INTERNET-DRAFT
Status: Informational
November 12, 1999
Expires May 12, 2000

Michael P. Armijo
Microsoft Corporation

Microsoft LDAP Control for Directory Synchronization
draft-armijo-ldap-dirsync-01.txt

1. Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

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

This document defines an LDAP Control for Directory Synchronization. This control allows a client to request changes made to a directory replica since a state of that replica identified by an opaque "cookie." This control is implemented by the Active Directory feature of Microsoft Windows 2000 Server. It is intended that other members of the Internet community be able to use this control if desired.

3. Overview

Many organizations today store information on multiple directories. For example, e-mail accounts and related information might be stored in one directory; information about files and networking in another;

certain data, such as financial or human resource data in yet another directory. Such an environment is referred to as a mixed-directory environment.

The LDAP Control for Directory Synchronization provides a method for dissimilar directories to share pertinent information.

4. Directory Synchronization Control

This control MUST only be used with a SearchRequest message. A server MUST ignore the control if used with any other message unless the criticality field is set to True, in which case the entire operation MUST fail and MUST instead return the resultCode unsupportedCriticalExtension as per section 4.1.12 of [RFC 2251]. The server MUST list that it recognizes this control in the supportedControl attribute in the root DSE.

The replication control is included in the searchRequest and searchResultDone messages as part of the server controls field of the LDAPMessage. The structure of this control is as follows:

```
Repl    Control ::= SEQUENCE {
                controlType             1.2.840.113556.1.4.841
                controlValue            replControlValue
                criticality             TRUE
}
```

The replControlValue in the SearchRequest is an OCTET STRING wrapping the BER-encoded version of the following:

```
realReplControlValue ::= SEQUENCE {
                parentsFirst            integer
                maxReturnlength         integer
                cookie                  OCTET STRING
}
```

parentsFirst: Setting parentsFirst to one ensures that all parents of the children come before their children.

maxReturnlength: This specifies the maximum length in bytes to be returned in the control response. This can be used to limit the amount of data returned. This field must be set to a number above zero for date to returned.

cookie: The cookie is an implementation specific opaque OCTET STRING that is updated by the directory during each search request. It allows the Dirsync control to read changes incrementally from the directory. The very first time the control is created, the cookie should be encoded as a NULL string with 0 length.

The replControlValue in the SearchResponse is an OCTET STRING wrapping the BER-encoded version of the following:

```
realReplControlValue ::= SEQUENCE {
                Flag                    integer
                maxReturnlength         integer
                cookie                  OCTET STRING
}
```

flag: If flag is set to a non-zero value, it implies that there is more data to retrieve.

maxReturnlength: This specifies the maximum length in bytes to be returned in the control response.

cookie: This is the opaque cookie returned by the server to be used by the client in subsequent searches.

5. Provider/Consumer Interaction

Server implementations (Providers) MUST return a globally unique identification (GUID) for each object returned with the Directory Synchronization control. This unique identifier MUST be returned in the value of the DN with the DNWithString Syntax (defined in section 5.2). If the DN is static then the DN can be used for unique identification of the object. Consumer applications MUST take the first value in the DN value set (encoded with the DNWithString syntax) to be the GUID to the object. A GUID MUST be matched to the existing objects on the consumer store. The values returned by the Provider server MUST be applied to the object, with the exception of reserved attributes defined in section 5.1. A GUID with no corresponding object on the consumer store MUST be treated as a new object.

The LDAP Directory Synchronization control allows a client to request changes made to a directory replica since a state of that replica identified by an opaque "cookie."

A typical consumer (dirsync agent) will work on a schedule to read changes from a supplier directory and write changes to a consumer directory. On this schedule the client will wake up, read the opaque cookie from a file, then enter a loop passing the current cookie to the supplier server and receive changes back. It computes updates to perform to the consumer directory based on the changes, and makes these updates. When these updates are committed it writes the new cookie to the file, and goes around the loop again if the setting of the 'flag' returned by the supplier states that there is additional information to be retreieved. If not it exits the loop and sleeps until the next scheduled cycle.

When the control is initally run the client should send the

cookie encoded as a NULL string with 0 length.

The server will respond to each Directory Synchronization search request with the changes since the last control was run (based on the cookie provided by the client) and a cookie to be stored and used by the client during the next synchronization cycle. The client MUST consider the cookie to be an opaque structure and not make any assumptions about its internal organization or value. The client may reuse older cookies, however this search request may result in changes being reported that have already been received by the client.

In the case of a complete server failure, a client may pass a cookie generated by one directory server to a different directory server hosting the same directory partition. This may result in the new server reporting changes already reported by the old server. The new server MAY report a full synchronization (all objects and attributes in the search request). The client MUST be able to handle changes already reported being returned again.

The directory server SHOULD limit use of this control to entities explicitly granted permission to use this control. The directory server SHOULD return objects and attributes based on the filters of the search request and based on the permissions of the authenticated entity.

Server implementations may have other restraints on which containers or objects may or may not use the Directory Synchronization control. If a client attempts to run the Directory Synchronization control on an object or container that does not support the control, the server SHOULD return the error unwillingToPerform(53).

5.1 Interpretation of Advanced Directory Operations

Certain directory changes and operations are not defined in an LDAP search response. The Directory Synchronization control will interpret these operations using defined object attributes. The directory synchronization consumer MUST understand and support these operations.

A Provider MUST return an attribute with a NULL value to signify that attribute has been removed. A DirSync consumer MUST interpret this as an attribute removal and process this accordingly.

If an object is deleted it will be returned in the search response message with the 'isDeleted' attribute set to value True. The client MUST interpret this as an object deletion and MUST perform the proper operation on the consumer directory.

```
isDeleted
(1.2.840.113556.1.2.48 NAME 'isDeleted'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7)
```

If an object is moved or renamed the attribute 'RDN' will be returned with the value set to the new object name. The client MUST interpret this as an object rename and perform the proper operation on the consumer directory.

```
RDN
(1.2.840.113556.1.4.1 NAME 'RDN'
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
```

5.2 DN With String Syntax

(1.2.840.113556.1.4.904 DESC 'DNWithString')
Values with this string are encoded as follows:

```
DNWithString  = StringTag ':' Count ':' String ':' DN

OctetTag = 'S' | 's'

Count = positive decimal number, counting number of bytes in
String

String = <normally encoded (i.e. UTF8 for V3) string>
// Note: the number
of bytes in the string encoding of the String is Count.

DN = <normal string encoding of a DN>

As an example, the string encoding of the combination of
"GUID=89876" and DC=foobar,DC=Com is

S:10:GUID=89876:DC=foobar,DC=Com

As an example, the string encoding of the combination of XYZ (where
X, Y, and Z all have two byte UTF-8 encodings) and
DC=foobar,DC=Com is

S:6:XYZ:DC=foobar,DC=Com

Note: Characters with multibyte UTF-8 encodings contribute more
than one to the count
```

6. Security Considerations

This document details a method for retreiving information from a directory server using an LDAP control. Server implementations utilizing this control SHOULD implement security mechanisms as defined in Authentication Methods for LDAP [AuthMeth].

Each implementation should take appropriate measures to insure that only

authorized entities can utilize this control.

7. References

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

[AuthMeth]
M. Wahl, H. Alvestrand, J. Hodges, R. Morgan. "Authentication Methods for LDAP". INTERNET-DRAFT, Work In Progress. draft-ietf-ldapext-authmeth-03.txt

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

8. Authors Address

Michael P. Armijo
One Microsoft Way
Redmond, WA
98052
USA

(425)882-8080
micharm@microsoft.com

Expires May 12, 2000