

X660LDAP
Internet-Draft
Intended status: Standards Track
Expires: November 15, 2021

J. Coretta
May 15, 2021

**Lightweight Directory Access Protocol (LDAP)
Procedures and Schema Definitions for the
Storage of X.660 Registration Information**
[draft-coretta-x660-ldap-06.txt](#)

Abstract

This specification defines models and schema definitions facilitating the storage of [[X.660](#)] registration data in a Lightweight Directory Access Protocol Directory Information Tree.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on November 15, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Coretta

Expires November 15, 2021

[Page 1]

Table of Contents

1. Introduction	4
1.1. Conventions	4
1.2. Acronyms Used	4
1.3. Intended Audience	4
1.4. OIDs Allocated	4
1.5. Well-Known OIDs	5
2. Schema Definitions	5
2.1. Attribute Types	5
2.1.1. 'n'	5
2.1.2. 'dottedNotation'	6
2.1.3. 'iRI'	6
2.1.4. 'asn1Notation'	7
2.1.5. 'unicodeValue'	7
2.1.6. 'identifier'	7
2.1.7. 'additionalIdentifier'	8
2.1.8. 'longArc'	8
2.1.9. 'arcInformation'	8
2.1.10. 'arcURI'	9
2.1.11. 'arcRegId'	9
2.1.12. 'arcCreateTimestamp'	10
2.1.13. 'arcModifyTimestamp'	10
2.1.14. 'currentRegAuthority'	11
2.1.15. 'currentRegAuthorityStartTimestamp'	11
2.1.16. 'currentRegAuthorityCommonName'	11
2.1.17. 'currentRegAuthorityCountryCode'	12
2.1.18. 'currentRegAuthorityCountryName'	12
2.1.19. 'currentRegAuthorityEmail'	13
2.1.20. 'currentRegAuthorityFax'	13
2.1.21. 'currentRegAuthorityLocality'	13
2.1.22. 'currentRegAuthorityMobile'	14
2.1.23. 'currentRegAuthorityOrg'	14
2.1.24. 'currentRegAuthorityPOBox'	15
2.1.25. 'currentRegAuthorityPostalAddress'	15
2.1.26. 'currentRegAuthorityPostalCode'	15
2.1.27. 'currentRegAuthorityState'	16
2.1.28. 'currentRegAuthorityStreet'	16
2.1.29. 'currentRegAuthorityTelephone'	17
2.1.30. 'currentRegAuthorityTitle'	17
2.1.31. 'currentRegAuthorityURI'	17
2.1.32. 'firstRegAuthority'	18
2.1.33. 'firstRegAuthorityStartTimestamp'	18
2.1.34. 'firstRegAuthorityEndTimestamp'	19
2.1.35. 'firstRegAuthorityCommonName'	19
2.1.36. 'firstRegAuthorityCountryCode'	19
2.1.37. 'firstRegAuthorityCountryName'	20
2.1.38. 'firstRegAuthorityEmail'	20

2.1.39	'firstRegAuthorityFax'	21
2.1.40	'firstRegAuthorityLocality'	21
2.1.41	'firstRegAuthorityMobile'	21
2.1.42	'firstRegAuthorityOrg'	22

2.1.43.	'firstRegAuthorityPOBox'	22
2.1.44.	'firstRegAuthorityPostalAddress'	23
2.1.45.	'firstRegAuthorityPostalCode'	23
2.1.46.	'firstRegAuthorityState'	23
2.1.47.	'firstRegAuthorityStreet'	24
2.1.48.	'firstRegAuthorityTelephone'	24
2.1.49.	'firstRegAuthorityTitle'	25
2.1.50.	'firstRegAuthorityURI'	25
2.1.51.	'sponsor'	25
2.1.52.	'sponsorStartTimestamp'	26
2.1.53.	'sponsorEndTimestamp'	26
2.1.54.	'sponsorCommonName'	26
2.1.55.	'sponsorCountryCode'	27
2.1.56.	'sponsorCountryName'	27
2.1.57.	'sponsorEmail'	28
2.1.58.	'sponsorFax'	28
2.1.59.	'sponsorLocality'	28
2.1.60.	'sponsorMobile'	29
2.1.61.	'sponsorOrg'	29
2.1.62.	'sponsorPOBox'	29
2.1.63.	'sponsorPostalAddress'	30
2.1.64.	'sponsorPostalCode'	30
2.1.65.	'sponsorState'	30
2.1.66.	'sponsorStreet'	31
2.1.67.	'sponsorTelephone'	31
2.1.68.	'sponsorTitle'	32
2.1.69.	'sponsorURI'	32
2.2.	Object Classes	32
2.2.1.	'x660RootArcEntry'	32
2.2.2.	'x660SubArcEntry'	33
2.2.3.	'x660ContactEntry'	33
3.	Directory Models	34
3.1.	Naming Context and Organization Entries	34
3.2.	Two-Dimensional Model	35
3.2.1.	Requirements	35
3.2.2.	Distinguished Name Convention	35
3.2.3.	Root Arc Entries	36
3.2.4.	Arc IRI and ASN.1 Value Storage	37
3.3.	Three-Dimensional Model	37
3.3.1.	Requirements	37
3.3.2.	Distinguished Name Convention	38
3.3.3.	Root Arc Entries	38
3.3.3.1.	Lack of Root Arc Entries	39
3.3.4.	Arc IRI and ASN.1 Value Storage	41
3.4.	Authority and Sponsorship Contact Info	41
3.4.1.	Examples	42
3.4.1.1.	Combined OID and Contact Entries	42
3.4.1.2.	Dedicated Contact Entries	42

4.	References	44
4.1.	Normative References	44
5.	IANA Considerations	45
6.	Security Considerations	45

Author's Address[43](#)

[1.](#) Introduction

This specification describes a means for storing [[X.660](#)] registration and contextual data within an LDAP [[RFC4510](#)] implementation.

[1.1.](#) Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

[1.2.](#) Acronyms Used

This specification makes reference to several acronyms, each of which are defined below.

DN Distinguished Name
RA Registration Authority
PEN IANA Private Enterprise Number
IRI Internationalized Resource Identifier
RDN Relative Distinguished Name
DUA Directory User Agent (an LDAP client)
DIT Directory Information Tree
OID ASN.1 Object Identifier
GUI Graphical User Interface
TUI Textual User Interface
LDAP Lightweight Directory Access Protocol
ASN.1 Abstract Syntax Notation one

[1.3.](#) Intended Audience

This specification is intended for use by any entity or individual in need of a means for storing and serving [[X.660](#)] data, in whole or in part. The most likely candidates for use are RAs, whether internal to an organization, or public.

[1.4.](#) OIDs Allocated

This specification provides a dedicated registered OID branch for all LDAP schema elements as defined in [Section 2](#).

- 1.3.6.1.4.1.56521 (author root)
- 1.3.6.1.4.1.56521.101 (specification OID)

- 1.3.6.1.4.1.56521.101.2 (schema OID)
- 1.3.6.1.4.1.56521.101.2.1 (attribute types OID)

Coretta

Expires November 15, 2021

[Page 4]

- 1.3.6.1.4.1.56521.101.2.2 (object classes OID)

1.5. Well-Known OIDs

This specification makes use of well-known OIDs defined by other parties or institutions. These OIDs are mentioned for example purposes and schema configuration only.

- 1.3 (Identified-Organization, per Section A.4.2 of [[X.660](#)])
- 1.3.6 (dod, per [Section 3.1 of \[RFC1155\]](#))
- 1.3.6.1 (Internet OID, per [Section 3.1 of \[RFC1155\]](#))
- 1.3.6.1.4.1.1466.115.121.1.12 (Distinguished Name syntax and matching rule, per [Section 4.2.15 of \[RFC4517\]](#))
- 1.3.6.1.4.1.1466.115.121.1.24 (Generalized Time syntax, per [Section 3.3.13 of \[RFC4517\]](#))
- 1.3.6.1.4.1.1466.115.121.1.27 (Integer syntax, per [Section 3.3.16 of \[RFC4517\]](#))
- 1.3.6.1.4.1.1466.115.121.1.38 (OID syntax, per [Section 3.3.26 of \[RFC4517\]](#))
- 1.3.6.1.4.1.1466.115.121.1.40 (Octet String syntax, per [Section 3.3.25 of \[RFC4517\]](#))

2. Schema Definitions

This section discusses the particulars of the LDAP schema definitions made available through this specification.

These schema definitions described in this section are provided using LDAP description formats [[RFC4512](#)]. These elements are line-wrapped and indented for readability.

2.1. Attribute Types

The following subsections detail LDAP attribute types created for use within implementations of this specification.

2.1.1. 'n'

The 'n' attribute type allows the assignment of an unsigned integer, meant to represent the primary identifier of the entry.

This attribute type plays a crucial role with regards to DN syntax

used in the three-dimensional directory model described in [Section 3.3](#).

```
( 1.3.6.1.4.1.56521.101.2.1.1
  NAME 'n'
  DESC 'A single unsigned integer value assigned to an OID arc
        to represent its primary integer identifier'
  EQUALITY integerMatch
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )
```

As a usage example, consider the RECOMMENDED distinguished name structure for three-dimensional implementations as described in [Section 3.3](#):

```
dn: n=3,n=1,ou=OID,ou=X660,dc=example,dc=com
```

[2.1.2](#). 'dottedNotation'

The 'dottedNotation' attribute type allows the assignment of an OID value [[X.680](#)] in dot-delimited form to an arc entry.

Please note this attribute type is only required in two-dimensional directory model implementations of this specification.

```
( 1.3.6.1.4.1.56521.101.2.1.2
  NAME 'dottedNotation'
  DESC 'Dotted ASN.1 Object Identifier for non-root OIDs'
  EQUALITY objectIdentifierMatch
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
```

Example:

```
dottedNotation: 1.3.6.1
```

[2.1.3](#). 'iRI'

The 'iRI' attribute type allows the assignment of one or more IRI values to an arc entry.

Please note this attribute type is only required in two-dimensional directory model implementations of this specification, or if clients will not automatically discover a given IRI value by traversal.

```
( 1.3.6.1.4.1.56521.101.2.1.3
  NAME 'iRI'
  DESC 'Internationalized Resource Identifiers for an OID'
  EQUALITY octetStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

Example:

iRI: /ISO/Identified-Organization

Coretta

Expires November 15, 2021

[Page 6]

2.1.4. 'asn1Notation'

The 'asn1Notation' attribute type allows the assignment of an OID in ASN.1 notation to an arc entry.

Please note this attribute type is only required in two-dimensional directory model implementations of this specification, or if clients will not automatically discover a given ASN.1 value by traversal.

```
( 1.3.6.1.4.1.56521.101.2.1.4
  NAME 'asn1Notation'
  DESC 'ASN.1 values, encapsulated between curly
        braces, for an OID'
  SINGLE-VALUE
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

Example:

```
asn1Notation: {iso(1) identified-organization(3)}
```

2.1.5. 'unicodeValue'

The 'unicodeValue' attribute type allows the assignment of the Unicode-based primary identifier (non-numeric) [[X.660](#)] to an arc entry.

```
( 1.3.6.1.4.1.56521.101.2.1.5
  NAME 'unicodeValue'
  DESC 'The primary non-numeric Unicode identifier
        for an arc'
  SINGLE-VALUE
  EQUALITY octetStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

Example:

```
unicodeValue: itu-t
```

2.1.6. 'identifier'

The 'identifier' attribute type allows the assignment of a single non-Unicode, non-numeric identifier [[X.660](#)] to an arc entry.

The attribute type 'name', as defined in [Section 2.18 of \[RFC4519\]](#), is a super type of this attribute type.

Coretta

Expires November 15, 2021

[Page 7]

```
( 1.3.6.1.4.1.56521.101.2.1.6
  NAME 'identifier'
  DESC 'The non-Unicode secondary identifier
        for an arc'
  SINGLE-VALUE
  SUP name )
```

Example:

```
identifier: itu-t
```

2.1.7. 'additionalIdentifier'

The 'additionalIdentifier' attribute type allows the assignment of one or more additional identifiers [[X.660](#)] in an arc entry.

The attribute type 'name', as defined in [Section 2.18 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.7
  NAME 'additionalIdentifier'
  DESC 'The non-Unicode additional identifier for an arc'
  SUP name )
```

Example:

```
additionalIdentifier: ccitt
```

2.1.8. 'longArc'

The 'longArc' attribute type allows the assignment of one or more so-called "Long Arc" well-known identifiers to an arc entry.

Per [[X.660](#)], entries that bear values of this attribute type MUST reside below the root joint-iso-itu-t (2) arc entry.

```
( 1.3.6.1.4.1.56521.101.2.1.8
  NAME 'longArc'
  DESC 'The well-known Long Arc names associated with, and
        registered to, a Joint-ISO-ITU-T subordinate arc'
  EQUALITY octetStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

Example:

```
longArc: /Example
```

2.1.9. 'arcInformation'

The 'arcInformation' attribute type allows the OPTIONAL assignment of octet based values intended for extended documentation or notes in an arc entry.

```
( 1.3.6.1.4.1.56521.101.2.1.9
  NAME 'arcInformation'
  DESC 'Extended information for an arc'
  EQUALITY octetStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

2.1.10. 'arcURI'

The 'arcURI' attribute type allows for the assignment of one or more URI values, with optional labels, to an arc entry.

The attribute type 'labeledURI', as defined in [[RFC2079](#)], is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.10
  NAME 'arcURI'
  DESC 'URI, with an optional label, leading to
       further related subject matter information'
  SUP labeledURI )
```

Example:

arcURI: http://www.example.com Example

2.1.11. 'arcRegId'

The 'arcRegId' attribute type is intended to allow for the singular assignment of a UUID or GUID to a contact entry. When used, this value would act as an absolute identifier for registration entries that may change in the future.

In larger, more complete implementations of this specification, it is RECOMMENDED that this attribute type be the primary identifier (or, RDN) for contact entries. This allows for an absolute and unambiguous reference to any registration entry by DN.

Please note the intended use of this attribute type SHOULD NOT be confused with the act of numbering an arc entry using the numerical form of a GUID or UUID value, such as:

2.25.483275873209587983492589328598493854833

Such an act can be achieved through standard use of the 'n' attribute type (defined in [Section 2.1.1](#)) as it allows an integer value of suitable size to accommodate such a value.

Coretta

Expires November 15, 2021

[Page 9]

```
( 1.3.6.1.4.1.56521.101.2.1.11
  NAME 'arcRegId'
  DESC 'GUID or UUID assigned to a past or
        present registration authority or a
        sponsor entry'
  SINGLE-VALUE
  EQUALITY octetStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
```

Example:

arcRegId: 4ab9c766-a875-4045-8df2-d4505193cca5

[2.1.12.](#) 'arcCreateTimestamp'

The 'arcCreateTimestamp' attribute type allows for the assignment of a generalized timestamp indicating the date and time at which an arc entry was created.

```
( 1.3.6.1.4.1.56521.101.2.1.12
  NAME 'arcCreateTimestamp'
  DESC 'Generalized timestamp for arc entry creation'
  SINGLE-VALUE
  EQUALITY generalizedTimeMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
```

Example:

arcCreateTimestamp: 20130109033116Z

[2.1.13.](#) 'arcModifyTimestamp'

The 'arcModifyTimestamp' attribute type allows for the assignment of one or more generalized timestamp values indicating the dates and times of all applied updates to an arc entry.

Whether multiple dates, or only most recent date, are stored is entirely up to the directory architect(s) involved.

```
( 1.3.6.1.4.1.56521.101.2.1.13
  NAME 'arcModifyTimestamp'
  DESC 'Generalized timestamps for arc entry modification'
  EQUALITY generalizedTimeMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
```

Example:

arcModifyTimestamp: 20130109033116Z

Coretta

Expires November 15, 2021

[Page 10]

2.1.14. 'currentRegAuthority'

The 'currentRegAuthority' attribute type allows for the assignment of one or more DN values to an arc entry.

The value(s) of this attribute type are meant to refer to distinct entries that contain current registration authority information for the indicated arc.

This attribute type is only required if such information is not stored within a given arc entry directly.

```
( 1.3.6.1.4.1.56521.101.2.1.14
  NAME 'currentRegAuthority'
  DESC 'LDAP Distinguished Name of an entry
        bearing current registration authority
        information'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
```

Example:

```
currentRegAuthority: arcRegId=4ab9c766-a875-4045-8df2-d450
                    5193cca5,ou=Registry,ou=X660,dc=example,dc=com
```

2.1.15. 'currentRegAuthorityStartTimestamp'

The 'currentRegAuthorityStartTimestamp' attribute type allows for the assignment of a generalized timestamp value to a current registration authority, indicative of the date and time at which the current registration authority was, or will be, officiated.

```
( 1.3.6.1.4.1.56521.101.2.1.15
  NAME 'currentRegAuthorityStartTimestamp'
  DESC 'Generalized timestamp indicating the date
        and time at which current authority commenced'
  EQUALITY generalizedTimeMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
```

Example:

```
currentRegAuthorityStartTimestamp: 20130105135904Z
```

2.1.16. 'currentRegAuthorityCommonName'

The 'currentRegAuthorityCommonName' attribute type allows for the assignment of a common name to a current registration authority entry.

The attribute type 'cn', as defined in [Section 2.3 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.16
  NAME 'currentRegAuthorityCommonName'
  DESC 'Common Name assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP cn )
```

Example:

currentRegAuthorityCommonName: Jesse Coretta

2.1.17. 'currentRegAuthorityCountryCode'

The 'currentRegAuthorityCountryCode' attribute type allows for the assignment of a country code to a current registration authority entry.

The attribute type 'c', as defined in [Section 2.2 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.17
  NAME 'currentRegAuthorityCountryCode'
  DESC 'Country Code assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP c )
```

Example:

currentRegAuthorityCountry: US

2.1.18. 'currentRegAuthorityCountryName'

The 'currentRegAuthorityCountryName' attribute type allows for the assignment of a country name to a current registration authority entry.

The attribute type 'co', as defined in [Section 2.4 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.18
  NAME 'currentRegAuthorityCountryName'
  DESC 'Country name assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP co )
```

Example:

currentRegAuthorityCountryName: United States

Coretta

Expires November 15, 2021

[Page 12]

2.1.19. 'currentRegAuthorityEmail'

The 'currentRegAuthorityEmail' attribute type allows for the assignment of an email address to the current registration authority entry.

The attribute type 'mail', as defined in [Section 2.16 of \[RFC4524\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.19
  NAME 'currentRegAuthorityEmail'
  DESC 'Email address assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP mail )
```

Example:

```
currentRegAuthorityEmail: jesse.coretta@icloud.com
```

2.1.20. 'currentRegAuthorityFax'

The 'currentRegAuthorityFax' attribute type allows for the assignment of a facsimile telephone number to a current registration authority entry.

The attribute type 'facsimileTelephoneNumber', as defined in [Section 2.10 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.20
  NAME 'currentRegAuthorityFax'
  DESC 'Facsimile telephone number assigned to
        a current registration authority entry'
  SINGLE-VALUE
  SUP facsimileTelephoneNumber )
```

Example:

```
currentRegAuthorityFax: +11234567890
```

2.1.21. 'currentRegAuthorityLocality'

The 'currentRegAuthorityLocality' attribute type allows for the assignment of a locality name to a current registration authority entry.

The attribute type 'l', as defined in [Section 2.16 of \[RFC4519\]](#), is a super type of this attribute type.

Coretta

Expires November 15, 2021

[Page 13]

```
( 1.3.6.1.4.1.56521.101.2.1.21
  NAME 'currentRegAuthorityLocality'
  DESC 'Locality name assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP 1 )
```

Example:

currentRegAuthorityLocality: Palm Springs

2.1.22. 'currentRegAuthorityMobile'

The 'currentRegAuthorityMobile' attribute type allows for the assignment of a mobile telephone number to a current registration authoritative entry.

The attribute type 'mobile', as defined in [Section 2.18 of \[RFC4524\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.22
  NAME 'currentRegAuthorityMobile'
  DESC 'Mobile telephone number assigned to a
        current registration authority entry'
  SINGLE-VALUE
  SUP mobile )
```

Example:

currentRegAuthorityMobile: +11234567890

2.1.23. 'currentRegAuthorityOrg'

The 'currentRegAuthorityOrg' attribute type allows for the assignment of an organization name to a current registration authority entry.

The attribute type 'o', as defined in [Section 2.19 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.23
  NAME 'currentRegAuthorityOrg'
  DESC 'Organization name assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP o )
```

Example:

currentRegAuthorityOrg: Acme, Co.

Coretta

Expires November 15, 2021

[Page 14]

2.1.24. 'currentRegAuthorityPOBox'

The 'currentRegAuthorityPOBox' attribute type allows for the assignment of a post office box number to a current registration authority entry.

The attribute type 'postOfficeBox', as defined in [Section 2.25 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.24
  NAME 'currentRegAuthorityPOBox'
  DESC 'Post office box number assigned to a
        current registration authority entry'
  SINGLE-VALUE
  SUP postOfficeBox )
```

Example:

```
currentRegAuthorityPOBox: 555
```

2.1.25. 'currentRegAuthorityPostalAddress'

The 'currentRegAuthorityPostalAddress' attribute type allows for the assignment of a complete postal address to a current registration authority entry. This single attribute may be used instead of other individual address component attribute types, but will require field parsing.

The attribute type 'postalAddress', as defined in [Section 2.23 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.25
  NAME 'currentRegAuthorityPostalAddress'
  DESC 'Full postal address assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP postalAddress )
```

Example:

```
currentRegAuthorityPostalAddress: 1 Fake St$Anytown$
CA$12345$US
```

2.1.26. 'currentRegAuthorityPostalCode'

The 'currentRegAuthorityPostalCode' attribute type allows for the assignment of a postal code to a current registration authority entry.

The attribute type 'postalCode', as defined in [Section 2.23 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.26
  NAME 'currentRegAuthorityPostalCode'
  DESC 'Postal code assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP postalCode )
```

Example:

currentRegAuthorityPostalCode: 92262

2.1.27. 'currentRegAuthorityState'

The 'currentRegAuthorityState' attribute type allows for the assignment of a state or province name to a current registration authority entry.

The attribute type 'st', as defined in [Section 2.33 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.27
  NAME 'currentRegAuthorityState'
  DESC 'State or province name assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP st )
```

Example:

currentRegAuthorityState: California

2.1.28. 'currentRegAuthorityStreet'

The 'currentRegAuthorityStreet' attribute type allows for the assignment of a street name and number to a current registration authority entry.

The attribute type 'street', as defined in [Section 2.34 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.28
  NAME 'currentRegAuthorityStreet'
  DESC 'Street name and number assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP street )
```

Example:

currentRegAuthorityStreet: 1 Fake Street

Coretta

Expires November 15, 2021

[Page 16]

2.1.29. 'currentRegAuthorityTelephone'

The 'currentRegAuthorityTelephone' attribute type allows for the assignment of a telephone number to a current registration authority entry.

The attribute type 'telephoneNumber', as defined in [Section 2.35 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.29
  NAME 'currentRegAuthorityTelephone'
  DESC 'Telephone number assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP telephoneNumber )
```

Example:

currentRegAuthorityTelephone: +11234567890

2.1.30. 'currentRegAuthorityTitle'

The 'currentRegAuthorityTitle' attribute type allows for the assignment of an official or professional title to a current registration authority entry.

The attribute type 'title', as defined in [Section 2.38 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.30
  NAME 'currentRegAuthorityTitle'
  DESC 'Title assigned to a current
        registration authority entry'
  SINGLE-VALUE
  SUP title )
```

Example:

currentRegAuthorityTitle: Chief Engineer

2.1.31. 'currentRegAuthorityURI'

The 'currentRegAuthorityURI' attribute type allows for the assignment of one or more URI values, with optional labels, to a current registration authority entry.

The attribute type 'labeledURI', as defined in [\[RFC2079\]](#), is a super type of this attribute type.

Coretta

Expires November 15, 2021

[Page 17]

```
( 1.3.6.1.4.1.56521.101.2.1.31
  NAME 'currentRegAuthorityURI'
  DESC 'URI, with an optional label, leading
        to a resource related to a current
        registration authority'
  SUP labeledURI )
```

Example:

currentRegAuthorityURI: http://www.example.com

2.1.32. 'firstRegAuthority'

The 'firstRegAuthority' attribute type allows for the assignment of one or more DN values to an arc entry.

The value(s) of this attribute type are meant to refer to distinct entries that contains previous registration authority information for the indicated arc.

This attribute type is only required if such information is not stored within a given arc entry directly.

```
( 1.3.6.1.4.1.56521.101.2.1.32
  NAME 'firstRegAuthority'
  DESC 'LDAP Distinguished Name of an entry
        bearing previous registration authority
        information'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
```

Example:

firstRegAuthority: arcRegId=4fb7ca10-b100-b33f-ae1-e101
1261ae79,ou=Registry,ou=X660,dc=example,dc=com

2.1.33. 'firstRegAuthorityStartTimestamp'

The 'firstRegAuthorityStartTimestamp' attribute type allows for the assignment of a generalized timestamp value to a previous registration authority, indicative of the date and time at which the previous registration authority was, or will be, officiated.

```
( 1.3.6.1.4.1.56521.101.2.1.33
  NAME 'firstRegAuthorityStartTimestamp'
  DESC 'Generalized timestamp indicating the date and
        time at which a previous registration authority
        commenced'
  EQUALITY generalizedTimeMatch
```

SYNTAX 1.3.6.1.4.1.1466.115.121.1.24)

Coretta

Expires November 15, 2021

[Page 18]

Example:

```
firstRegAuthorityStartTimestamp: 20130105135904Z
```

2.1.34. 'firstRegAuthorityEndTimestamp'

The 'firstRegAuthorityEndTimestamp' attribute type allows for the assignment of a generalized timestamp value to a previous registration authority, indicative of the date and time at which an entity's authoritative role was, or will be, terminated.

```
( 1.3.6.1.4.1.56521.101.2.1.34
  NAME 'firstRegAuthorityEndTimestamp'
  DESC 'Generalized timestamp indicating the date and
        time at which a previous registration authority
        terminated'
  EQUALITY generalizedTimeMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
```

Example:

```
firstRegAuthorityEndTimestamp: 20170528110555Z
```

2.1.35. 'firstRegAuthorityCommonName'

The 'firstRegAuthorityCommonName' attribute type allows for the assignment of a common name to a previous registration authority entry.

The attribute type 'cn', as defined in [Section 2.3 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.35
  NAME 'firstRegAuthorityCommonName'
  DESC 'Common Name assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP cn )
```

Example:

```
firstRegAuthorityCommonName: Jesse Coretta
```

2.1.36. 'firstRegAuthorityCountryCode'

The 'firstRegAuthorityCountryCode' attribute type allows for the assignment of a country code to a previous registration authority entry.

The attribute type 'c', as defined in [Section 2.2 of \[RFC4519\]](#),
is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.36
  NAME 'firstRegAuthorityCountryCode'
  DESC 'Country Code assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP c )
```

Example:

firstRegAuthorityCountry: US

2.1.37. 'firstRegAuthorityCountryName'

The 'firstRegAuthorityCountryName' attribute type allows for the assignment of a country name to a previous registration authority entry.

The attribute type 'co', as defined in [Section 2.4 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.37
  NAME 'firstRegAuthorityCountryName'
  DESC 'Country name assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP co )
```

Example:

firstRegAuthorityCountryName: United States

2.1.38. 'firstRegAuthorityEmail'

The 'firstRegAuthorityEmail' attribute type allows for the assignment of an email address to a previous registration authority entry.

The attribute type 'mail', as defined in [Section 2.16 of \[RFC4524\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.38
  NAME 'firstRegAuthorityEmail'
  DESC 'Email address assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP mail )
```

Example:

firstRegAuthorityEmail: jesse.coretta@icloud.com

Coretta

Expires November 15, 2021

[Page 20]

2.1.39. 'firstRegAuthorityFax'

The 'firstRegAuthorityFax' attribute type allows for the assignment of a facsimile telephone number to a previous registration authority entry.

The attribute type 'facsimileTelephoneNumber', as defined in [Section 2.10 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.39
  NAME 'firstRegAuthorityFax'
  DESC 'Facsimile telephone number assigned to
        a previous registration authority entry'
  SINGLE-VALUE
  SUP facsimileTelephoneNumber )
```

Example:

firstRegAuthorityFax: +11234567890

2.1.40. 'firstRegAuthorityLocality'

The 'firstRegAuthorityLocality' attribute type allows for the assignment of a locality name to a previous registration authority entry.

The attribute type 'l', as defined in [Section 2.16 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.40
  NAME 'firstRegAuthorityLocality'
  DESC 'Locality name assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP l )
```

Example:

firstRegAuthorityLocality: Palm Springs

2.1.41. 'firstRegAuthorityMobile'

The 'firstRegAuthorityMobile' attribute type allows for the assignment of a mobile telephone number to a previous registration authoritative entry.

The attribute type 'mobile', as defined in [Section 2.18 of \[RFC4524\]](#), is a super type of this attribute type.

Coretta

Expires November 15, 2021

[Page 21]

```
( 1.3.6.1.4.1.56521.101.2.1.41
  NAME 'firstRegAuthorityMobile'
  DESC 'Mobile telephone number assigned to a
        previous registration authority entry'
  SINGLE-VALUE
  SUP mobile )
```

Example:

firstRegAuthorityMobile: +11234567890

2.1.42. 'firstRegAuthorityOrg'

The 'firstRegAuthorityOrg' attribute type allows for the assignment of an organization name to a previous registration authority entry.

The attribute type 'o', as defined in [Section 2.19 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.42
  NAME 'firstRegAuthorityOrg'
  DESC 'Organization name assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP o )
```

Example:

firstRegAuthorityOrg: Acme, Co.

2.1.43. 'firstRegAuthorityPOBox'

The 'firstRegAuthorityPOBox' attribute type allows for the assignment of a post office box number to a previous registration authority entry.

The attribute type 'postOfficeBox', as defined in [Section 2.25 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.43
  NAME 'firstRegAuthorityPOBox'
  DESC 'Post office box number assigned to a
        previous registration authority entry'
  SINGLE-VALUE
  SUP postOfficeBox )
```

Example:

firstRegAuthorityPOBox: 555

Coretta

Expires November 15, 2021

[Page 22]

2.1.44. 'firstRegAuthorityPostalAddress'

The 'firstRegAuthorityPostalAddress' attribute type allows for the assignment of a complete postal address to a previous registration authority entry. This single attribute may be used instead of other individual address component attribute types, but will require field parsing.

The attribute type 'postalAddress', as defined in [Section 2.23 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.44
  NAME 'firstRegAuthorityPostalAddress'
  DESC 'Full postal address assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP postalAddress )
```

Example:

```
firstRegAuthorityPostalAddress: 1 Fake St$Anytown$
                                CA$12345$US
```

2.1.45. 'firstRegAuthorityPostalCode'

The 'firstRegAuthorityPostalCode' attribute type allows for the assignment of a postal code to a previous registration authority entry.

The attribute type 'postalCode', as defined in [Section 2.23 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.45
  NAME 'firstRegAuthorityPostalCode'
  DESC 'Postal code assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP postalCode )
```

Example:

```
firstRegAuthorityPostalCode: 92262
```

2.1.46. 'firstRegAuthorityState'

The 'firstRegAuthorityState' attribute type allows for the assignment of a state or province name to a previous registration authority entry.

The attribute type 'st', as defined in [Section 2.33 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.46
  NAME 'firstRegAuthorityState'
  DESC 'State or province name assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP st )
```

Example:

firstRegAuthorityState: California

2.1.47. 'firstRegAuthorityStreet'

The 'firstRegAuthorityStreet' attribute type allows for the assignment of a street name and number to a previous registration authority entry.

The attribute type 'street', as defined in [Section 2.34 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.47
  NAME 'firstRegAuthorityStreet'
  DESC 'Street name and number assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP street )
```

Example:

firstRegAuthorityStreet: 1 Fake Street

2.1.48. 'firstRegAuthorityTelephone'

The 'firstRegAuthorityTelephone' attribute type allows for the assignment of a telephone number to a previous registration authority entry.

The attribute type 'telephoneNumber', as defined in [Section 2.35 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.48
  NAME 'firstRegAuthorityTelephone'
  DESC 'Telephone number assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP telephoneNumber )
```

Example:

firstRegAuthorityTelephone: +11234567890

Coretta

Expires November 15, 2021

[Page 24]

2.1.49. 'firstRegAuthorityTitle'

The 'firstRegAuthorityTitle' attribute type allows for the assignment of an official or professional title to a previous registration authority entry.

The attribute type 'title', as defined in [Section 2.38 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.49
  NAME 'firstRegAuthorityTitle'
  DESC 'Title assigned to a previous
        registration authority entry'
  SINGLE-VALUE
  SUP title )
```

Example:

firstRegAuthorityTitle: Chief Engineer

2.1.50. 'firstRegAuthorityURI'

The 'firstRegAuthorityURI' attribute type allows for the assignment of one or more URI values, with optional labels, to a previous registration authority entry.

The attribute type 'labeledURI', as defined in [\[RFC2079\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.50
  NAME 'firstRegAuthorityURI'
  DESC 'URI, with an optional label, leading
        to a resource related to a previous
        registration authority'
  SUP labeledURI )
```

Example:

firstRegAuthorityURI: http://www.example.com

2.1.51. 'sponsor'

The 'sponsor' attribute type allows for the assignment of one or more DN values to an arc entry.

The value(s) of this attribute type are meant to refer to distinct entries that contains sponsorship-related information for a given arc.

This attribute type is only required if such information is not stored within a given arc entry directly.

```
( 1.3.6.1.4.1.56521.101.2.1.51
  NAME 'sponsor'
  DESC 'LDAP Distinguished Name of an entry
        bearing sponsorship information'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
```

Example:

```
sponsor: arcRegId=1ea6ac09-c200-c11a-c430-c3p0
        3211ffff,ou=Registry,ou=X660,dc=example,dc=com
```

[2.1.52.](#) 'sponsorStartTimestamp'

The 'sponsorStartTimestamp' attribute type allows for the assignment of a generalized timestamp value to a sponsor entry, indicative of the date and time at which sponsorship was, or will be, officiated.

```
( 1.3.6.1.4.1.56521.101.2.1.52
  NAME 'sponsorStartTimestamp'
  DESC 'Generalized timestamp indicating the date
        and time sponsorship commenced'
  EQUALITY generalizedTimeMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
```

Example:

```
sponsorStartTimestamp: 20130105135904Z
```

[2.1.53.](#) 'sponsorEndTimestamp'

The 'sponsorEndTimestamp' attribute type allows for the assignment of a generalized timestamp value to a sponsor entry, indicative of the date and time at which sponsorship was, or will be, terminated.

```
( 1.3.6.1.4.1.56521.101.2.1.53
  NAME 'sponsorEndTimestamp'
  DESC 'Generalized timestamp indicating the date
        and time sponsorship terminated'
  EQUALITY generalizedTimeMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
```

Example:

```
sponsorEndTimestamp: 20170528110555Z
```

[2.1.54.](#) 'sponsorCommonName'

The 'sponsorCommonName' attribute type allows for the assignment

of a common name to a sponsor entry.

Coretta

Expires November 15, 2021

[Page 26]

The attribute type 'cn', as defined in [Section 2.3 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.54
  NAME 'sponsorCommonName'
  DESC 'Common Name of a sponsor entry'
  SINGLE-VALUE
  SUP cn )
```

Example:

```
sponsorCommonName: Jane Sponsor
```

2.1.55. 'sponsorCountryCode'

The 'sponsorCountryCode' attribute type allows for the assignment of a two-letter country code to a sponsor entry.

The attribute type 'c', as defined in [Section 2.2 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.55
  NAME 'sponsorCountryCode'
  DESC 'Country code for a sponsor entry'
  SINGLE-VALUE
  SUP c )
```

Example:

```
sponsorCountryCode: US
```

2.1.56. 'sponsorCountryName'

The 'sponsorCountryName' attribute type allows the assignment of a country name to a sponsor entry.

The attribute type 'co', as defined in [Section 2.4 of \[RFC4524\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.56
  NAME 'sponsorCountryName'
  DESC 'Country name for a sponsor entry'
  SINGLE-VALUE
  SUP co )
```

Example:

```
sponsorCountryName: United States
```

Coretta

Expires November 15, 2021

[Page 27]

2.1.57. 'sponsorEmail'

The 'sponsorEmail' attribute type allows for the assignment of an email address to a sponsor entry.

The attribute type 'mail', as defined in [Section 2.16 of \[RFC4524\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.57
  NAME 'sponsorEmail'
  DESC 'Email address for a sponsor entry'
  SINGLE-VALUE
  SUP mail )
```

Example:

```
sponsorEmail: sponsor@example.com
```

2.1.58. 'sponsorFax'

The 'sponsorFax' attribute type allows for the assignment of a facsimile telephone number to a sponsor entry.

The attribute type 'facsimileTelephoneNumber', as defined in [Section 2.10 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.58
  NAME 'sponsorFax'
  DESC 'Facsimile telephone number for a sponsor entry'
  SINGLE-VALUE
  SUP facsimileTelephoneNumber )
```

Example:

```
sponsorFax: +11234567890
```

2.1.59. 'sponsorLocality'

The 'sponsorLocality' attribute type allows for the assignment of a locality name to a sponsor entry.

The attribute type 'l', as defined in [Section 2.16 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.59
  NAME 'sponsorLocality'
  DESC 'Locality name for a sponsor entry'
  SINGLE-VALUE
  SUP l )
```

Coretta

Expires November 15, 2021

[Page 28]

Example:

sponsorLocality: Anytown

2.1.60. 'sponsorMobile'

The 'sponsorMobile' attribute type allows for the assignment of a mobile telephone number to a sponsor entry.

The attribute type 'mobile', as defined in [Section 2.18 of \[RFC4524\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.60
  NAME 'sponsorMobile'
  DESC 'Mobile telephone number for a sponsor entry'
  SINGLE-VALUE
  SUP mobile )
```

Example:

sponsorMobile: +11234567890

2.1.61. 'sponsorOrg'

The 'sponsorOrg' attribute type allows for the assignment of an organization name to a sponsor entry.

The attribute type 'o', as defined in [Section 2.19 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.61
  NAME 'sponsorOrg'
  DESC 'Organization name for a sponsor entry'
  SINGLE-VALUE
  SUP o )
```

Example:

sponsorOrg: Sponsor Co.

2.1.62. 'sponsorPOBox'

The 'sponsorPOBox' attribute type allows for the assignment of a post office box number to a sponsor entry.

This attribute type 'postOfficeBox', as defined in [Section 2.25 of \[RFC4519\]](#), is a super type of this attribute type.

Coretta

Expires November 15, 2021

[Page 29]

```
( 1.3.6.1.4.1.56521.101.2.1.62
  NAME 'sponsorPOBox'
  DESC 'Post office box number for a sponsor entry'
  SINGLE-VALUE
  SUP postOfficeBox )
```

Example:

sponsorPOBox: 1255

2.1.63. 'sponsorPostalAddress'

The 'sponsorPostalAddress' attribute type allows for the assignment of a complete postal address sponsor entry. This single attribute may be used instead of other individual address component attribute types, but will require field parsing.

The attribute type 'postalAddress', as defined in [Section 2.23 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.63
  NAME 'sponsorPostalAddress'
  DESC 'Full postal address for a sponsor entry'
  SINGLE-VALUE
  SUP postalAddress )
```

Example:

sponsorPostalAddress: 1 Fake St\$Anytown\$CA\$12345\$US

2.1.64. 'sponsorPostalCode'

The 'sponsorPostalCode' attribute type allows for a postal code to be assigned to a sponsor entry.

The attribute type 'postalCode', as defined in [Section 2.23 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.64
  NAME 'sponsorPostalCode'
  DESC 'Postal code for a sponsor entry'
  SINGLE-VALUE
  SUP postalCode )
```

Example:

sponsorPostalCode: 92262

2.1.65. 'sponsorState'

The 'sponsorState' attribute type allows for the assignment of a state or province name to a sponsor entry.

Coretta

Expires November 15, 2021

[Page 30]

The attribute type 'st', as defined in [Section 2.33 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.65
  NAME 'sponsorState'
  DESC 'State or province name for a sponsor entry'
  SINGLE-VALUE
  SUP st )
```

Example:

```
sponsorState: California
```

[2.1.66.](#) 'sponsorStreet'

The 'sponsorStreet' attribute type allows for the assignment of a street name and number to a sponsor entry.

The attribute type 'street', as defined in [Section 2.34 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.66
  NAME 'sponsorStreet'
  DESC 'Street name and number for a sponsor entry'
  SINGLE-VALUE
  SUP street )
```

Example:

```
sponsorStreet: 1 Fake Street
```

[2.1.67.](#) 'sponsorTelephone'

The 'sponsorTelephone' attribute type allows for the assignment of a telephone number to a sponsor entry.

The attribute type 'telephoneNumber', as defined in [Section 2.35 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.67
  NAME 'sponsorTelephone'
  DESC 'Telephone number for a sponsor entry'
  SINGLE-VALUE
  SUP telephoneNumber )
```

Example:

```
sponsorTelephone: +11234567890
```

Coretta

Expires November 15, 2021

[Page 31]

2.1.68. 'sponsorTitle'

The 'sponsorTitle' attribute type allows for the assignment of an official or professional title to a sponsor entry.

The attribute type 'title', as defined in [Section 2.38 of \[RFC4519\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.68
  NAME 'sponsorTitle'
  DESC 'Title for a sponsor entry'
  SINGLE-VALUE
  SUP title )
```

Example:

```
sponsorTitle: Executive Sponsor
```

2.1.69. 'sponsorURI'

The 'sponsorURI' attribute type allows for the assignment of one or more URI values, each with an optional label, to a sponsor entry.

The attribute type 'labeledURI', as defined in [\[RFC2079\]](#), is a super type of this attribute type.

```
( 1.3.6.1.4.1.56521.101.2.1.69
  NAME 'sponsorURI'
  DESC 'URI, with an optional label, for a
    sponsor entry'
  SUP labeledURI )
```

Example:

```
sponsorURI: http://www.example.com Sponsor
```

2.2. Object Classes

The following subsections describes LDAP object classes made available by this specification.

2.2.1. 'x660RootArcEntry'

The 'x660RootArcEntry' class is meant to define a maximum of three (3) root arcs within a DIT, per Rec. ITU-T X.660 (ISO/IEC 9834-1).

Coretta

Expires November 15, 2021

[Page 32]

```
( 1.3.6.1.4.1.56521.101.2.2.1
  NAME 'x660RootArcEntry'
  DESC 'Top-level class for entries meant to represent
        ITU-T, ISO or Joint-ISO-ITU-T root arcs as defined
        in Section A.2 of the X.660 specification'
  SUP top
  STRUCTURAL
  MUST ( n $ unicodeValue )
  MAY ( identifier $ firstRegAuthority $ arcURI $
        arcCreateTimestamp $ arcModifyTimestamp $
        arcInformation $ additionalIdentifier $
        additionalIdentifier $ asn1Notation $
        currentRegAuthority $ sponsor $
        description ) )
```

[2.2.2.](#) 'x660SubArcEntry'

The 'x660SubArcEntry' object class makes a collection of attribute types available for use when crafting sub arc entries within a DIT.

```
( 1.3.6.1.4.1.56521.101.2.2.2
  NAME 'x660SubArcEntry'
  DESC 'A generalized class meant to represent
        subordinate arcs beneath any root, as defined
        in X.660 Sections A.3-A.5'
  SUP top
  STRUCTURAL
  MUST ( n )
  MAY ( asn1Notation $ iRI $ dottedNotation $
        unicodeValue $ currentRegAuthority $
        firstRegAuthority $ arcInformation $
        identifier $ additionalIdentifier $
        arcModifyTimestamp $ sponsor $
        longArc $ arcCreateTimestamp $
        description $ arcURI ) )
```

[2.2.3.](#) 'x660ContactEntry'

The 'x660ContactEntry' object class allows for current, previous (first) and/or sponsorship contact information to be stored within an entry.

In larger, more complete implementations of this specification, it is RECOMMENDED that registration data be stored in dedicated entries that bear this class. In contrast, sparse implementations MAY opt to assign this class directly to entries bearing 'x660RootArcEntry' and 'x660SubArcEntry' object classes, though this is not required.

Coretta

Expires November 15, 2021

[Page 33]

```
( 1.3.6.1.4.1.56521.101.2.2.3
  NAME 'x660ContactEntry'
  DESC 'A generalized auxiliary class for arc
        registration contact information'
  SUP top
  AUXILIARY
  MAY ( currentRegAuthorityCommonName $ firstRegAuthorityMobile $
        currentRegAuthorityTitle $ firstRegAuthorityCountryCode $
        currentRegAuthorityPostalAddress $ firstRegAuthorityOrg $
        currentRegAuthorityFax $ firstRegAuthorityPostalAddress $
        currentRegAuthorityURI $ sponsorCommonName $ sponsorOrg $
        currentRegAuthorityCountryCode $ firstRegAuthorityState $
        currentRegAuthorityCountryName $ firstRegAuthorityEmail $
        currentRegAuthorityMobile $ firstRegAuthorityCommonName $
        firstRegAuthorityPOBox $ sponsorPOBox $ sponsorLocality $
        firstRegAuthorityCountryName $ currentRegAuthorityState $
        sponsorStreet $ sponsorMobile $ sponsorState $ arcRegId $
        sponsorPostalCode $ sponsorPostalAddress $ sponsorEmail $
        currentRegAuthorityPostalCode $ firstRegAuthorityStreet $
        currentRegAuthorityTelephone $ currentRegAuthorityEmail $
        currentRegAuthorityLocality $ firstRegAuthorityLocality $
        currentRegAuthorityStreet $ firstRegAuthorityPostalCode $
        sponsorCountryCode $ firstRegAuthorityURI $ description $
        firstRegAuthorityStartTimestamp $ sponsorStartTimestamp $
        firstRegAuthorityFax $ sponsorCountryName $ sponsorURI $
        sponsorTelephone $ currentRegAuthorityOrg $ sponsorFax $
        firstRegAuthorityTelephone $ currentRegAuthorityPOBox $
        firstRegAuthorityEndTimestamp $ sponsorEndTimestamp $
        currentRegAuthorityStartTimestamp $ sponsorTitle $
        firstRegAuthorityTitle ) )
```

3. Directory Models

This specification offers two (2) distinct models by which directory architects and application developers SHOULD be guided during their efforts for implementation.

Note that in various examples shown, some DNs are particularly long and are line-wrapped and indented for readability.

3.1. Naming Context and Organization Entries

In these examples, a naming context of "ou=X660, dc=example, dc=com" is used as the "suffix". Within this suffix are two (2) entries:

- "ou=OID, ou=X660, dc=example, dc=com" - Storage of all arc registration entries.

- "ou=Registry, ou=X660, dc=example, dc=com" - Storage of all arc authority and sponsorship registration contact entries (OPTIONAL).

Directory architects MAY choose to use models of their own design, so long as noted requirements in the following sections are satisfied.

3.2. Two-Dimensional Model

This model suggests that arc entries reside as siblings within an LDAP DIT in singular, non-hierarchical locations.

This model is RECOMMENDED for small and/or sparse implementations. The three-dimensional model (See [Section 3.3](#)) may be more appropriate for larger, more robust implementations.

Use of this model is entirely at the discretion of the directory architect(s) involved. It should be noted that if users will be managing OID data directly through use of standard LDAP TUI or GUI applications, this model would seem to be more convenient as opposed to the three-dimensional model.

3.2.1. Requirements

One requirement of this model is strict use of the 'dottedNotation' attribute type, covered in [Section 2.1.2](#). This attribute MUST be used on all non-root arc entries.

Root arc entries SHALL NOT bear an 'dottedNotation' value, as the syntax for OIDs (see [Section 3.3.26 of \[RFC4517\]](#)) requires at least two (2) arcs in a given value.

Uniqueness of 'dottedNotation' values within a directory structure MUST always be enforced to ensure unambiguous results. The simplest way to meet this requirement would be to adopt a DN structure based on this attribute type, as shown in the next section.

3.2.2. Distinguished Name Convention

Because all LDAP search requests can be conducted using a "one-level scope" below the circumscribing directory branch, a hierarchical DN structure is unnecessary. While the three-dimensional model (shown in [Section 3.3](#)) uses the integer-based 'n' attribute type (as defined in [Section 2.1.1](#)) to form the effective LDAP RDN of an entry, it is not practical in this model.

The most sensible convention for DN involves use of the attribute type 'dottedNotation' as shown:

```
dn: dottedNotation=1.3,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 3
```

unicodeValue: identified-organization
dottedNotation: 1.3

Coretta

Expires November 15, 2021

[Page 35]

Subsequent entries, regardless of hierarchical superiority, manifest as sibling entries. For example, the addition of deeper arcs would be procedurally identical:

```
dn: dottedNotation=1.3.6.1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 1
unicodeValue: internet
dottedNotation: 1.3.6.1
```

3.2.3. Root Arc Entries

A maximum of three (3) root arcs MAY exist within the directory landscape. If one or more are created, they SHOULD be identifiable as follows:

- itu-t (0)
- iso (1)
- joint-iso-itu-t (2)

As sibling entries, these root arcs MUST use the 'x660RootArcEntry' class, as shown in [Section 2.2.1](#):

```
dn: n=0,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660RootArcEntry
n: 0
unicodeValue: itu-t
```

```
dn: n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660RootArcEntry
n: 1
unicodeValue: iso
```

```
dn: n=2,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660RootArcEntry
n: 2
unicodeValue: joint-iso-itu-t
```

Using root arc entries is only useful in the two-dimensional model if the administrator wishes to organize lists of OIDs beneath their respective root arcs. This is likely unnecessary in implementations that are small and sparse. In larger implementations, however, this model may be convenient in situations where DIT content segmentation

is in effect.

Coretta

Expires November 15, 2021

[Page 36]

3.2.4. Arc IRI and ASN.1 Value Storage

Following this directory model implementation, storage of literal IRI and/or ASN.1 values is required if such values are expected to be present for a given arc entry.

Unlike the three-dimensional model defined in [Section 3.3](#), there is no inherent hierarchy to traverse for the purpose of reading interim 'unicodeValue' values and composing a resultant IRI or ASN.1 value in this directory model. As a result, use of the 'iRI' and/or 'asn1Notation' attribute types is necessary.

3.3. Three-Dimensional Model

This model is hierarchical by nature, providing a means for storing arc entries in "nested" fashion, thereby reflecting the hierarchical logic of the [\[X.660\]](#) specification itself.

This model is RECOMMENDED for thorough or complete implementations, or implementations in which custom solutions (applications) have been tailored for this purpose. This model is NOT RECOMMENDED for sparse and/or small implementations.

Use of this model is entirely at the discretion of the directory architect(s) involved. It should be noted that end-users that will directly access or manage this data through standard LDAP TUI or GUI applications alone may find this model tedious, and may prefer the two-dimensional model as described in [Section 3.2](#).

3.3.1. Requirements

In this model, interim arcs MUST exist even if they are otherwise unnecessary.

For example, in order to add the well-known arc "internet" OID, directory administrators MUST ensure these registrations exist beforehand:

```
dn: n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660RootArcEntry
n: 1
unicodeValue: iso
```

```
dn: n=3,n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 3
unicodeValue: identified-organization
```

Coretta

Expires November 15, 2021

[Page 37]

```
dn: n=6,n=3,n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 6
unicodeValue: dod
```

Only once this requirement is satisfied would the administrators be able to create the desired registration, such as a registration entry for the "internet" OID, as shown in [[RFC1155](#)]:

```
dn: n=1,n=6,n=3,n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 1
unicodeValue: internet
```

3.3.2. Distinguished Name Convention

Under a strict interpretation of this model, its implementation will provide a means for bidirectional resolution of registered arc OIDs. LDAP DNs can be deduced from OIDs, and vice versa.

This is achieved by using the 'n' attribute type (as defined in [Section 2.1.1](#)) as components in the effective LDAP DN, but in reverse order to reflect the directory hierarchy.

For example: the "internet" OID would exist as an entry with a DN as depicted below:

```

      1.3.6.1
      -----
      |   |   |   |
dn: n=1, n=6, n=3, n=1, ou=OID, ou=X660, dc=example, dc=com
```

As a result, use of the 'dottedNotation' attribute type becomes unnecessary.

3.3.3. Root Arc Entries

A maximum of three (3) root arcs SHOULD exist within the directory landscape. If one or more are created, they MUST be identifiable as follows:

- itu-t (0)
- iso (1)
- joint-iso-itu-t (2)

As sibling entries, these root arcs MUST use the 'x660RootArcEntry' class, as shown in [Section 2.2.1](#):

```
dn: n=0,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660RootArcEntry
n: 0
unicodeValue: itu-t
```

```
dn: n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660RootArcEntry
n: 1
unicodeValue: iso
```

```
dn: n=2,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660RootArcEntry
n: 2
unicodeValue: joint-iso-itu-t
```

Depending on the breadth and scope of an implementation, creation and use of root arc entries is RECOMMENDED, but not required for every situation.

3.3.3.1. Lack of Root Arc Entries

In situations where a three-dimensional model is used, but only contains a subset of sub arc entries, true root arc entries may not be necessary.

Instead, directories architects MAY choose to create a sub arc entry as a false root, and store the relevant contents there.

For example, if a directory architect only wanted to store IANA PEN registrations that are allocated below the OID 1.3.6.1.4.1, the relevant DIT entries could manifest as follows:

```
dn: dottedNotation=1.3.6.1.4.1,ou=OID,ou=X660,dc=example,
   dc=com
objectClass: top
objectClass: x660SubArcEntry
arcURI: https://www.iana.org/assignments/enterprise-numbers/enterprise-numbers
```

```
dn: n=56521,dottedNotation=1.3.6.1.4.1,ou=OID,ou=X660,
   dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
unicodeValue: Jesse Coretta
```

... other sibling PEN entries ...

Coretta

Expires November 15, 2021

[Page 39]

Alternatively, if the above non-standard DN syntax is undesirable, directory architects MAY choose to honor the three-dimensional model DN syntax, but limit created entries to those that are direct ancestors of the intended subset. In this situation use of a root arc entry is necessary, but only relevant entries would need to be created.

Keeping with the above scenario, the relevant DIT entries could manifest as follows:

```
dn: n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660RootArcEntry
n: 1
unicodeValue: iso
```

```
dn: n=3,n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 3
unicodeValue: identified-organization
```

```
dn: n=6,n=3,n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 6
unicodeValue: dod
```

```
dn: n=1,n=6,n=3,n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 1
unicodeValue: internet
```

```
dn: n=4,n=1,n=6,n=3,n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 4
unicodeValue: enterprise
```

```
dn: n=1,n=4,n=1,n=6,n=3,n=1,ou=OID,ou=X660,dc=example,dc=com
objectClass: top
objectClass: x660SubArcEntry
n: 1
unicodeValue: private
arcURI: https://www.iana.org/assignments/enterprise-numbers/enterprise-numbers
```

Coretta

Expires November 15, 2021

[Page 40]

```
dn: n=56521,n=1,n=4,n=1,n=6,n=3,n=1,ou=OID,ou=X660,dc=example,  
   dc=com  
n: 56521  
objectClass: top  
objectClass: x660SubArcEntry  
unicodeValue: Jesse Coretta  
  
... other sibling PEN entries ...
```

3.3.4. Arc IRI and ASN.1 Value Storage

Following this directory model implementation, storage of IRI and ASN.1 values is not required, but may still be desirable.

It is assumed that a suitable DUA (one optimized with this specific specification in mind) would be capable of extrapolating any fully qualified IRI or ASN.1 value by way of traversal of all arcs defined in a given branch path. In other words, a DUA can deduce the iRI for the OID '1.3' is '/iso/identified-organization' by reading the value of the attribute type 'unicodeValue' from both the iso root 'n' (1) as well as the specified subordinate 'n' (3), in that order.

However this may not be a desirable action from some points of view. It may be more administratively feasible to simply store the literal IRI and/or ASN.1 values for any given arc entry by way of 'iRI' or 'asn1Notation' attribute type usage respectively.

In terms of a reasonably sound DUA design, it is RECOMMENDED the client check if the 'iRI' and/or 'asn1Notation' attribute types are present and, if not, attempt to extrapolate such values as a fall back or optional action.

3.4. Authority and Sponsorship Contact Info

Directory architects MAY choose to store authority and/or sponsor contact information in one of two main ways:

- Store sponsor or authority contact information within arc entries themselves, or ...
- Store sponsor or authority contact information within dedicated entries, and reference the DNS of these entries via the 'currentRegAuthority', 'sponsor' and/or 'firstRegAuthority' attribute types assigned to arc entries.

The nature of these contacts will typically encompass three (3) kinds of entities:

Coretta

Expires November 15, 2021

[Page 41]

- An individual
- An organization, institution or working group
- A document (e.g.: a standard or draft)

3.4.1. Examples

3.4.1.1. Combined OID and Contact Entries

This is a basic two-dimensional example entry comprised of both OID and contact attribute types.

```
dn: dottedNotation=1.3.6.1.4.1.56521,n=1,ou=OID,ou=X660,
   dc=example,dc=com
objectClass: x660SubArcEntry
objectClass: x660ContactEntry
objectClass: top
currentRegAuthorityPostalAddress: 1 Fake St$Anywhere$CA$92262
currentRegAuthorityCommonName: Jesse Coretta
currentRegAuthorityEmail: jesse.coretta@example.com
currentRegAuthorityMobile: +11234567890
dottedNotation: 1.3.6.1.4.1.56521
unicodeValue: Jesse Coretta
n: 56521
```

This is a basic three-dimensional example entry of the same design.

```
dn: n=56521,n=1,n=4,n=1,n=6,n=3,n=1,ou=OID,ou=X660,
   dc=example,dc=com
objectClass: x660SubArcEntry
objectClass: x660ContactEntry
objectClass: top
currentRegAuthorityPostalAddress: 1 Fake St$Anywhere$CA$92262
currentRegAuthorityCommonName: Jesse Coretta
currentRegAuthorityEmail: jesse.coretta@example.com
currentRegAuthorityMobile: +11234567890
unicodeValue: Jesse Coretta
n: 56521
```

3.4.1.2. Dedicated Contact Entries

This is a basic example of a single authority-based contact entry.

Please note that use of the 'organizationalRole' object class (per [Section 3.10 of \[RFC4519\]](#)) is purely incidental here. Directory architects MAY opt for another STRUCTURAL object class.

Coretta

Expires November 15, 2021

[Page 42]

```
dn: arcRegId=draft-coreтта-x660-ldap,ou=Registry,
   ou=X660,dc=example,dc=com
arcRegId: draft-coreтта-x660-ldap
cn: draft-coreтта-x660-ldap
objectClass: organizationalRole
objectClass: x660ContactEntry
objectClass: top
currentRegAuthorityPostalAddress: 1 Fake St$Palm Springs$
   CA$92262
currentRegAuthorityCommonName: Jesse Coretta
currentRegAuthorityEmail: jesse.coretta@icloud.com
currentRegAuthorityMobile: +11234567890
currentRegAuthorityStartTimestamp: 20200229134901Z
```

In cases where multiple distinct individuals or addresses are used, they can all be combined into a single record:

```
dn: arcRegId=draft-coreтта-x660-ldap,ou=Registry,
   ou=X660,dc=example,dc=com
arcRegId: draft-coreтта-x660-ldap
cn: draft-coreтта-x660-ldap
objectClass: organizationalRole
objectClass: x660ContactEntry
objectClass: top
currentRegAuthorityPostalAddress: 1 Fake St$Palm Springs$
   CA$92262
currentRegAuthorityCommonName: Jesse Coretta
currentRegAuthorityEmail: jesse.coretta@icloud.com
currentRegAuthorityMobile: +11234567890
currentRegAuthorityStartTimestamp: 20200229134901Z
sponsorOrg: Sponsor, Co.
sponsorEmail: sponsor@example.com
sponsorMobile: +11234560987
sponsorStartTimestamp: 20010104120144Z
sponsorPostalAddress: 456 Fugazi Ln$Anywhere$CA$92262
firstRegAuthorityPostalAddress: 1 Fake St$Palm Springs$
   CA$92262
firstRegAuthorityCommonName: Jesse Coretta
firstRegAuthorityEmail: jesse.coretta@icloud.com
firstRegAuthorityMobile: +11234567890
firstRegAuthorityStartTimestamp: 20200229134901Z
```

Keeping with the example arc described in [Section 3.4.1.1](#), the three-dimensional arc entry would manifest as follows:

Coretta

Expires November 15, 2021

[Page 43]

```
dn: n=56521,n=1,n=4,n=1,n=6,n=3,n=1,ou=0ID,ou=X660,  
    dc=example,dc=com  
objectClass: x660SubArcEntry  
objectClass: top  
currentRegAuthority: arcRegId=draft-coreтта-x660-ldap,  
    ou=Registry,ou=X660,dc=example,dc=com  
unicodeValue: Jesse Coretta  
n: 56521
```

4. References

4.1. Normative References

- [RFC1155] Rose, M., "Structure and Identification of Management Information for TCP/IP-based Internets", [RFC 1155](#), May 1990.
- [RFC2079] Smith, M., "Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)", [RFC 2079](#), January 1997.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC4510] Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", [RFC 4510](#), June 2006.
- [RFC4512] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Directory Information Models", [RFC 4512](#), June 2006.
- [RFC4517] Legg, Ed., S., "Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules", [RFC 4517](#), June 2006.
- [RFC4519] Sciberras, Ed., A., "Lightweight Directory Access Protocol (LDAP): Schema for User Applications", [RFC 4519](#), June 2006.
- [RFC4524] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): COSINE LDAP/X.500 Schema", [RFC 4524](#), June 2006.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [RFC 8174](#), May 2017.
- [X.660] International Telecommunication Union - Telecommunication Standardization Sector, "General procedures and top arcs of the international object identifier tree", ITU-T X.660,

July 2011.

Coretta

Expires November 15, 2021

[Page 44]

[X.680] International Telecommunication Union - Telecommunication Standardization Sector, "Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T X.680, July 2002.

5. IANA Considerations

There are no requests to IANA in this document.

6. Security Considerations

This document focuses on providing flexible directory models and LDAP schema elements in order to serve OID-related entries, and to allow for an LDAP-based means for OID resolution.

If some or all of the data in the directory is sensitive in nature, directory architects **MUST** take appropriate steps to secure this information. This concept is out of scope for this document.

Beyond this, there are no specific concerns in the area of security.

Author's Address

Jesse Coretta
Palm Springs, CA 92262
United States

Email: jesse.coretta@icloud.com

Coretta

Expires November 15, 2021

[Page 45]