Appendix E

LDAP Services for NDS

Overview

This appendix discusses LDAP Services for NDS, which publishes information stored in Novell Directory Services (NDS) to Lightweight Directory Access Protocol (LDAP) clients. LDAP is a developing Internet protocol for accessing directory information such as user and host data. LDAP client software will probably become a standard feature of Internet browsers and the standard protocol for accessing directory information over the Internet.

LDAP Services for NDS enables new uses of NDS in network environments. In the past, NDS has been used primarily to control access to network services. Although NDS excels at this, it also can provide other kinds of directory information to all the clients in a network environment. For example, you can use LDAP services for NDS to publish user's business phone numbers.

LDAP Services for NDS supports most NDS security features and adds an LDAP access control layer that provides additional security features. These security features allow you to make some types of directory information available to the public, make other types available to your organization, and make certain types available only to those groups or individuals that have a need to know.

This appendix discusses the following LDAP Services for NDS concepts:

- Security
- NDS schema changes
- LDAP Functions and Parameters

Concepts to Know

To understand this appendix you should be familiar with the following NDS concepts:

- Directory tree structure
- Objects, classes, and attributes
- Partitions
- Replica management
- General NDS synchronization
- NDS Authentication

Security

LDAP Services for NDS allows LDAP clients to access data in Novell Directory Services (NDS). LDAP clients access a directory by building a request and sending it to the directory. When an LDAP client sends a request through LDAP Services for NDS to NDS, NDS completes the

request for only those attributes to which the LDAP client has the appropriate access rights.

The default rights that most users receive provide limited rights to the user's own object. To provide access to other objects and their attributes, you must change the rights assigned in NDS.

If the NDS rights assignment options provide too high a level of access, you can implement additional controls using the optional LDAP Access Control List (ACL) feature. LDAP ACLs limit what attributes LDAP clients can access based on the client address. Currently NDS does not permit inherited rights on attributes, so you may use LDAP ACLs to limit attribute access. For example, you may wish to publish E-Mail addresses for your directory to those outside your company, but publish both E-Mail addresses and telephone numbers to internal users.

LDAP ACLs do not grant access at a higher level than NDS access. This allows ACls to be based on the network address and specific attributes.

LDAP Bind (Authentication)

All LDAP clients are authenticated to NDS as one of the following types of users:

- [Public] (Anonymous bind)
- Proxy user (Proxy user anonymous bind)
- An NDS user (simple bind)

Note LDAP refers to authentication as a *bind.*

Anonymous Bind

An anonymous bind is a bind from an LDAP user that does not contain a user name or password. If an LDAP client binds, or authenticates, to LDAP Services for NDS and the service is not configured to use a proxy user, the user is authenticated to NDS as user [Public] . The default Browse right for user [Public] allows users to browse NDS objects and blocks user access to most object attributes.

To enable [Public] user access to object attributes, you must make user [Public] a trustee of the appropriate container or containers and assign the appropriate object attribute rights.

Note The rights assigned to user [Public] are available to anyone who accesses NDS without a username and password. This access can be through LDAP Services for NDS or through any NetWare utility such as NetWare Administrator.

Proxy User Anonymous Bind

A proxy user anonymous bind is an anonymous bind that is linked to a proxy NDS username. If an LDAP client uses a proxy username to bind to LDAP Services for NDS and the service is configured to use a proxy user, the user is authenticated to NDS as the proxy user, whose name is configured in LDAP Services for NDS and in NDS.

To implement proxy user anonymous binds, the adminstrator must create the proxy user in NDS and assign the appropriate rights to that user. You also need to enable the proxy user in LDAP Services for NDS by specifying a proxy user name.

The key concepts to remember about proxy user anonymous binds are as follows:

- All LDAP client access through anonymous binds is assigned through the proxy user object.
- The proxy user cannot have a password or any password restrictions (such as password change intervals) because LDAP clients do not supply passwords during anonymous binds. You should not allow the proxy user to change passwords.
- The proxy user must be created in NDS and assigned rights to the NDS objects you want to publish. The default user rights provide read access to a limited set of objects and attributes.
- The proxy user must be enabled on the General page of the LDAP Group object that configures LDAP Services for NDS.

Note You can configure Access Control Lists in the LDAP Services for NDS LDAP Group object to add access controls for the proxy user. For example, you can create an Access Control List that allows the proxy user access through one IP address or a group of IP addresses.

NDS User Bind (Simple Bind)

An NDS user bind, or simple bind, is a bind that an LDAP client makes using an NDS username and password. The user must supply the full LDAP Distinguished Name of the NDS user to authenticate. The LDAP user is authenticated in NDS using the supplied password, and the LDAP client is allowed access to whatever information the NDS user is allowed to access.

The NDS user bind feature is disabled when LDAP Services for NDS is installed because passwords entered by LDAP clients are not encrypted on the communications path between the LDAP client and LDAP Services for NDS. It is possible for these passwords to be captured, with the NDS username, by network monitoring equipment. Anyone who captures an NDS username and password has immediate access, through an LDAP

or NDS client, to all the NDS objects to which the captured username has access.

Note Disabling NDS User Binds does not prevent users from trying to bind with their username and password, and an attempted bind exposes the username and password just the same as a successful bind. Disabling NDS User Binds does, however, discourage users from using their name and password because this method is unsuccessful.

One way to enable NDS user binds and protect NDS user passwords is to create a special NDS user for LDAP client access. This allows you to enforce passwords for all LDAP client binds. The risk with this approach is that users may try to authenticate using their login user name and password instead of the LDAP client user name and password. In this situation, the user will successfully authenticate and the password can be captured. The best way to encourage users to use the LDAP client username and password is to limit the directory rights for their login username and password. If users find that they have more access with the LDAP client username and password, they will be more likely to use it.

To authenticate an LDAP user:

Module	Action
Requesting module	1. Sends the server an LDAP Bind request containing the complete LDAP name and password of the object authenticating.
Server	2. Parses the LDAP Bind request. 3. Converts the LDAP DN to an NDS DN. • If there is no Distinguished Name, sets the bind as an anonymous bind or sets up the proxy user name. • If a Distinguished Name is present, sets the bind as a User bind 4. Resolves the Distinguished Name to check that the object exists. 5. Uses NDS authentication to authenticate the user. 6. Returns success or failure.

Assigning NDS Rights for LDAP Clients

When an LDAP client requests access to an NDS object and attribute, the LDAP client's request is accepted or rejected by NDS based on the LDAP client's rights to NDS. The following tables list NDS rights clients must have to make LDAP requests.

Table E.1
NDS Rights Required for LDAP Operations

LDAP Access Requested	NDS Rights Required
Search (object)	Browse [Entry Rights]
Add (object)	Create [Entry Rights]

Table E.1
NDS Rights Required for LDAP Operations

Delete (object)	Delete [Entry Rights]
Compare (attribute)	Compare attribute rights
Search (attribute)	Compare attribute rights
Read (attribute)	Read attribute rights
Write (attribute)	Write attribute rights

Note NDS currently does not allow objects to inherit individual attribute rights from containers. One way to assign individual attribute rights that are inherited is to use the Access Control List feature in LDAP Services for NDS.

Schema Support for LDAP Services

LDAP Services for NDS requires that the NDS schema support the LDAP schema. Support for LDAP includes:

- NDS schema extensions
- Support for LDAP Distinguished Names
- Mapping LDAP objects and classes to NDS objects and classes

NDS Schema Extensions

LDAP Services for NDS requires that the NDS schema be extended to support LDAP. LDAP Services for NDS extends the NCP Server object. In addition, LDAP Services for NDS creates the following NDS objects when it is installed on a server:

- An LDAP Server object called LDAP Server
- An LDAP Group object called LDAP Group

Note These objects are stored in the same container as the server on which the product is installed. If you install LDAP Services for NDS on additional servers, the installation will check for the presence of these objects and will not create additional objects if they are found.

For information about these schema extensions, see *Novell Directory Services Schema Specification.*

LDAP Distinguished Names

The OSI Directory uses Distinguished Names as the primary keys to entries in the directory. The following paragraphs define the LDAP functions that require DNs to identify objects or attribute values and the delimiters used in their specification. The primary focus of this document is the interaction of LDAP Services for NDS and Novell Directory Services. Information contained in this documented was obtained from RFC

1779, “A String Representation of Distinguished Names” and RFC 1823 “The LDAP Application Program Interface”, which can be found on the University of Michigan web site “http://www.umich.edu/~rsug/ldap/”.

The following LDAP Functions use a DN to specify the object:

- Add Entry
- Bind
- Compare
- Delete
- Modify
- ModifyRDN
- Search

The following LDAP Functions uses DNs as attribute values:

- AddEntry
- Compare
- Modify
- Search

LDAP Representation of Distinguished Names (RFC 1779)

Because LDAP is based on X.500, on which NDS is also based, it represents Distinguished Names in much the same way, with a few differences

Separators

Like NDS, LDAP uses separators to delimit objects within a Distinguished Name. The following characters are used as LDAP separators:

- Comma (,)
- Semicolon (;)

Distinguished Names may contain optional spaces and carriage returns before and after the separator.

Object Names

LDAP object names may contain any ANSI character string except the reserved characters specified below. Currently, LDAP works only in the server’s code page. In future versions, LDAP will use UTF-8 to map characters not in the low ASCII range.

Naming Attributes

LDAP naming attributes are similar to NDS attributes. LDAP naming attributes include the following:

- CN. Common Name
- OU. Organizational Unit
- O. Organization
- C. Country

- L. Locality
- ST. State or Province

Escape Character

LDAP uses the backslash (\) as an escape character.

Reserved Characters

Reserved characters are valid within object names only if they are used in conjunction with the escape character.

- Equals (=). Separates the object type and object name
- Plus (+). Delimits multi-attribute names
- Comma (,). Object separator
- Semicolon (;). Object separator
- Pound (#). Delimits a hexi-decimal character
- Backslash (\). Escape character
- Less Than (<). A special delimiter defined in RFC 1779
- Greater Than (>). A special delimiter defined in RFC 1779
- Double Quote (“). Specifies a literal string

Special NDS Characters

NDS uses the following special characters:

- Period (.) Used by DS as a separator.
- Plus(+). Delimits multi-attribute names
- Equal(=). Separates the object type and object name
- Backslash(\). Escape character

LDAP to Directory Services Mappings

The LDAP Schema supports only a subset of the available NDS syntaxes, attributes and object classes. Containment of object classes in LDAP corresponds to the containment of the NDS classes the LDAP classes map to. The NDS schema does not currently include all LDAP schema classes and attributes.

LDAP to NDS Attribute Mappings

The following LDAP attributes map directly to NDS attributes:

Table E.2
LDAP to NDS Attribute Mappings

LDAP Attribute	NDS Attribute
objectClass	Object Class
aliasedObjectName	Aliased Object Name
c	C
cn	CN
description	Description

Table E.2
LDAP to NDS Attribute Mappings

LDAP Attribute	NDS Attribute
facsimileTelephoneNumber	Facsimile Telephone Number
givenName	Given Name
initials	Initials
l	L
mail	Internet Email Address
member	Member
o	O (Organization)
ou	OU (Organizational Unit)
owner	Owner
physicalDeliveryOfficeName	Physical Delivery Office Name
postalAddress	Postal Address
postalCode	Postal Code
postOfficeBox	Postal Office Box
roleOccupant	Role Occupant
seeAlso	See Also
sn	Surname
st	S (State or Province Name)
street	SA (Street Address)
telephoneNumber	Telephone Number
title	Title
uid	UID (User ID)

LDAP to NDS Object Class Mappings

The following LDAP object classes map directly to NDS object classes:

Table E.3
LDAP to NDS Object Class Mappings

LDAP Object Class	NDS Object Class
alias	Alias
country	Country
groupOfNames	Group
locality	Locality
newPilotPerson	User
organization	Organization
organalizationPerson	Organization Person (noneffective class)
organizationalRole	Organizational Role
organizationalUnit	Organizational Unit
person	Person (noneffective class)
residentialPerson	User

LDAP Functions and Parameters

LDAP Services for NDS allows LDAP to make calls into NDS. LDAP Services for NDS supports the following LDAP function calls:

- LDAP Add Entry
- LDAP Delete Entry
- LDAP Compare Entry
- LDAP Modify Entry
- LDAP ModifyRDN
- LDAP Search
- LDAP Open
- LDAP Bind
- LDAP Unbind

Each of these functions is described in the following sections.

Structures

LDAP Services for NDS uses several structures, some of which are modified from the original LDAP implementation. These structures are defined in SLAP.H.

OP Structure

This structure is sent with all LDAP Services for NDS operations and contains information specific to the operation. It contains the following elements:

Table E.4
OP Structure Fields

Field	Type	Description
o_ber	BerElement	The request
o_msgid	long	The message ID of the request
o_tag	unsigned long	Type of request
o_time	time_t	The time the operation was initiated
*o_dn	char	The Distinguished Name that was bound when the operation was initiated
o_authtype	int	Authentication method used to bind the Distinguished Name
o_opid	int	The ID for this operation
o_connid	int	The ID of the connection initiating this operaiton
o_cldap	int	!= 0 if this came in via CLDAP
o_clientaddr	struct sockaddr	Client address if via CLDAP
o_searchbase	char	Search base if via CLDAP
o_next	struct op	The next operation pending
o_tid	pthread_t	The thread handling this operation

Table E.4
OP Structure Fields

Field	Type	Description
o_abandon	int	Indicates that the operation has been abandoned
o_abandonmutex	pthread_mutex	Indicates that the operation has been abandoned
o_private	int	Reserved for future use.

Conn Structure

This structure contains information about the current LDAP connection. It has the following fields:

Field	Type	Description
c_sb	Sockbuf	Connection information
*c_dn	char	The current Distinguished Name bound to this connection
c_dnmutex	pthread_mutex_t	Mutex for c_dn field
c_authtype	int	The authentication method used to bind c_dn
c_version	int	Used for compatibility with versions 2.0 and 3.0
*c_addr	char	The address of the client on this connection
*c_domain	char	The domain of the client on this connection.
*c_ops	Operation	List of pending operations
c_opsmutex	pthread_mutex	Mutex for c_ops lists and stats
c_pdumutex	pthread_mutex	Used to write to the network for a connection.
c_wcv	pthread_cond_t	Used to wait for sd write-ready
c_gettingber	int	Indicates that the operation is in the process of the ber_get_next operation.
c_currentber	BerElement	The current ber being retrieved in the ber_get_next operation
c_writewaiter	int	Signals write-ready sd waiter
c_pduwaiters	int	Signals that the threads are waiting for pdu
c_starttime	time_t	The time the connection was opened.
c_connid	int	ID of this connection for stats
c_opsinitiated	int	Number of operations initiated and the nex operation ID
c_opscompleted	int	The number of operations completed
c_context	NWDSContextHandle	NDS context
c_lastactivitytime	time_t	The time the connection was last used
c_closemutex	pthread_mutex	Mutext for closing connections

Field	Type	Description
c_conn_logged_out	int	Boolean to indicate connection logout status
c_conn_should_close	int	Boolean to indicate that this connection should be closed when the worker thread is ready to terminate
c_userhandle	int	NDS user handle for this connection and NDS context
c_fromaddr	struct sockaddr	IP address of the caller
c_useaddr	struct sockaddr	IP address to write to.
c_ber_list	BerElement	Linked list of search response entries
c_monitor_thread	void	Void pointing to the monitor_thread_info_t, which monitors this connection
conn_entry_index	int	Index into conn_entry array for the monitor thread
c_conn_closing	int	Boolean to indicate that the connection is closing

Backend Structure

It has the following format:

Field	Type	Description
**be_suffix	char	The DN suffixes of data in this backend
*be_rootdn	char	Not used.
*be_rootpw	char	Not used.
be_readonly	int	Not used.
be_sizelimit	int	Size limit for this backend
be_timelimit	int	Time limit for this backend
*be_acl	struct acl	Not used.
be_dfltaccess	int	Not used.
**be_replica	char	Not used.
*be_replogfile	char	Not used.
*be_updatedn	char	Not used.
be_lastmod	int	Not used.
*be_type	char	Type of database
*be_private	void	Not used.
be_bind	IFP	Backend bind routine
be_unbind	IFP	Backend unbind routine
be_search	IFP	Backend search routine
be_compare	IFP	Backend compare routine
be_modify	IFP	Backen modify routine
be_modrdn	IFP	Backend modrdn routine
be_add	IFP	Backend add routine
be_delete	IFP	Backend delete routine
be_abandon	IFP	Backend abandon routine

Field	Type	Description
be_config	IFP	Backend config routine
be_init	IFP	Backend init routine
be_close	IFP	Backend close routine

Abandon

This operation is called when a flag in the op structure indicates that an operation is to be abandoned, in which case the operation terminates and the operation state is changed. Currently only the *Search* operation supports the abandon flag.

Add Entry

The Add Entry function adds an entry to the Directory. To add an entry from an LDAP client:

Module	Action
Requesting module	1. Sends the server an LDAP Add Entry request containing the Distinguished Name of the new entry's parent, and the new entry's attributes and attribute values.
Server	2. Checks that the parent object exists. 3. Checks that the requesting client has LDAP *Add* [Entry Rights]. 4. Maps the request to the NDS *Add Entry* verb, with the LDAP object class, names, and attributes being mapped to NDS object class, names and attributes. 5. Returns a reply containing a completion code indicating success or failure.

Close

This function closes an LDAP connection and cleans up allocated memory and the LDAP configuration.

Compare Entry

This function compares attribute values within the Directory. To compare attribute values:

Module	Action
Requesting module	1. Sends the server an LDAP *Compare* Entry request containing the information to be compared (such as a Distinguished Name or attributes and values).

Module	Action
Server	2. Checks that the object exists. 3. Checks that the requesting client has *Compare* attribute rights on the attribute being compared. 4. Maps the request to the *Compare* verb, with the LDAP attribute values being mapped to NDS attribute values. 5. Returns a reply containing a completion code indicating success or failure.

Config

The Config operation is a front-end LDAP operation that passes the object class and attribute mappings to the back-end operations listed here. LDAP Services for NDS uses the slapd.conf file defined in the University of Michigan's LDAP specification. NLDAP reads the config file when it starts up. If there are any configuration changes, NLDAP must be restarted. In addition to the keywords defined by LDAP, LDAP Services for NDS uses the following keywords:

Keyword	Description
MAX_CONCURRENT_CONNECTIONS	The maximum of concurrent connections 0. Unlimited (default)
MAX_NO_ACTIVITY_TIME	The maximum time of inactivity before a connection is ended. 0. Unlimited (default is 15 minutes)
TCP_PORT	TCP Port
UDP_PORT	Not used.
MAX_LOG_FILE_SIZE	The maximum size for the log file
LOG_FILE_NAME	The name of the log file
BACKUP_LOG_FILE_NAME	The name of the backup file
PROXY	The proxy user name
ALLOW_CLEAR_TEXT_PASS	Indicates whether or not to allow the password to be sent as clear text
NDS_OC	NDS object class
NDS_ATTRIBUTE	NDS attribute

Delete Entry

This function removes entries from the Directory. It maps to the NDS *Remove Entry* function. To delete an entry from the Directory:

Module	Action
Requesting module	1. Sends the server an LDAP Delete Entry request containing the Distinguished Name of the entry to be deleted.

Module	Action
Server	2. Checks that the parent object exists. 3. Checks that the requesting client has LDAP *Delete* [Entry Rights]. 4. Maps the request to the NDS *Remove Entry* verb, with the LDAP Distinguished Name being mapped to the NDS Distinguished Name. 5. Returns a reply containing a completion code indicating success or failure.

Modify Entry

This function modifies objects that already exist in the Directory. To modify an existing entry:

Module	Action
Requesting module	1. Sends the server an LDAP Modify Entry request containing the Distinguished Name of the entry to be deleted.
Server	2. Checks that theobject exists. 3. Checks that the requesting client has LDAP *Write* attribute rights on the attribute to be modified. 4. Maps the request to the NDS *Modify Entry* verb, with the LDAP Distinguished Name and attribute value being mapped to the NDS Distinguished Name and attribute value. 5. Returns a reply containing a completion code indicating success or failure.

Modify RDN

This function modifies the Relative Distinguished Name (RDN) of an entry in the Directory. To modify an entry's Relative Distinguished Name:

Module	Action
Requesting module	1. Sends the server an LDAP Modify RDN request containing the Distinguished Name of the entry to be deleted.
Server	2. Checks that the object exists. 3. Maps the request to the NDS *Modify RDN* verb, with the LDAP Distinguished Name being mapped to the NDS Distinguished Name. 4. Returns a reply containing a completion code indicating success or failure.

Search

The LDAP Search function searches the NDS Directory, returning a requested set of attributes and values for each entry matched. To search the NDS Directory:

Module	Action
Requesting module	1. Checks that the client has sufficient rights to do the operation. 2. Sends the server an LDAP Search request containing: • The base object from which to start the search • The scope of the search • A dereference flag that indicates whether or not to dereference aliases • A size limit, indicating the maximum size of the returned information • A time limit indicating the maximum time before the operation stops • A search filter
Server	3. Maps the request to the NDS *Search* verb, with the search filters and flags mapping to NDS search filters and flags. 4. Returns a reply containing a completion code indicating success or failure.

Bind

The Bind function authenticates LDAP users to the Directory. Currently, only the LDAP_AUTH_SIMPLE method of authenticating is supported. To bind to NDS:

Module	Action
Requesting module	1. Sends the server an *LDAP Bind* request containing the Distinguished Name of the entry authenticating along with a password.
Server	2. Checks that the object exists. If no DN is included in the request, authenticates as an anonymous bind or sets up a proxy user. 3. Authenticates the LDAP user to NDS. 4. Returns a reply containing a completion code indicating success or failure.

Unbind

This function unbinds a user from the Directory and closes the connection.

Error Codes

A list of all LDAP error codes is found in the LDAP header file LDAP.H.