IETF ASID Working Group
Internet Draft

Sanjay Jain
Oracle Corporation
Uppili Srinivasan
Oracle Corporation
Gordon Good
Netscape Communications Corporation
July 1997

Schema for Replication Information
<draft-ietf-asid-ldap-repl-info-01.txt>

I. Status of this Memo

This document is an Internet-Draft. 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.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet- Drafts Shadow Directories on ds.internic.net (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

Distribution of this memo is unlimited. Editorial comments should be sent to skjain@us.oracle.com. Technical discussion should take place on the IETF ASID mailing list (ietf-asid@umich.edu).

This Internet Draft expires February 1st, 1998.

II. Abstract

This document defines a new attribute syntax and an operational attribute type to store replication agreements within the directory. In addition it defines a framework to detect whether an entry is a master or replica.

The replication agreement structure defined here includes a placeholder to specify the replication protocol associated with an agreement. This document itself does not define any replication protocol. Replication protocols and replication agreements are seen as orthogonal issues.

III. Definition Of Terms

Consumer An LDAP server which receives replica updates.

Master An LDAP server holding a master copy of an entry.

Master Entry The copy of an entry where all direct LDAP modifications are performed. In a multi-master environment, there could exist multiple master copies for an entry.

Replication Agreement
An agreement between a supplier and a consumer regarding replication of a full/partial naming context. The main components of a replication agreement are:

- Reference to the supplier
- Reference to the consumer
- Specification of the directory information to be replicated.
- Schedule for updating the replicas.

Replica A copy of an entry where modifications are allowed only through replication updates. I.e. direct LDAP modifications are not allowed on a replica of an entry. A replica will typically issue a referral to a supplier if a client attempts an update operation.

Supplier An LDAP server which supplies replica updates.

IV. Introduction

The broadening of interest in LDAP directories beyond their use as front ends to X.500 and proliferation of stand alone LDAP implementations has created a need to define standards that would facilitate deployment of distributed and replicated directory involving multi-vendor implementations.

The "referrals" draft (Ref[4]) defines ways to create and use knowledge references. It lays the foundation for distributed directories by defining mechanisms that can be used to "glue" the parts of directory information tree (DIT) together. The framework defined here is a step in the direction to standardize the directory replication process. Together, these proposals establish the infrastructure to distribute and replicate the directory information.

This document defines a new attribute syntax and an operational attribute type to store replication agreements within the directory. The attribute is a root DSE attribute.

A replication agreement is established between a consumer and a supplier. Establishment of an agreement is expected to precede any replication. An agreement specifies, among other things, the replication area and the update frequency. It also identifies the replication protocol and the parameters that are specific to that protocol.

While it is desirable for a single LDAP replication protocol to be defined and standardized, this draft accommodates the possibility of variat-

ions and versions to be specified in the agreement through a replication protocol identifier (oid) and associated parameters. Some of the possible variations include characteristics such as supplier initiated (push), consumer initiated (pull) etc.

Proprietary replication protocols already exist and LDAP directories will be rolled out into these environments. Replication protocol oids can be registered for such protocols as well. The intent is not to promote proprietary protocols, but to facilitate painless roll-out of standards based implementations within pre-existing directory environments. It is believed that this accommodation in the replication agreements would ease adoption of the standards.

V. Replication Agreement Establishment and Usage Model

Creation of a replication agreement would either be initiated by a human administrator or by a consumer. Identical copies of an agreement would exist on the supplier and the consumer. A replication agreement has to exist before replication can take place. The servers will enforce replication exchanges per the terms of an agreement.

The replication agreements should be added, deleted or modified using LDAP "modify" operation on the root DSE. The server should check the consistency and acceptability of an agreement based on local administrative policies. An agreement is considered to be in effect when both the supplier and consumer have identical copies of the agreement in their root DSEs. Two agreements are deemed identical when they match according to the replBindingMatch matching rule defined in this document.

It is possible to achieve the effect of the above administrative actions (creation of replication agreements) through protocol exchanges between servers, equivalent of the X.500 DOP. Such protocols may get standardized in the future. This draft does not define such a protocol. It only defines a standard for the structure and semantics of resulting replication agreement information.

A supplier can maintain different replication agreements with different

consumer servers for a single naming context. It is also possible for a consumer to have multiple agreements with one or more suppliers for same or different naming contexts.

VI. Attribute Definitions

A. replAgreement

(< ora_onldap_replica_oid1> NAME 'replAgreement' DESC 'describes a replication agreement between 2 servers' EQUALITY replBindingMatch SYNTAX 'replBinding' USAGE distributedOperation)

'replAgreement' attribute stores the information about a replication agreement reached between two servers (consumer and supplier). It is a multi-valued attribute. It is a root DSE attribute.

A replication agreement specifies, among other details, the user entries and attributes to be replicated from the supplier LDAP server to the consumer LDAP server. Normally, the administrative information (Schema, ACLs etc.), associated with the selected user entries, would also be replicated along with the user entries.

B. supportedReplProtocols

(< ora_onldap_replica_oid2> NAME 'supportedReplProtocols' DESC 'Stores the OIDS of the supported replication protocols" SYNTAX 'OID' USAGE distributedOperation)

'supportedReplProtocols' attribute stores the OIDS of the supported replication protocols. It is a multi-valued attribute. It is a root DSE attribute.

C. myRef

(< ora_onldap_replica_oid3> NAME 'myRef' DESC 'LDAP URI without any DN part. Reference to self.' EQUALITY caseExactIA5Match SYNTAX 'IA5String' SINGLE_VALUE USAGE dSAOperation)

'myRef' is a root DSE attribute . It is a single valued attribute. It contains the LDAP reference (without the DN part) to the server itself.

D. masterRef

(< ora_onldap_replica_oid4> NAME 'masterRef' DESC 'LDAP URI without any DN part' EQUALITY caseExactIA5Match SYNTAX 'IA5String' USAGE dSAOperation)

'masterRef' is an operational attribute for every entry (except the DSE root entry) in the LDAP directory. It is a multi-valued attribute. The values of this attribute refer to the servers which master this entry.

By comparing 'myRef' attribute value and the 'masterRef' attribute values for an entry, a server and a directory user can determine whether the particular copy is a master copy or a replica. When an LDAP client tries a modify operation on a replica then the server by looking at the 'masterRef' attribute can return a referral to the master LDAP server.

VII. replBinding Syntax

(< ora_onldap_replica_oid5> DESC 'Replication Agreement Syntax')

Attribute values with 'replBinding' syntax are written according to following BNF:

```
<replBinding>     ::= "("
                      "AGREEMENTID"            <NumericString>
                      "NAMINGCONTEXT"          <DirectoryString>
                      "SUPPLIER_REFERENCE"     <IA5STRING>
```



```
                      "CONSUMER_REFERENCE"     <IA5STRING>
                      ["SUPPLIER_IDENTITY"     <DirectoryString>]
                      ["CONSUMER_IDENTITY"     <DirectoryString>]
                      ["BASEDN"                <DirectoryString>]
                      ["FILTER"                <DirectoryString>]
                      ["SCOPE"                 <0|1|2>]
                      ["ATTRIBUTE_SELECTION    <AttributeSelectionList>]
                      "UPDATE_SCHEDULE"        <UpdateSchedule>
                      ["REPLICATION_PROTOCOL"  <ReplProtocol>]
                      ")"
```

AGREEMENTID is local to the supplier. The server must verify that the combination of AGREEMENTID and SUPPLIER_REFERENCE is unique. Otherwise the LDAP "modify" operation to enter an agreement should fail with an LDAP_CONSTRAINT_VIOLATION (0x13) LDAP error.

NAMINGCONTEXT is the DN of the root of the naming context with which the replication agreement is associated.

SUPPLIER_REFERENCE is an LDAP URI [6] (without the DN part) to the supplier.

CONSUMER_REFERENCE is an LDAP URI [6] (without the DN part) to the consumer.

SUPPLIER_IDENTITY is the identity which the supplier must use to authenticate itself to the consumer for replica update exchanges. This parameter would be most useful in the push model.

CONSUMER_IDENTITY is the identity which the consumer must use to authenticate itself to the supplier for replica update exchanges. This parameter would be most useful in the pull model.

Note: For security reasons, bind credentials MUST NOT be stored in the replication agreement attribute.

SUPPLIER_IDENTITY, CONSUMER_IDENTITY and associated credentials may not actually exist in the directory. When binding for sending/receiving replication updates, the target server would recognize the SUPPLIER_IDENTITY /CONSUMER_IDENTITY and adjust its semantics appropriately.

SUPPLIER_IDENTITY and CONSUMER_IDENTITY are both optional parameters of a replication agreement.

BASEDN, FILTER and SCOPE have same semantics as in a LDAP search request . Default value for BASEDN is the root of the naming context. Default value for SCOPE is 2 (whole subtree).

Through ATTRIBUTE_SELECTION one can specify the list of attributes of a particular object class which should be included/excluded in the replication.


```
AttributeSelectionList      ::=    <AttributeSelection> |
                                   '('<AttributeSelectionList>')'

AttributeSelectionList      ::=    <AttributeSelectionList> '$'
                                   <AttributeSelection> |
                                   <AttributeSelection>

AttributeSelection          ::=    "(" [<ldapFilter>]
                                       [<AttributeSelectionSpec>]
                                   ")"

<ldap-Filter>               ::=    An LDAP filter as described in [5]
```

If the filter is absent, then the attribute list selection applies to all replicated entries.

```
<AttributeSelectionSpec>    ::=     <IncludeList> |
                                    <ExcludeList>
```

```
<IncludeList>               ::=     "INCLUDE"  <AttributeList>

<ExcludeList>               ::=     "EXCLUDE"  <AttributeList>
<AttributeList>             ::=     <AttributeType> |
                                    '(' <AttributeList> ')'

 <AttributeList>            ::=     <AttributeList> '$'
                                    <AttributeType> |
                                    <AttributeType>
```

UPDATE_SCHEDULE specifies the schedule for updating the replica.

```
<UpdateSchedule>            ::=      <ScheduleList>

<ScheduleList>              ::=      <ScheduleItem> |
                                     '('<ScheduleList>')'

<ScheduleList>              ::=      < ScheduleList> '$'
                                     <ScheduleItem> |
                                     <ScheduleItem>

<ScheduleItem>              ::=      <minute> <hour> <day_of_month>
                                     <month_of_year> <day_of_week>
```

<ScheduleItem> has UNIX crontab style syntax and semantics. The schedule is specified using five fields separated by spaces. The fields are integer patterns that specify the following:

```
            minute (0-59)
            hour (0-23)
            day of the month (1-31)
            month of the year (1-12)
            day of the week (0-6 with 0=Sunday)
```


Each of these five patterns may be either an asterisk (meaning all legal values) or a list of elements separated by commas. An element is either a number or two numbers separated by a minus sign (meaning an inclusive range). Note that the specification of days may be made by two fields (day of the month and day of the week). Both are adhered to if specified as a list of elements. Times are always interpreted as Greenwich Mean Time.

Examples:

1.Always keep replicas updated immediately:

```
            UPDATE_SCHEDULE  * * * * *
```

2. Update replicas at 1 am every day:

UPDATE_SCHEDULE * 1 * * *

3. Update replicas every hour between 8am and 5 pm on weekdays, and at 1am on weekends:

UPDATE_SCHEDULE (* 8-17 * * 1-5 $ * 1 * * 0, 6)

4. Update replicas every 30 minutes:

UPDATE_SCHEDULE 0,30 * * * *

5. Update replicas on first and fifteenth of each month as well as on every Monday (Example of using two fields for the specification of days)

UPDATE_SCHEDULE * * 1,15 * 1

REPLICATION_PROTOCOL is the protocol to be used to exchange replica updates.

```
ReplProtocol       ::=        "("
                                "OID"                <oid>
                                ["PROTOCOL_SPECIFIC" <DirectoryString>]
                              ")"
```

VIII. replBindingMatch definition

(< ora_onldap_shadow_oid6> NAME ' replBindingMatch' SYNTAX 'replBinding ')

Two replication agreements are equal if and only if the AGREEMENTID and SUPPLIER_REFERENCE match for equality, as determined by the NumericStringMatch and CaseExactIA5Match matching rules defined in X.520 (1993).

IX. Relationship To X.500 Shadowing Agreements

LDAP replication agreements are independent of X.500 shadowing agreements, although terminology has been borrowed from X.500 and, in general, the meaning of these terms is equivalent.

It is anticipated that LDAP servers which front-end and X.500 server will advertise the fact that they support X.500 DISP by placing in the 'supportedReplProtocols' attribute of the root DSE the OID which represents X.500 DISP (this OID has not been defined at this time).

X. Examples

Following are examples of attribute values for replAgreement attribute:

1. This is an example of a replication agreement between two cooperating organizations, "ABC Inc." and "XYZ AG". ABC's LDAP server runs on the host "ldap.abc.com" while XYZ's LDAP server runs on the host "ldap.mch.xyz.de". The objective is for both organizations to have a copy of the subtree "ou=Sales, o=ABC, c=US". The server "ldap.mch.xyz.de", then, is a consumer for the subtree "ou=Sales, o=ABC, c=US", which is part of the "c=US" naming context held by the supplier. The server "ldap.abc.com" is the supplier for that subtree. The replica is updated by the supplier (since only supplier identity isthere in the agreement) continuously, using the replication protocol identified by the OID '1.2.3'. When connecting to the consumer, the supplier will bind as "cn=replAdmin,dc=replTree,dc=ldap,dc=mch,dc=xyz,dc=de". The following replAgreement attribute describes this agreement:

```
( AGREEMENTID 123456 NAMINGCONTEXT 'c=us' SUPPLIER_REFERENCE
  'ldap://ldap.abc.com' CONSUMER_REFERENCE
  'ldap://ldap.mch.xyz.de' SUPPLIER_IDENTITY
  'cn=replAdmin,dc=replTree,dc=ldap,dc=mch,dc=xyz,dc=de'
  BASEDN 'ou=Sales, o=ABC, c=US' SCOPE 2 UPDATE_SCHEDULE * * *
  * * REPLICATION_PROTOCOL 1.2.3 )
```

2. This example is a refinement of the above agreement. In addition to the parameters described above, entries are only replicated if they match the filter "(objetclass=person)". Furthermore, only the attributes cn, sn, telephoneNumber, and mail are replicated, and replication may only occur between 1 am and 2 am GMT.

```
( AGREEMENTID 123456 NAMINGCONTEXT 'c=us' SUPPLIER_REFERENCE
  'ldap://ldap.abc.com' CONSUMER_REFERENCE
  'ldap://ldap.mch.xyz.de' SUPPLIER_IDENTITY
  'cn=replAdmin,dc=replTree,dc=ldap,dc=mch,dc=xyz,dc=de'
  BASEDN 'ou=Sales, o=ABC, c=US' FILTER '(objectlass=person)'
  ATTRIBUTE_SELECTION '(objectclass=*) INCLUDE cn $ sn $
  telephoneNumber $ mail)' SCOPE 2 UPDATE_SCHEDULE * 1-2 * * *
  REPLICATION_PROTOCOL 1.2.3 )
```

XI. Security Considerations

This document defines a mechanism that can be used to describe the replication agreements between suppliers and consumers. The replication process would rely on this information to schedule and control the replica updates. Hence the replication agreement attribute should be protected from unauthorized access. If the identity of consumers and/or the nature of their subscription to directory information is not public informa-

tion, the relevant directory information should be protected from unauthorized access as well.

Servers will generally need to persistently store authentication credentials used to bind to consumer or supplier servers when performing replica updates. These credentials MUST be adequately protected from unauthorized access.

XII. References

[1] Good, G., "The LDAP Data Interchange Format (LDIF) - Technical Specification", INTERNET-DRAFT draft-ietf-asid-ldif-01.txt, Netscape Communications Corp.,
<URL:ftp://ietf.org/internet-drafts/draft-ietf-asid-ldif-01.txt>

[2] Good, G., "Definition of an Object Class to Hold LDAP Change Records", INTERNET-DRAFT draft-ietf-asid-changelog-01.txt, Netscape Communications Corp.,
<URL:ftp://ietf.org/internet-drafts/draft-ietf-asid-changelog-01.txt>

[3] C. Weider, J. Strassner, "LDAP Multi-master Replication Protocol", INTERNET-DRAFT draft-ietf-asid-ldap-mult-mast-rep-00.txt, Microsoft Corporation, Cisco Systems Corporation,
<URL:ftp://ietf.org/internet-drafts/draft-ietf-asid-ldap-mult-mast-rep-00.txt>

[4] T. Howes, M. Wahl, "Referrals and Knowledge References in LDAP Directories", INTERNET-DRAFT draft-ietf-asid-ldapv3-referral-00.txt, Netscape Communications Corp., Critical-Angle, Inc., <URL:ftp://ietf.org/internet-drafts/draft-ietf-asid-ldapv3-referral-00.txt>

[5] T. Howes, "The String Representation of LDAP Search Filters", INTERNET-DRAFT draft-ietf-asid-ldapv3-filter-02.txt, Netscape Communications Corp.,
<URL:ftp://ietf.org/internet-drafts/draft-ietf-asid-ldapv3-filter-02.txt>

[6] T. Howes, M. Smith, "The LDAP URL Format", INTERNET-DRAFT draft-ietf-asid-ldapv3-url-03.txt, Netscape Communications Corp.,
<URL:ftp://ietf.org/internet-drafts/draft-ietf-asid-ldapv3-url-03.txt>

XIII. Authors' Addresses

Sanjay Jain
Oracle Corporation
200 Oracle Parkway
Box 659210
Redwood Shores, CA 94065
USA
Phone: +1 415-506-9325
E-mail: skjain@us.oracle.com

Uppili Srinivasan
Oracle Corporation
200 Oracle Parkway
Box 659210
Redwood Shores, CA 94065
USA
Phone: +1 415-506-3039
E-mail: usriniva@us.oracle.com

Gordon Good
Netscape Communications Corporation
501 E. Middlefield Rd.
Mail Stop MV068
Mountain View, CA 94043
USA
Phone: +1 415 937 3825
E-mail: ggood@netscape.com