INTERNET-DRAFT
Intended Category: Experimental
Expires in six months

Kurt D. Zeilenga
OpenLDAP Foundation
27 February 2006

The LDAP Manage Directory Information Tree Control
<draft-zeilenga-ldap-managedit-00.txt>

Status of this Memo

Abstract

This document defines the Lightweight Directory Access Protocol (LDAP) Manage Directory Information Tree (DIT) Control which allows a directory user agent (a client) to request the directory service temporarily relax enforcement of constraints of the X.500 models.

1. Background and Intended Use

Directory servers accessible via Lightweight Directory Access Protocol (LDAP) [Roadmap] are expected to act in accordance with the X.500 series of ITU-T Recommendations. In particular, servers are expected to ensure the X.500 data and service models are not violated.

An LDAP server is expected to prevent modification of the structural object class of an object [Models]. That is, the X.500 models do not allow a 'person' object to be transformed into an 'organizationalPerson' object through modification of the object. Instead, the 'person' object must be deleted and then a new 'organizationalPerson' object created in its place. This approach, aside from being inconvient, is problematic for a number reasons. First, as LDAP does not have a standardized method for performing the two operations in a single transaction, the intermediate directory state (after the delete, before the add) is visible to other clients, which may lead to undesirable client behavior. Second, attributes such as entryUUID [entryUUID] will reflect the object was replaced, not transformed.

An LDAP server is expected to prevent clients from modifying values of NO-USER-MODIFICATION attributes [Models]. For instance, an entry is not allowed to assign or modify the value of the entryUUID attribute. However, where an administrator is restoring a previously existing object, for instance when repartitioning data between directory servers or when migrating from one vendor server product to another, it may be desirable to allow the client to assign or modify the value of the entryUUID attribute.

This document specifies the Manage Directory Information Tree (DIT) control. The Manage DIT control may be attached to LDAP requests to update the DIT to request DIT restrictions be temporarily relaxed during the performance of the requested DIT update. The server is however to ensure the resulting directory state is valid.

Use of this control is expected that use of this extension will be restricted by administrative and/or access controls. It is intended to be used by directory administrators.

This extension is considered experimental as it is not yet clear whether it adequately addresses directory administrators' needs for flexible mechanisms for managing directory objects. It is hoped that after suitable amount of time, either this extension or a suitable replacement will be standardization.

1.1. Terminology

Protocol elements are described using ASN.1 [X.680] with implicit tags. The term "BER-encoded" means the element is to be encoded using the Basic Encoding Rules [X.690] under the restrictions detailed in Section 5.2 of [Protocol].

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].

DSA stands for Directory System Agent, a server. DSE stands for DSA-specific Entry.

2. The Manage DIT Control

The Manage DIT control is an LDAP Control [Protocol] whose controlType is IANA-ASSIGNED-OID, controlValue is empty, and the criticality of TRUE.

There is no corresponding response control.

The control is appropriate for all LDAP update requests, including add, delete, modify, and modifyDN (rename) [Protocol].

The presence of the Manage DIT control in an LDAP update request indicates the server temporarily relax X.500 model contraints during performance of the directory update.

The server may restrict use of this control and/or limit the extent of the relaxation provided based upon local policy or factors.

The server is obligated to ensure the resulting directory state is consistent with the X.500 models. For instance, the server ensure that values of attributes conform to the value syntax.

It is noted that while this extension may be used to add or modify objects in a manner which violate the controlling subschema, the presence of objects in the DIT is not inconsistent with the X.500 models. For instance, an object created prior to establshment of a

DIT content rule may contain an attribute now precluded by the current controlling DIT Content Rule.

Servers implementing this technical specification SHOULD publish the object identifier IANA-ASSIGNED-OID as a value of the 'supportedControl' attribute [Models] in their root DSE. A server MAY choose to advertise this extension only when the client is authorized to use it.

3. Use Cases

3.1. Object metamorphism

In absence of this control, an attempt to modify an object's 'objectClass' in a manner which cause a change in the structural object class of the object would normally lead to an objectClassModsProhibited error [Protocol]. The presence of the Manage DIT control in the modify request requests the change be allowed. If the server is willing and able to allow the change in the structural object class of the object.

For instance, to change an 'organization' object into an 'organizationalUnit' object, a client could issue the following LDAP request:

```
dn: o=Unit,dc=example,dc=net
control: IANA-ASSIGNED-OID
changetype: modify
delete: objectClass
objectClass: organization
-
add: objectClass
objectClass: organizationalUnit
-
```

In this case, the server is expected to either effect the requested change in the structural object class, including updating of the value of the structural object class, or fail the operation.

3.2. Inactive Attribute Types

In absence of the Manage DIT control, an attempt to add or modify values to an attribute whose type has been marked inactive in the controlling subschema (its attribute type description contains the OBSOLETE field) [Models] normally results in a failure.

In the presence of the Manage DIT control, the server performs the update operation as if the attribute's type is marked active in the controlling subschema (its attribute type description does not contain the OBSOLETE field).

3.3. DIT Content Rules

In absence of the Manage DIT control, an attempt to include the name (or OID) of an auxiliary class to an object's 'objectClass' which is not allowed by the controlling DIT Content Rule would be disallowed [Models]. Additionally, an attempt to add values of an attribute not allowed (or explicitly precluded) by the DIT Content Rule would fail.

In presence of the Manage DIT control, the server performs the update operation as if the controlling DIT Content Rule allowed any and all known auxiliary classses to be present and allowed any and all known attributes to be present (and precluded no attributes).

3.4. DIT Structure Rules and Name Forms

In absence of the Manage DIT control, the service enforces DIT structure rules and name form requirements of the controlling subschema [Models].

In the presence of the Manage DIT control, the server performs the update operation ignoring all DIT structure rules and name forms in the controlling subschema.

3.5. Modification of Nonconformant Objects

It is also noted that in absense of this control, modification of an object which presently violates the controlling subschema will fail unless the modification would result in the object conforming to the controlling subschema. That is, modifications of an non-conformant object should result in a conformant object.

In the presence of this control, modifications of a non-conformant object need not result in a conformant object.

3.6. NO-USER-MODIFICATION attribute modification

In absence of this control, an attempt to modify values of a NO-USER-MODIFICATION attribute would normally lead to a constraintViolation or other appropriate error [Protocol]. In the

presence of the Manage DIT control in the update request requests the modification be allowed.

Relaxation of the NO-USER-MODIFICATION constraint is not appropriate for some operational attribute types. For instance, as the value of the 'structuralObjectClass' is derived by the values of the 'objectClass' attribute, the 'structuralObjectClass' attribute type's NO-USER-MODIFICATION contraint MUST NOT be relaxed. To effect a change in the structuralObjectClass class, values of objectClass should be changed as discussed in Section 3.1. Other attributes for which the NO-USER-MODIFICATION constraint should not be relaxed include 'entryDN' [EntryDN], 'subschemaSubentry' [Models], and 'collectiveAttributeSubentries' [RFC3671].

The subsections of this section discuss modification of various operational attributes where their NO-USER-MODIFICATION constraint may be relaxed. Future documents may specify where NO-USER-MODIFICATION constraints on other operational attribute may be relaxed. In absence of a document detailing that the NO-USER-MODIFICATION constraint on a particular operational attribute may be relaxed, implementors SHOULD assume relaxation of the constraint is not appropriate for that attribute.

3.1.1. entryUUID

To provide a value for the 'entryUUID' attribute on entry creation, the client should issue an LDAP Add request with a Manage DIT control providing the desired value. For instance:

```
dn: ou=Unit,dc=example,dc=net
control: IANA-ASSIGNED-OID
changetype: add
objectClass: organizationalUnit
ou: Unit
entryUUID: 597ae2f6-16a6-1027-98f4-d28b5365dc14
```

In this case, the server is either to add the entry using the provided 'entryUUID' value or fail the request.

To provide a replacement value for the 'entryUUID' after entry creation, the client should issue an LDAP Modify request with a Manage DIT control including an approrpiate change. For instance:

```
dn: ou=Unit,dc=example,dc=net
control: IANA-ASSIGNED-OID
changetype: modify
replace: entryUUID
```

```
    entryUUID: 597ae2f6-16a6-1027-98f4-d28b5365dc14
    -
```

In this case, the server is either to replace the 'entryUUID' value as requested or fail the request.

3.2.2. createTimestamp

To provide a value for the 'createTimestamp' attribute on entry creation, the client should issue an LDAP Add request with a Manage DIT control providing the desired 'createTimestamp' value. For instance:

```
    dn: ou=Unit,dc=example,dc=net
    control: IANA-ASSIGNED-OID
    changetype: add
    objectClass: organizationalUnit
    ou: Unit
    createTimestamp: 20060101000000Z
```

In this case, the server is either to add the entry using the provided 'createTimestamp' value or fail the request.

To provide a replacement value for the 'createTimestamp' after entry creation, the client should issue an LDAP Modify request with a Manage DIT control including an approrpiate change. For instance:

```
    dn: ou=Unit,dc=example,dc=net
    control: IANA-ASSIGNED-OID
    changetype: modify
    replace: createTimestamp
    createTimestamp: 20060101000000Z
    -
```

In this case, the server is either to replace the 'createTimestamp' value as requested or fail the request.

The server should ensure the requested 'createTimestamp' value is appropriate. In particular, it should fail the request if the requested 'createTimestamp' value is in the future or is greater than the value of the 'modifyTimestamp' attribute.

3.2.3. modifyTimestamp

To provide a value for the 'modifyTimestamp' attribute on entry creation, the client should issue an LDAP Add request with a Manage

DIT control providing the desired 'modifyTimestamp' value. For instance:

```
dn: ou=Unit,dc=example,dc=net
control: IANA-ASSIGNED-OID
changetype: add
objectClass: organizationalUnit
ou: Unit
modifyTimestamp: 20060101000000Z
```

In this case, the server is either to add the entry using the provided 'modifyTimestamp' value or fail the request.

To provide a replacement value for the 'modifyTimestamp' after entry creation, the client should issue an LDAP Modify request with a Manage DIT control including an approrpiate change. For instance:

```
dn: ou=Unit,dc=example,dc=net
control: IANA-ASSIGNED-OID
changetype: modify
replace: modifyTimestamp
modifyTimestamp: 20060101000000Z
-
```

In this case, the server is either to replace the 'modifyTimestamp' value as requested or fail the request.

The server should ensure the requested 'modifyTimestamp' value is appropriate. In particular, it should fail the request if the requested 'modifyTimestamp' value is in the future or is less than the value of the 'createTimestamp' attribute.

3.2.3. creatorsName and modifiersName

To provide a value for the 'creatorsName' and/or 'modifiersName' attribute on entry creation, the client should issue an LDAP Add request with a Manage DIT control providing the desired values. For instance:

```
dn: ou=Unit,dc=example,dc=net
control: IANA-ASSIGNED-OID
changetype: add
objectClass: organizationalUnit
ou: Unit
creatorsName: cn=Jane Doe,dc=example,net
modifiersName: cn=Jane Doe,dc=example,net
```

In this case, the server is either to add the entry using the provided values or fail the request.

To provide a replacement values after entry creation for either of the 'creatorsName' or 'modifiersName' attributes or both, the client should issue an LDAP Modify request with a Manage DIT control including the apprORpiate changes. For instance:

```
dn: ou=Unit,dc=example,dc=net
control: IANA-ASSIGNED-OID
changetype: modify
replace: creatorsName
creatorsName: cn=Jane Doe,dc=example,net
-
replace: modifiersName
modifiersName: cn=Jane Doe,dc=example,net
-
```

In this case, the server is either to replace the provided values as requested or fail the request.

4. Security Considerations

Use of this extension should be subject to appropriate administrative and access controls. Use of this mechanism is intended to be restricted to directory administrators.

Security considerations for the base operations [Protocol] extended by this control, as well as general LDAP security considerations [Roadmap], generally apply to implementation and use of this extension.

5. IANA Considerations

5.1. Object Identifier

It is requested that IANA assign a LDAP Object Identifier [BCP64bis] to identify the LDAP Assertion Control defined in this document.

```
Subject: Request for LDAP Object Identifier Registration
Person & email address to contact for further information:
    Kurt Zeilenga <kurt@OpenLDAP.org>
Specification: RFC XXXX
Author/Change Controller: Kurt Zeilenga <kurt@openldap.org>
Comments: Identifies the LDAP Manage DIT Control
```

5.2 LDAP Protocol Mechanism

Registration of this protocol mechanism [BCP64bis] is requested.

```
Subject: Request for LDAP Protocol Mechanism Registration
Object Identifier: IANA-ASSIGNED-OID
Description: Manage DIT Control
Person & email address to contact for further information:
    Kurt Zeilenga <kurt@openldap.org>
Usage: Control
Specification: RFC XXXX
Author/Change Controller: Kurt Zeilenga <kurt@openldap.org>
Comments: none
```

6. Author's Address

Kurt D. Zeilenga
OpenLDAP Foundation

Email: Kurt@OpenLDAP.org

7. References

[[Note to the RFC Editor: please replace the citation tags used in referencing Internet-Drafts with tags of the form RFCnnnn where possible.]]

7.1. Normative References

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

[Roadmap] Zeilenga, K. (editor), "LDAP: Technical Specification Road Map", draft-ietf-ldapbis-roadmap-xx.txt, a work in progress.

[Models] Zeilenga, K. (editor), "LDAP: Directory Information Models", draft-ietf-ldapbis-models-xx.txt, a work in progress.

7.2. Informative References

[BCP64bis] Zeilenga, K., "IANA Considerations for LDAP",

draft-ietf-ldapbis-bcp64-xx.txt, a work in progress.

[EntryUUID] Zeilenga, K., "The LDAP EntryUUID Operational Attribute", draft-zeilenga-ldap-uuid-xx.txt, a work in progress.

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

Intellectual Property Rights

Full Copyright

ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.