", draft-ietf-
                   ldapbis-authmeth-xx (a work in progress)

    [LDAP-PKI]   Zeilenga, K., "Lightweight Directory Access Protocol
                   (LDAP) schema definitions for X.509 Certificates",
                   draft-zeilenga-ldap-x509-xx (a work in progress)

    [RFC1274]   Barker, P., Kille, S.,"The COSINE and Internet X.500
                   Schema", RFC 1274, November 1991

    [RFC2247]   Kille, S., Wahl, M., Grimstad, A., Huber, R., and
                   Sataluri, S., "Using Domains in LDAP/X.500
                   Distinguished Names", RFC 2247, January 1998

    [RFC2377]   Grimstad, A., Huber, R., Sataluri, S., and Wahl, M.,
                   "Naming Plan for Internet-Enabled Applications", RFC
                   2377, September 1998.

    [RFC2798]   Smith, M., "Definition of the inetOrgPerson LDAP Object
                   Class", RFC 2798, April 2000

    [X.500]   ITU-T Recommendations X.5000 (1993) | ISO/IEC
                   9594-1:1994, Information Technology - Open Systems
                   Interconnection - The Directory: Overview of concepts,
                   models and services.

8. Author's Address

    Andrew Sciberras
    eB2Bcom
    Suite 3, Woodhouse Corporate Centre,
    935 Station Street,
    Box Hill North, Victoria 3129
    AUSTRALIA

    Phone: +61 3 9896 7833

     Email: andrew.sciberras@eb2bcom.com

9. **Intellectual Property Statement**

   The IETF takes no position regarding the validity or scope of any
   Intellectual Property Rights or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; nor does it represent that it has
   made any independent effort to identify any such rights. Information
   on the procedures with respect to rights in RFC documents can be
   found in BCP 78 and BCP 79.

   Copies of IPR disclosures made to the IETF Secretariat and any
   assurances of licenses to be made available, or the result of an
   attempt made to obtain a general license or permission for the use of
   such proprietary rights by implementers or users of this
   specification can be obtained from the IETF on-line IPR repository at
   http://www.ietf.org/ipr.

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights that may cover technology that may be required to implement
   this standard. Please address the information to the IETF at
   ietf-ipr@ietf.org.

10. **Full Copyright Statement**

Appendix A   Changes Made Since RFC 2256

This appendix lists the changes that have been made from RFC 2256 to
this I-D.

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

1.   Replaced the document title.

2.   Removed the IESG Note.

3.   Dependencies on RFC 1274 have been eliminated.

4.   Added a Security Considerations section and an IANA
     considerations section.

5.   Deleted the conformance requirement for subschema object
     classes in favor of a statement in [Syntaxes].

6.   Added explanation to attribute types and to each object class.

7.   Removed Section 4, Syntaxes, and Section 6, Matching Rules,
     (moved to [Syntaxes]).

8.   Removed the certificate-related attribute types:
     authorityRevocationList, cACertificate,
     certificateRevocationList, crossCertificatePair,
     deltaRevocationList, supportedAlgorithms, and userCertificate.

     Removed the certificate-related Object Classes:
     certificationAuthority, certificationAuthority-V2,
     cRLDistributionPoint, strongAuthenticationUser, and
     userSecurityInformation

     LDAP PKI is now discussed in [LDAP-CRL] and [LDAP-CERT].

9.   Removed the dmdName, knowledgeInformation,
     presentationAddress, protocolInformation, and
     supportedApplicationContext attribute types and the dmd,
     applicationEntity, and dSA object classes.

10. Deleted the aliasedObjectName and objectClass attribute type
     definitions. Deleted the alias and top object class
     definitions. They are included in [Models].

11. Added the 'dc' attribute type from RFC 2247.

12. Numerous edititorial changes.

13. Removed upper bound after the SYNTAX oid in all attribute
    definitions where it appeared.

14. Added text about Unicode, SASLprep and UTF-8 for userPassword.

changes since 07:

15. Corrected examples in preferredDeliveryMethod, uniqueMember,
    postalAddress, and registeredAddress attribute types.

16. Clarified and corrected examples in owner and roleOccupant
    attribute types.

17. Added RFC 2234 to normative references.

18. Added RFC 1274 and RFC 2798 to informative references.

19. Removed the statement about RFC 2026 conformance.

20. Added the IPR Disclosure and Notice

21. Updated the Copyright text.

changes since 08:

22. Included RFC 2377 into Updates header and Informative
    References

23. Changed Editor information to Andrew Sciberras.

24. Updated I-D Template information.

25. References made consistent with other LDAPbis ID's. [ROADMAP]
    -> [RoadMap] and [AUTHMETH] -> [AuthMeth].

26. Changed Introduction to include an (LDAP) acronym after the
    first usage.

27. Renamed section 1.1 to "Relationship with other
    specifications" from "Situation".

28. Included definitions, comments and references for 'dcObject'
    and 'uidObject'.

29. Replaced PKI schema references to use draft-zeilenga-ldap-
    x509-xx

30. Spelt out and referenced ABNF on first usage.

31. Removed Section 2.4 (Source). Replaced the source table with
    explicit references for each definition.

32. All references to an attribute type or object class are
    enclosed in single quotes.

33. The layout of attribute type definitions has been changed to
    provide consistency throughout the document:
    > Section Heading
    > Description of Attribute type
    > Multivalued description
    > Source Information
    > Definition
    > Example
    > Additional Comments

    Adding this consistent output included the addition of
    examples to some definitions.

34. References to alternate names for attributes types are
    provided with a reference to where they were originally
    specified.

35. Clarification of the description of 'distinguishedName' and
    'name', in regards to these attribute types being supertypes.

36. Spelt out ISDN on first usage.

37. Inserted a reference to [Syntaxes] for the
    'teletexTerminalIdentifier' definition's SYNTAX OID.

38. Additional names were added to the IANA Considerations. Names
    include 'commonName', 'dcObject', 'domainComponent', 'GN',
    'localityName', 'organizationName', 'organizationUnitName',
    'surname', 'uidObject' and 'userid'.

39. Renamed all instances of supercede to supersede.

40. Moved [F.1], [F.30] and [SASLprep] from informative to
    normative references.

41. Changed the 'c' definition to be consistent with X.500.

42. Added text to 'dc', making the distinction between 'stored'
    and 'query' values when preparing IDN strings.