INTERNET-DRAFT
draft-ietf-ldup-subentry-07.txt

Ed Reed
Reed-Matthews, Inc.
March 1, 2001

LDAP Subentry Schema

1 Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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.

This Internet-Draft expires on September 1, 2001.

2 Abstract / Description

This document describes two object classes called ldapSubEntry and inheritableLDAPSubEntry, and a control, ldapSubentriesControl (to control the visibility of entries of type ldapSubEntry) which MUST be used by directory servers claiming support for the features of this document to indicate operations and management related entries in the directory, called LDAP Subentries. Scope rules are defined for LDAP Subentries.

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

Expires September 1, 2001

INTERNET-DRAFT 1 March 2001

LDAP Subentry Schema

described in [RFC2119]. The sections below reiterate these definitions and include some additional ones.

3 Table of Contents

4 Object Class Definitions

4.1 ldapSubEntry Class

```
( 2.16.840.1.113719.2.142.6.1.1 NAME 'ldapSubEntry'
   DESC 'LDAP Subentry class, version 1'
     SUP top STRUCTURAL
```

MAY (cn))

The class ldapSubEntry is intended to be used as a super-class when defining other structural classes to be used as LDAP Subentries, and as the structural class to which Auxiliary classes may be added for application specific subentry information. Where possible, the use of Auxiliary classes to extend LDAP Subentries is strongly preferred.

The presence of ldapSubEntry in the list of super-classes of an entry in the directory makes that entry an LDAP Subentry. Object classes derived from ldapSubEntry are themselves considered ldapSubEntry classes, for the purpose of this discussion.

LDAP Subentries MAY be named by their commonName attribute [RFC2251]. Other naming attributes are also permitted.

LDAP Subentries MAY be containers, unlike their [X.501] counterparts.

LDAP Subentries MAY be contained by, and will usually be located in the directory information tree immediately subordinate to, administrative points. Further (unlike X.500 subentries), LDAP Subentries MAY be contained by other LDAP Subentries (the way organizational units may be contained by other organizational units). Deep nesting of LDAP Subentries are discouraged, but not prohibited. Developers are warned that deep nesting of LDAP Subentries may not be supported by all (or indeed, by any) LDAP server implementations.

4.1.1Scope Rules

The default scope of an LDAP Subentry is limited to the administrative area in which it is defined. Specifically, the subtree of the directory namespace based at the administrative point most immediately superior to the LDAP

Subentry, down to but not including any subordinate administrative points or areas. Policy defined in an LDAP Subentry is not inheritable, unless such inheritance is explicitly defined (see the object class definition for InheritableLDAPSubEntry, below, for such an example).

If an LDAP Subentry is subordinate to another LDAP Subentry, it takes the same default scope as the parent LDAP Subentry.

Applications MAY define alternative scope semantics for classes they define which are derived from the ldapSubEntry class. This means that an application can derive a new class from the ldapSubEntry class and add an attribute, like subtreeSpecification [X.501] or inheritance controls (see below), to define a new scope rule for that application to use.

Applications MUST NOT define alternative scope rules for auxiliary classes used to decorate entries of the ldapSubEntry class. This restriction is required to avoid having conflicting or contradictory scope definitions applied by different applications to the same LDAP Subentry.

4.2 InheritableLDAPSubEntry Class

```
( 1.3.6.1.4.1.7628.5.6.1.1 NAME 'inheritableLDAPSubEntry'
   DESC 'Inheritable LDAP Subentry class, version 1'
   SUP ldapSubEntry STRUCTURAL
   MUST ( inheritable )
   MAY  ( blockInheritance )
```

The InheritableLDAPSubentry class is derived from the ldapSubEntry class and provides modified scope semantics to permit and control inheritance from one administrative area to one or more subordinate administrative areas.

If the 'inheritable' attribute is TRUE (1), then the policy

information contained in the InheritableLDAPSubEntry is intended to apply to any (and all) subordinate administrative areas. Subordinate administrative areas MUST include Inheritable LDAP Subentries from their immediately superior administrative area (unless blocked, see below). The means of such inclusion (that is, whether via replication, caching, or explicitly walking the tree to locate and "include" them, are left to the application that consumes the inheritable policy information contained on the inheritableLDAPSubEntry.

If the 'inheritable' attribute is FALSE (0), the policy is NOT inheritable, and subordinate administrative areas MUST treat the associated policy information as UNDEFINED (that is, absent) unless explicitly defined within their own administrative area.

If a subordinate administrative area defines an Inheritable LDAP Subentry for an application with the same name as one defined in a superior administrative area, and if the subordinateÆs Inheritable LDAP Subentry has the attribute 'blockInheritance' with the value TRUE, then inheritance is blocked from the superior administrative area to that subordinate administrative area, and the effect is the same as if the superior Inheritable LDAP Subentry contained the 'inheritable' attribute set to FALSE.

Expires September 1, 2001

The value of the 'blockInheritance' attribute in a superior administrative area Inheritable LDAP Subentry is irrelevant to a subordinate administrative area for this object class.

No mechanism is defined (at this time) to signal to subordinate administrative areas that they may not block inheritable policy from superior administrative areas.

4.2.1Illustration

An illustration may help clarify the use of the class and these attributes.

Suppose the administrative area based at 'dc=com' has an Inheritable LDAP Subentry for an application defined with the 'inheritable' attribute set to TRUE. Subordinate administrative areas, for instance 'dc=widget, dc=com' might or might not want to accept the inherited policy from the 'dc=com' administrative area.

If the administrator of the 'dc=widget, dc=com' administrative area creates an Inheritable LDAP Subentry (say, 'cn=example, dc=widget, dc=com') with the same relative distinguished name as used in the 'dc=com' administrative area (that is, 'cn=example, dc=com') setting the 'blockInheritance' attribute set to TRUE, then the inheritance of the policy defined (on 'cn=example, dc=com') is effectively blocked from affecting the 'dc=widget, dc=com' administrative area. WeÆll call this a blocking subentry for our discussion here.

If the administrator of the 'dc=widget, dc=com' administrative area creates a blocking subentry (as above) with some locally defined policy information, that policy information effectively replaces the policy information defined by the superior administrative area. WeÆll call this an over-riding subentry for our discussion here.

An over-riding subentry MAY itself be inheritable, in which case the 'inheritable' attribute on the locally defined Inheritable LDAP Subentry MAY be set to TRUE or FALSE, at the discretion of the local administrative authority, with appropriate implications for inheritance of the new, locally defined policy, on any other subordinate administrative areas. In this way, the 'dc=widget, dc=com' administrator can set inheritable policy for organizational units (like 'ou=eng, dc=widget, dc=com') for an application

---

Expires September 1, 2001

while over-riding inheritable policy from the superior 'dc=com' administrative area.

5 Attribute Definitions

5.1 inheritable Attribute

```
( 1.3.6.1.4.1.7628.5.4.1 NAME 'inheritable'
   SYNTAX BOOLEAN
   SINGLE-VALUE NO-USER-MODIFICATION USAGE dSAOperation )
```

Used to signal whether an inheritableLDAPSubEntry is intended to be inherited by subordinate administrative areas, or not. TRUE indicates that the subentry and the policy it contains is inheritable.

FALSE indicates that information from the inheritableLDAPSubEntry is not to be inherited by subordinate administrative areas.

5.2 blockInheritance Attribute

```
( 1.3.6.1.4.1.7628.5.4.2 NAME 'blockInheritance'
   SYNTAX BOOLEAN
   SINGLE-VALUE NO-USER-MODIFICATION USAGE dSAOperation )
```

Used by administrators of subordinate administrative areas to over-ride, or block, the inheritance of inheritableLDAPSubEntry policy from superior administrative areas.

A value of TRUE indicates that inheritance is to be blocked.

A value of FALSE is implies that inheritance is not to be blocked, but specific semantic interpretation is left to applications (who may specify any of a variety of policy aggregation mechanisms to define how inherited policy is to be mixed with locally defined policy, which mechanisms are explicitly outside the scope of this specification).

6 Visibility Controls

6.1 ldapSubentriesControl

This control is included in the searchRequest message as part of the controls field of the LDAPMessage, as defined in Section 4.1.12 of [RFC2251].

The controlType is set to "1.3.6.1.4.1.7628.5.101.1". The criticality MAY be set to either TRUE or FALSE. The controlValue is absent.

There is no corresponding response control defined.

LDAP servers that support this control MUST treat LDAP Subentries as "operational objects" in much the same way that "operational attributes" are not returned in search results and [X.511] read operations when only user attributes are requested.

Entries which are not LDAP Subentries may still be referenced in the base object of search operations where the ldapSubentriesControl is present in the request.

6.1.1LDAP Search with scope other than baseObject

The ldapSubentriesControl is defined for LDAP to signal to LDAP Search operations that ONLY LDAP Subentries are to be included in the return set of entries for the Search, provided other Search criteria (such as scope and filter) are satisfied. When ldapSubentriesControl is NOT included in a Search request on a server that supports the control, LDAP Subentries MUST be omitted from the return set (with the single exception described in Search Filter Visibility, below).

6.1.2LDAP Search with scope of baseObject

For Search operations with a scope value of baseObject, the presence or absence of the ldapSubentriesControl MUST be ignored. Specifically, baseObject searches applied to ldapSubEntry entries MUST be evaluated by Search as if the ldapSubentriesControl is present, even if it is absent.

Expires September 1, 2001

This provision is intended to preserve the behavior of [X.511] Read operations, which are not affected by the [X.511] subentries control (see Correspondence to X.500, below), and because it would seem silly to behave otherwise.

6.1.3Other LDAP operations

The ldapSubentriesControl is not defined for any LDAP operation other than Search. However, an LDAPv3 Extension MAY define a use of this control with that extension as long as such use is consistent with this specification.

6.1.4Correspondence to X.500 [X.511]

In [X.511] a ServiceControl option is used to govern the visibility of [X.501] subentries. The subentry ServiceControl option is a specific bit of a bitstring that, when set to TRUE in the common arguments of an X.500 Search or List operation, indicates that the operation is to access ONLY the subentries found in the context of the list or search. In fact, normal entries are explicitly NOT returned in the result of a list or search operation when the X.500 subentries ServiceControl is set.

Entries which are not subentries may still be referenced in the base object of list and search operations where the subentries control is set.

The [X.511] subentries ServiceControl has no meaning for operations other than Search and List (i.e., Read, Modify, Delete, etc.).

In [X.501], the scope of a subentry is a subtree or subtree refinement. The ldapSubEntry class defined in this

document provides no mechanism to define a subtree refinement.

7 Security Considerations

LDAP Subentries will frequently be used to hold data which reflects either the actual or intended behavior of the directory service. As such, permission to read such entries MAY need to be restricted to authorized users.

Expires September 1, 2001

More importantly, IF a directory service treats the information in an LDAP Subentry as the authoritative source of policy to be used to control the behavior of the directory, then permission to create, modify, or delete such entries MUST be carefully restricted to authorized administrators.

This specification defines a policy inheritance model that allows subordinate administrators to over-ride policy defined by administrators of administrative areas superior to the local administrative area. No mechanism is defined here to keep local administrators from over-riding such inherited policy. Implementations that intend to provide such control over the actions of subordinate administrators will require additional semantics (and possibly syntax).

8 References

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

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

[X.501] ITU-T Rec. X.501, "The Directory: Models", 1993 and subsequent versions

[X.511] ITU-T Rec. X.501, "The Directory: Abstract Service Definition", 1993 and subsequent versions

9 Copyright Notice

---

10 Acknowledgements

The utility of subEntry object class was originally suggested as a means to store Replica and Replication Agreement information with a the lucid explanation by Mark Wahl, (then of Innosoft), of how they could be used and extended.

Expires September 1, 2001

11 Author's Address

Edwards E. Reed
Reed-Matthews, Inc.
1064 E 140 North
Lindon, UT 84042

USA
E-mail: eer@oncalldba.com

LDUP Mailing List: ietf-ldup@imc.org
LDAPEXT Mailing List: ietf-ldapext@netscape.com

Expires September 1, 2001