

INTERNET-DRAFT
Intended Category: Standard Track
Expires: 20 May 2002
Obsoletes: RFC [1274](#)
Updates: RFC [2798](#)

Editor: Kurt D. Zeilenga
OpenLDAP Foundation
20 November 2001

LDAPv3: A Collection of User Schema
<[draft-zeilenga-ldap-user-schema-04.txt](#)>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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 Directory Interest mailing list <directory@apps.ietf.org>. Please send editorial comments directly to the author <Kurt@OpenLDAP.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 2001, The Internet Society. All Rights Reserved.

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

Abstract

This document provides a collection of user schema elements for use with LDAP collected from numerous sources including [RFC 1274](#), X.501, and X.520.

Conventions

Schema definitions are provided using LDAPv3 description formats [[RFC2252](#)]. Definitions provided here are formatted (line wrapped) for readability.

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

Table of Contents (to be expanded by editor)

Status of this Memo	1
Abstract	
Conventions	2
Table of Contents	
1. Background and Intended Use	3
2. Matching Rules	
2.1. booleanMatch	4
2.2. caseExactMatch	
2.3. caseExactOrderingMatch	
2.4. caseExactSubstringsMatch	
2.5. caseIgnoreListSubstringsMatch	
2.6. directoryStringFirstComponentMatch	5
2.7. integerOrderingMatch	
2.7. keywordMatch	
2.9. numericStringOrderingMatch	6
2.10. octetStringOrderingMatch	
2.11. storedPrefixMatch	
2.12. wordMatch	7
3. Attribute Types	
3.1. associatedDomain	
3.2. associatedName	
3.3. buildingName	
3.3. co	8
3.4. destinationIndicator	
3.5. documentAuthor	
3.6. documentIdentifier	9
3.7. documentLocation	
3.8. documentPublisher	
3.9. documentTitle	
3.10. documentVersion	
3.11. drink	10
3.12. houseIdentifier	
3.13. homePhone	
3.14. homePostalAddress	
3.15. host	11

3.16.	info	
3.17.	mail	
3.18.	manager	12
3.19.	mobile	
3.20.	organizationalStatus	
3.21.	otherMailbox	
3.22.	pager	13
3.23.	personalTitle	
3.24.	roomNumber	
3.25.	secretary	
3.26.	uid	14
3.27.	uniqueIdentifier	
3.28.	userClass	
4.	Object Classes	15
4.1.	account	
4.2.	document	
4.3.	documentSeries	
4.4.	domainRelatedObject	16
4.5.	friendlyCountry	
4.6.	rFC822LocalPart	
4.7.	room	17
4.8.	simpleSecurityObject	
5.	Security Considerations	
6.	Acknowledgments	
7.	Author's Address	
8.	Normative References	18
9.	Informative References	
	Full Copyright	19

1. Background and Intended Use

This document provides descriptions [[RFC2252](#)] of user schema for use with LDAP [[LDAPTS](#)] collected from numerous sources.

This document includes a summary of select schema introduced for the COSINE and Internet X.500 pilot projects [[RFC1274](#)]. This document obsoletes [RFC 1274](#).

This document contains a summary of X.500 user schema [[X.520](#)] not included in LDAPv3 [[RFC2252](#)][[RFC2256](#)]. Some of these items were described in the inetOrgPerson [[RFC2798](#)] schema. This document supersedes these descriptions, replacing sections [9.1.3](#) and [9.3.3](#) of [RFC 2798](#).

2. Matching Rules

This section introduces LDAP matching rules based upon descriptions of their X.500 counterparts.

2.1. booleanMatch

BooleanMatch compares for equality a asserted Boolean value with an attribute value of BOOLEAN syntax. The rule returns TRUE if and only if the values are the same, i.e. both are TRUE or both are FALSE.
(Source: X.520)

```
( 2.5.13.13 NAME 'booleanMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 )
```

2.2. caseExactMatch

CaseExactMatch compares for equality the asserted value with an attribute value of DirectoryString syntax. The rule is identical to the caseIgnoreMatch [[RFC2252](#)] rule except that case is not ignored.
(Source: X.520)

```
( 2.5.13.5 NAME 'caseExactMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

2.3. caseExactOrderingMatch

CaseExactOrderingMatch compares the collation order of the asserted string with an attribute value of DirectoryString syntax. The rule is identical to the caseIgnoreOrderingMatch [[RFC2252](#)] rule except that letters are not folded. (Source: X.520)

```
( 2.5.13.6 NAME 'caseExactOrderingMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

2.3. caseExactSubstringsMatch

CaseExactSubstringsMatch determines whether the asserted value are substrings of an attribute value of DirectoryString syntax. The rule is identical to the caseIgnoreSubstringsMatch [[RFC2252](#)] rule except that case is not ignored. (Source: X.520)

```
( 2.5.13.7 NAME 'caseExactSubstringsMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )
```


2.4. caseIgnoreListSubstringsMatch

CaseIgnoreListSubstringMatch compares the asserted substring with an attribute value which is a sequence of DirectoryStrings, but where the case (upper or lower) is not significant for comparison purposes. The asserted value matches a stored value if and only if the asserted value matches the string formed by concatenating the strings of the stored value. This matching is done according to the caseIgnoreSubstringsMatch [RFC2252] rule; however, none of the initial, any, or final values of the asserted value are considered to match a substring of the concatenated string which spans more than one of the strings of the stored value. (Source: X.520)

```
( 2.5.13.12 NAME 'caseIgnoreListSubstringsMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.58 )
```

2.5. directoryStringFirstComponentMatch

DirectoryStringFirstComponentMatch compares for equality the asserted DirectoryString value with an attribute value of type SEQUENCE whose first component is mandatory and of type DirectoryString. The rule returns TRUE if and only if the attribute value has a first component whose value matches the asserted DirectoryString using the rules of caseIgnoreMatch [RFC2252]. A value of the assertion syntax is derived from a value of the attribute syntax by using the value of the first component of the SEQUENCE. (Source: X.520)

```
( 2.5.13.31 NAME 'directoryStringFirstComponentMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

2.6. integerOrderingMatch

The integerOrderingMatch rule compares the ordering of the asserted integer with an attribute value of Integer syntax. The rule returns True if the attribute value is less than the asserted value. (Source: X.520)

```
( 2.5.13.15 NAME 'integerOrderingMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )
```

2.7. keywordMatch

The keywordMatch rule compares the asserted string with keywords in an attribute value of DirectoryString syntax. The rule returns TRUE if and only if the asserted value matches any keyword in the attribute

value. The identification of keywords in an attribute value and of the exactness of match are both implementation specific. (Source: X.520)

```
( 2.5.13.32 NAME 'keywordMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

2.8. numericStringOrderingMatch

NumericStringOrderingMatch compares the collation order of the asserted string with an attribute value of NumericString syntax. The rule is identical to the caseIgnoreOrderingMatch [[RFC2252](#)] rule except that all space characters are skipped during comparison (case is irrelevant as characters are numeric). (Source: X.520)

```
( 2.5.13.9 NAME 'NumericStringOrderingMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.36 )
```

2.9. octetStringOrderingMatch

OctetStringOrderingMatch compares the collation order of the asserted octet string with an attribute value of OCTET STRING syntax. The rule compares octet strings from first octet to last octet, and from the most significant bit to the least significant bit within the octet. The first occurrence of a different bit determines the ordering of the strings. A zero bit precedes a one bit. If the strings are identical but contain different numbers of octets, the shorter string precedes the longer string. (Source: X.520)

```
( 2.5.13.18 NAME 'octetStringOrderingMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

2.10. storedPrefixMatch

StoredPrefixMatch determines whether an attribute value, whose syntax is DirectoryString, is a prefix (i.e. initial substring) of the asserted value, without regard to the case (upper or lower) of the strings. The rule returns TRUE if and only if the attribute value is an initial substring of the asserted value with corresponding characters identical except possibly with regard to case. (Source: X.520)

```
( 2.5.13.41 NAME 'storedPrefixMatch'  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```


Note: This rule can be used, for example, to compare values in the Directory which are telephone area codes with a purported value which is a telephone number.

2.11. wordMatch

The wordMatch rule compares the asserted string with words in an attribute value of DirectoryString syntax. The rule returns TRUE if and only if the asserted word matches any word in the attribute value. Individual word matching is as for the caseIgnoreMatch [\[RFC2252\]](#) matching rule. The precise definition of a "word" is implementation specific. (Source: X.520)

```
( 2.5.13.32 NAME 'wordMatch'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

3. Attribute Types

This section details attribute types for use in LDAP based upon their X.500 descriptions.

3.1. associatedDomain

The associatedDomain attribute type specifies a DNS domain [\[RFC1034\]](#) which is associated with an object. For example, the entry in the DIT with a distinguished name "DC=example,DC=com" might have an associated domain of "example.com". (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.37 NAME 'associatedDomain'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

3.2. associatedName

The associatedName attribute type specifies an entry in the organizational DIT associated with a DNS domain [\[RFC1034\]](#). (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.38 NAME 'associatedName'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
```


3.3. buildingName

The buildingName attribute type specifies the name of the building where an organization or organizational unit is based. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.48 NAME 'buildingName'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

3.3. co

The co (Friendly Country Name) attribute type specifies names of countries in human readable format. The standard attribute country name must be one of the two-letter codes defined in [ISO 3166]. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.43
  NAME ( 'co' 'friendlyCountryName' )
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

3.4. destinationIndicator

The destinationIndicator attribute type specifies (according to CCITT Recommendation F.1 and CCITT Recommendation F.31) the country and city associated with the object (the addressee) needed to provide the Public Telegram Service. An attribute value for Destination Indicator is a printable string containing only alphabetical characters. (Source: X.520)

```
( 2.5.4.27 NAME 'destinationIndicator'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.44{128} )
```

3.5. documentAuthor

The documentAuthor attribute type specifies the distinguished name of the author of a document. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.14 NAME 'documentAuthor'
  EQUALITY distinguishedNameMatch
```



```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
```

3.6. documentIdentifier

The documentIdentifier attribute type specifies a unique identifier for a document. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.11 NAME 'documentIdentifier'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

3.7. documentLocation

The documentLocation attribute type specifies the location of the document original. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.15 NAME 'documentLocation'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

3.8. documentPublisher

The documentPublisher attribute is the person and/or organization that published a document. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.56 NAME 'documentPublisher'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

3.9. documentTitle

The documentTitle attribute type specifies the title of a document. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.12 NAME 'documentTitle'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

3.10. documentVersion

The documentVersion attribute type specifies the version number of a document. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.13 NAME 'documentVersion'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

[3.11.](#) drink

The drink (Favourite Drink) attribute type specifies the favorite drink of an object (or person). (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.5 NAME ( 'drink' 'favouriteDrink' )
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

[3.12.](#) houseIdentifier

The houseIdentifier attribute type specifies a linguistic construct used to identify a particular building, for example a house number or house name relative to a street, avenue, town or city, etc. An attribute value for houseIdentifier is a string, e.g. "14". (Source: X.520)

```
( 2.5.4.51 NAME 'houseIdentifier'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

[3.13.](#) homePhone

The homePhone (Home Telephone Number) attribute type specifies a home telephone number (e.g., "+44 71 123 4567") associated with a person. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.20
  NAME ( 'homePhone' 'homeTelephoneNumber' )
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )
```

[3.14.](#) homePostalAddress

The homePostalAddress attribute type specifies a home postal address for an object. This should be limited to up to 6 lines of 30 characters each. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.39
  NAME 'homePostalAddress'
  EQUALITY caseIgnoreListMatch
  SUBSTR caseIgnoreListSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.41 )
```

[3.15.](#) host

The host attribute type specifies a host computer. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.9
  NAME 'host'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

[3.16.](#) info

The info (Information) attribute type specifies any general information pertinent to an object. It is RECOMMENDED that specific usage of this attribute type is avoided, and that specific requirements are met by other (possibly additional) attribute types. It is noted the description attribute [[RFC2256](#)] for specifying descriptive information pertinent to an object. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.4
  NAME 'info'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{2048} )
```

[3.17.](#) mail

The mail (rfc822mailbox) attribute type holds an the electronic mail address in [[RFC822](#)] form (e.g.: user@example.com). Note that this attribute SHOULD NOT be used to hold non-Internet addresses. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.3
  NAME ( 'mail' 'rfc822Mailbox' )
```



```

EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} )

```

3.18. manager

The Manager attribute type specifies the manager of an object represented by an entry. (Source: [RFC 1274](#))

```

( 0.9.2342.19200300.100.1.10
  NAME 'manager'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

```

3.19. mobile

The mobile (Mobile Telephone Number) attribute type specifies a mobile telephone number (e.g., "+44 71 123 4567") associated with a person. (Source: [RFC 1274](#))

```

( 0.9.2342.19200300.100.1.41
  NAME ( 'mobile' 'mobileTelephoneNumber' )
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )

```

3.20. organizationalStatus

The organizationalStatus attribute type specifies a category by which a person is often referred to in an organization. Examples of usage in academia might include undergraduate student, researcher, lecturer, etc.

A Directory administrator should probably consider carefully the distinctions between this and the title and userClass attributes. (Source: [RFC 1274](#))

```

( 0.9.2342.19200300.100.1.45
  NAME 'organizationalStatus'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )

```

3.21. otherMailbox

The otherMailbox attribute type specifies values for electronic mailbox types other than X.400 and [RFC822](#). (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.22
  NAME 'otherMailbox'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.39 )
```

[3.22.](#) pager

The pager (Pager Telephone Number) attribute type specifies a pager telephone number (e.g., "+44 71 123 4567") for an object. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.42
  NAME ( 'pager' 'pagerTelephoneNumber' )
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )
```

[3.23.](#) personalTitle

The personalTitle attribute type specifies a personal title for a person. Examples of personal titles are "Frau", "Dr", "Herr", and "Prof". (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.40
  NAME 'personalTitle'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

[3.24.](#) roomNumber

The roomNumber attribute type specifies the room number of an object. Note that the cn (commonName) attribute should be used for naming room objects. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.6
  NAME 'roomNumber'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

[3.25.](#) secretary

The secretary attribute type specifies the secretary of a person. The attribute value for Secretary is a distinguished name. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.21
  NAME 'secretary'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
```

[3.26. uid](#)

The uid (userid) attribute type specifies a computer system login name. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.1
  NAME ( 'uid' 'userid' )
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

[3.27. uniqueIdentifier](#)

The Unique Identifier attribute type specifies a "unique identifier" for an object represented in the Directory. The domain within which the identifier is unique, and the exact semantics of the identifier, are for local definition. For a person, this might be an institution-wide payroll number. For an organizational unit, it might be a department code. An attribute value for uniqueIdentifier is a directoryString. (Source: [RFC 1274](#))

```
( 2.5.4.45 NAME 'uniqueIdentifier'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

Note: X.520 describes an attribute also called 'uniqueIdentifier' (2.5.4.45) which is called 'x500UniqueIdentifier' in LDAP [[RFC2256](#)]. The attribute detailed here ought not be confused with x500UniqueIdentifier.

[3.28. userClass](#)

The userClass attribute type specifies a category of computer user. The semantics placed on this attribute are for local interpretation. Examples of current usage of this attribute in academia are

undergraduate student, researcher, lecturer, etc. Note that the `organizationalStatus` attribute may now often be preferred as it makes no distinction between computer users and others. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.1.8 NAME 'userClass'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

4. Object Classes

This section details attribute types for use in LDAP based upon their X.500 descriptions.

4.1. account

The account object class is used to define entries representing computer accounts. The `uid` (`userid`) attribute should be used for naming entries of this object class. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.4.5
  NAME 'account'
  SUP top STRUCTURAL
  MUST uid
  MAY ( description $ seeAlso $ l $ o $ ou $ host ) )
```

4.2. document

The document object class is used to define entries which represent documents. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.4.6
  NAME 'document'
  SUP top STRUCTURAL
  MUST documentIdentifier
  MAY ( cn $ description $ seeAlso $ l $ o $ ou $
        documentTitle $ documentVersion $ documentAuthor $
        documentLocation $ documentPublisher ) )
```

4.3. documentSeries

The documentSeries object class is used to define an entry which represents a series of documents (e.g., The Request For Comments memos). (Source: [RFC 1274](#))


```
( 0.9.2342.19200300.100.4.9
  NAME 'documentSeries'
  SUP top STRUCTURAL
  MUST cn
  MAY ( description $ l $ o $ ou $ seeAlso $
        telephonenumber ) )
```

[4.4.](#) **domainRelatedObject**

The domainRelatedObject object class is used to define entries which represent DNS domains which are "equivalent" to an X.500 domain: e.g., an organization or organizational unit. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.4.17
  NAME 'domainRelatedObject'
  SUP top AUXILIARY
  MUST associatedDomain )
```

[4.5.](#) **friendlyCountry**

The friendlyCountry object class is used to define country entries in the DIT. The object class is used to allow friendlier naming of countries than that allowed by the object class country. The naming attribute of object class country, c (countryName), has to be a 2 letter string defined in [[ISO3166](#)]. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.4.18
  NAME 'friendlyCountry'
  SUP country STRUCTURAL
  MUST co )
```

[4.6.](#) **rFC822LocalPart**

The rFC822LocalPart object class is used to define entries which represent the local part of [[RFC822](#)] mail addresses. This treats this part of an [RFC 822](#) address as a domain [[RFC2247](#)]. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.4.14
  NAME 'rFC822localPart'
  SUP domain STRUCTURAL
  MAY ( cn $ description $ destinationIndicator $
        facsimileTelephoneNumber $ internationaliSDNNumber $
        physicalDeliveryOfficeName $ postalAddress $
        postalCode $ postOfficeBox $ preferredDeliveryMethod $
        registeredAddress $ seeAlso $ sn $ street $
```



```
telephoneNumber $ teletexTerminalIdentifier $  
telexNumber $ x121Address ) )
```

4.7. room

The room object class is used to define entries representing rooms. The cn (commonName) attribute should be used for naming entries of this object class. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.4.7 NAME 'room'  
  SUP top STRUCTURAL  
  MUST cn  
  MAY ( roomNumber $ description $  
        seeAlso $ telephoneNumber ) )
```

4.8. simpleSecurityObject

The simpleSecurityObject object class is used to allow an entry to have a userPassword attribute when an entry's principal object classes do not allow userPassword as an attribute type. (Source: [RFC 1274](#))

```
( 0.9.2342.19200300.100.4.19 NAME 'simpleSecurityObject'  
  SUP top AUXILIARY  
  MUST userPassword )
```

Note: Security considerations related to the use of simple authentication mechanisms in LDAP are discussed in [RFC 2829](#) [[RFC2829](#)].

5. Security Considerations

General LDAP security considerations [[LDAPTS](#)] is applicable to the use of this schema. Additional considerations are noted above where appropriate.

6. Acknowledgments

This document borrows from a number of IETF documents including [RFC 1274](#) by Paul Barker and Steve Kille. This document also borrows from a number of ITU documents including X.520.

7. Author's Address

Kurt D. Zeilenga
OpenLDAP Foundation
<Kurt@OpenLDAP.org>

8. Normative References

- [ISO3166] International Standards Organization, "Codes for the representation of names of countries", ISO 3166.
- [RFC822] D. Crocker, "Standard for the format of ARPA Internet text messages", STD 11 (also [RFC 822](#)), August 1982.
- [RFC1034] P.V. Mockapetris, "Domain names - concepts and facilities", STD 13 (also [RFC 1034](#)), November 1987.
- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#) (also [RFC 2119](#)), March 1997.
- [RFC2247] S. Kille, M. Wahl, A. Grimstad, R. Huber, S. Sataluri, "Using Domains in LDAP/X.500 Distinguished Names", January 1998.
- [RFC2252] M. Wahl, A. Coulbeck, T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", [RFC 2252](#), December 1997.
- [RFC2256] M. Wahl, "A Summary of the X.500(96) User Schema for use with LDAPv3", [RFC 2256](#), December 1997.
- [RFC2829] M. Wahl, H. Alvestrand, J. Hodges, R. Morgan, "Authentication Methods for LDAP", [RFC 2829](#), May 2000.
- [LDAPTS] J. Hodges, R. Morgan, "Lightweight Directory Access Protocol (v3): Technical Specification", [draft-ietf-ldapbis-ldapv3-ts-00.txt](#).

9. Informative References

- [RFC1274] P. Barker, S. Kille, "The COSINE and Internet X.500 Schema", November 1991.
- [RFC2798] M. Smith, "The LDAP inetOrgPerson Object Class", [RFC 2798](#), April 2000.
- [X.520] International Telephone Union, "The Directory: Selected Attribute Types", X.520, 1997.

Full Copyright

Copyright 2001, The Internet Society. 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 AUTHORS, 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.

