LDAP and eDirectory™ Integration

Developer Kit

April 2, 2012

Novell

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. For more information on exporting Novell software, see the Novell International Trade Services Web page (http://www.novell.com/info/exports/). Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at Novell Legal Patents (http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell developer products, and to get updates, see Novell Developer Kit (http://developer.novell.com/wiki/index.php/Category:Novell_Developer_Kit). To access online documentation for Novell products, see Novell Documentation (http://www.novell.com/documentation/).

Novell Trademarks

For a list of Novell trademarks, see Trademarks (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners

Contents

LDAP and eDirectory Integration

This book describes the following Novell® eDirectory™ and LDAP integration features:

- Terminology differences between eDirectory and LDAP
- LDAP controls and extensions for accessing eDirectory features
- eDirectory support for the LDAP schema
- LDAP support for eDirectory classes and attributes
- Authenticating to eDirectory through LDAP
- LDAP return codes common to all LDAP implementations
- eDirectory Event System

This guide contains the following sections:

Audience

This guide is intended for the developers to understand the Novell's LDAP server and eDirectory.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comment feature at the bottom of each page of the online documentation.

Additional Information

For the related developer support postings for LDAP and eDirectory Integration (eDirectory Libraries for C), see the Developer Support Forums (http://developer.novell.com/ndk/devforums.htm).

Documentation Updates

For the most recent version of this guide, see the LDAP and eDirectory Integration NDK page (http://developer.novell.com/ndk/doc/ldapover/ldap_enu/data/a3wyu4m.html).

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

1 LDAP and eDirectory Integration

A Novell® eDirectory™ tree or directory can be accessed either through NDAP/NCP or LDAP. This document exposes the specific details you need to know to use LDAP to access eDirectory. It describes details about Novell's LDAP server and eDirectory. It assumes that you are familiar with the following:

- **LDAP v2 and v3:** If you are new to LDAP and need to develop a basic understanding of LDAP and the standard features of LDAP v3, see Section 1.13, "Other Sources of LDAP Information," on page 46.
- **eDirectory:** If you are new to eDirectory and need to develop an understanding of basic eDirectory functionality, see *NDK: Novell eDirectory Technical Overview*).

This document provides an overview of the following topics:

1.1 Supported Platforms

Novell currently supplies the following LDAP development tools:

- LDAP Libraries for C (http://developer.novell.com/ndk/cldap.htm)
- LDAP Classes for Java (http://developer.novell.com/ndk/jldap.htm)
- LDAP Extensions and Controls for JNDI (http://developer.novell.com/ndk/extjndi.htm)
- Novell LDAP JDBC Driver (http://developer.novell.com/ndk/ldapjdbc.htm)
- Novell Controls for ActiveX (http://developer.novell.com/ndk/ocx.htm)

The Novell LDAP server is currently available on NetWare 5.x, NDS for NT, NDS for Solaris, and NDS eDirectory (for NetWare, NT, Solaris, and Linux).

1.1.1 eCommerce Beans

The eCommerce beans provide easy-to-use Java components to integrate Web applications using LDAP.

1.1.2 LDAP Libraries for C

The LDAP Libraries for C kit enables application developers to write applications to access, manage, update, and search for information stored in eDirectory and other LDAP-aware directories. It currently supports development on the following NDS server platforms:

- NetWare 5.1™
- NetWare 5™ with SP4 or higher
- Windows NT* server 4.0 with SP 4 or higher
- Solaris 2.6 and 2.7*
- Linux (tested on Red Hat* 6.1, 6.2, and 7.0)

It supports development on the following client platforms:

- Windows NT* workstation 4.0 with SP 3 and SP 4
- Windows 95*
- Windows 98*
- Windows 2000*
- Solaris 2.6 and 2.7*
- Linux (tested on Red Hat* 6.1, 6.2, and 7.0)

1.1.3 LDAP Classes for Java

The LDAP Classes for Java kit enables application developers to write applications to access, manage, update, and search for information stored in eDirectory and other LDAP-aware directories. It provides LDAP extensions for accessing and managing eDirectory-specific information such as effective rights, replicas, and partitions.

1.1.4 LDAP Extensions and Controls for JNDI

The LDAP Extensions and Controls for JNDI are designed to be used in applications that run on any operating system that has a JVM (Java virtual machine). The provider comes from Sun. Although other platforms are possible, this provider has been tested against the Novell LDAP server and eDirectory directory from the following platforms:

- NetWare™
- Windows* (NT, 95, 98, 2000, and XP)

1.1.5 LDAP JDBC Driver

The LDAP JDBC Driver is a Java database connectivity driver that enables Java programs to execute SQL (Structured Query Language) statements to access eDirectory or Novell Directory Services® (NDS).

1.1.6 Novell Controls for ActiveX

You can use the Internet Directory Query Control to query and retrieve information about directory entries from LDAP-based directories.

1.2 LDAP and eDirectory Terminology

The following sections describe some of the terminology and conceptual differences between LDAP and eDirectory:

1.2.1 Objects and Entries, Attributes and Properties

In LDAP documentation, an entry consistently means a record in the directory database. In eDirectory documentation, such a record is fairly consistently called an object. Since object becomes ambiguous when describing object-oriented programming languages, eDirectory developer documentation is beginning to use the LDAP term, entries. However, some functions, attribute names, and class names use object as part of the name. Novell product documentation for eDirectory utilities and applications continue to use the eDirectory term, objects.

In most directory documentation, attributes refer to the fields of a record. LDAP and X.500 conform to this standard. However, a lot of LDAP documentation also uses the word attribute to refer to the attribute's value. eDirectory programming documentation uses attribute. However, Novell product documentation for eDirectory utilities and applications use the term properties to describe fields in a record. The following table highlights some of these differences:

Table 1-1 *Different Terminologies*

LDAP Term	eDirectory Developer Term	eDirectory Product Term	Meaning
Entry	Both Object and Entry	Object	A record in a database
Attribute	Attribute	Property	A field in a database. The value of the field.

1.2.2 Naming Conventions

Distinguished Names: eDirectory and LDAP support the same character set for entry names and their distinguished names. However, eDirectory supports a number of styles for rendering distinguished names. For example, a DN for an eDirectory function can:

- Be typeful or typeless

  CN=Kim.OU=Eng.O=Novell *or* Kim.Eng.Novell

- Contain dots or slashes as the delimiter.
- Use Unicode or the code-page character set of the computer

LDAP supports one convention: typeful names with a comma delimiter and code-page characters.

"CN=Kim, OU=Eng, O=Novell"

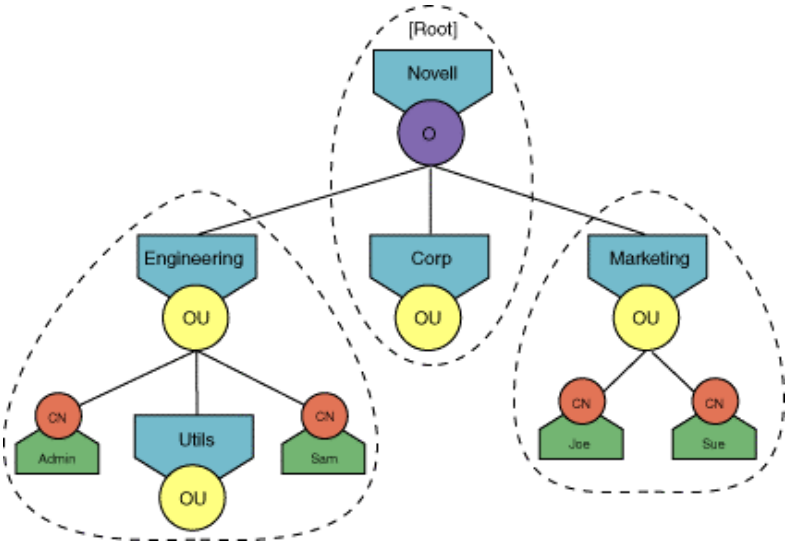
When using LDAP to access eDirectory, distinguished names must always be typeful and in LDAP format.

Schema Names: eDirectory and LDAP support different character sets for object class definitions and attribute definitions. For more information, see "Schema Naming Rules" on page 26.

1.2.3 LDAP Naming Contexts and eDirectory Partitions

NOTE: The naming context terminology is now obsolete.

In LDAP, a naming context is the same thing as an eDirectory partition. Just as an eDirectory partition is a branch of the eDirectory tree with only one parent, an LDAP naming context specifies a branch of a hierarchical tree. The following figure illustrates a simple hierarchical tree with three naming contexts ("O=Novell"; "OU=Engineering, O=Novell"; and "OU=Marketing, O=Novell").



Just as with eDirectory partitions, the naming context is named by the distinguished name of the root container.

1.2.4 Access Control and Effective Rights

Access control determines who has rights to entry information in a directory. In an eDirectory tree, every entry has an ACL (access control list) attribute which contains the explicit trustee assignments that have been made to the entry and its attributes. In addition, eDirectory allows rights to be inherited, so that an assignment in a parent container can allow additional trustees to have access to an entry. Functions that calculate effective rights gather information from these parent containers as well as from the ACL attribute. LDAP queries for eDirectory effective rights return the explicit assignments as well as the inherited rights. Directories that do not allow the inheritance of rights implement the functions to return only explicit trustee assignments.

1.2.5 Containment in Flat and Hierarchical Directories

Flat directories allow an entry to be added without specifying a location. Hierarchical directories require a location to be specified, and the purpose of the distinguished name is to fully describe the location of the entry. However, not all hierarchical directories have rules about where entries can be located. Many directories, such as eDirectory, require a location and enforce rules about where classes of entries can be located in the directory. For example, user or inetOrgPerson entries cannot be created in a Country container. Directories without such rules allow entries to be created any where in the tree. Sometimes these directories give the appearance of hierarchy, but in reality place all entries in flat database with a link to the entry's assigned parent.

1.3 Novell LDAP Servers

Novell has provided an LDAP server for accessing the eDirectory directory since the release of NetWare 4.1. The LDAP server in NetWare 5 added support for LDAP v3. The following table compares the features of the newest release of the LDAP server with previous releases of the LDAP server.

Table 1-2 *Comparison Between Different eDirectory Releases*

LDAP Feature	Novell eDirectory	Novell eDirectory	NDS eDirectory	NDS eDirectory	NDS 8	NDS
	Version 8.7	Version 8.6	Version 8.5	Version 8.3x	Version 8.1x	Version 7.xx
Authentication (anonymous, clear text, and SSL)	Yes	Yes	Yes	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	No	No	No
SASL Authentication	Yes	Yes, simple	No	No	No	No
MD5 Bind	Yes	No	No	No	No	No
NMAS Bind	Yes	No	No	No	No	No
Configure port for clear text and SSL binds	Yes	Yes	Yes	No	No	No
Enforcement of connection management policies (concurrent connections and time restrictions)	Yes	Yes	Yes	No	No	No
Enforcement of password restrictions (length, grace logins, expiration, and uniqueness)	Yes	Yes	Yes	No	No	No

LDAP Feature	Novell eDirectory	Novell eDirectory	NDS eDirectory	NDS eDirectory	NDS 8	NDS
	Version 8.7	Version 8.6	Version 8.5	Version 8.3x	Version 8.1x	Version 7.xx
Entry management (search, modify, compare, rename, adding, deleting)	Yes	Yes	Yes	Yes	Yes	Yes
Partition and replica management	Yes	Yes	Yes	No	No	No
Thread support	Yes	Yes	Yes	Yes	Yes	Yes
Set indexes for faster searching	Yes	Yes	Yes	No	No	No
LDAP controls (query root DSE for supported controls)	Yes	Yes	Yes	Yes	Yes	No
LDAP extensions (query root DSE for supported extensions)	Yes	Yes	Yes	Yes	No	No
Readable root DSE	Yes	Yes	Yes	Yes	Yes	Yes
Referrals and traversals	Yes	Yes	Yes	Yes	Yes	Restrictions
Read and write schema	Yes	Yes	Yes	Yes	Read—Yes	No
Modify existing schema definitions	Yes	Yes	Yes	No	No	No
Auxiliary classes	Yes	Yes	Yes	Yes	Yes	No
Valid LDAP names require no mapping	Yes	Yes	Yes	Yes	No	No
Generated LDAP name for all NDS names which are not mapped or which are not valid	Yes	Yes	Yes	No	No	No
Access to eDirectory compound syntaxes	Yes	Yes	Yes	Selected	No	No
Operational attributes	Yes	Yes	Yes	No	No	No
Dynamic Groups	Yes	Yes	No	No	No	No
Persistent Search	Yes	Yes	No	No	No	No
Refresh LDAP server from LDAP	Yes	Yes	Yes	No (only from NDAP))	No (only from NDAP)	No (only from NDAP)

LDAP Feature	Novell eDirectory	Novell eDirectory	NDS eDirectory	NDS eDirectory	NDS 8	NDS
	Version 8.7	Version 8.6	Version 8.5	Version 8.3x	Version 8.1x	Version 7.xx
Superior Referrals	Yes	No	No	No	No	No
Referrals for non-search operations	Yes	No	No	No	No	No
TLS (SSL) encryption	Yes	Yes	Yes	Yes	Yes	No
Start/Stop TLS	Yes	No	No	No	No	No
Extensible Match	Yes	No	No	No	No	No
Events	Yes	No	No	No	No	No

For an application to use the LDAP features in NDS 8 or eDirectory, the application must attach to an LDAP server running that version of NDS/eDirectory.

1.3.1 Deployed Versions of NDS and eDirectory

Table 1-3 *Deployed Versions of NDS and eDirectory*

Product Version	Build Version	Platforms
NetWare 5.1 SP4 (NDS 7)	DS.nlm v7.57	NetWare 5.1
NetWare 5.1 SP 4 (NDS 8)	DS.nlm v8.79	NetWare 5.1
eDirectory 8	DS.nlm & DS.dlm v8.79	NetWare 5.0, Win NT/2K
eDirectory 8.5.x	DS v85.23	NetWare 5.x, Win, Solaris
NetWare 6 (eDirectory 8.6)	DS.nlm v10110.20	NetWare 6
eDirectory 8.6.1	DS v10210.43	NW 5.1, NW 6, Win, Solaris, Linux
NetWare 6 SP1 (eDirectory 8.6.2)	DS.nlm v10310.17	NetWare 6
eDirectory 8.6.2	DS v103xx.xx	NW 5.1, NW 6, Win, Solaris, Linux
eDirectory 8.7	DS v10410.xx	NW 5.1, NW6, Win, Solaris, Linux, AIX

1.4 LDAP Support for eDirectory Features

LDAP supports access to the following eDirectory features:

- Authentication: LDAP authentication includes anonymous binds, clear text binds, SSL and SASL binds. For eDirectory, these LDAP authentication methods mean the following:
  - An anonymous bind is an unauthenticated connection with [Public] access to the directory.

- A clear text bind is an authentication over an unencrypted channel. The client sends a user name and a clear-text password. The LDAP server must be configured to accept unencrypted passwords.
- An SSL bind is an authentication over an encrypted channel. All data, including the password, is encrypted. eDirectory clients have access to SSL binds only through LDAP.

- Adding, modifying, and deleting entries and attributes in the directory.
- Reading, sorting, and searching entries and attributes in the directory.
- Reading schema definitions (object classes and attributes).
- Adding and deleting schema definitions (object classes and attributes). The LDAP server in NDS eDirectory 8.5 supports the modifying of class definitions and attribute definitions as long as the modifications increase functionality rather than restrict it.

The LDAP protocol does not yet support access to replication, partition, and synchronization services. These services require LDAP extensions that have been developed for NDS eDirectory 8.5 (for more information, see Section 1.6, "LDAP Extensions," on page 14).

1.5 LDAP Controls

eDirectory as a directory provides more services than are available with the standard LDAP functions or Java classes. LDAP v3 provides a method for extending its functionality through LDAP controls and extensions. The LDAP server in NDS eDirectory contains a registration mechanism for LDAP extensions. In future releases, Novell will write LDAP extensions to access such features as ACLs, replication, partition, and synchronization services. Novell is currently working with standards bodies to define ways to access such services.

For a list of currently supported controls, query the rootDSE.

1.6 LDAP Extensions

Novell has added LDAP extensions so that LDAP clients can manage naming contexts (partitions) and replicas. These extensions allow LDAP clients to manage the following:

- Split and join naming contexts (partitions)
- Add, delete, list, and change replicas
- Synchronize replicas and the schema

NDS supports the following extensions in NDS eDirectory 8.5.

1.6.1 LDAP Extensions

Table 1-4 *Supported LDAP Extensions*

OID	Name
2.16.840.1.113719.1.27.100.1	ndsToLdapResponse
2.16.840.1.113719.1.27.100.2	ndsToLdapRequest
2.16.840.1.113719.1.27.100.3	createNamingContextRequest

OID	Name
2.16.840.1.113719.1.27.100.4	createNamingContextResponse
2.16.840.1.113719.1.27.100.5	mergeNamingContextRequest
2.16.840.1.113719.1.27.100.6	mergeNamingContextResponse
2.16.840.1.113719.1.27.100.7	addReplicaRequest
2.16.840.1.113719.1.27.100.8	addReplicaResponse
2.16.840.1.113719.1.27.100.9	refreshLDAPServerRequest
2.16.840.1.113719.1.27.100.10	refreshLDAPServerResponse
2.16.840.1.113719.1.27.100.11	removeReplicaRequest
2.16.840.1.113719.1.27.100.12	removeReplicaResponse
2.16.840.1.113719.1.27.100.13	namingContextEntryCountRequest
2.16.840.1.113719.1.27.100.14	namingContextEntryCountResponse
2.16.840.1.113719.1.27.100.15	changeReplicaTypeRequest
2.16.840.1.113719.1.27.100.16	changeReplicaTypeResponse
2.16.840.1.113719.1.27.100.17	getReplicaInfoRequest
2.16.840.1.113719.1.27.100.18	getReplicaInfoResponse
2.16.840.1.113719.1.27.100.19	listReplicaRequest
2.16.840.1.113719.1.27.100.20	listReplicaResponse
2.16.840.1.113719.1.27.100.21	receiveAllUpdatesRequest
2.16.840.1.113719.1.27.100.22	receiveAllUpdatesResponse
2.16.840.1.113719.1.27.100.23	sendAllUpdatesRequest
2.16.840.1.113719.1.27.100.24	sendAllUpdatesResponse
2.16.840.1.113719.1.27.100.25	requestNamingContextSyncRequest
2.16.840.1.113719.1.27.100.26	requestNamingContextSyncResponse
2.16.840.1.113719.1.27.100.27	requestSchemaSyncRequest
2.16.840.1.113719.1.27.100.28	requestSchemaSyncResponse
2.16.840.1.113719.1.27.100.29	abortNamingContextOperationRequest
2.16.840.1.113719.1.27.100.30	abortNamingContextOperationResponse
2.16.840.1.113719.1.27.100.31	getContextIdentityNameRequest
2.16.840.1.113719.1.27.100.32	getContextIdentityNameResponse
2.16.840.1.113719.1.27.100.33	getEffectivePrivilegesRequest
2.16.840.1.113719.1.27.100.34	getEffectivePrivilegesResponse
2.16.840.1.113719.1.27.100.35	SetReplicationFilterRequest
2.16.840.1.113719.1.27.100.37	getReplicationFilterRequest

OID	Name
2.16.840.1.113719.1.27.100.39	createOrphanPartitionrequest
2.16.840.1.113719.1.27.100.41	removeOrphanPartitionRequest
2.16.840.1.113719.1.27.100.43	triggerBKLinkerRequest
2.16.840.1.113719.1.27.100.45	triggerDRLProcessRequest
2.16.840.1.113719.1.27.100.47	triggerJanitorRequest
2.16.840.1.113719.1.27.100.49	triggerLimberRequest
2.16.840.1.113719.1.27.100.51	triggerSkulkerRequest
2.16.840.1.113719.1.27.100.53	triggerSchemaSyncRequest
2.16.840.1.113719.1.27.100.55	triggerPartitionPurgeRequest
1.3.6.1.4.1.1466.20037	Start TLS
2.16.840.1.113719.1.27.100.79	EventMonitorRequest
2.16.840.1.113719.1.27.100.84	filteredEventMonitorRequest
2.16.840.1.113719.1.27.103.1	createGroupingRequest
2.16.840.1.113719.1.27.103.2	endGroupingRequest
2.16.840.1.113719.1.148.100.1	Put Login Configuration
2.16.840.1.113719.1.148.100.3	Get Login Configuration
2.16.840.1.113719.1.148.100.5	Delete Login Configuration
2.16.840.1.113719.1.148.100.7	Put Login Secret
2.16.840.1.113719.1.148.100.9	Delete Login Secret
2.16.840.1.113719.1.148.100.11	Set Universal Password
2.16.840.1.113719.1.148.100.13	Get Universal Password
2.16.840.1.113719.1.148.100.15	Delete Universal Password
2.16.840.1.113719.1.148.100.17	Check password against password policy
2.16.840.1.113719.1.39.42.100.19	Get password policy information
2.16.840.1.113719.1.39.42.100.21	Change Universal Password
2.16.840.1.113719.1.39.42.100.23	Graded Authentication management
2.16.840.1.113719.1.39.42.100.25	NMAS management (new with NMAS 3.1.0)
2.16.840.1.113719.1.148.100.1	SSLDAP_GET_SERVICE_INFO_REQUEST
2.16.840.1.113719.1.148.100.2	SSLDAP_GET_SERVICE_INFO_REPLY
2.16.840.1.113719.1.148.100.3	SSLDAP_READ_SECRET_REQUEST
2.16.840.1.113719.1.148.100.4	SSLDAP_READ_SECRET_REPLY
2.16.840.1.113719.1.148.100.5	SSLDAP_WRITE_SECRET_REQUEST
2.16.840.1.113719.1.148.100.6	SSLDAP_WRITE_SECRET_REPLY

OID	Name
2.16.840.1.113719.1.148.100.7	SSLDAP_ADD_SECRET_ID_REQUEST
2.16.840.1.113719.1.148.100.8	SSLDAP_ADD_SECRET_ID_REPLY
2.16.840.1.113719.1.148.100.9	SSLDAP_REMOVE_SECRET_REQUEST
2.16.840.1.113719.1.148.100.10	SSLDAP_REMOVE_SECRET_REPLY
2.16.840.1.113719.1.148.100.11	SSLDAP_REMOVE_SECRET_STORE_REQUEST
2.16.840.1.113719.1.148.100.12	SSLDAP_REMOVE_SECRET_STORE_REPLY
2.16.840.1.113719.1.148.100.13	SSLDAP_ENUMERATE_SECRET_IDS_REQUEST
2.16.840.1.113719.1.148.100.14	SSLDAP_ENUMERATE_SECRET_IDS_REPLY
2.16.840.1.113719.1.148.100.15	SSLDAP_UNLOCK_SECRETS_REQUEST
2.16.840.1.113719.1.148.100.16	SSLDAP_UNLOCK_SECRETS_REPLY
2.16.840.1.113719.1.148.100.17	SSLDAP_SET_EP_MASTER_PASSWORD_REQUEST
2.16.840.1.113719.1.148.100.18	SSLDAP_SET_EP_MASTER_PASSWORD_REPLY
2.16.840.1.113719.1.27.100.103	Get Privileges List Request
2.16.840.1.113719.1.27.100.104	Get Privileges List Response

LDAP Extensions Used by the Novell Import Convert Export Utility

The Novell Import Convert Export utility uses the following extensions. They are not general extensions designed for developer use but are designed to support the LDAP Bulk Update Replication Protocol (LBURP).

Table 1-5 *Extensions Used by the Novell Import Convert Export Utility*

OID	Name
2.16.840.1.113719.1.142.100.1	startFramedProtocolRequest
2.16.840.1.113719.1.142.100.2	startFramedProtocolResponse
2.16.840.1.113719.1.142.100.4	endFramedProtocolRequest
2.16.840.1.113719.1.142.100.5	endFramedProtocolResponse
2.16.840.1.113719.1.142.100.6	lburpOperationRequest
2.16.840.1.113719.1.142.100.7	lburpOperationResponse
2.16.840.1.113719.1.27.100.96	LDAPBackupRequest
2.16.840.1.113719.1.27.100.97	LDAPBackupResponse
2.16.840.1.113719.1.27.100.98	LDAPRestoreRequest

1.7 Differences Among LDAP Servers

You can use LDAP to implement an application that works only on a specific type of LDAP server with a specific type of directory. No two LDAP servers and their directories function exactly alike. For example, the LDAP servers available from Novell, Netscape, IBM, and Oracle all function slightly differently. If you want your application to be generic and work with most LDAP servers and their directories, you need to be aware of differences and the causes of those differences.

The following guidelines are not a definitive list of the differences between Novell and other implementations of an LDAP server and directory. These are intended to help make you code your application so that it will work on more than one directory. The definitive list will come as you test your application against multiple directories. Our suggestions are divided into the following topics:

1.7.1 Schema

No two directory schemas contain the same class and attribute definitions. Be aware of the following schema operations.

Guideline: Never assume that a particular attribute is available in a directory's schema.

The application should first read the schema definition before trying to use an attribute. For example, some non-standard LDAP attributes have not been added to the eDirectory schema. Therefore, if you code your application to expect a particular attribute, the application could fail.

Guideline: When extending a schema, select the most flexible method.

In some schemas, the Person-type class definitions allows an entry to be a container object. In the eDirectory schema, person-type definitions allow only leaf entries. The schema specification for LDAP v3 includes support for auxiliary classes. If you select to add attributes to a Person-type entry through an auxiliary class rather than as objects subordinate to the Person class, the application can work with both LDAP servers.

Guideline: Order schema extensions to accommodate referential validation.

If you define a class that references specific attributes, eDirectory verifies that all referenced attributes exist before it allows the class to be added. Therefore, when extending the schema, you must add the attributes before adding the class. Other directories perform no referential validation. To accommodate both types of directories, always add attribute definitions before adding class definitions.

1.7.2 Validation and Security Checking

LDAP servers have different rules for processing data. Be aware of the following.

Guideline: Order operations to accommodate integrity checking.

The Novell LDAP server uses eDirectory security and supports eDirectory validity and integrity checking. eDirectory verifies all attribute values that are DNs, a feature that increases the security and integrity of the directory. For example, when the DN of a user is added to a group membership list, eDirectory verifies that this user exists in the directory and creates a reference from the user to the group. This reference is used to remove the user from the group when the user is deleted from the directory. The delete operation removes all references to that user. If the system administrator creates another user with the same name, there is no residual information from the old user for the new user to connect with.

Some directories do not do this type of verification, and without the implementation of a reference, the user can be deleted from the directory while the user's name remains in a group membership list. If the system administrator fails to manually remove the user from the group list and then later adds another user with the same name, that new user becomes a member of the group the old user belonged to, regardless of the system administrator's intentions.

In directories that do not verify DNs, users and groups can be added to the directory in any order. You can add the groups with their membership lists and then the users that are associated with the groups. eDirectory will not allow this. The users must be added first, and then the groups with their membership lists. You can make your application work with both types of directories if you order the operation to accommodate eDirectory integrity checking.

Guideline: Request an LDAP v3 binding when the LDAP specification requires it.

Novell has the overall philosophy of restricting rights and enforcing those restrictions. According to the LDAP specifications, clients should have an LDAP v3 bind in order to read the root DSE object. Novell's LDAP server enforces this restriction. Not all LDAP servers enforce this restriction. To make your application work with most LDAP servers, you need to request an LDAP v3 bind when reading the root DSE.

1.7.3 Directory Structure

LDAP directories vary in structure and structure rules. Be aware of the following.

Guideline: Do not assume that all directories have the same containment structure and naming rules.

Some LDAP servers access a directory with a flat database structure on a single computer. Novell's LDAP server accesses eDirectory, which has a hierarchical database that is distributed on multiple computers.

In maintaining the hierarchical structure, eDirectory enforces

- Containment rules that determine which types of objects another object can contain
- Naming rules that determine which attributes can be used to name objects

Some LDAP directories, which claim to have containment and naming rules, do not enforce these rules. Therefore, do not build your application with the assumption that a certain entry must use a specific attribute for naming or with the assumption that certain entries can be located anywhere in a directory or only in one type of container. Applications built with such assumptions will not work with all LDAP directories.

Guideline: Build your application to handle referrals.

eDirectory is a distributed database, which means that all available data is not usually local to an LDAP server. In most eDirectory trees, the data is stored in replicas on multiple servers. Therefore, the data that is local is the data that is in the replicas stored on the LDAP server, and the rest of the data is remote.

Because eDirectory has been designed to be distributed and to hide that fact from most end users, the LDAP server can be configured to search automatically for the data in remote locations. Such a search takes longer than a local search that returns referrals. To remove the impression that Novell's LDAP server is slow, the default configuration for the LDAP server in NDS eDirectory has been changed. It is now configured as most other LDAP servers, to return a referral when the data isn't local. Your application should be designed to handle referrals.

1.7.4 LDAP Control Support

Guideline: If your application uses LDAP controls, use only those features that all LDAP servers support.

LDAP controls have not been implemented alike. See the documentation for each LDAP server you want your application to run on for a list of those restrictions.

1.7.5 LDAP Extension Support

Guideline: If your application uses LDAP extensions, query the LDAP server to get the supported extensions.

LDAP extensions are specific to an LDAP server. For example, each LDAP server in an eDirectory could support different extensions. LDAP servers from different vendors will support different extensions. Currently, all LDAP extensions are in a draft state with the IETF and none has had the time to move from proposal to acceptance as a standard. Novell is proposing its naming context (partition) extensions to IETF.

1.8 LDAP Search Filters

The LDAP search filter grammar is specified in RFC 2254 and 2251. The grammar uses ABNF notation.

```
filter = " ( " filtercomp " ) "
filtercomp = and / or /not /item

and = "&" filterlist
   filterlist = 1*filter

or = "|" filterlist
   filterlist = 1*filter

not = "!" filterlist
   filterlist = 1*filter

item = simple/present/substring/extensible

simple = attr filtertype value
   attr =  name | name;binary
   filtertype = equal/approx/greater/less
   value = data valid for the attribute's syntax

equal = "="
approx = "~="
greater = ">="
less = "<="

present = attr "=*"
   attr =  name | name;binary

substing = attr "=" [initial] any [final]
   attr =  name | name;binary
   initial = value
   any = "*" *(value "*")
   final = value

extensible = attr [":dn"] [":" matchingrule] ":="value
            /[":dn] ":" matchingrule ":=" value
            /matchingrule = name | OID
```

For additional options for the attr option, see Section 4.1.5 of RFC 2251.

For additional information on the value option, see Section 4.1.6 of RFC 2251.

IMPORTANT

- eDirectory does not support LDAP approximate (~=) matching or extensible matching rules.
- You cannot use the `dn` attribute in an LDAP search filter. Filters using either `distinguishedName=` or `dn=` in the filter syntax will not function correctly.

1.8.1 Operators

Table 1-6 *LDAP Filter Operators*

Operator	Description
=	Used for presence and equality matching. To test if an attribute exists in the directory, use (attributename=*). All entries that have the specified attribute will be returned. To test for equality, use (attributename=value). All entries that have attributename=value are returned. For example, (cn=Kim Smith) would return entries with Kim Smith as the common name attribute. (cn=*) would return all entries that contained a cn attribute. The = operator can also be used with wildcards to find a substring, (cn=*ary*) would return mary, hillary, and gary.
>=	Used to return attributes that are greater than or equal to the specified value. For this to work, the syntax type of the attribute must have defined a mechanism to make this comparison. For example, (cn>=Kim Smith) would return all entries from Kim Smith to Z.
<=	Used to return attributes that are less than or equal to the specified value. For this to work, the syntax type of the attribute must have defined a mechanism to make this comparison. For example, (cn<=Kim Smith) would return all entries from A to Kim Smith.
~=	Used for approximate matching. The algorithm used for approximate matching varies with different LDAP implementations.

The following boolean operators can be combined with the standard operators to form more complex filters. Note that boolean operator syntax is used different in search filters than in the C and Java programming languages, but the concepts are the same.

Table 1-7 *LDAP Filter Boolean Operators*

Boolean Operators	Description
&	And. For example, (&(cn=Kim Smith) (telephonenumber=555-5555)) would return entries with common name of Kim Smith and a telephone number of 555-5555.
|	Or. For example, (|(cn=Kim Smith)(cn=Kimberly Smith)) would return entries with common name Kim Smith or Kimberly Smith.
!	Not. For example, (!(cn=Kim Smith)) would return entries with any cn other than Kim Smith. Note that the ! operator is unary.

Examples:

Filter and Description
(cn = Kim Smith) Returns entries with a common name of Kim Smith.
(&(cn=Kim Smith)(telephonenumber=555*)(emailaddress=*acme.com)) Returns entries with a common name of Kim Smith, a telephone number that starts with 555, and an e-mail address that ends in acme.com
(!(cn = Chris Jones)) Returns entries that do not have a common name of Chris Jones.
(&(objectClass=inetOrgPerson) (| (sn=Smith) (cn=Chris S*))) Returns entries that are of type inetOrgPerson with a surname of Smith or a common name beginning with Chris S.
(&(o=acme)(objectclass=Country)(!(|(c=spain)(c=us)) Returns entries that are of type Country from the organization Acme, that are not countries spain or us.

1.9 LDAP URLs

LDAP URLs provide a uniform method to access information on an LDAP server. Defined in RFC 2255, LDAP URLs begin with the prefix LDAP:// or LDAPS://. The following provides the syntax and descriptions of an LDAP URL.

```
ldap[s]://<hostname>:<port>/<base_dn>?<attributes>?<scope>?<filter>?<extension>
```

Note that ldaps is a common enhancement used to denote SSL, and is not defined in an RFC.

Table 1-8 *Field descriptions for an LDAP URL*

URL Element	Default Value	Description
hostname	none	DNS name or IP address of the LDAP server.
port	389	Port of the LDAP server.
base_dn	root	Base DN for the LDAP operation.
attributes	all attributes	A comma delimited list of attributes to return.
scope	base	Search scope.
filter	(objectClass=*)	Search filter.
extension	none	LDAP extended operations.

NOTE: An attribute list is required if you want to provide a scope (even if the attribute list is blank). To return all attributes within a specific scope you must include <base_dn>??<scope>.

1.10 Connecting and Authenticating to eDirectory with LDAP

The LDAP connection and authentication model is a two-step process.

1. The client first establishes a connection to the LDAP server.
   - Non-secure, clear-text connections are established by default on port 389.
   - Data encrypted connections are established by default on port 636.

   The port numbers are configurable and may vary from LDAP server to LDAP server.
2. After the connection is established, the client then binds with the appropriate credentials for the port. The bind authenticates the client to eDirectory and establishes an identity.

The Novell LDAP server listens on both ports for connections and can be configured to accept both types of connections.

1.10.1 Connections on Port 389

LDAP supports various types of binds on port 389:

- Anonymous binds require the client to send in an empty string for both the user name and the password. The Novell LDAP server accepts such a bind request and establishes the client as the NDS entry [Public].
- Clear-text password binds require a valid distinguished name and a password. Since clear-text passwords compromise security, eDirectory by default accepts only encrypted passwords and must be configured to accept clear-text passwords. The client is authenticated with the identity of the distinguished name and given all of its eDirectory rights.

1.10.2 SSL Connections on Port 636

The SSL init functions exchange encryption keys as part of the connection process, and these keys are then used to encrypt all data sent across the connection.

The eDirectory server currently supports only server-side authentication. This means that the client must obtain a certificate file from each LDAP server to which the client will authenticate. Each Novell LDAP server must be configured for SSL authentication and export a certificate for clients to use.

The client sends this certificate with an SSL init function. The client then binds using a distinguished name and password, which establishes the client as the eDirectory identity of the distinguished name. The client can also bind as an anonymous user by sending empty strings for the distinguished name and password. Such a bind request establishes the client as the NDS identity [Public].

1.10.3 Anonymous Binds

An anonymous bind request establishes the client as the NDS identity [Public].When using the ldap_get_context_identity_name function on an anonymous bind, the function returns an empty string rather than [Public].

1.10.4 userPassword Attribute

When creating a user, a value must be set for the userPassword attribute in order for the user to log in to the directory.

When using an LDAP compare function to verify a userPassword value, the function can return true and the client can still be locked out of the account. eDirectory uses the following attributes, not just the userPassword attribute, to control access to an account:

- Locked By Intruder
- Login Allowed Time Map
- Login Disabled
- Login Expiration Time
- Login Maximum Simultaneous
- Password Expiration Interval
- Password Required

If the password is verified as valid, these other attributes should be checked to determine why the client cannot access the account. See the *NDK: Novell eDirectory Schema Reference* for a description of these attributes.

In NDS eDirectory 8.5, these attributes can be accessed from LDAP by using the name without the spaces. In previous versions of NDS, these attributes must be mapped to valid LDAP names to access them.

1.11 LDAP and eDirectory Schema

Although both the LDAP schema and the eDirectory schema are based on X.500, some of the implementation details are quite different with each being more restrictive in some areas and less restrictive in other areas. The following sections describe some of these differences:

- "Schema Naming Rules" on page 26
- "Schema Mapping" on page 26
- "LDAP Operational Attributes" on page 27
- "Attribute Flags" on page 30
- "Object Class Flags" on page 32
- "Syntax Definitions" on page 34
- "Auxiliary Classes versus Modifications to Class Definitions" on page 36

1.11.1 Schema Naming Rules

eDirectory supports different rules for naming object class definitions and attribute definitions. For example, eDirectory allows spaces and punctuation characters in the name. These conventions are incompatible with LDAP naming conventions, which restrict the name to alphabetical characters, no spaces, and one punctuation character, the hyphen.

The LDAP server in NDS eDirectory automatically maps the attributes and classes defined by RFC 2256 to their LDAP equivalent names. (For a list, see the LDAP Name index in the *NDK: Novell eDirectory Schema Reference*.) If LDAP clients need access to the other eDirectory classes and attributes that have incompatible names, the system administrator needs to use ConsoleOne to manually map them to an LDAP compatible name.

Even if the name is compatible with LDAP conventions, an LDAP client may not be able to access the attribute because the attribute's syntax is not supported by the LDAP server. For a list of supported syntax definitions, see "Syntax Definitions" on page 34.

The following paragraphs describe a couple of special cases for object class definitions.

User and inetOrgPerson: By default, inetOrgPerson is mapped to the NDS User object class, and you can access this class using the LDAP names of inetOrgPerson or user. By default, the User class definition does not contain all the standard attributes for inetOrgPerson. To add these attributes to the User class definition, the system administrator must run a schema file (`nov_inet.sch`). In NDS eDirectory, Novell also supplies a schema file that creates an inetOrgPerson class with the standard LDAP attributes. Your application will need to read the schema and the directory to determine which class contains the users for the system.

residentialPerson: In NDS eDirectory, Novell supplies a schema file that adds the residentialPerson class definition to the schema with the standard LDAP attributes. Your application will need to read the schema to determine if the system administrator has added this class to the schema.

1.11.2 Schema Mapping

Since the eDirectory schema has many class and attribute definitions with illegal LDAP names, these definitions must be mapped to a legal LDAP name. All LDAP servers in the eDirectory tree must map the same LDAP class or attribute to the same eDirectory class or attribute. Each version of eDirectory has made this mapping easier:

- In NDS eDirectory 8.3x, all standard LDAP attribute and class definitions are mapped to the corresponding eDirectory attribute or class definitions when eDirectory is installed. The *NDK: Novell eDirectory Schema Reference* contains an index to these LDAP names. If the schema is extended with classes and attributes that use invalid LDAP names, these classes and attributes need to be manually mapped to valid LDAP names.
- In NDS eDirectory 8.5, all eDirectory attribute and class definitions can be accessed from LDAP by using the eDirectory name with the colon and spaces removed. Also, any required mappings can be processed from an LDIF file during installation.

From ConsoleOne, system administrators can add mappings for classes and attributes that are not automatically mapped. The LDAP server needs to be stopped and started to make the mapped attributes and classes visible from LDAP.

1.11.3 LDAP Operational Attributes

In eDirectory, not all information about an entry is kept in attributes, for example, an entry's base class, last modified time, or creation time. Through NDAP, eDirectory has made this information available through DSI flags. In NDS eDirectory 8.5, this information is available through LDAP as operational attributes. The information is read-only. Clients cannot modify the attributes.

These attributes are not returned in search results unless explicitly requested by name. NDS eDirectory 8.5 supports the following operational attributes on all entries in the directory.

Table 1-9 *LDAP Operational Attributes*

Operation Attribute	Description
"createTimeStamp"	Contains when the entry was created. Standard LDAP attribute (RFC 2252).
"creatorsName"	Contains the distinguished name of the user that created this entry. Standard LDAP attribute (RFC 2252).
"entryFlags"	Contains information about an entry's state, for example whether the entry is an alias, a partition, or a container. eDirectory-specific attribute.
"federationBoundary"	Contains where the federation boundary begins for a DNS-rooted tree. eDirectory-specific attribute.
"localEntryID"	Contains the record number for the entry in the server's local database. eDirectory-specific attribute.
"modifiersName"	Contains the distinguished name of the last user that modified this entry. Standard LDAP attribute (RFC 2252).
"modifyTimeStamp"	Contains when the entry was last modified. Standard LDAP attribute (RFC 2252).
"structuralObjectClass"	Contains the base class of the entry. Standard LDAP attribute.
"subordinateCount"	Contains the number of entries immediately subordinate to this entry. eDirectory-specific attribute.
"subschemaSubentry"	Contains the LDAP name for the schema. Standard LDAP attribute (RFC 2252). For more information, see "Subschema Subentry Attributes" on page 29.

For more information, see "LDAP Operational Attributes" in the *NDK: Novell eDirectory Schema Reference*.

1.11.4 Root DSE Attributes

The LDAP server maintains the root DSE attributes, and clients can read these attributes but cannot modify them. The Novell LDAP server supports the following attributes.

Table 1-10 *Root DSE Attributes*

Attribute	Description	NDS/eDirectory Version
namingContexts	Contains the distinguished names of the naming contexts (or replicas) on the LDAP server. Standard LDAP attribute (RFC 2252).	8.x
altServer	Contains the name of alternative servers if this one is unavailable in subsequent requests Standard LDAP attribute (RFC 2252)..	8.5
supportedExtension	Contains a list of OIDs that identify supported extended operations. Standard LDAP attribute (RFC 2252).	8.x
supportedControl	Contains a list of OIDs that identify supported controls. Standard LDAP attribute (RFC 2252).	8.x . In version 85, controls are not advertized.
supportedSASLMechanisms	Contains a list of supported SASL security features. Standard LDAP attribute (RFC 2252).	8.5
supportedLDAPVersion	Contains a list of LDAP versions implemented by the server. Standard LDAP attribute (RFC 2252).	8.x
subschemaSubentry	Contains the distinguished name of the subschema subentry which is the name of the schema for this server. For eDirectory, this is always cn=schema. Standard LDAP attribute (RFC 2252).	8.x

Attribute	Description	NDS/eDirectory Version
vendorName	Contains a string with the name of the company. eDirectory-specific attribute.	8.5
vendorVersion	Contains the LDAP server version. eDirectory-specific attribute.	8.5
directoryTreeName	Contains the eDirectory tree name. eDirectory-specific attribute.	8.5
dsaName	Contains the distinguished name of the server. eDirectory-specific attribute.	8.5

In a DNS-rooted tree, you can find the federation boundary by one of the following methods:

- Using the namingContext attribute, search one of the naming contexts for its federationBoundary attribute.
- Using the dsaName attribute, search the server for its federationBoundary attribute. Use this method only if a naming context is not available.

1.11.5 Subschema Subentry Attributes

Clients must specifically request attribute information about the subschema entry. It is never returned in a generic request to read all information about all entries. Clients must request a base object search with the search filter set to the following: "objectClass=subschema"

The Novell LDAP server supports the following attributes.

Table 1-11 *Subschema Subentry Attributes*

Attribute	Description	NDS/eDirectory Version
cn	Contains the RDN of the subschema entry.	8.x
objectClasses	Contains a value for each object class known by the LDAP server.	8.x
objectClass	Contains the parent object classes of the subschema subentry. It always contains two classes: top and subschema.	8.x
attributeTypes	Contains a value for each attribute definition known by the LDAP server.	8.x
ldapSyntaxes	Contains a value for each syntax definition known by the LDAP server.	8.5

Clients add attribute definitions and object classes to the schema by adding values to the attributeTypes and objectClasses attributes of the subentry.

1.11.6 Attribute Flags

eDirectory supports numerous attribute definition flags that affect the type of data an attribute can contain and that control its synchronization schedule. The LDAP server in NDS eDirectory can set and get information about the following eDirectory extended flags. By default, when creating an attribute definition, none of these flags are set. To obtain the information when reading the schema, you must send the flag with the request to read the attribute.

Table 1-12 *eDirectory Extended Attribute Flags*

Extended LDAP Flags	Description
X-NDS_PUBLIC_READ	When set, allows anyone to read the attribute's value even though such rights have not been granted or inherited. Using this flag makes access to the attribute extremely efficient because eDirectory performs no rights checking. When not set, users must have been granted rights or inherit rights to read the attribute's value. In NDAP, this is equivalent to the DS_PUBLIC_READ flag set to True.
X-NDS_SERVER_READ	When set, allows the NCP Server object to read the attribute's value even though such rights have not been granted or inherited. When not set, the NCP Server object must be granted rights or inherit rights to read the attribute's value. In NDAP, this is equivalent to the DS_SERVER_READ flag set to True.
X-NDS_NEVER_SYNC	When set, prevents changes to this attribute from synchronizing with other replicas. The information in the attribute is specific to the replica. When not set, changes to the attribute are synchronized to other replicas. In NDAP, this is equivalent to the DS_PER_REPLICA flag set to True.
X-NDS_NOT_SCHED_SYNC_IMMEDIATE	When set, allows the attribute's value to change without scheduling synchronization, and synchronization will start within 30 minutes. When not set, causes any changes to the attribute to schedule immediate synchronization (within 10 seconds). In NDAP, this is equivalent to the DS_SYNC_IMMEDIATE flag set to False.

Extended LDAP Flags	Description
X-NDS_SCHED_SYNC_NEVER	When set, allows the attribute's value to change without scheduling synchronization. The attribute can wait until the next scheduled synchronization cycle to propagate its changes. When not set, causes any changes to the attribute to schedule synchronization. Developers can only read this flag. In NDAP, this is equivalent to the DS_SCHEDULE_SYNC_NEVER flag set to True.
X-NDS_LOWER_BOUND	When set, specifies the lower boundary for a string or integer syntax. When not set, the attribute has no lower boundary. In NDAP, this is equivalent to the DS_SIZED_ATTR flag set to True
X-NDS_NAME_VALUE_ACCESS	This flag only works on attributes which use a DN syntax and contain a list of entries, such as groupMembership. When set, requires users to have supervisor rights to the entry before they can add or delete the entry as a value for this attribute. When not set, requires the user to have read rights to read the values and write rights to modify the values. In NDAP, this is equivalent to the DS_WRITE_MANAGED flag set to True.
X-NDS_NAME	When creating an attribute, specifies the legacy eDirectory attribute that automatically maps to this LDAP attribute. This is new in NDS eDirectory 8.5 and should be used to make attributes available to previous versions of the LDAP server in an eDirectory tree. When reading the attribute definition, returns the legacy eDirectory attribute name.

Extended LDAP Flags	Description
X-NDS_ACL_TEMPLATES	Every object in the NDS tree has an ACL attribute. This attribute holds information about which trustees have access to the object itself (entry rights) and which trustees have access to the attributes for the object. This information is stored in sets of information containing the following: ◆ The trustee name ◆ The affected attribute-[Entry Rights], [All Attributes Rights], or a specific attribute ◆ The privileges ACL templates helps us in defining ACLs for specific classes in the base schema and provide a minimum amount of access security for newly created objects. This flag was added in 8.7.0.

The standard LDAP attribute flags can also be used. The following table lists the LDAP name and the corresponding NDAP name.

Table 1-13 *Standard LDAP Attribute Flags*

Standard LDAP Flags	NDAP Flag
SINGLE-VALUE	DS_SINGLE_VALUED_ATTR set to True
COLLECTIVE	Not supported
NO-USER-MODIFICATION	DS_READ_ONLY_ATTR set to True
USAGE userApplications	None required. This sets the attribute as a normal attribute. The other USAGE flags can only be set by eDirectory.
USAGE directoryOperation	DS_OPERATIONAL (set by eDirectory)
USAGE distributedOperation	DS_OPERATIONAL (set by eDirectory)
USAGE dSAOperation	DS_OPERATIONAL (set by eDirectory)

1.11.7 Object Class Flags

eDirectory uses a set of flags to define allowable class operations. When adding a new object class definition to the schema, you can set the following flags. When reading definitions, you send the flags to obtain the information.

Table 1-14 *eDirectory Extended Object Class Flags*

Extended LDAP Flags	Description
X-NDS_NOT_CONTAINER	When set, indicates that this object class cannot contain other entries and is thus a leaf entry. When not set, indicates that this object class can contain other entries and is thus a container class. In NDAP, this is equivalent to the DS_CONTAINER_CLASS flag set to False.
X-NDS_CONTAINMENT	When included, this flag is followed by a list of object classes that can be the parent container of the object class that is being defined. When not included, the object class that is being defined is automatically assigned containment classes of country, organization, organizationalUnit, locality, and domain. In NDAP, this is equivalent to the DS_AMBIGUOUS_CONTAINMENT flag set to False.
X-NDS_NAMING	When included, this flag is followed by the list of attributes that can be used to name entries based on this object class definition. When not included, the naming attributes for the object class are all of the MAY and MUST attributes that use a string-type syntax. In NDAP, this is equivalent to the DS_AMBIGUOUS_NAMING flag set to False.
X-NDS_NONREMOVABLE	When set, indicates that the class cannot be removed even if no entries are using the definition. This flag is placed on all classes in the eDirectory operational schema. NDS 8 and higher allow application developers to set this flag. When not set, indicates that the class can be removed from the schema if no entries are using the definition. In NDAP, this is equivalent to the DS_NONREMOVABLE_CLASS flag set to True.
X-NDS_NAME	When defining an object class, specifies the legacy eDirectory object class that automatically maps to this LDAP class. This is new in NDS eDirectory 8.5 and should be used to make classes available to previous versions of the LDAP server in an eDirectory tree. When reading object class definitions, returns the legacy eDirectory name for this object class.

The standard LDAP class flags can also be used. The following table lists the LDAP name and the corresponding NDAP name.

Table 1-15 Standard LDAP Flags

Standard LDAP Flags	NDAP FLag
ABSTRACT	DS_EFFECTIVE_CLASS set to False
STRUCTURAL	DS_EFFECTIVE_CLASS set to True
AUXILIARY	DS_AUXILIARY_CLASS set to True

1.11.8 Syntax Definitions

The LDAP server allows LDAP access to eDirectory attributes if the eDirectory attribute uses an LDAP compatible syntax. For example, this makes all eDirectory attributes that use the Case Ignore String syntax available through LDAP because LDAP supports the Case Ignore String syntax. eDirectory attributes that use a compound syntax (such as the Timestamp syntax with its fields for time, replica number, and event identifier) are not automatically accessible through LDAP. The LDAP server has made most of these syntax definitions available.

The LDAP server supports the following eDirectory syntax definitions. The eDirectory column lists the minimum version of eDirectory for the LDAP server to support the syntax.

Table 1-16 eDirectory Syntax Definitions

NDS Name	LDAP Descriptive Name	OID	NDS
SYN_STREAM	Binary	1.3.6.1.4.1.1466.115.121.1.5	7.xx
SYN_BOOLEAN	Boolean	1.3.6.1.4.1.1466.115.121.1.7	7.xx
SYN_DIST_NAME	DN	1.3.6.1.4.1.1466.115.121.1.1 1.3.6.1.4.1.1466.115.121.1.12	7.xx
SYN_CI_STRING	Directory String	1.3.6.1.4.1.1466.115.121.1.3 1.3.6.1.4.1.1466.115.121.1.11 1.3.6.1.4.1.1466.115.121.1.15 1.3.6.1.4.1.1466.115.121.1.16 1.3.6.1.4.1.1466.115.121.1.17 1.3.6.1.4.1.1466.115.121.1.54 1.3.6.1.4.1.1466.115.121.1.57 1.3.6.1.4.1.1466.115.121.1.30 1.3.6.1.4.1.1466.115.121.1.31 1.3.6.1.4.1.1466.115.121.1.32 1.3.6.1.4.1.1466.115.121.1.33 1.3.6.1.4.1.1466.115.121.1.34 1.3.6.1.4.1.1466.115.121.1.35 1.3.6.1.4.1.1466.115.121.1.37 1.3.6.1.4.1.1466.115.121.1.38 1.3.6.1.4.1.1466.115.121.1.39	7.xx
SYN_FAX_NUMBER	Facsimile Telephone Number	1.3.6.1.4.1.1466.115.121.1.22	7.xx
SYN_TIME	Generalized Time	1.3.6.1.4.1.1466.115.121.1.24	7.xx
SYN_CE_STRING	IA5 String	1.3.6.1.4.1.1466.115.121.1.26	7.xx
SYN_INTEGER	Integer	1.3.6.1.4.1.1466.115.121.1.27	7.xx

NDS Name	LDAP Descriptive Name	OID	NDS
SYN_INTERVAL	Integer	1.3.6.1.4.1.1466.115.121.1.21 1.3.6.1.4.1.1466.115.121.1.27	7.xx
SYN_NU_STRING	Numeric String	1.3.6.1.4.1.1466.115.121.1.36	7.xx
SYN_CLASS_NAME	OID	1.3.6.1.4.1.1466.115.121.1.38	7.xx
SYN_OCTET_STRING	Octet String	1.3.6.1.4.1.1466.115.121.1.1 1.3.6.1.4.1.1466.115.121.1.2 1.3.6.1.4.1.1466.115.121.1.4 1.3.6.1.4.1.1466.115.121.1.6 1.3.6.1.4.1.1466.115.121.1.8 1.3.6.1.4.1.1466.115.121.1.9 1.3.6.1.4.1.1466.115.121.1.10 1.3.6.1.4.1.1466.115.121.1.13 1.3.6.1.4.1.1466.115.121.1.14 1.3.6.1.4.1.1466.115.121.1.18 1.3.6.1.4.1.1466.115.121.1.19 1.3.6.1.4.1.1466.115.121.1.20 1.3.6.1.4.1.1466.115.121.1.21 1.3.6.1.4.1.1466.115.121.1.23 1.3.6.1.4.1.1466.115.121.1.25 1.3.6.1.4.1.1466.115.121.1.28 1.3.6.1.4.1.1466.115.121.1.56 1.3.6.1.4.1.1466.115.121.1.29 1.3.6.1.4.1.1466.115.121.1.55 1.3.6.1.4.1.1466.115.121.1.40 1.3.6.1.4.1.1466.115.121.1.43 1.3.6.1.4.1.1466.115.121.1.42 1.3.6.1.4.1.1466.115.121.1.58 1.3.6.1.4.1.1466.115.121.1.45 1.3.6.1.4.1.1466.115.121.1.46 1.3.6.1.4.1.1466.115.121.1.47 1.3.6.1.4.1.1466.115.121.1.48 1.3.6.1.4.1.1466.115.121.1.49 1.3.6.1.4.1.1466.115.121.1.51 1.3.6.1.4.1.1466.115.121.1.52 1.3.6.1.4.1.1466.115.121.1.53	7.xx
SYN_PO_ADDRESS	Postal Address	1.3.6.1.4.1.1466.115.121.1.41	7.xx
SYN_PR_STRING	Printable String	1.3.6.1.4.1.1466.115.121.1.44	7.xx
SYN_TEL_NUMBER	Telephone Number	1.3.6.1.4.1.1466.115.121.1.50	7.xx
SYN_UNKNOWN	Unknown	2.16.840.1.113719.1.1.5.1.0	7.xx
SYN_CI_LIST	Case Ignore List	2.16.840.1.113719.1.1.5.1.6	8.5
SYN_NET_ADDRESS	Tagged Data	2.16.840.1.113719.1.1.5.1.12	8.3x
SYN_OCTET_LIST	Octet List	2.16.840.1.113719.1.1.5.1.13	8.5
SYN_EMAIL_ADDRESS	Tagged String	2.16.840.1.113719.1.1.5.1.14	8.5
SYN_PATH	Tagged Name and String	2.16.840.1.113719.1.1.5.1.15	8.3x
SYN_REPLICA_POINTER	NDS Replica Pointer	2.16.840.1.113719.1.1.5.1.16	8.5

NDS Name	LDAP Descriptive Name	OID	NDS
SYN_OBJECT_ACL	NDS ACL	2.16.840.1.113719.1.1.5.1.17	8.3x
SYN_TIMESTAMP	NDS Timestamp	2.16.840.1.113719.1.1.5.1.19	8.5
SYN_COUNTER	Counter	2.16.840.1.113719.1.1.5.1.22	8.5
SYN_BACK_LINK	Tagged Name	2.16.840.1.113719.1.1.5.1.23	8.5
SYN_TYPED_NAME	Typed Name	2.16.840.1.113719.1.1.5.1.25	8.5

Most of the definitions with Novell OIDs (2.16.840.1.113719) are structured and have multiple components. The LDAP server converts such a syntax to case ignore strings, using dollar ($) signs to separate fields of the same data type and (#) signs to separate fields of different data types. See "Attribute Syntax Definitions" in the *NDK: Novell eDirectory Schema Reference* for more information.

The SYN_HOLD syntax is not supported through LDAP and is being discontinued in eDirectory.

1.11.9 Auxiliary Classes versus Modifications to Class Definitions

eDirectory and LDAP have had very different conventions about modifying existing object class definitions. Once an object class has been defined in the schema,

- LDAP conventions assume (1) that the attributes for the object class do not change, and (2) that any new attributes will be added through auxiliary classes to the entry rather than to the class definition.
- eDirectory conventions allow applications to add new attributes to the class definitions, or with the release of NDS 8, to add new attributes to the entry through auxiliary classes rather than the class definition.

To maintain backwards compatibility with NDS releases prior to NDS 8, most eDirectory applications are still adding attributes to class definitions rather than to entries through auxiliary classes.

When accessing eDirectory, you must read the schema to discover all possible attributes for a class definition. If your application accesses more than one eDirectory tree, it must read the schema from each tree because the chance of the schemas being the same is very small.

1.12 LDAP Client Functionality

This section lists a large number of common LDAP operations and a corresponding code sample detailing how to perform the operation using the LDAP Libraries for C (http://developer.novell.com/ndk/cldap.htm), LDAP Classes for Java (http://developer.novell.com/ndk/jldap.htm), or the LDAP Extensions and Controls for JNDI (http://developer.novell.com/ndk/extjndi.htm).

Table 1-17 *LDAP Client Functionality*

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java	JNDI
Add entry	addentry.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/addentry.c.html)	AddEntry.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/AddEntry.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Add replica	addrepl.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/addrepl.c.html)	AddReplica.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/AddReplica.java.html)	AddReplica.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/AddReplica.java.html)
Add user to group	addUserToGroup.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/addUserToGroup.c.html)	AddUserToGroup.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/AddUserToGroup.java.html)	AddUserToGroup.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/AddUserToGroup.java.html)
Background process, trigger	ldaptrigger.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/ldaptrigger.c.html)	TriggerBackground.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/TriggerBackground.java.html)	TriggerBackgroundProcess.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/TriggerBackgroundProcess.java.html)
Bind	bind.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/bind.c.html)	CheckBind.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/CheckBind.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Bind, check	CheckBind.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/CheckBind.c.html)	CheckBind.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/CheckBind.java.html)	CheckBind.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/CheckBind.java.html)
Bind, Interactive SSL	sslbindi.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/sslbindi.c.html)	TLSTrustManager.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/security/TLSTrustManager.java.html)	

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java	JNDI
Bind, Get DN	getbinddn.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/getbinddn.c.html)	GetBindDN.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/GetBindDN.java.html)	
Md5 bind	md5bind.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/md5bind.c.html)		
Nmas bind	nmasbind.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/nmasbind.c.html)		
SASL external bind	saslExternal.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/saslExternal.c.html)		JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
SSL bind	sslbind.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/sslbind.c.html)	SSLConnection.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/security/SSLConnection.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Get authenticated		GetAuthenticated.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/GetAuthenticated.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Compare attributes	cmpattrs.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/cmpattrs.c.html)	CompareAttrs.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/CompareAttrs.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Delete entry	delentry.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/delentry.c.html)	DeleteEntry.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/DeleteEntry.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java	JNDI
Dynamic group	dyngroup.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/dyngroup.c.html)	DynamicGroup.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/DynamicGroup.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Events	monitorevents.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/monitorevents.c.html)		
Extended Partial Response			
Extensible match	extmatch.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/extmatch.c.html)	ExtensibleMatch.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/ExtensibleMatch.java.html)	
Get DSE	getdse.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/getdse.c.html)	GetDSE.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/GetDSE.java.html)	GetDSE.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/GetDSE.java.html)
Get effective rights	getpriv.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/getpriv.c.html)	GetEffectivePrivileges.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/GetEffectivePrivileges.java.html)	GetEffectiveRights.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/GetEffectiveRights.java.html)
Get name from OID	nameandoid.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/nameandoid.c.html)	NameAndOID.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/NameAndOID.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java	JNDI
Get replication filter	getReplFilter.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/getReplFilter.c.html)	GetReplicationFilter.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/GetReplicationFilter.java.html)	GetReplicationFilter.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/GetReplicationFilter.java.html)
Grace login	gracelog.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/gracelog.c.html)	GraceLogin.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/GraceLogin.java.html)	
Indexes, Create, delete and list eDirectory database indexes.	index.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/index.c.html)		
List	list.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/list.c.html)	List.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/List.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
List group	listgroup.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/listgroup.c.html)	ListGroups.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/ListGroups.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
List schema	schema.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/schema.c.html)	ListSchema.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/ListSchema.java.html)	ListSchema.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/ListSchema.java.html)
Make container	mkcontainer.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/mkcontainer.c.html)	MakeContainer.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/MakeContainer.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Modify ACL	modifyACL.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/modifyACL.c.html)	ModifyACL.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/ModifyACL.java.html)	ModifyACL.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/ModifyACL.java.html)

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java	JNDI
Modify attributes	modattrs.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/modattrs.c.html)	ModifyAttrs.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/ModifyAttrs.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Modify password	modpass.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/modpass.c.html)	ModifyPassword.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/ModifyPassword.java.html)	ModifyPassword.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/ModifyPassword.java.html)
Modify time stamp	modifyTimeStamp.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/modifyTimeStamp.c.html)	ModifyTimeStamp.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/ModifyTimeStamp.java.html)	ModifyTimeStamp.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/ModifyTimeStamp.java.html)
Mutual authentication	mutual.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/mutual.c.html)		
OID from name	nameandoid.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/nameandoid.c.html)	NameAndOID.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/NameAndOID.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Operational attributes	operationalAttrs.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/operationalAttrs.c.html)	OperationalAttrs.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/OperationalAttrs.java.html)	OperationalAttrs.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/OperationalAttrs.java.html)
Partition, abort operation	abortpo.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/abortpo.c.html)	AbortPartitionOperation.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/AbortPartitionOperation.java.html)	AbortPartitionOperation.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/AbortPartitionOperation.java.html)

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java	JNDI
Partition, entry count	getcount.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/getcount.c.html)	PartitionEntryCount.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/PartitionEntryCount.java.html)	
Partition, merge	mergepart.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/mergepart.c.html)	MergePartitions.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/MergePartitions.java.html)	
Partition, remove orphan		RemoveOrphanPartition.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/RemoveOrphanPartition.java.html)	RemoveOrphanPartition.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/RemoveOrphanPartition.java.html)
Partition, split orphan		SplitOrphanPartition.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/SplitOrphanPartition.java.html)	
Partition, split	splitpart.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/splitpart.c.html)	SplitPartition.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/SplitPartition.java.html)	
Partition, sync	parsync.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/parsync.c.html)	PartitionSync.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/PartitionSync.java.html)	
Password, set	setpass.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/setpass.c.html)	SetPassword.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/SetPassword.java.html)	SetPassword.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/SetPassword.java.html)

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java	JNDI
Password, verify	verpass.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/verpass.c.html)	VerifyPassword.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/VerifyPassword.java.html)	VerifyPassword.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/VerifyPassword.java.html)
Persistent search	searchPersist.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/searchPersist.c.html)	SearchPersist.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/controls/SearchPersist.java.html)	Psearch.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/Psearch.java.html)
Rebind	rebind.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/rebind.c.html)	SearchUtil.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/SearchUtil.java.html)	
Receive updates	recvupd.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/recvupd.c.html)	ReceiveAllUpdates.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/ReceiveAllUpdates.java.html)	ReceiveAllUpdates.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/ReceiveAllUpdates.java.html)
Refresh LDAP server	recvupd.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/recvupd.c.html)	RefreshLDAPServer.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/RefreshLDAPServer.java.html)	RefreshLDAPServer.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/RefreshLDAPServer.java.html)
Rename RDN	renamerdn.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/renamerdn.c.html)	RenameRDN.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/RenameRDN.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Replica, remove	remrepl.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/remrepl.c.html)	RemoveReplica.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/RemoveReplica.java.html)	RemoveReplica.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/RemoveReplica.java.html)

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java	JNDI
Replica, type change	chgrepl.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/chgrepl.c.html)	ChangeReplicaType.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/ChangeReplicaType.java.html)	ChangeReplicaType.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/ChangeReplicaType.java.html)
Replica, get information	getrinfo.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/getrinfo.c.html)	GetReplicaInfo.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/GetReplicaInfo.java.html)	GetReplicaInfo.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/GetReplicaInfo.java.html)
Replica, list	listrepl.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/listrepl.c.html)	ListReplicas.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/ListReplicas.java.html)	ListReplicas.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/ListReplicas.java.html)
Replication filter, set	setReplFilter.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/setReplFilter.c.html)	SetReplicationFilter.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/SetReplicationFilter.java.html)	SetReplicationFilter.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/SetReplicationFilter.java.html)
Schema, extend	schema.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/schema.c.html)	ExtendSchema.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/ExtendSchema.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Schema, sync	schsync.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/schsync.c.html)	SchemaSync.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/SchemaSync.java.html)	SchemaSync.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/SchemaSync.java.html)

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java	JNDI
Search	search.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/search.c.html), searchmsg.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/searchmsg.c.html)	Search.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/Search.java.html), SearchUtil.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/SearchUtil.java.html)	JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Search, binary	searchBinary.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/searchBinary.c.html)		JNDI Tutorial (http://java.sun.com/products/jndi/tutorial/index.html)
Search, URL	searchURL.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/searchURL.c.html)	UrlSearch.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/UrlSearch.java.html)	
Send updates	sendupd.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/extensions/sendupd.c.html)	SendAllUpdates.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/extensions/SendAllUpdates.java.html)	SendAllUpdates.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/ext/SendAllUpdates.java.html)
Sort, client-side	multisort.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/multisort.c.html)	ClientSideSort.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/ClientSideSort.java.html)	
Sort, server-side sort control	sortcntl.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/sortcntl.c.html)	SortControl.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/controls/SortControl.java.html)	SearchControl.java (http://developer.novell.com/ndk/doc/samplecode/extjndi_sample/extjndi_sample/SearchControl.java.html)
TLS, Start/Stop	starttls.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/starttls.c.html)	StartTLS.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/security/StartTLS.java.html)	

LDAP Operation	LDAP Libraries for C	LDAP Classes for Java JNDI
Unsolicited Listener		UnsolicitedListener.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/jldap_sample/UnsolicitedListener.java.html)
Virtual list view control	vlvcntl.c (http://developer.novell.com/ndk/doc/samplecode/cldap_sample/cldap_sample/vlvcntl.c.html)	VLVControl.java (http://developer.novell.com/ndk/doc/samplecode/jldap_sample/controls/VLVControl.java.html)

1.13 Other Sources of LDAP Information

If you are new to LDAP and Novell development, check out the following sources:

- Novell's LDAP site (http://developer.novell.com/edirectory). This site contains links to generic LDAP information and to specific information about the Novell LDAP server.
- Novell's DeveloperNet University (http://developer.novell.com/education/index.html). This site contains source code for specific LDAP operations and the labs take you step-by-step through the process.

2 LDAP Return Codes

This chapter describes the following:

- Return codes sent by the LDAP client or libraries
- Return codes sent by the LDAP server
- Result code structure used by the return codes

2.1 LDAP Client Return Codes

The LDAP client or libraries return codes set the following return codes.

Table 2-1 *LDAP Client Return Codes*

Hex	Decimal	Constant: Description
0x51	81	LDAP_SERVER_DOWN: Indicates that the LDAP libraries cannot establish an initial connection with the LDAP server. Either the LDAP server is down or the specified host name or port number is incorrect.
0x52	82	LDAP_LOCAL_ERROR: Indicates that the LDAP client has an error. This is usually a failed dynamic memory allocation error.
0x53	83	LDAP_ENCODING_ERROR: Indicates that the LDAP client encountered errors when encoding an LDAP request intended for the LDAP server.
0x54	84	LDAP_DECODING_ERROR: Indicates that the LDAP client encountered errors when decoding an LDAP response from the LDAP server.
0x55	85	LDAP_TIMEOUT: Indicates that the time limit of the LDAP client was exceeded while waiting for a result.
0x56	86	LDAP_AUTH_UNKNOWN: Indicates that the ldap_bind or ldap_bind_s function was called with an unknown authentication method.
0x57	87	LDAP_FILTER_ERROR: Indicates that the ldap_search function was called with an invalid search filter.
0x58	88	LDAP_USER_CANCELLED: Indicates that the user cancelled the LDAP operation.
0x59	89	LDAP_PARAM_ERROR: Indicates that an LDAP function was called with an invalid parameter value (for example, the ld parameter is NULL).
0x5A	90	LDAP_NO_MEMORY: Indicates that a dynamic memory allocation function failed when calling an LDAP function.
0x5B	91	LDAP_CONNECT_ERROR: Indicates that the LDAP client has lost either its connection or cannot establish a connection to the LDAP server.

Hex	Decimal	Constant: Description
0x5C	92	LDAP_NOT_SUPPORTED: Indicates that the requested functionality is not supported by the client. For example, if the LDAP client is established as an LDAPv2 client, the libraries set this error code when the client requests LDAPv3 functionality.
0x5D	93	LDAP_CONTROL_NOT_FOUND: Indicates that the client requested a control that the libraries cannot find in the list of supported controls sent by the LDAP server.
0x5E	94	LDAP_NO_RESULTS_RETURNED: Indicates that the LDAP server sent no results. When the ldap_parse_result function is called, no result code is included in the server's response.
0x5F	95	LDAP_MORE_RESULTS_TO_RETURN: Indicates that more results are chained in the result message. The libraries set this code when the call to the ldap_parse_result function reveals that additional result codes are available.
0x60	96	LDAP_CLIENT_LOOP: Indicates the LDAP libraries detected a loop. Usually this happens when following referrals.
0x61	97	LDAP_REFERRAL_LIMIT_EXCEEDED: Indicates that the referral exceeds the hop limit. The hop limit determines how many servers the client can hop through to retrieve data. For example, suppose the following conditions: ◆ The hop limit is two. ◆ The referral is to server D which can be contacted only through server B (1 hop) which contacts server C (2 hops) which contacts server D (3 hops) With these conditions, the hop limit is exceeded and the LDAP libraries set this code.

2.2 LDAP Server Return Codes

The LDAP server sets the following return codes.

Table 2-2 *LDAP Server Return Codes*

Hex	Decimal	Constant: Description
0x00	0	LDAP_SUCCESS: Indicates the requested client operation completed successfully.
0x01	1	LDAP_OPERATIONS_ERROR: Indicates an internal error. The server is unable to respond with a more specific error and is also unable to properly respond to a request. It does not indicate that the client has sent an erroneous message. In NDS 8.3x through NDS 7.xx, this was the default error for eDirectory errors that did not map to an LDAP error code. To conform to the new LDAP drafts, NDS 8.5 uses 80 (0x50) for such errors.
0x02	2	LDAP_PROTOCOL_ERROR: Indicates that the server has received an invalid or malformed request from the client.
0x03	3	LDAP_TIMELIMIT_EXCEEDED: Indicates that the operation's time limit specified by either the client or the server has been exceeded. On search operations, incomplete results are returned.

Hex	Decimal	Constant: Description
0x04	4	LDAP_SIZELIMIT_EXCEEDED: Indicates that in a search operation, the size limit specified by the client or the server has been exceeded. Incomplete results are returned.
0x05	5	LDAP_COMPARE_FALSE: Does not indicate an error condition. Indicates that the results of a compare operation are false.
0x06	6	LDAP_COMPARE_TRUE: Does not indicate an error condition. Indicates that the results of a compare operation are true.
0x07	7	LDAP_AUTH_METHOD_NOT_SUPPORTED: Indicates that during a bind operation the client requested an authentication method not supported by the LDAP server.
0x08	8	LDAP_STRONG_AUTH_REQUIRED: Indicates one of the following: ◆ In bind requests, the LDAP server accepts only strong authentication. ◆ In a client request, the client requested an operation such as delete that requires strong authentication. ◆ In an unsolicited notice of disconnection, the LDAP server discovers the security protecting the communication between the client and server has unexpectedly failed or been compromised.
0x09	9	Reserved.
0x0A	10	LDAP_REFERRAL: Does not indicate an error condition. In LDAPv3, indicates that the server does not hold the target entry of the request, but that the servers in the referral field may.
0x0B	11	LDAP_ADMINLIMIT_EXCEEDED: Indicates that an LDAP server limit set by an administrative authority has been exceeded.
0x0C	12	LDAP_UNAVAILABLE_CRITICAL_EXTENSION: Indicates that the LDAP server was unable to satisfy a request because one or more critical extensions were not available. Either the server does not support the control or the control is not appropriate for the operation type.
0x0D	13	LDAP_CONFIDENTIALITY_REQUIRED: Indicates that the session is not protected by a protocol such as Transport Layer Security (TLS), which provides session confidentiality.
0x0E	14	LDAP_SASL_BIND_IN_PROGRESS: Does not indicate an error condition, but indicates that the server is ready for the next step in the process. The client must send the server the same SASL mechanism to continue the process.
0x0F	15	Not used.
0x10	16	LDAP_NO_SUCH_ATTRIBUTE: Indicates that the attribute specified in the modify or compare operation does not exist in the entry.
0x11	17	LDAP_UNDEFINED_TYPE: Indicates that the attribute specified in the modify or add operation does not exist in the LDAP server's schema.
0x12	18	LDAP_INAPPROPRIATE_MATCHING: Indicates that the matching rule specified in the search filter does not match a rule defined for the attribute's syntax.
0x13	19	LDAP_CONSTRAINT_VIOLATION: Indicates that the attribute value specified in a modify, add, or modify DN operation violates constraints placed on the attribute. The constraint can be one of size or content (string only, no binary).

Hex	Decimal	Constant: Description
0x14	20	LDAP_TYPE_OR_VALUE_EXISTS: Indicates that the attribute value specified in a modify or add operation already exists as a value for that attribute.
0x15	21	LDAP_INVALID_SYNTAX: Indicates that the attribute value specified in an add, compare, or modify operation is an unrecognized or invalid syntax for the attribute.
	22-31	Not used.
0x20	32	LDAP_NO_SUCH_OBJECT: Indicates the target object cannot be found. This code is not returned on following operations: ◆ Search operations that find the search base but cannot find any entries that match the search filter. ◆ Bind operations.
0x21	33	LDAP_ALIAS_PROBLEM: Indicates that an error occurred when an alias was dereferenced.
0x22	34	LDAP_INVALID_DN_SYNTAX: Indicates that the syntax of the DN is incorrect. (If the DN syntax is correct, but the LDAP server's structure rules do not permit the operation, the server returns LDAP_UNWILLING_TO_PERFORM.)
0x23	35	LDAP_IS_LEAF: Indicates that the specified operation cannot be performed on a leaf entry. (This code is not currently in the LDAP specifications, but is reserved for this constant.)
0x24	36	LDAP_ALIAS_DEREF_PROBLEM: Indicates that during a search operation, either the client does not have access rights to read the aliased object's name or dereferencing is not allowed.
	37-47	Not used.
0x30	48	LDAP_INAPPROPRIATE_AUTH: Indicates that during a bind operation, the client is attempting to use an authentication method that the client cannot use correctly. For example, either of the following cause this error: ◆ The client returns simple credentials when strong credentials are required. ◆ The client returns a DN and a password for a simple bind when the entry does not have a password defined.
0x31	49	LDAP_INVALID_CREDENTIALS: Indicates that during a bind operation one of the following occurred: ◆ The client passed either an incorrect DN or password. ◆ The password is incorrect because it has expired, intruder detection has locked the account, or some other similar reason.
0x32	50	LDAP_INSUFFICIENT_ACCESS: Indicates that the caller does not have sufficient rights to perform the requested operation.
0x33	51	LDAP_BUSY: Indicates that the LDAP server is too busy to process the client request at this time but if the client waits and resubmits the request, the server may be able to process it then.
0x34	52	LDAP_UNAVAILABLE: Indicates that the LDAP server cannot process the client's bind request, usually because it is shutting down.

Hex	Decimal	Constant: Description
0x35	53	LDAP_UNWILLING_TO_PERFORM: Indicates that the LDAP server cannot process the request because of server-defined restrictions. This error is returned for the following reasons: ◆ The add entry request violates the server's structure rules. ◆ The modify attribute request specifies attributes that users cannot modify. ◆ Password restrictions prevent the action. ◆ Connection restrictions prevent the action.
0x36	54	LDAP_LOOP_DETECT: Indicates that the client discovered an alias or referral loop, and is thus unable to complete this request.
	55-63	Not used.
0x40	64	LDAP_NAMING_VIOLATION: Indicates that the add or modify DN operation violates the schema's structure rules. For example, ◆ The request places the entry subordinate to an alias. ◆ The request places the entry subordinate to a container that is forbidden by the containment rules. ◆ The RDN for the entry uses a forbidden attribute type.
0x41	65	LDAP_OBJECT_CLASS_VIOLATION: Indicates that the add, modify, or modify DN operation violates the object class rules for the entry. For example, the following types of request return this error: ◆ The add or modify operation tries to add an entry without a value for a required attribute. ◆ The add or modify operation tries to add an entry with a value for an attribute which the class definition does not contain. ◆ The modify operation tries to remove a required attribute without removing the auxiliary class that defines the attribute as required.
0x42	66	LDAP_NOT_ALLOWED_ON_NONLEAF: Indicates that the requested operation is permitted only on leaf entries. For example, the following types of requests return this error: ◆ The client requests a delete operation on a parent entry. ◆ The client request a modify DN operation on a parent entry.
0x43	67	LDAP_NOT_ALLOWED_ON_RDN: Indicates that the modify operation attempted to remove an attribute value that forms the entry's relative distinguished name.
0x44	68	LDAP_ALREADY_EXISTS: Indicates that the add operation attempted to add an entry that already exists, or that the modify operation attempted to rename an entry to the name of an entry that already exists.
0x45	69	LDAP_NO_OBJECT_CLASS_MODS: Indicates that the modify operation attempted to modify the structure rules of an object class.
0x46	70	LDAP_RESULTS_TOO_LARGE: Reserved for CLDAP.
0x47	71	LDAP_AFFECTS_MULTIPLE_DSAS: Indicates that the modify DN operation moves the entry from one LDAP server to another and thus requires more than one LDAP server.

Hex	Decimal	Constant: Description
	72-79	Not used.
0x50	80	LDAP_OTHER: Indicates an unknown error condition. This is the default value for eDirectory error codes which do not map to other LDAP error codes.

2.3 LDAP Result Code Structure

An LDAP result code has the following format:

```
LDAPResult   ::= SEQUENCE {
   resultCode      ENUMERATED,
   matchedDN       LDAPDN,
   errorMessage    LDAPString,
   referral        [3] Referral OPTIONAL
}
```

These fields contain the following information.

resultCode

Set to the return code by the LDAP server or the LDAP libraries.

matchedDN

Set to the lowest entry (object or entry) in the directory that was matched when the resultCode field is set to LDAP_NO_SUCH_OBJECT, LDAP_ALIAS_PROBLEM, LDAP_INVALID_DN_SYNTAX, or LDAP_ALIAS_DEREF_PROBLEM. If the resultCode field is set to other result codes, this field is set to a zero-length string.

For example, if the resultCode field is set to LDAP_NO_SUCH_OBJECT, and if the name "CN=Chris, OU=LDAP, O=Novell" is passed in, and if the LDAP server could find the Novell and LDAP entries but not Chris, then this field is set to "OU=LDAP, O=Novell".

errorMessage

Set to a human-readable string that explains the error code.

referral

Optional field that can be set to the URLs of servers that may hold the target entry.

LDAP result codes are returned with each message from the server in a BER-encoded form to the caller of the LDAP function. To retrieve the return code information from a result, use the ldap_parse_result function.

3 LDAP Event Services

LDAP Event Services provide a way for applications to monitor the activity of eDirectory on an individual server using LDAP. LDAP Event Services are available on eDirectory 8.7.

3.1 Concepts

LDAP Event Services utilizes the standard LDAP extension mechanism to expose the eDirectory event system. The LDAP Libraries for C (http://developer.novell.com/ndk/cldap.htm) are enhanced to provide support functions to simplify the use of the event system extension.

The event system extension allows the client to specify the events for which it wants to receive notification. This information is sent in the extension request. If the extension request specifies valid events, the LDAP server keeps the connection open and uses the intermediate extended response to notify the client when events occur. Any data associated with an event is also sent in the response. If an error occurs when processing the extended request or during the subsequent processing of events, the server sends an extended response to the client containing error information and then terminates the processing of the request.

3.1.1 Configuring the eDirectory Event System

The eDirectory Event System extension is configured on a per LDAP server basis using the iManager utility (for information, see the iManager Documentation (http://www.novell.com/documentation/lg/imanage10/index.html)). There are two parameters that need to be set. The "allow event monitoring" parameter will turn event monitoring on or off on that particular server. If event monitoring is turned off, the monitor events request will fail. The second parameter is the maximum event monitoring load for the server. A zero value indicates no load limit. Each event type is assigned an integer valued load factor. The load factor is a representation of the loading effect monitoring this event has on the server relative to all other event types. The load is calculated based on each monitored event's load factor and the number of clients registered for that event.

Client Access Rights to Event Data

Any LDAP client can register to monitor any event. Access retrictions are enforced at the time of notification. If the authenticated client does not have access rights to view all of the information in the event, the event will not be sent. The one exception to this rule is the perpetrator DN. If the client does not have rights to the perpertrator object it will be sent as a zero length string and represented as a NULL pointer value at the client. The event notification will still be sent.

3.2 Event Types

The eDirectory event system supports over 200 events. Each event is identified by an integer eventType and most events have associated event data with additional information about the event. This information is returned in one of several structs depending upon the event type.

The eDirectory event system events are grouped according to the structure of the associated event data. This event grouping is outlined in the following table:

Table 3-1 *eDirectory Event Types*

Event Type	Description
Entry Events	These events indicate the occurrence of individual entry operations such as creating or deleting an entry. The event data is contained in an “EVT_EntryInfo” structure.
Value Events	These events indicate the occurrence of attribute value operations such as deleting or adding a value. The event data is contained in a “EVT_ValueInfo” structure.
General DS Events	These are general events used to indicate a wide variety of DS operations. A generic structure, “EVT_EventData”, is used to hold the event data which needs to be interpreted based on the event type.
Bindery Events	These events occurrence of bindery emulation operations. The event data is contained in a “EVT_BinderyObjectInfo” structure.
Security Equivalence Event	This event indicates an entry’s security equivalence vector is being checked. The event data is contained in a “EVT_SEVInfo” structure.
Network Address Events	The data for these events is contained in a “EVT_NetAddress” structure.
Events without Data	This classification includes all events that do not have associated data.

3.2.1 Entry Events

The following table lists the event types that are associated with changes to individual attributes.

Table 3-2 *Entry Events*

Val	Event Type	Description
1	EVT_CREATE_ENTRY	A new eDirectory object has been created. This event does not include className and is set to null.
2	EVT_DELETE_ENTRY	An existing eDirectory object has been deleted.
3	EVT_RENAME_ENTRY	An existing eDirectory object has been renamed.
4	EVT_MOVE_SOURCE_ENTRY	This is the second of two events reported for a move operation. This event specifies the deletion of a eDirectory object from its original location in the Directory tree. (See EVT_MOVE_DEST_ENTRY).
14	EVT_MOVE_DEST_ENTRY	This is the first of two events reported for a move operation. This event specifies the placement of the eDirectory object into its new location in the Directory tree. (See EVT_MOVE_SOURCE_ENTRY.) This generates EVT_ADD_VALUE events for all of the values associated with the object.
15	EVT_DELETE_UNUSED_EXTREF	An unused external reference has been deleted.
228	EVT_PRE_DELETE_ENTRY	A pre-delete event posted when an entry is about to be deleted.

3.2.2 Value Events

The following table lists the events that are associated with changes to individual attributes:

Table 3-3 *Value Events*

	Event Type	Structure Returned
5	EVT_ADD_VALUE	A value has been added to an object attribute.
6	EVT_DELETE_VALUE	A value has been deleted from an object attribute.
7	EVT_CLOSE_STREAM	A Stream attribute has been closed.
8	EVT_DELETE_ATTRIBUTE	An attribute has been deleted from an object. This generates EVT_DELETE_VALUE events for values associated with the attribute. The EVT_DELETE_VALUE events occur after the EVT_DELETE_ATTRIBUTE event.

3.2.3 Debug Events

The following table lists the events that are associated with debug events:

Table 3-4 *Debug Events*

	Event Type	Description	Data Returned
26	EVT_DB_AUTHEN	An authentication debug message has been sent.	"EVT_DebugInfo"
27	EVT_DB_BACKLINK	A backlink debug message has been sent.	"EVT_DebugInfo"
28	EVT_DB_BUFFERS	A request buffer debug message has been sent.	"EVT_DebugInfo"
29	EVT_DB_COLL	A collision debug message has been sent.	"EVT_DebugInfo"
30	EVT_DB_DSAGENT	A low-level DSAgent debug message has been sent.	"EVT_DebugInfo"
31	EVT_DB_EMU	A Bindery emulation debug message has been sent.	"EVT_DebugInfo"
32	EVT_DB_FRAGGER	A Fragger debug message has been sent.	"EVT_DebugInfo"
33	EVT_DB_INIT	An initialization debug message has been sent.	"EVT_DebugInfo"
34	EVT_DB_INSPECTOR	An inspector debug message has been sent.	"EVT_DebugInfo"
35	EVT_DB_JANITOR	A Janitor process debug message has been sent.	"EVT_DebugInfo"
36	EVT_DB_LIMBER	A Limber process debug message has been sent.	"EVT_DebugInfo"
37	EVT_DB_LOCKING	A locking debug message has been sent.	"EVT_DebugInfo"
38	EVT_DB_MOVE	A move debug message has been sent.	"EVT_DebugInfo"

	Event Type	Description	Data Returned
39	EVT_DB_MIN	A default DSTrace (equivalent to ON) debug message has been sent.	“EVT_DebugInfo”
40	EVT_DB_MISC	A miscellaneous debug message has been sent	“EVT_DebugInfo”
41	EVT_DB_PART	A partition operations debug message has been sent.	“EVT_DebugInfo”
42	EVT_DB_RECMAN	A Record Manager debug message has been sent.	“EVT_DebugInfo”
44	EVT_DB_RESNAME	A Resolve Name debug message has been sent.	“EVT_DebugInfo”
45	EVT_DB_SAP	A SAP debug message has been sent.	“EVT_DebugInfo”
46	EVT_DB_SCHEMA	A schema debug message has been sent.	“EVT_DebugInfo”
47	EVT_DB_SKULKER	A synchronization debug message has been sent.	“EVT_DebugInfo”
48	EVT_DB_STREAMS	A streams debug message has been sent.	“EVT_DebugInfo”
49	EVT_DB_SYNC_IN	An incoming synchronization debug message has been sent.	“EVT_DebugInfo”
50	EVT_DB_THREADS	An eDirectory thread scheduling debug message has been sent.	“EVT_DebugInfo”
52	EVT_DB_TIMEVECTOR	A time vectors debug message has been sent.	“EVT_DebugInfo”
53	EVT_DB_VCLIENT	A virtual client debug message has been sent.	“EVT_DebugInfo”
166	EVT_DB_NCPENG	An NCPENG debug message has been sent.	“EVT_DebugInfo”

	Event Type	Description	Data Returned
175	EVT_DB_AUDIT	An auditing debug message has been sent.	“EVT_DebugInfo”
176	EVT_DB_AUDIT_NCP	An auditing NCP debug message has been sent.	“EVT_DebugInfo”
177	EVT_DB_AUDIT_SKULK	An auditing debug message concerning synchronization has been sent.	“EVT_DebugInfo”
184	EVT_DB-CHANGE_CACHE	A change cache debug message has been sent.	“EVT_DebugInfo”
186	EVT_DB_PURGE	A purge debug message has been sent.	“EVT_DebugInfo”
189	EVT_DB_CLIENT_BUFFERS	A client buffers debug message has been sent.	“EVT_DebugInfo”
190	EVT_DB_WANMAN	A WAN Traffic Manager debug message has been sent	“EVT_DebugInfo”
198	EVT_DB_DRL	A Distribute Reference Link (DRL) has been created.	“EVT_DebugInfo”
202	EVT_DB_ALLOC	A memory allocation debug message has been generated.	“EVT_DebugInfo”
204	EVT_DB_SERVER_PACKET	Not implemented.	“EVT_DebugInfo”
207	EVT_DB_OBIT	An obituary debug message has been generated.	“EVT_DebugInfo”
209	EVT_DB_SYNC_DETAIL	A synchronization detail debug message has been generated.	“EVT_DebugInfo”
210	EVT_DB_CONN_TRACE	A connection trace debug message has been generated.	“EVT_DebugInfo”
214	EVT_DB_DIRXML	A DirXML debug message has been sent.	“EVT_DebugInfo”
217	EVT_DB_DIRXML_DRIVERS	A DirXML Drivers debug message has been sent.	“EVT_DebugInfo”

	Event Type	Description	Data Returned
218	EVT_DB_NDSMON	A NDSMON debug message has been sent.	“EVT_DebugInfo”
220	EVT_DB_DNS	A DNS debug message has been sent.	“EVT_DebugInfo”
221	EVT_DB_REPAIR	A DS Repair debug message has been sent.	“EVT_DebugInfo”
222	EVT_DB_REPAIR_DEBUG	A Repair Debug debug message has been sent.	“EVT_DebugInfo”
225	EVT_DB_SCHEMA_DETAIL	A Schema Detail debug message has been sent.	“EVT_DebugInfo”
227	EVT_DB_IN_SYNC_DETAIL	A Sync Detail debug message has been sent.	“EVT_DebugInfo”
229	EVT_DB_SSL	An SSL debug message has been sent.	“EVT_DebugInfo”
230	EVT_DB_PKI	A PKI debug message has been sent.	“EVT_DebugInfo”
231	EVT_DB_HTTPSTK	A HTTPSTK debug message has been sent.	“EVT_DebugInfo”
232	EVT_DB_LDAPSTK	An LDAPSTK debug message has been sent.	“EVT_DebugInfo”
233	EVT_DB_NICIEXT	A NICI Ext debug message has been sent.	“EVT_DebugInfo”
234	EVT_DB_SECRET_STORE	A Secret Store debug message has been sent.	“EVT_DebugInfo”
235	EVT_DB_NMAS	A NMAS debug message has been sent.	“EVT_DebugInfo”
236	EVT_DB_BACKLINK_DETAIL	A Backlink Detail debug message has been sent.	“EVT_DebugInfo”
237	EVT_DB_DRL_DETAIL	A DRL debug message has been sent.	“EVT_DebugInfo”

	Event Type	Description	Data Returned
238	EVT_DB_OBJECT_PRODUCER	An Object Producer debug message has been sent.	"EVT_DebugInfo"
239	EVT_DB_SEARCH	A Search debug message has been sent.	"EVT_DebugInfo"
240	EVT_DB_SEARCH_DETAIL	A Search Detail debug message has been sent.	"EVT_DebugInfo"
242	EVT_DB_NPKI_API	An NPKI debug message has been sent.	"EVT_DebugInfo"

3.2.4 General DS Events

A large number of events are classified as general events. The meaning of the data returned is dependent on the type of the event.

For example, when a EVT_LOGIN (event type 100) occurs, the EVT_EventData struct contains several general data fields about the event (dstime, milliseconds, curProcess, and verb). The final two output parameters (intValues[], and strValues[]) contain information specific to the EVT_LOGIN event. The description of this information for each event is contained in the Data Returned column in the following table.

In the following example, each integer and string value from the Data Returned column is paired with its corresponding value in the EVT_EventData struct:

```
Integer Values
[0] (corresponds to) intValues[0]
[1] (corresponds to) intValues[1]
...

String Values:
[0] (corresponds to) strValues[0]
[1] (corresponds to) strValues[1]
...
```

The Data Returned column for the EVT_LOGIN event contains the following:

```
Integer Values
[0] 0 if a non-null password was used, 1 if a null password was used
[1] 0 if a bindery login was used, -1 if an NDS login was performed

String Values:
[0] DN of the parent of the entry that performed the login
[1] DN of the entry that performed the login
```

The data described by Integer Values [0] is contained in the first location in the intValues array, intValues[0]. If you wanted to determine if a non-null password was used during the login, intValues[0] would be checked to determine if it is 1 or 0 (with a value of 0 indicating that a non-null password was used).

Table 3-5 *General DS Events*

	Event Type	Description	Data Returned
53	EVT_AGENT_OPEN_LOCAL	The local Directory agent has been opened.	Integer Values: [0] end state of the open operation String Values: [0] end [1] start [2] audit
54	EVT_AGENT_CLOSE_LOCAL	The local Directory agent has been closed.	Integer Values: [0] the state of the operation String Values: [0] end [1] start
55	EVT_DS_ERR_VIA_BINDERY	An error was returned from the bindery.	Integer Values: [0] error code returned from the bindery
56	EVT_DSA_BAD_VERB	An incorrect verb number was given in a DSAgent request.	Integer Values: [0] bad verb number given to the DSA Request (NCP 104, 2)
57	EVT_DSA_REQUEST_START	A DSAgent request has been started.	Integer Values: [0] verb number (NCP 104, 2)
58	EVT_DSA_REQUEST_END	A DSAgent request has completed.	Integer Values: [0] verb number [1] primary ID [2] request size [3] reply size
59	EVT_MOVE_SUBTREE	A container and its subordinate objects have been moved.	String Values: [0] DN of source container [1] DN of destination container
60	EVT_NO_REPLICA_PTR	A replica exists that has no replica pointer associated with it.	String Values: [0] DN of associated partition object
61	EVT_SYNC_IN_END	Inbound synchronization has finished.	Integer Values: [0] number of entries sent String Values: [0] DN of the server entry associted with the server sending the changes [1] DN of root entry of the partition

	Event Type	Description	Data Returned
62	EVT_BKLINK_SEV	A backlink operation has updated an object's Security Equivalence Vector.	String Values: [0] DN of the updated object
63	EVT_BKLINK_OPERATOR	A backlink operation has changed an object's console operator privileges.	String Values: [0] DN of updated entry [1] DN of server entry for which the privleges were changed
64	EVT_DELETE_SUBTREE	A container and its subordinate objects have been deleted.	Integer Values: [0] number of entries deleted String Values: [0] DN of the root object of the deleted subtree
67	EVT_REFERRAL	A referral has been created.	Integer Values: [0] Referral type String Values: [0] DN of the partition [1] DN of the entry
68	EVT_UPDATE_CLASS_DEF	A schema class definition has been updated.	String Values: [0] Name of updated class
69	EVT_UPDATE_ATTR_DEF	A schema attribute definition has been updated.	String Values: [0] Name of updated or added attribute
70	EVT_LOST_ENTRY	eDirectory has encountered a lost entry. A lost entry is an entry for which updates are being received, but no entry exists on the local server.	Integer Values: [0] Seconds field of entry's timestamp [1] replicaNumber field of entry's timestamp [2] Event field of the entry's timestamp String Values: [0] DN of the entry's parent
71	EVT_PURGE_ENTRY_FAIL	A purge operation on an entry has failed.	String Values: [0] DN of the entry for which the purge operation failed
72	EVT_PURGE_START	A purge operation has started.	Integer Values: [0] Replica type String Values: [0] DN of the partition being purged

	Event Type	Description	Data Returned
73	EVT_PURGE_END	A purge operation has ended.	Integer Values: [0] Number of entries purged [1] Number of values purged String Values: [0] DN of the purged partition
76	EVT_LIMBER_DONE	A Limber operation has completed.	Integer Values: [0] 1 indicates all initialized, 0 indicates not all initialized [1] 1 indicates that a new RDN was found, 0 indicates that a new RDN was not found
77	EVT_SPLIT_DONE	A Split Partition operation has completed.	String Values: [0] DN of the parent partition's root [1] DN of the child partition's root
78	EVT_SYNC_SVR_OUT_START	Outbound synchronization has begun from a particular server.	Integer Values: [0] Replica number [1] Replica state, type, and flags String Values: [0] DN of the associated server entry [1] DN of the partition root
79	EVT_SYNC_SVR_OUT_END	Outbound synchronization from a particular server has finished.	Integer Values: [0] Number of objects sent [1] Number of values sent String Values: [0] DN of the associated server entry [1] DN of the partition root
80	EVT_SYNC_PART_START	Synchronization of a partition has begun.	Integer Values: [0] Partition state [1] Replication type String Values: [0] DN of associated partition entry

	Event Type	Description	Data Returned
81	EVT_SYNC_PART_END	Synchronization of a partition has finished.	Integer Values: [0] 1 indicates all processed, 0 indicates not all processed String Values: [0] DN of the associated partition entry
82	EVT_MOVE_TREE_START	A Move Subtree operation has started.	String Values: [0] DN of the root of the subtree to be moved [1] DN of the destination parent entry [2] DN of the server the operation is starting with
83	EVT_MOVE_TREE_END	A Move Subtree operation has finished.	String Values: [0] DN of the root of the moved subtree [1] DN o fthe server the operation started from
86	EVT_JOIN_DONE	A Join Partitions operation has completed.	String Values: [0] DN of the parent partition root entry [1] DN of the child partition root entry
87	EVT_PARTITION_LOCKED	A partition has been locked.	String Values: [0] DN of the locked partition
88	EVT_PARTITION_UNLOCKED	A partition has been unlocked.	String Values: [0] DN of the unlocked partition
89	EVT_SCHEMA_SYNC	The schema has been synchronized.	Integer Values: [0] 1 indicates all changes processed, 0 indicates not all changes processed
90	EVT_NAME_COLLISION	A name collision (two entries with the same name) has occurred.	String Values: [0] DN of the original entry [1] DN of the duplicate entry
91	EVT_NLM_LOADED	An NLM™ has been loaded.	Integer Values: [0] Module handle of the loaded NLM.
96	EVT_SERVER_RENAME	A server has been renamed.	String Values: [0] New server name

	Event Type	Description	Data Returned
97	EVT_SYNTHETIC_TIME	To bring eDirectory servers into synchronization, synthetic time has been invoked.	Integer Values: [0] Number of time stamps requested String Values: [0] DN of the root entry of the parition issuing the time stamp [1] DN of the partiton
99	EVT_DSA_READ	A Read operation has been performed on an entry.	String Values: [0] DN of read entry
100	EVT_LOGIN	A user has logged in.	Integer Values: [0] 0 if a non-null password was used, 1 if a null password was used [1] 0 if a bindery login was used, -1 if an eDirectory login was performed String Values: [0] DN of the parent [1] DN of the entry
101	EVT_CHGPASS	A user's password has changed.	String Values: [0] DN of the parent entry of the changed entry [1] DN of the entry whose password was changed
102	EVT_LOGOUT	A user has logged out.	String Values: [0] DN of the parent entry of entry that logged out [1] DN of the entry that logged out
103	EVT_ADD_REPLICA	A replica of a partition has been added to a server.	Integer Values: [0] Replica type String Values: [0] DN of the root entry of the partition [1] DN of the server entry [2] Name of the server
104	EVT_REMOVE_REPLICA	A replica of a partition has been removed from a server.	String Values: [0] DN of the root entry of the partition [1] DN of the server entry [2] Name of the server

	Event Type	Description	Data Returned
105	EVT_SPLIT_PARTITION	A partition has been split.	String Values: [0] DN of the root entry of the parent partition [1] DN of the root entry of the new partition [2] Name of the new partition entry
106	EVT_JOIN_PARTITIONS	A parent partition has been joined with a child partition.	String Values: [0] DN of the root entry of the parent partition [1] DN of the root entry of the child partition
107	EVT_CHANGE_REPLICA_TYPE	A partition replica’s type has been changed.	Integer Values: [0] Old replica type [1] New replica type String Values: [0] DN of the partition's root entry [1] DN of the target server's entry
108	EVT_REMOVE_ENTRY	An entry has been removed beneath a container.	String Values: [0] DN of the container entry [1] DN of the deleted entry
109	EVT_ABORT_PARTITION_OP	A partition operation has been aborted.	String Values: [0] DN of the partition's parent entry [1] DN of the partition entry
110	EVT_RECV_REPLICA_UPDATES	A replica has received an update during synchronization.	String Values: [0] DN of the replica's root entry
111	EVT_REPAIR_TIMESTAMPS	A replica’s time stamps have been repaired.	String Values: [0] DN of the replica's root entry
112	EVT_SEND_REPLICA_UPDATES	A replica has sent an update during synchronization.	String Values: [0] DN of the replica's root entry
113	EVT_VERIFY_PASS	A password has been verified.	String Values: [0] DN of the entry's parent [0] DN of the entry
114	EVT_BACKUP_ENTRY	An entry has been backed up.	String Values: [0] Backed-up entry's DN

	Event Type	Description	Data Returned
115	EVT_RESTORE_ENTRY	An entry has been restored.	String Values: [0] DN of the entry's parent [1] RDN of the entry
116	EVT_DEFINE_ATTR_DEF	An attribute definition has been added to the schema.	String Values: [0] Attribute's name
117	EVT_REMOVE_ATTR_DEF	An attribute definition has been removed from the schema.	String Values: [0] Attribute name
118	EVT_REMOVE_CLASS_DEF	A class definition has been removed from the schema.	String Values: [0] Class name
119	EVT_DEFINE_CLASS_DEF	A class definition has been added to the schema.	String Values: [0] Class name
120	EVT_MODIFY_CLASS_DEF	A class definition has been modified.	String Values: [0] Class name
121	EVT_RESET_DS_COUNTERS	The internal eDirectory counters have been reset.	String Values: [0] DN of the server entry
122	EVT_REMOVE_ENTRY_DIR	A file directory associated with an entry has been removed.	String Values: [0] DN of the entry's parent [1] DN of the entry
123	EVT_COMPARE_ATTR_VALUE	A Compare operation has been performed on an attribute.	String Values: [0] DN of the entry's parent [1] DN of the entry [2] Attribute name
124	EVT_STREAM	A stream attribute has been opened or closed.	Integer Values: [0] 0 if the stream was opened, 1 if the stream was closed [1] requested rights (only present if the stream was opened) String Values: [0] DN of the entry [1] Attribute name
125	EVT_LIST_SUBORDINATES	A List Subordinate Entries operation has been performed on a container object.	String Values: [0] DN of the entry's parent [1] DN of the entry
126	EVT_LIST_CONT_CLASSES	A List Containable Classes operation has been performed on an entry.	String Values: [0] DN of the entry's parent [1] DN of the entry
127	EVT_INSPECT_ENTRY	An Inspect Entry operation has been performed on an entry.	String Values: [0] DN of the entry's parent [1] DN of the entry

	Event Type	Description	Data Returned
128	EVT_RESEND_ENTRY	A Resend Entry operation has been performed on an entry.	String Values: [0] DN of the entry's parent [1] DN of the entry
129	EVT_MUTATE_ENTRY	A Mutate Entry operation has been performed on an entry.	String Values: [0] DN of the entry [1] OID of the new class [2] Name of the new class
130	EVT_MERGE_ENTRIES	Two entries have been merged.	String Values: [0] DN of the parent of the winner entry [1] DN of the winner entry [2] DN of the loser entry
131	EVT_MERGE_TREE	Two eDirectory trees have been merged.	String Values: [0] DN of the root entry
132	EVT_CREATE_SUBREF	A subordinate reference has been created.	String Values: [0] subordinate reference ID
133	EVT_LIST_PARTITIONS	A List Partitions operation has been performed.	String Values: [0] DN of the partitions root entry
134	EVT_READ_ATTR	An entry’s attributes have been read.	String Values: [0] DN of the entry [1] Attribute's name
135	EVT_READ_REFERENCES	The references on a given object have been read.	String Values: [0] DN of the entry
136	EVT_UPDATE_REPLICA	An Update Replica operation has been performed on a partition replica.	String Values: [0] DN of the partition's root entry [1] DN of the partition entry
137	EVT_START_UPDATE_REPLICA	A Start Update Replica operation has been performed on a partition replica.	String Values: [0] DN of the partition's root entry
138	EVT_END_UPDATE_REPLICA	An End Update Replica operation has been performed on a partition replica.	String Values: [0] DN of the partition's root entry
139	EVT_SYNC_PARTITION	A Synchronize Partition operation has been performed on a partition replica.	String Values: [0] DN of the partition's root entry

	Event Type	Description	Data Returned
141	EVT_CREATE_BACKLINK	A backlink has been created.	Integer Values: [0] Remote entry ID String Values: [0] DN of the server entry making the request [1] DN of the local entry
142	EVT_CHECK_CONSOLE_OPERATOR	An object has been checked for Console Operator rights.	Integer Values: [0] 0 if the entry does not have console rights, 1 if it does String Values: [0] DN of server entry [1] DN of entry being checked
143	EVT_CHANGE_TREE_NAME	The tree name has been changed.	String Values: [0] New tree name
144	EVT_START_JOIN	A Start Join operation has been performed.	String Values: [0] DN of the parent partition's root entry [1] DN of the child partition's root entry
145	EVT_ABORT_JOIN	A Join operation has been aborted.	String Values: [0] DN of the parent partition root [1] DN of the child partition root
146	EVT_UPDATE_SCHEMA	An Update Schema operation has been performed.	String Values: [0] DN of the server entry
147	EVT_START_UPDATE_SCHEMA	A Start Update Schema operation has been performed.	String Values: [0] Name of the Tree root [1] DN of the server entry
148	EVT_END_UPDATE_SCHEMA	An End Update Schema operation has been performed.	String Values: New: [0] Name of the Tree root [1] DN of the server entry

	Event Type	Description	Data Returned
149	EVT_MOVE_TREE	A Move Tree operation has been performed.	Integer Values: [0] Type of string. String Values: [0] DN of the source parent [1] DN of the destination parent [2] If integer[0] = 0, Source DN. If integer [0] = 1, new name.
150	EVT_RELOAD_DS	DS has been reloaded.	String Values: [0] DN of tree root.
151	EVT_ADD_PROPERTY	An attribute (property) has been added to an object.	Integer Values: [0] Security [1] Flags String Values: [0] DN of the entry
152	EVT_DELETE_PROPERTY	An attribute (property) has been removed from an object.	String Values: [0] DN of the entry
153	EVT_ADD_MEMBER	A member has been added to a Group object.	String Values: [0] DN of the group entry [1] DN of the new member [2] Attribute name
154	EVT_DELETE_MEMBER	A member has been deleted from a Group object.	String Values: [0] DN of the group entry [1] DN of the deleted entry [2] Attribute name
155	EVT_CHANGE_PROP_SECURITY	Security for a bindery object's property has been changed.	Integer Values: [0] New security String Values: [0] DN of the object [1] Property's name
156	EVT_CHANGE_OBJ_SECURITY	A bindery object's security has been changed.	Integer Values: [0] New security String Values: [0] DN of the object's parent [1] DN of the object
158	EVT_CONNECT_TO_ADDRESS	A connection has been established with a particular address.	Integer Values: [0] taskID [1] Address type [2] Address_size String Values: [0] Address

	Event Type	Description	Data Returned
159	EVT_SEARCH	A Search operation has been performed.	Integer Values: [0] Scope [1] Indicates the nodes to search [2] Info type String Values: [0] DN of the base entry
160	EVT_PARTITION_STATE_CHG	A partition's state has changed.	Integer Values: [0] Function [1] Type [2] State String Values: [0] DN of the partition's root entry [1] DN of the partner partition entry
161	EVT_REMOVE_BACKLINK	A backlink has been removed.	String Values: [0] DN of the affected object [1] DN of the affected server entry [2] DN of the remote server entry
162	EVT_LOW_LEVEL_JOIN	A low-level join has been performed.	String Values: [0] DN of the parent's root entry [1] DN of the child partition's root entry
164	EVT_CHANGE_SECURITY_EQUALS	An object's Security Equals attribute has been changed.	Integer Values: [0] 0=delete equivalence, 1=add equivalence String Values: [0] DN of the obect whos security has changed. [1] DN of the equivalent object.
167	EVT_CRC_FAILURE	A CRC failure has occurred when fragmented NCP requests were reconstructed.	Integer Values: [0] Failure type, 0=server, 1=client [1] Server/Client CRC error count
168	EVT_ADD_ENTRY	A new object has been added under a container object.	String Values: [0] Container entry's DN [1] Entry's DN

	Event Type	Description	Data Returned
169	EVT_MODIFY_ENTRY	An attribute has been modified on an object.	String Values: [0] Dn of the entry's parent entry [1] Entry's DN
178	EVT_MODIFY_RDN	A Modify RDN operation has been performed.	String Values: [0] Dn of the entry's parent entry [1] Entry's DN [2] Entry's former DN
181	EVT_ENTRYID_SWAP	A Swap Entry ID operation has been performed.	String Values: [0] DN of the source entry [1] DN of the destination entry
185	EVT_LOW_LEVEL_SPLIT	A low-level partition split has been performed.	String Values: [0] DN of the parent partition root entry [1] DN of the child partition root entry
188	EVT_ALLOW_LOGIN	A user has been allowed to log in.	Integer Values: [0] Flags String Values: [0] Entry's DN

3.2.5 Events Without Data

The following events do not have any associated data. When these events occur, the eventData field of the EventMonitorResponse is not present.

Table 3-6 *Events Without Data*

	Event	Description
9	EVT_SET_BINDERY_CONTEXT	
13	EVT_UPDATE_SEV	
94	EVT_LUMBER_DONE	Signals that a lumber operation has finished.
95	EVT_BACKLINK_PROC_DONE	Signals that a backlink process has finished.
98	EVT_SERVER_ADDRESS_CHANGE	Signals that a server address has changed.
140	EVT_SYNC_SCHEMA	Signals that the schema has been synchronized.
150	EVT_RELOAD_DS	Signals that eDirectory has been reloaded.
163	EVT_CREATE_NAMEBASE	Signals that a directory namebase has been created.
171	EVT_OPEN_BINDERY	Signals that the bindery has been opened.
172	EVT_CLOSE_BINDERY	Signals that the bindery has been closed.
174	EVT_NEW_SCHEMA_EPOCH	Signals that a new schema epoch has been declared.
182	EVT_INSIDE_NCP_REQUEST	
187	EVT_END_NAMEBASE_TRANSACTION	
213	EVT_BEGIN_NAMEBASE_TRANSACTION	

3.2.6 Bindery Events

The following table lists the events that are associated with bindery events:

Table 3-7 *Bindery Events*

	Event	Description
10	EVT_CREATE_BINDERY_OBJECT	Signals that a bindery object has been created.
11	EVT_DELETE_BINDERY_OBJECT	Signals that a bindery object has been deleted.

3.2.7 Security Equivalence Event

The security equivalence event that is indicated by the following eventType value:

Table 3-8 *Security Equivalence Event*

	Event	Description
12	EVT_CHECK_SEV	

3.2.8 Module State Event

The following table lists the event that is associated with module state events:

Table 3-9 *Module State Event*

	Event	Description
21	EVT_CHANGE_MODULE_STATE	

3.2.9 Network Address Events

The following table lists the events that are associated with network address events:

Table 3-10 *Network Address Events*

	Event	Description
17	EVT_REMOTE_SERVER_DOWN	
18	EVT_NCP_RETRY_EXPENDED	

3.2.10 Connection Change Events

The following table lists the events that are associated with connection change events:

Table 3-11 *Connection Change Events*

	Event	Description
173	EVT_CHANGE_CONN_STATE	Signals that the connection state has changed
212	EVT_COMPUTE_CONN_SEV_INLINE	

3.2.11 Change Server Address Event

The following table lists the event that is associated with change server address events:

Table 3-12 *Change Server Address Event*

	Event	Description
219	EVT_CHANGE_SERVER_ADDRS	

3.2.12 LDAP Events

The following table lists the events that are associated with LDAP Events.

Table 3-13 *LDAP Events*

	Event	Description	Data Returned
247	EVT_LDAP_BIND	Signals the occurrence of bind operation with the ldap server.	EVT_AuthEventData
249	EVT_LDAP_UNBIND	Signals the occurrence of unbind operation with the ldap server.	EVT_AuthEventData
250	EVT_LDAP_CONNECTION	Signals the occurrence of Connection operation with the ldap server.	EVT_ConnectionEventData
251	EVT_LDAP_SEARCH	Signals the occurrence of Search operation with the ldap server.	EVT_SearchEventData
253	EVT_LDAP_SEARCHENTRYRESPONSE	Signals the occurrence of Entry of Search operation with the ldap server.	EVT_SearchEntryResponse EventData
254	EVT_LDAP_ADD	Signals the occurrence of Add operation with the ldap server.	EVT_UpdateEventData
258	EVT_LDAP_MODIFY	Signals the occurrence of Modify operation with the ldap server.	EVT_UpdateEventData
260	EVT_LDAP_DELETE	Signals the occurrence of Delete operation with the ldap server.	EVT_UpdateEventData
256	EVT_LDAP_COMPARE	Signals the occurrence of Compare operation with the ldap server.	EVT_CompareEventData
262	EVT_LDAP_MODDN	Signals the occurrence of Modify DN operation with the ldap server.	EVT_ModDNEventData
264	EVT_LDAP_ABANDON	Signals the occurrence of Abandon operation with the ldap server.	EVT_AbandonEventData
265	EVT_LDAP_EXTOP	Signals the occurrence of Extension operation with the ldap server.	EVT_ExtOpEventData
266	EVT_LDAP_SYSEXTOP	Signals the occurrence of System extensioin operation with the ldap server.	EVT_SysExtOpEventData
268	EVT_LDAP_MODLDAPSERVER	Signals the occurrence of modification of the ldap server object.	

	Event	Description	Data Returned
269	EVT_LDAP_PASSWORDMODIFYTYPE	SIgnals the occurrence of Password Modify operation with the ldap server.	EVT_PasswordModifyEvent Data
270	EVT_LDAP_UNKNOWNOP	Signals the occurrence of Unknown operation.	EVT_UnknownEventData
248	EVT_LDAP_BINDRESPONSE	Signals the occurrence of the Response Event caused by the Bind Operation.	EVT_ResponseEventData
252	EVT_LDAP_SEARCHRESPONSE	Signals the occurrence of the Response Event caused by the Search Operation.	EVT_ResponseEventData
255	EVT_LDAP_ADDRESPONSE	Signals the occurrence of the Response Event caused by the Add Operation.	EVT_ResponseEventData
257	EVT_LDAP_COMPARERESPONSE	Signals the occurrence of the Response Event caused by the Compare Operation.	EVT_ResponseEventData
259	EVT_LDAP_MODIFYRESPONSE	Signals the occurrence of the Response Event caused by the Modify Operation.	EVT_ResponseEventData
261	EVT_LDAP_DELETERESPONSE	Signals the occurrence of the Response Event caused by the Delete Operation.	EVT_ResponseEventData
263	EVT_LDAP_MODDNRESPONSE	Signals the occurrence of the Response Event caused by the Modify DN Operation.	EVT_ResponseEventData
267	EVT_LDAP_EXTOP_RESPONSE	Signals the occurrence of the Response Event caused by the Extension Operation.	EVT_ResponseEventData

A Revision History

The following table lists all changes made to the LDAP and eDirectory documentation.

July 2008	Updated LDAP Events Table 3-13 on page 75.
March 2006	◆ Added some more LDAP Extensions under Section 1.6, “LDAP Extensions,” on page 14. ◆ Removed the ssl_any.c sample code link. ◆ Fixed formatting issues.
March 2005	Changed the component name to LDAP and eDirectory Integration.
June 2004	Added more information to “Syntax Definitions” on page 34 and on “X-NDS_ACL_TEMPLATES” on page 32.
February 2004	Made the following changes: ◆ Rectified the example for “!” in Table 1-7 on page 22. ◆ Added more information for EVT_CREATE_ENTRY under “Entry Events” on page 54. ◆ Renamed the product name from “NDS” to “Novell eDirectory” at relevant instances.
June 2003	Changed LDAP Event System to eDirectory Event System.
March 2003	Added Debug Events to the LDAP Event System.
May 2002	Added LDAP Event System.
June 2001	Added eCommerce Beans to Section 1.1, “Supported Platforms,” on page 7.
July 2000	Added LDAP operational attributes, OIDs for LDAP extensions, and other NDS eDirectory 8.5 information.
May 2000	Added names, OIDs, NDS version for all NDS syntax definitions.