             Subentries in the Lightweight Directory Access Protocol (LDAP)

Status of this Memo

Copyright Notice

Abstract

   In X.500 directories, subentries are special entries used to hold
   information associated with a subtree or subtree refinement. This
   document adapts X.500 subentries mechanisms for use with the
   Lightweight Directory Access Protocol (LDAP).

1. Overview

   From [X.501]:

       A subentry is a special kind of entry immediately subordinate to
       an administrative point. It contains attributes that pertain to
       a subtree (or subtree refinement) associated with its
       administrative point. The subentries and their administrative
       point are part of the same naming context.

       A single subentry may serve all or several aspects of
       administrative authority. Alternatively, a specific aspect of
       administrative authority may be handled through one or more of
       its own subentries.

   Subentries in the Lightweight Directory Access Protocol (LDAP)
   [RFC3377] SHALL behave in accordance with X.501 unless noted
   otherwise in this specification.

In absence of the subentries control (detailed in Section 3),
subentries SHALL NOT be considered in one-level and subtree scope
search operations. For all other operations, including base scope
search operations, subentries SHALL be considered.

1.1. Conventions

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

Protocol elements are described using ASN.1 [X.680]. The term "BER-
encoded" means the element is to be encoded using the Basic Encoding
Rules [X.690] under the restrictions detailed in Section 5.1 of
[RFC2251].

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

2. Subentry Schema

2.1. Subtree Specification Syntax

The Subtree Specification syntax provides a general purpose mechanism
for the specification of a subset of entries in a subtree of the
Directory Information Tree (DIT). A subtree begins at some base
entry and includes the subordinates of that entry down to some
identified lower boundary, possibly extending to the leaf entries. A
subtree specification is always used within a context or scope which
implicitly determines the bounds of the subtree. For example, the
scope of a subtree specification for a subschema administrative area
does not include the subtrees of any subordinate administrative point
entries for subschema administration. Where a subtree specification
does not identify a contiguous subset of the entries within a single
subtree the collection is termed a subtree refinement.

This syntax corresponds to the SubtreeSpecification ASN.1 type
described in [X.501], Section 11.3. This ASN.1 data type definition
is reproduced here for completeness.

```
SubtreeSpecification ::= SEQUENCE {
    base                 [0] LocalName DEFAULT { },
                             COMPONENTS OF ChopSpecification,
    specificationFilter [4] Refinement OPTIONAL }

LocalName ::= RDNSequence
```

```
    ChopSpecification ::= SEQUENCE {
        specificExclusions  [1] SET OF CHOICE {
                                chopBefore [0] LocalName,
                                chopAfter [1] LocalName } OPTIONAL,
        minimum                 [2] BaseDistance DEFAULT 0,
        maximum                 [3] BaseDistance OPTIONAL }

    BaseDistance ::= INTEGER (0 .. MAX)

    Refinement ::= CHOICE {
        item                    [0] OBJECT-CLASS.&id,
        and                     [1] SET OF Refinement,
        or                      [2] SET OF Refinement,
        not                     [3] Refinement }
```

   The components of SubtreeSpecification are: base, which identifies
   the base entry of the subtree or subtree refinement, and
   specificExclusions, minimum, maximum and specificationFilter, which
   then reduce the set of subordinate entries of the base entry. The
   subtree or subtree refinement contains all the entries within scope
   that are not excluded by any of the components of the subtree
   specification. When all of the components of SubtreeSpecification
   are absent (i.e., when a value of the Subtree Specification syntax is
   the empty sequence, {}), the specified subtree implicitly includes
   all the entries within scope.

   Any particular use of this mechanism MAY impose limitations or
   constraints on the components of SubtreeSpecification.

   The LDAP syntax specification is:

      (1.3.6.1.4.1.1466.115.121.1.45 DESC 'SubtreeSpecification')

   The LDAP-specific encoding of values of this syntax is defined by the
   Generic String Encoding Rules [RFC3641]. Appendix A provides an
   equivalent Augmented Backus-Naur Form (ABNF) [RFC2234] for this
   syntax.

2.1.1. Base

   The base component of SubtreeSpecification nominates the base entry
   of the subtree or subtree refinement. The base entry may be an entry
   which is subordinate to the root entry of the scope in which the
   subtree specification is used, in which case the base component
   contains a sequence of Relative Distinguished Names (RDNs) relative
   to the root entry of the scope, or may be the root entry of the scope
   itself (the default), in which case the base component is absent or
   contains an empty sequence of RDNs.

Entries that are not subordinates of the base entry are excluded from
the subtree or subtree refinement.

2.1.2. Specific Exclusions

The specificExclusions component of a ChopSpecification is a list of
exclusions that specify entries and their subordinates to be excluded
from the subtree or subtree refinement. The entry is specified by a
sequence of RDNs relative to the base entry (i.e., a LocalName).
Each exclusion is of either the chopBefore or chopAfter form. If the
chopBefore form is used then the specified entry and its subordinates
are excluded from the subtree or subtree refinement. If the
chopAfter form is used then only the subordinates of the specified
entry are excluded from the subtree or subtree refinement.

2.1.3. Minimum and Maximum

The minimum and maximum components of a ChopSpecification allow the
exclusion of entries based on their depth in the DIT.

Entries that are less than the minimum number of RDN arcs below the
base entry are excluded from the subtree or subtree refinement. A
minimum value of zero (the default) corresponds to the base entry.

Entries that are more than the maximum number of RDN arcs below the
base entry are excluded from the subtree or subtree refinement. An
absent maximum component indicates that there is no upper limit on
the number of RDN arcs below the base entry for entries in the
subtree or subtree refinement.

2.1.4. Specification Filter

The specificationFilter component is a boolean expression of
assertions about the values of the objectClass attribute of the base
entry and its subordinates. A Refinement assertion item evaluates to
true for an entry if that entry's objectClass attribute contains the
OID nominated in the assertion. Entries for which the overall filter
evaluates to false are excluded from the subtree refinement. If the
specificationFilter is absent then no entries are excluded from the
subtree or subtree refinement because of their objectClass attribute
values.

2.2. Administrative Role Attribute Type

   The Administrative Model defined in [X.501], clause 10 requires that
   administrative entries contain an administrativeRole attribute to
   indicate that the associated administrative area is concerned with
   one or more administrative roles.

   The administrativeRole operational attribute is specified as follows:

        (2.5.18.5 NAME 'administrativeRole'
            EQUALITY objectIdentifierMatch
            USAGE directoryOperation
            SYNTAX 1.3.6.1.4.1.1466.115.121.1.38)

   The possible values of this attribute defined in X.501 are:

        OID   NAME
        --------   ------------------------------
        2.5.23.1   autonomousArea
        2.5.23.2   accessControlSpecificArea
        2.5.23.3   accessControlInnerArea
        2.5.23.4   subschemaAdminSpecificArea
        2.5.23.5   collectiveAttributeSpecificArea
        2.5.23.6   collectiveAttributeInnerArea

   Other values may be defined in other specifications. Names
   associated with each administrative role are Object Identifier
   Descriptors [RFC3383].

   The administrativeRole operational attribute is also used to regulate
   the subentries permitted to be subordinate to an administrative
   entry. A subentry not of a class permitted by the administrativeRole
   attribute cannot be subordinate to the administrative entry.

2.3. Subtree Specification Attribute Type

   The subtreeSpecification operational attribute is defined as follows:

        (2.5.18.6 NAME 'subtreeSpecification'
            SINGLE-VALUE
            USAGE directoryOperation
            SYNTAX 1.3.6.1.4.1.1466.115.121.1.45)

   This attribute is present in all subentries. See [X.501], clause 10.
   Values of the subtreeSpecification attribute nominate collections of
   entries within the DIT for one or more aspects of administrative
   authority.

2.4. Subentry Object Class

   The subentry object class is a structural object class.

        (2.5.17.0 NAME 'subentry'
            SUP top STRUCTURAL
            MUST (cn $ subtreeSpecification))

3. Subentries Control

   The subentries control MAY be sent with a searchRequest to control
   the visibility of entries and subentries which are within scope.
   Non-visible entries or subentries are not returned in response to the
   request.

   The subentries control is an LDAP Control whose controlType is
   1.3.6.1.4.1.4203.1.10.1, criticality is TRUE or FALSE (hence absent),
   and controlValue contains a BER-encoded BOOLEAN indicating
   visibility. A controlValue containing the value TRUE indicates that
   subentries are visible and normal entries are not. A controlValue
   containing the value FALSE indicates that normal entries are visible
   and subentries are not.

   Note that TRUE visibility has the three octet encoding { 01 01 FF }
   and FALSE visibility has the three octet encoding { 01 01 00 }.

   The controlValue SHALL NOT be absent.

   In absence of this control, subentries are not visible to singleLevel
   and wholeSubtree scope Search requests but are visible to baseObject
   scope Search requests.

   There is no corresponding response control.

   This control is not appropriate for non-Search operations.

4. Security Considerations

   Subentries often hold administrative information or other sensitive
   information and should be protected from unauthorized access and
   disclosure as described in [RFC2829][RFC2830].

   General LDAP [RFC3377] security considerations also apply.

5. IANA Considerations

5.1. Descriptors

   The IANA has registered the LDAP descriptors detailed in this
   technical specification. The following registration template is
   suggested:

       Subject: Request for LDAP Descriptor Registration
       Descriptor (short name): see comment
       Object Identifier: see comment
       Person & email address to contact for further information:
           Kurt Zeilenga <kurt@OpenLDAP.org>
       Usage: see comment
       Specification: RFC3672
       Author/Change Controller: IESG
       Comments:

          NAME   Type OID
          -----------------------   ---- --------
          accessControlInnerArea   R   2.5.23.3
          accessControlSpecificArea   R   2.5.23.2
          administrativeRole   A   2.5.18.5
          autonomousArea   R   2.5.23.1
          collectiveAttributeInnerArea   R   2.5.23.6
          collectiveAttributeSpecificArea R   2.5.23.5
          subentry   O   2.5.17.0
          subschemaAdminSpecificArea   R   2.5.23.4
          subtreeSpecification   A   2.5.18.6

       where Type A is Attribute, Type O is ObjectClass, and Type R is
       Administrative Role.

5.2. Object Identifiers

   This document uses the OID 1.3.6.1.4.1.4203.1.10.1 to identify an
   LDAP protocol element defined herein. This OID was assigned [ASSIGN]
   by OpenLDAP Foundation, under its IANA-assigned private enterprise
   allocation [PRIVATE], for use in this specification.

   Other OIDs which appear in this document were either assigned by the
   ISO/IEC Joint Technical Committee 1 - Subcommittee 6 to identify
   elements of X.500 schema or assigned in RFC 2252 for the use
   described here.

5.3. Protocol Mechanisms

   The IANA has registered the LDAP protocol mechanisms [RFC3383]
   detailed in this specification.

   Subject: Request for LDAP Protocol Mechanism Registration

   Description: Subentries

   Person & email address to contact for further information:
       Kurt Zeilenga <kurt@openldap.org>

   Usage: Control

   Specification: RFC3672

   Author/Change Controller: IESG

   Comments: none

6. Acknowledgment

   This document is based on engineering done by IETF LDUP and LDAPext
   Working Groups including "LDAP Subentry Schema" by Ed Reed. This
   document also borrows from a number of ITU documents including X.501.

7. Intellectual Property Statement

   The IETF takes no position regarding the validity or scope of any
   intellectual property or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; neither does it represent that it
   has made any effort to identify any such rights. Information on the
   IETF's procedures with respect to rights in standards-track and
   standards-related documentation can be found in BCP-11. Copies of
   claims of rights made available for publication and any assurances of
   licenses to be made available, or the result of an attempt made to
   obtain a general license or permission for the use of such
   proprietary rights by implementors or users of this specification can
   be obtained from the IETF Secretariat.

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights which may cover technology that may be required to practice
   this standard. Please address the information to the IETF Executive
   Director.

A. Subtree Specification ABNF

   This appendix is non-normative.

   The LDAP-specific string encoding for the Subtree Specification
   syntax is specified by the Generic String Encoding Rules [RFC3641].
   The ABNF [RFC2234] in this appendix for this syntax is provided only
   as a convenience and is equivalent to the encoding specified by the
   application of [RFC3641]. Since the SubtreeSpecification ASN.1 type
   may be extended in future editions of [X.501], the provided ABNF
   should be regarded as a snapshot in time. The LDAP-specific encoding
   for any extension to the SubtreeSpecification ASN.1 type can be
   determined from [RFC3641].

   In the event that there is a discrepancy between this ABNF and the
   encoding determined by [RFC3641], [RFC3641] is to be taken as
   definitive.

   SubtreeSpecification = "{"   [sp ss-base]
                                 [sep sp ss-specificExclusions]
                                 [sep sp ss-minimum]
                                 [sep sp ss-maximum]
                                 [sep sp ss-specificationFilter]
                                   sp "}"

   ss-base   = id-base   msp LocalName
   ss-specificExclusions = id-specificExclusions msp
                             SpecificExclusions
   ss-minimum   = id-minimum   msp BaseDistance
   ss-maximum   = id-maximum   msp BaseDistance
   ss-specificationFilter = id-specificationFilter msp Refinement

   id-base   = %x62.61.73.65 ; "base"
   id-specificExclusions = %x73.70.65.63.69.66.69.63.45.78.63.6C.75.73
                             %x69.6F.6E.73 ; "specificExclusions"
   id-minimum   = %x6D.69.6E.69.6D.75.6D ; "minimum"
   id-maximum   = %x6D.61.78.69.6D.75.6D ; "maximum"
   id-specificationFilter = %x73.70.65.63.69.66.69.63.61.74.69.6F.6E.46
                             %x69.6C.74.65.72 ; "specificationFilter"

   SpecificExclusions = "{" [sp SpecificExclusion
                          *("," sp SpecificExclusion)] sp "}"
   SpecificExclusion = chopBefore / chopAfter
   chopBefore   = id-chopBefore ":" LocalName
   chopAfter   = id-chopAfter ":" LocalName
   id-chopBefore   = %x63.68.6F.70.42.65.66.6F.72.65 ; "chopBefore"
   id-chopAfter   = %x63.68.6F.70.41.66.74.65.72   ; "chopAfter"

```
   Refinement  = item / and / or / not
   item        = id-item ":" OBJECT-IDENTIFIER
   and         = id-and  ":" Refinements
   or          = id-or   ":" Refinements
   not         = id-not  ":" Refinement
   Refinements = "{" [ sp Refinement
                    *( "," sp Refinement ) ] sp "}"
   id-item     = %x69.74.65.6D ; "item"
   id-and      = %x61.6E.64    ; "and"
   id-or       = %x6F.72       ; "or"
   id-not      = %x6E.6F.74    ; "not"

   BaseDistance = INTEGER-0-MAX
```

   The <sp>, <msp>, <sep>, <INTEGER>, <INTEGER-0-MAX>, <OBJECT-
   IDENTIFIER> and <LocalName> rules are defined in [RFC3642].

Normative References

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

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

   [X.690]   ITU-T, "Specification of ASN.1 encoding rules: Basic,
               Canonical, and Distinguished Encoding Rules", X.690,
               1994.

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
               Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC2251]   Wahl, M., Howes, T. and S. Kille, "Lightweight Directory
               Access Protocol (v3)", RFC 2251, December 1997.

   [RFC2252]   Wahl, M., Coulbeck, A., Howes, T. and S. Kille,
               "Lightweight Directory Access Protocol (v3): Attribute
               Syntax Definitions", RFC 2252, December 1997.

   [RFC2829]   Wahl, M., Alvestrand, H., Hodges, J. and R. Morgan,
               "Authentication Methods for LDAP", RFC 2829, May 2000.

   [RFC2830]   Hodges, J., Morgan, R. and M. Wahl, "Lightweight
               Directory Access Protocol (v3): Extension for Transport
               Layer Security", RFC 2830, May 2000.

   [RFC3377]   Hodges, J. and R. Morgan, "Lightweight Directory Access
               Protocol (v3): Technical Specification", RFC 3377,
               September 2002.

    [RFC3383]   Zeilenga, K., "Internet Assigned Numbers Authority (IANA)
                Considerations for the Lightweight Directory Access
                Protocol (LDAP)", RFC 3383, September 2002.

    [RFC3641]   Legg, S., "Generic String Encoding Rules (GSER) for ASN.1
                Types", RFC 3641, October 2003.

Informative References

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

    [RFC3642]   Legg, S., "Common Elements of Generic String Encoding
                Rules (GSER) Encodings", RFC 3642, October 2003.

    [ASSIGN]   OpenLDAP Foundation, "OpenLDAP OID Delegations",
                http://www.openldap.org/foundation/oid-delegate.txt

    [PRIVATE]   IANA, "Private Enterprise Numbers",
                http://www.iana.org/assignments/enterprise-numbers

Authors' Addresses

    Kurt D. Zeilenga
    OpenLDAP Foundation

    EMail: Kurt@OpenLDAP.org

    Steven Legg
    Adacel Technologies Ltd.
    250 Bay Street
    Brighton, Victoria 3186
    AUSTRALIA

    Phone: +61 3 8530 7710
    Fax: +61 3 8530 7888
    EMail: steven.legg@adacel.com.au

Full Copyright Statement

Acknowledgement