Internet Draft
Document: draft-hahn-schemapart-00.txt
Expires: January 2002

T. Hahn
IBM

July 2001

Approach for identifying different schemas in effect across a Directory Name-space

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

2. Abstract

IETF RFC 2251 [RFC2251] provides a mechanism for indicating, given any particular entry in the directory tree, what entry in the directory tree holds the directory schema information for that particular entry. RFC 2251 does not, however, provide guidance on how different directory servers, each of which might have their own active directory schema, should ôpublicizeö this directory schema such that the different active schemas are distinct from one another when viewing the entire directory name-space. This document describes a way to name sub-schema sub-entry entries such that different active schemas can be distinguished from one another across the entire directory name-space.

3. Table of Contents

4. Conventions used in this document

In this document, directory entries will be described using LDAP Data Interchange Format (LDIF). See RFC 2849 [RFC2849] for details on LDIF.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

5. Review of RFC 2251 and RFC 2252 definition of subschemasubentry

In RFC 2251 and RFC 2252, an operational attribute was defined called subschemasubentry. This attribute can be requested from any entry in the directory tree. When requested, the attributeÆs value will be a distinguished name which ôpoints toö another entry in the name-space. The entry ôpointed toö contains the definition of the directory schema which controls that entry.

While this allows the schema that controls an entry to be found given any entry in the name-space, it does not give guidance on how servers that manage multiple ôactive schemasö or multiple servers would make their active schemas appear ôuniqueö from other schemas that are active in the name-space.

Based on implementation experience, the distinguished names that have been chosen for the subschemasubentry have ranged from fixed names such as ôcn=schemaö to names relative to the namingContexts attribute values in the root DSE entry such as ôcn=schema, o=Your Company, c=USö. It is clear that to promote interoperability and organization of the directory name-space (within single servers and across multiple server environments), more specification of how to name the subschema sub-entry entries is required. If multiple servers name their schema ôcn=schemaö and each subschema sub-entry is different from one another, applications which access data in each of those servers will have difficulty determining which ôcn=schemaö entry is in effect for the name-space. This problem is further confounded with the use of LDAP referrals where the LDAP server on which a request originates may not be the server on which the request is processed.

This document will provide a means of naming subschema sub-entry entries such that each ôactive schemaö has a unique name in the directory name-space.

6. Contents of subschemasubentry

RFC 2251 provides a description of the attributes which should be contained in the subschemasubentry entry. These attributes are ôattributetypesö, ôobjectclassesö, ôldapsyntaxesö, and

ômatchingrulesö. Implementation experience has shown that implementers have added additional attributes including attributes that further define attribute type definitions as well as attributes

to describe naming formats and structure rules. The definition of the subschema sub-entry entry in RFC 2251 has its roots in the X.500 directory model and its definition of a sub-entry which defines the schema for an area of the directory name-space.

A few implementations have named the subschema sub-entry entries based on the tree of information that is controlled by that schema. In these implementations, the subschema sub-entry is also defined as an object class that is derived from the X.500 ôsubentryö object class. The ôsubentryö object class has since been modeled in LDAP schema as a ôldapSubEntryö object class [LDAPSUBENTRY].

By using the ôldapSubEntryö construct coupled with the notion that different portions of the directory name-space may be controlled by different schemas, we can define a mechanism for uniquely naming subschema sub-entry entries across single and multiple server environments.

7. Method of naming subschemasubentry entries as distinct from one another

7.1. Potential problems with ambiguous subschemasubentry values

RFC 2252 defines the ôsubschemasubentryö attribute value. It does not require all entries in the directory to return the same value for this attribute. Indeed, an implementation could choose to define a separate value for every entry in the directory name-space it is controlling and still conform to the requirements for the subschemasubentry attribute from RFC 2251.

Most implementations today take a ôsingle serverö view of the directory name-space. With this view, the choice of naming the ôsubschemasubentryö entry as ôcn=schemaö does not appear to cause any difficulty. After all, if there is only one server serving the directory, there need not be more than one schema. When multiple servers are serving the overall directory name-space (for example, when multiple servers are tied together using LDAP referrals [LDAPREFERRALS], then different servers might contain different active schemas. At this point, if all servers name their schema as ôcn=schemaö, problems can arise as applications access the subschema sub-entry. The same distinguished name refers to different entries, depending upon the server that is contacted. If a server is contacted through following a referral, a subsequent request to retrieve the subschema sub-entry may not follow the referral, causing the wrong subschema sub-entry entry to be returned to the application.

As an example, consider two LDAP servers, server A and server B. If server A has namingContexts in the root DSE entry of:

```
namingContexts: ou=Marketing, o=Your Company
namingContexts: ou=Research, o=Your Company
```

While server B has namingContexts of:

namingContexts: ou=Dept 14, ou=Marketing, o=Your Company

Further, assume that server A holds a referral to server B such that applications which request information below ôou=Dept 14, ou=Marketing, o=Your Companyö will be re-routed to server B for processing.

Also assume that both server A and server B use the same distinguished name, ôcn=schemaö for the subschemasubentry attribute value.

If an application requests the ôsubschemasubentryö attribute from ôou=Dept 14, ou=Marketing, o=Your Companyö from server A, referrals will be followed (presumably), and the value ôcn=schemaö will be returned from server B (unbeknownst to the application). If the application then requests the subschema sub-entry from server A, it will get the ôcn=schemaö entry from server A (not from server B). If the two subschema sub-entry entries were named uniquely, this situation would not occur.

It is within the bounds of RFC 2251 that server A and server B use different distinguished names for the subschema sub-entry. For example, server A could use ôcn=schema, ou=Research, o=Your Companyö and server B could use ôcn=schema, ou=Dept 14, ou=Marketing, o=Your Companyö. If this were done, then when the application requested the subschemasubentry attribute in the prior example, it would be returned a distinguished name that was also in server BÆs ôname-spaceö. If the request for this entry was sent to server A, then the LDAP referral which re-routed the first request to server B would do so again, re-directing the request for the subshema sub-entry to the server on which the schema exists.

There are two other possibilities as well: multiple servers all use the same schema or a single server uses multiple schemas. In either of these cases, if the subschema sub-entry entry is named uniquely (relative to other subschema sub-entry entries that might exist in the directory name-space) then the ôrightö schema can be retrieved unambiguously.

7.2. Subschema sub-entry is really an administrative element

The active schema (or schemas) in a directory server is (are) really an administrative element of that server. This information, similar to replication information or namingContext information, is related to administering the directory server(s) and the directory name-space(s) that those servers are serving.

As an administrative element, it seems a good fit that the subschema sub-entry entry use the object classes and structures that have been

defined for modeling administrative elements in the directory name-space, namely the ldapSubEntry object class defined in [LDAPSUBENTRY]. Using ldapSubEntry also provides the notion of a

ôspan of controlö for the subschema sub-entry entry, something that has been missing from RFC 2251.

There is a slight problem today with defining only the subschemasubentry attribute per RFC 2251. This has to do with predicting which sub-schema subentry will be used when an entry is added to the directory. Since the directory entry does not exist yet, it has no subschemasubentry attribute û thus, there is no way to point to the subschema sub-entry entry that ôwould beö used to verify the entryÆs structure during the processing of the add operation.

Further, when found in the root DSE entry, the single-valued subschemasubentry attribute does not refer to the schema across the server but rather to the subschema entry that contains the definition of the attribute types in the root DSE entry.

By using the ldapSubEntry construct, applications would get a ôhintö regarding what subschema sub-entry ôwould beö in effect when adding an entry to the directory as the ldapSubEntry construct defines its span of control ôdownwardö in the tree until an overriding ldapSubEntry is encountered. Note that this is only a ôhintö since the active schema could change right at the point in the directory name-space where the new entry is being added. This could occur, for example, when the entry at the top of a namingContext is being added and the namingContext is located on a different server.

7.3. Subschema sub-entry entries as ldapSubEntry entries

With the justification in the last two sections, the proposal for naming subschema sub-entry entries across the directory name-space is to

1) define the subschema sub-entry entry to be derived from the ldapSubEntry object class:

```
( 1.3.18.0.2.6.x NAME ældapSubSchemaSubEntryÆ
   SUP ldapSubEntry
   STRUCTURAL
   DESC ÆLDAP sub-entry which represents the active schema
         that is in effect across a sub-tree of the directory
         name-space.  The subschema AUXILIARY object class
         is attached to this sub-entry to reflect the schema
         information.Æ
   )
```

By using the ldapSubSchemaSubEntry object class above, the naming attribute for the entry is ôcnö (per the ldapSubEntry object class). Further, the entry should exist just below the entry at which the subschema sub-entry starts controlling entries in the directory

name-space. Subschema entries are named in relation to the portion of the overall directory name-space to which they apply.

2) recommend that directory servers use this construct to define their subschema sub entry entries and that the ôsubschemasubentryö attribute for an entry should point to the schema that ôcontrolsö the entry (per RFC 2251), and that the name of the subschema sub-entry entry should be specific to what information it controls (if the schema only applies to information in one or a set of servers, then the subschema sub-entry should have a name specific to that server or set of servers).

Using the example from the previous section with server A and server B, server A would have two subschema sub-entry entries:

```
dn: cn=schema, ou=Marketing, o=Your Company
objectclass: ldapSubEntry
objectclass: ldapSubSchemaSubEntry
objectclass: subschema
attributetypes: . . .
objectclasses: . . .
matchingrules: . . .

dn: cn=schema, ou=Research, o=Your Company
objectclass: ldapSubEntry
objectclass: ldapSubSchemaSubEntry
objectclass: subschema
attributetypes: . . .
objectclasses: . . .
matchingrules: . . .
```

There is nothing preventing server A from using the same ôactive schemaö for both of these entries while ôpublicizingö them at both locations in the directory name-space.

Server B from the previous example would have a subschema sub-entry named:

```
dn: cn=schema, ou=Dept 14, ou=Marketing, o=Your Company
objectclass: ldapSubEntry
objectclass: ldapSubSchemaSubEntry
objectclass: subschema
attributetypes: . . .
objectclasses: . . .
matchingrules: . . .
```

By basing this object class on the ldapSubEntry construct, the active schema is presumed to be ôin effectö in the directory name-space starting at the directory entry directly above the ldapSubSchemaSubEntry/ldapSubEntry, until another ldapSubSchemaSubEntry object is encountered lower in the directory name-space.

3) define a new root DSE attribute which ôpoints toö the subschema sub-entry entries that are active within that specific server (since it is possible that multiple schemas may be active within a single server).

Since multiple active schemas may exist across the directory name-space, it would be useful for applications to be able to query the root DSE entry in a directory server to find the names of all ôactive schemasö in that server. The ôsubschemasubentryö attribute in the root DSE is not used for this purpose since this attribute should be used to refer to the subschema sub-entry attribute which controls the formats of the attributes used in the root DSE.

A new attribute must be defined to hold this information:

```
( 1.3.18.0.2.4.x NAME æsubschemasubentriesÆ
   SYNTAX distinguishedName
   EQUALITY distinguishedNameMatch
   DESC æmulti-valued attribute in the root DSE which points to
         all ldapSubSchemaSubEntry entries that are in effect/used
         on this serverÆ )
```

8. Summary

This document has described the current problem of naming subschema sub-entry entries with identical names across multiple LDAP servers that are using different ôactive schemasö. Problems can occur for applications that are attempting to access and/or modify the currently ôactive schemaö, especially when LDAP referrals are used in the environment to build a directory name-space that spans multiple directory servers.

This document recommends that subschema sub-entries build on the ldapSubEntry construct to unambiguously name subschema sub-entry entries across the directory name-space as well as provide a ôhintö for applications in determining the ôactive schemaö that will be used when a new entry is added to the directory. The name of the scubschema sub-entry is distinct in the overall directory name-space from other subschema sub-entries by their placement in the name-space. In addition, this document defines a new root DSE attribute to allow directory servers to ôpublicizeö the set of subschema sub-entries that are controlling entries in the portion of the directory name-space being served by that server.

9. Security Considerations

There are no additional security considerations introduced by the recommendations made in this document. It should be noted that access to and update of the active schema in a directory server should be controlled by some means of access control to ensure that only qualified entities are able to access and/or update the active schema. Unauthorized updates to the active schema could cause existing information in the directory to become unreachable.

10. References

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

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

[RFC2849]
G. Good, ôThe LDAP Data Interchange Format (LDIF) - Technical Specificationö, RFC 2849, June 2000.

[LDAPREFERRAL]
K. Zeilenga, ôNamed Subordinate References in LDAP Directoriesö, Internet Draft, draft-zeilenga-ldap-namedref-03.txt.

[LDAPSUBENTRY]
E. Reed, ôLDAP SubEntry Definitionö, Internet Draft, draft-ietf-ldup-subentry-08.txt, April 2001.

11. Copyright Notice

12. Author's Address

Tim Hahn
IBM
Bldg 256-2, Dept. C8NG
1701 North St.
Endicott, NY 13760 USA
E-mail: hahnt@us.ibm.com