CHAPTER 3

Novell eDirectory Management

This chapter covers the following testing objectives for *Novell Course 3000: Upgrading to NetWare 6*:

- Describe the Function and Features of eDirectory 8.6.
- Integrate eDirectory 8.6 into an Existing Network.
- Use the eDirectory Import/Export Wizard to Manage LDIF Files.
- Use Index Manager to Improve Directory Performance.
- Configure Replica Filters.
- Maintain eDirectory 8.6.

NetWare 6 introduces eDirectory 8.6, the greatest version to date of Novell's world-class directory service.

eDirectory is the world's leading directory service. It provides a unifying, cross-platform infrastructure for managing, securing, accessing, and developing all major components of your network. eDirectory scales to the largest network environments, including the Internet. Because it is based on the X.500 standard, eDirectory supports LDAP, HTTP, and the Java programming environment.

eDirectory can store and manage millions of objects in a seamless ballet of communications. It also provides the foundation network service for all NetWare servers and network resources. In fact, after network communications, it is the most fundamental network service offered by NetWare 6.

With all of this in mind, I'm sure you would agree that eDirectory management is one of your key responsibilities as a Novell CNE (Certified Novell Engineer). In this chapter, we will explore three important lessons regarding eDirectory management:

- *Understanding eDirectory 8.6*—First, we'll begin with a brief lesson in the architecture of eDirectory 8.6 and compare it to its predecessor, Novell Directory Services (NDS).
- *Maintaining eDirectory 8.6*—In the final lesson of this chapter, we will explore a comprehensive eDirectory maintenance plan to help ensure the health of your directory. First, you'll learn how to use the Index Manager capabilities of ConsoleOne to maintain eDirectory indexes and to increase the performance of your network. Then, you'll learn how to reduce synchronization traffic by configuring filtered replicas of specific eDirectory 8.6 partitions. Finally, you'll learn how to optimize eDirectory by configuring eDirectory cache.

As you can see, there's a lot to learn in this chapter and when it's all done, you'll be an accomplished eDirectory engineer. So, let's get started at the beginning—Understanding eDirectory 8.6.

Understanding eDirectory 8.6

Test Objective Covered:

1. Describe the function and features of eDirectory 8.6.

eDirectory 8.6 is a highly scalable, high-performing, secure directory service. Along with replication and partitioning capabilities, eDirectory provides the basic foundation for multiplatform networking. eDirectory also includes cryptography services to protect confidential data; it natively supports LDAP 3 over SSL (Secure Socket Layer).

Earlier versions of eDirectory were called Novell Directory Services (NDS). At first glance, eDirectory appears to have the same underlying architecture as NDS. That is, a distributed, object-oriented database organized as a hierarchical tree. Upon closer inspection, however, you'll find that eDirectory 8.6 is built on a much more sophisticated database structure than NDS.

Let's take a closer look at the underlying architectural differences between NDS and eDirectory, starting with NDS.

NDS Architecture

NDS was first introduced in NetWare 4. Prior to NetWare 4, NetWare operating systems relied on a server-centric model—in which each NetWare server had its own flat-file database for tracking network resources (called the *bindery*). The bindery consisted of three files: one that held object, one that held property, and one that held value information.

NDS offered a gigantic leap forward by evolving the server-centric model into a network-centric model. In this architecture (shown in Figure 3.1), the NetWare 4 operating system relies on four data files and multiple streams files located in a hidden directory on the server's SYS: volume. This database is referred to as the RECMAN database.

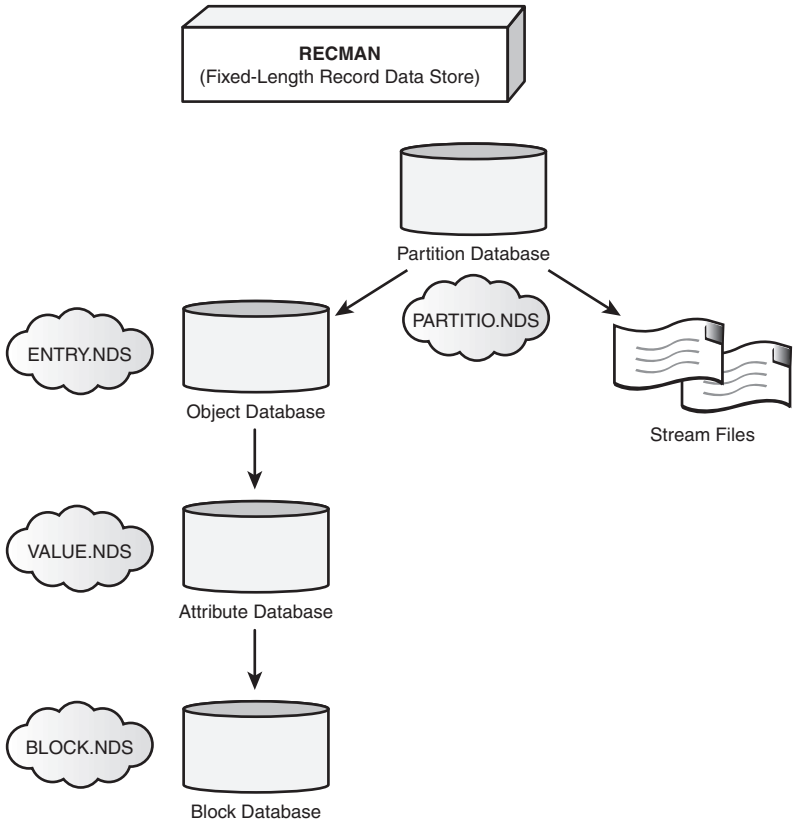


FIGURE 3.1
NDS architecture.

The four files that make up the NDS architecture in Figure 3.1 perform these functions:

- `PARTITIO.NDS`—The partition database contains a list of database partitions including system, schema, external reference, and bindery.
- `ENTRY.NDS`—The object database contains records for each object in a given server's replicas.
- `VALUE.NDS`—The attribute database contains property values for each object in `ENTRY.NDS`.
- `BLOCK.NDS`—The block database contains overflow data for the attribute database.

NDS *streams* files are named with hexadecimal characters (0-9, A-F) and hold information such as print job configurations and login scripts. Earlier versions of NDS used Novell's Transactional Tracking System (TTS) to ensure that database transactions were either completed or backed out in the event of a system failure.

TIP

The NetWare 5 version of NDS uses the same architecture as described above; however, the names of the files are different. In NetWare 5, `ENTRY.NDS` **is called** `0.DSD`**,** `VALUE.NDS` **is called** `1.DSD`**,** `BLOCK.NDS` **is called** `2.DSD`**, and** `PARTITIO.NDS` **is called** `3.DSD`**.**

eDirectory 8.6 Architecture

eDirectory 8.6 improves on NDS' fixed-length record data store model by introducing a highly scalable indexed database called *FLAIM* (FLexible and Adaptable Information Manager). The FLAIM database uses three different types of files instead of four, but still relies on streams files for print job configurations and login scripts. Check out the eDirectory 8.6 architecture in Figure 3.2.

Following is a description of each of the three different types of files that make up eDirectory's FLAIM database:

- `NDS.DB`—The control file is the centerpiece of the eDirectory architecture. This file contains the rollback log and is used to abort incomplete transactions.
- `NDS.01`—The primary database file contains all records and indexes found on a given server. When this data file reaches 2GB in size, `NDS.02` is created for the remaining data. New files are then created as

necessary to keep database files from growing beyond 2GB. This allows the database files to remain scalable while retaining their quick-search capabilities.

- `NDS*.LOG`—The transaction log file acts as a roll-forward log to reapply completed transactions that might not have been fully written to disk because of a system interruption.

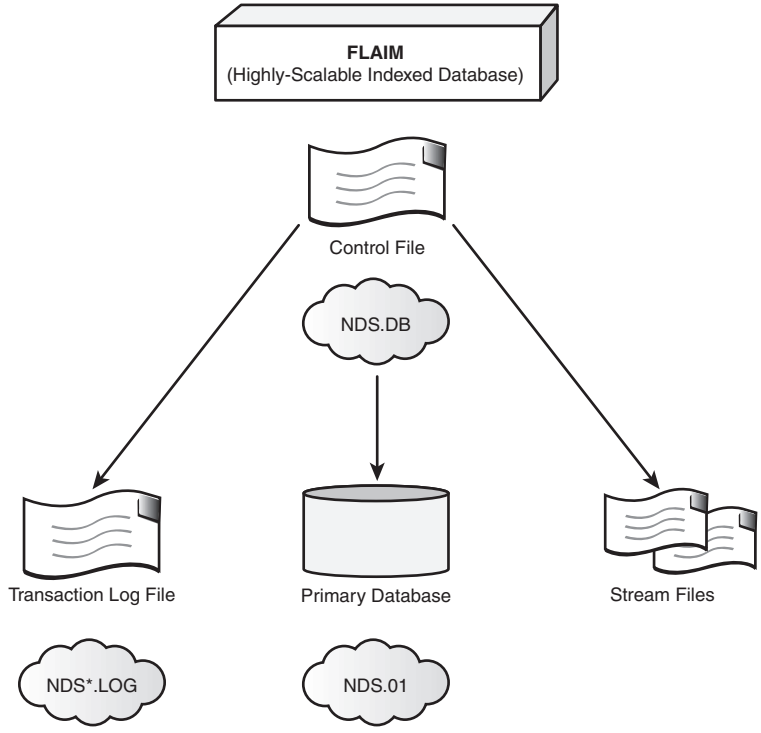


FIGURE 3.2
eDirectory 8.6 architecture.

eDirectory streams files perform the same function as they do in NDS and have an `.NDS` extension. However, unlike NDS, eDirectory does not use TTS; instead, it uses log files to back out and roll forward transactions in the event of a system failure. Refer to Table 3.1 for a summary of the differences between NDS and eDirectory architecture.

TIP

The primary eDirectory database file, `NDS.01`, includes a number of indexes to enhance performance. First, it includes attribute substring indexes for the "CN" and "uniqueID" fields. Second, it includes attribute indexes for the "Object Class" and "dc" fields. Finally, it includes attribute indexes for positioning that include strings beginning with CN, uniqueID, Given Name, and Surname.

TABLE 3.1 Comparing NDS and eDirectory Architecture

COMPONENT	NDS	EDIRECTORY
Database Name	RECMAN	FLAIM
Database Function	Fixed-Length Record Data Store	Highly Scalable Indexed Database
NetWare Version	4.*x* and 5.*x*	6.0
Number of Files	4	3
Data Records File	ENTRY.NDS	NDS.01
Roll-Back Mechanism	TTS	Log Files
Streams	Yes	Yes

The eDirectory architecture described here provides an exceptional foundation for all of eDirectory 8.6's new features and benefits. Following is a brief list of some of eDirectory's greatest new advancements:

- eDirectory 8.6 can be implemented on any of these operating system platforms: NetWare, Windows NT, Windows 2000, Linux, Solaris, and Tru64 UNIX. Client libraries and LDAP tools are available for Linux, Solaris, and Tru64 UNIX. LDAP support provides an open structure for integration with applications that are written to the LDAP standard.
- The Index Manager tool enables you to manage database indexes easily. The Filtered Replica Configuration Wizard enables you to easily create filtered replicas.
- The eDirectory Import/Export Wizard enables you to import or export LDIF files and to perform a server-to-server data migration.
- eDirectory includes a merge utility that enables you to merge one directory tree into another or to graft one tree onto another.
- iMonitor provides monitoring and diagnostic capabilities for all servers in your eDirectory tree from a web browser.

This completes our architectural lesson of eDirectory 8.6. We hope that you have gained an appreciation for the sophisticated directory services platform that eDirectory 8.6 provides for your NetWare 6 network. Now that you understand how it's built, you're ready to learn how to integrate it into your existing network.

Implementing eDirectory 8.6

Test Objectives Covered:

2. Integrate eDirectory 8.6 into an existing network.
3. Use the eDirectory Import/Export Wizard to manage LDIF files.

Now that you understand the fundamental architecture of the eDirectory tree, it's time to explore how it works. As you manage network objects within eDirectory, pay particular attention to its treelike structure. A well-designed tree will make resource access and management much easier. The structure of the eDirectory tree is both organizational and functional. The location of an object in the tree can affect how users access it and how network administrators manage it.

In this lesson, you will learn how to integrate eDirectory 8.6 objects in two simple steps:

- *Step 1: eDirectory Integration*—You must complete four tasks to prepare your network for eDirectory 8.6.
- *Step 2: eDirectory Import/Export Wizard*—You can use the eDirectory Import/Export Wizard to create large groups of eDirectory objects from existing LDAP databases.

Step 1: eDirectory Integration

When you install NetWare 6, eDirectory 8.6 is installed by default. If you upgrade to NetWare 6 from an existing network, however, you must carefully complete the following four tasks to prepare your network for eDirectory 8.6:

1. Apply the latest support packs.
2. Update the eDirectory schema.
3. Configure the Novell Certificate Server.
4. Perform an eDirectory health check.

Let's explore step 1 in more depth, staring with support packs.

Applying the Latest Support Packs

eDirectory 8.6 operates at the core of your network. Thus, you should ensure that the latest NetWare Support Packs have been installed on all of your NetWare servers before implementing eDirectory 8.6. These updates can be downloaded from the Novell Web site at `http://support.Novell.com`.

Updating the eDirectory Schema

eDirectory uses a mechanism called the *schema* to define the object naming structure for all network resources. The schema is distributed to all NetWare servers and follows specific rules. Think of the schema as the pulse of eDirectory 8.6.

Prior to installing NetWare 6 and updating your network to eDirectory 8.6, you must update your network's eDirectory schema. This is easily accomplished using NetWare Deployment Manager (which is located in the root of the *NetWare 6 Operating System* CD). As you recall from Chapter 2, NetWare Deployment Manager is a graphical tool that guides you through the steps required to ensure that all of your servers are using the latest version of the eDirectory schema. The good news is you only have to complete this procedure once!

Configuring the Novell Certificate Server

Prior to installing NetWare 6 and upgrading your network to eDirectory 8.6, you must configure the Novell Certificate Server.

The Novell Certificate Server allows you to mint, issue, and manage digital certificates from within eDirectory by using two key objects:

- *Security container object*—The Security container holds security-related objects for the eDirectory tree, including the Organizational CA object. This container physically resides at the very top of the eDirectory tree. The first server installed in eDirectory creates and stores the Security container.
- *Organizational CA object*—The Organizational CA object enables secure data transmissions. This object is stored inside the Security container and thus, also resides at the very top of the eDirectory tree. Only one Organizational CA object can exist in an eDirectory tree. Once this object is created, it should not be moved to another server. Deleting and re-creating an organizational CA will invalidate any certificates associated with it.

Make sure that the first eDirectory server is the most reliable one in the tree. This special server will host the Organizational CA object and must be operational during the installation of all other servers into the tree.

REAL WORLD

You must be running the latest version of the Novell Certificate Server in order to implement eDirectory 8.6. To upgrade your network, follow these simple steps:

1. *Identify the server that is acting as the organizational CA*—Use ConsoleOne to browse to your tree's Security container. Double-click the organizational CA and select the **Other** tab. The server acting as the CA is listed in the Host Server field.
2. *Verify that the CA server is running Novell Certificate Server 2.0 or later*—Move to the server that you identified in step 1. From the server console, execute `NWCONFIG`. Select **Product Options**, then **View/Configure/Remove Installed Products**. Finally, look for the PKIS entry to validate the version of Novell Certificate Server you are running.
3. *Verify that the necessary security-related objects exist in your Security container*—Inside the Security container, you should find the following three security-related objects: a KAP container object, a W0 security object within the KAP container, and an Organizational CA object. If these objects don't exist, the first NetWare 6 server will create them. The network administrator performing the installation, however, must have Supervisor rights in the Security container, as well as at the [Root] of the eDirectory tree.
4. *Establish the necessary eDirectory rights for operating the CA*—To properly administer the Novell Certificate Server, you must have Supervisor eDirectory rights to the W0 object and to the host server's container. In addition, you must have Read entry rights to the `NDSPKI:Private Key` attribute of the organizational CA.
5. *Download and install the client NICI on the ConsoleOne administrative workstation*—The Client NICI can be downloaded from the Novell Web site at `www.Novell.com/products/cryptograpy`.

After you have successfully accomplished these five tasks, updated the directory schema, and applied the latest support packs, your network is ready to accommodate eDirectory 8.6. Ready, set, go!

REAL WORLD

If you use the Novell Certificate Server 2.20 ConsoleOne Snap-In (which is included with NetWare 6), you will need to ensure that Client NICI 2.02 (or later) is installed on the ConsoleOne administrative workstation.

Performing an eDirectory Health Check

After you install eDirectory 8.6 on your new network, you should run a health check on each NetWare server to ensure that the integration was successful.

TIP

Regular health checks will help keep your directory running smoothly and make upgrades and troubleshooting much easier. In fact, one of the most frequent problems encountered by Novell Technical Support engineers is caused by network administrators who fail to run a health check on their eDirectory tree after a new server has been installed.

A complete health check begins with verifying the version of eDirectory that you are using. Every NetWare server on your network should be running the same version of `DS.NLM`. Next, you should check time synchronization because all object and property updates rely on consistent time stamps. Then, you should check partition continuity to ensure that all replicas of a partition are in sync. Finally, you should ensure that all NDS `SET` parameters are operating correctly.

Following are the detailed steps for the four most important eDirectory health checks, as well as a step-by-step guide to repairing the local database if anything goes wrong.

TIP

You must perform these health check procedures for every server in the eDirectory tree. You can start by performing the steps on the server holding the Master replica for each partition (starting with the Tree partition) and working down the Directory tree.

Time Synchronization Check

Start at the NetWare server holding the Master replica for the Tree partition. At the server console, execute `DSREPAIR`, and then select **Time Synchronization** to check the version of `DS.NLM` on each server synchronizing with this one. Also, verify that time stamps are properly synchronized.

Server-to-Server Synchronization Check

At the server console, enter the following `DSTRACE` commands to check server-to-server synchronization:

- `SET DSTRACE=ON`—Activates the trace screen for eDirectory transactions.
- `SET DSTRACE=+S`—Permits you to view the synchronization of objects.
- `SET DSTRACE=*H`—Initiates synchronization between servers.

Next press **Ctrl+Esc** and select **Directory Services** from the Current Screens list to view the Directory Services Trace screen. If there are no errors, a message will appear indicating that `All Processed=YES`. This message should appear for each partition on this server.

Replica Check

In `DSREPAIR`, you can perform four different health check procedures to ensure that replicas are synchronizing correctly. Follow these simple procedures:

- *Replica Synchronization*—Select **Report Synchronization Status** to view replica synchronization. A server must have a replica for this operation to work.
- *External References*—In the Advanced Options menu, select **Check External References**. This option shows external references, obituaries, and the states of all servers in the backlink list for the obituaries.
- *Replica State*—In the Advanced Options menu, select **Replica and Partition Operations**. Verify that the replica state is on.
- *Replica Ring*—In the Advanced Options menu, select **Replica and Partition Operations**. Then choose a particular partition and select **View Replica Ring**. Verify that the servers holding replicas of that partition are on and correct.

Obituaries are objects that are deleted from the tree and waiting to be purged. TIP

Schema Check

At the server console, enter the following `DSTRACE` commands to check the health of your eDirectory schema:

- `SET DSTRACE=ON`—Activates the trace screen for eDirectory transactions.
- `SET DSTRACE=+SCHEMA`—Displays schema information.
- `SET DSTRACE=*SS`—Initiates schema synchronization.

At the server console, press **Ctrl+Esc** and select **Directory Services** from the Current Screens list to view the Directory Services Trace screen. If there are no errors, a message will appear indicating that `All Processed=YES`.

Repair the Local Database

If you find errors in your eDirectory database after performing the health checks described above, you can attempt to repair the local database using `DSREPAIR`. This process may take a considerable amount of time and does lock the database during repair, so make sure that you perform the repair procedure after normal business hours.

In `DSREPAIR`,

1. Select the **Advanced Options** menu.
2. Choose **Repair Local DS Database**.
3. Mark the options on this page as follows:

   Check Local References—Yes

   Rebuild Operational Schema—Yes

   All Other Options—No

   This option locks the eDirectory database.
4. `DSREPAIR` displays a message stating that authentication cannot occur with this server when the eDirectory database is locked. Press **F10** and select **Yes**.
5. When the repair process is complete, exit `DSREPAIR`.

After you have completed all of the eDirectory health checks and repaired the local database, you're done. Now you can rest easy knowing that your eDirectory database is in the best possible condition it can be. And the good news is that you are ready to begin populating your tree with users, servers, containers, and other network objects.

Let's shift our focus to step 2 of eDirectory Implementation—the eDirectory Import/Export Wizard.

REAL WORLD

After you have completed all of the health check procedures described above, you will need to enter the following commands at the server console to turn off DSTRACE**:**

- `SET DSTRACE=nodebug`**—Erases all** DSTRACE SET **commands.**
- `SET DSTRACE=+min`**—Sets** DSTRACE **to minimum settings.**
- `SET DSTRACE=off`**—Turns off the** DSTRACE **screen.**

If left running, DSTRACE uses server resources that can slow down critical procedures. So when in doubt, turn it off.

Lab Exercise 3.1: Implement Novell eDirectory 8.6

In Chapter 2, you used the NetWare 6 migration process to move data from a NetWare 5.1 (source) server across the network to a new temporary (destination) NetWare 6 server. After the migration, the temporary NetWare 6 (destination) server then assumed the identity of the source server.

In this lab exercise, you will run the following types of tests to verify that the LABS-SRV1 server is operating properly after the migration:

- Part I: Verify that Time Synchronization Is Properly Configured
- Part II: Run a Health Check

In this lab exercise, you will need the following servers:

- LABS-SRV1 server created in Lab Exercise 2.2.
- WHITE-SRV1 server created in Lab Exercise 2.2.

Part I: Verify that Time Synchronization Is Properly Configured

Complete the following tasks:

1. Verify that the LABS-SRV1 server is configured as a Single Reference time provider.
   a. At the LABS-SRV1 server prompt, enter **`MONITOR`**.
   b. When the Available Options menu appears, select **Server Parameters**.

TIP **If you hesitate a too long when making your selection, you'll notice that the General Information window automatically expands, and in the process, hides the Available Options menu. If this occurs, simply press Tab to gain access to the Available Options menu.**

   c. When the Select a Parameter Category menu appears, select **Time**.

d. When the Time Parameters window appears

- Verify that the TIMESYNC Type is SINGLE.
- Verify that the Default Time Server Type is SINGLE.

e. Exit MONITOR.

Part II: Run a Health Check

Complete the following tasks:

1. Check server-to-server synchronization:

   a. At the LABS-SRV1 server console prompt, enter each of these commands:

   ```
   SET DSTRACE=ON
   SET DSTRACE=+S
   SET DSTRACE=*H
   ```

TIP

At the server console, you can press Alt+Esc to toggle between screens or Ctrl+Esc to display a list of active screens.

   b. Press **Ctrl+Esc**.

   c. When the Current Screens menu appears, select **Directory Services**.

   d. When the `DSTRACE` screen appears, review the information on the screen:

   - If no errors were found, skip to step 2.
   - If any errors were found, try reentering the following commands at the server console prompt:

     `SET DSTRACE=+S`
     `SET DSTRACE=*H`

     and then return to step 1b.

2. Check schema information:

   a. At the LABS-SRV1 server console prompt, enter these commands:

   `SET DSTRACE=+SCHEMA`
   `SET DSTRACE=*SS`

   b. Press **Ctrl+Esc**.

c. When the Current Screens menu appears, select **Directory Services**.

d. When the `DSTRACE` screen appears, verify that the following message is displayed: `All Processed = YES`.

3. Verify the `DS.NLM` version and check time synchronization:

   a. At the LABS-SRV1 server console prompt, enter **`DSREPAIR`**.

   b. When the Available Options menu appears, select **Time Synchronization**.

   c. When the View Log File (Last Entry): SYS:SYSTEM\DSREPAIR.LOG window appears:

      - Verify that the `DS.NLM` version is 10110.20 or later.
      - Verify that time is synchronized.

   d. Press **Esc** to return to the Available Options menu.

4. Check replica synchronization:

   a. When the Available Options menu appears, select **Report Synchronization Status**.

   b. When the View Log File (Last Entry): SYS:SYSTEM\DSREPAIR.LOG window appears, verify that the replicas on all servers are synchronized up to time for each partition.

   c. Press **Esc** to return to the Available Options menu.

5. Check external references:

   a. When the Available Options menu appears, select **Advanced Options Menu**.

   b. When the Advanced Options menu appears, select **Check External References**.

   c. When the View Log File (Last Entry): SYS:SYSTEM\DSREPAIR.LOG window appears, you'll notice that no external references were checked.

   d. Press **Esc** to return to the Advanced Options menu.

6. Check the replica state:
   a. When the Advanced Options menu appears, select **Replica and Partition Operations**.
   b. When the Replicas Stored on This Server window appears, verify that the Replica State is On for all partitions.
   c. Press **Esc** to return to the Advanced Options menu.

7. Check the replica ring:
   a. In the Advanced Options menu, select **Replica and Partition Operations**.
   b. When the Replicas Stored on This Server window appears, select the **[Root] partition**.
   c. When the Replica Options, Partition: .[Root]. menu appears, select **View Replica Ring**.
   d. When the Replicas of Partition .[Root]. window appears:
      - Verify that the servers holding replicas of this partition are correct.
      - Verify that the replica state of the [Root] partition is On.
   e. Press **Esc** three times to return to the Advanced Options menu.

8. Repair the local database:
   a. When the Advanced Options menu appears, select **Repair Local DS Database**.
   b. When the Repair Local Database Options window appears
      - In the Rebuild Operational Schema field, you'll notice there is a warning indicating that you should not enable this option unless directed by Technical Support. Change the value to Yes anyway. (To do so, press **Y**, and then press **Enter**.)
      - In the Repair All Local References field, verify that Yes is displayed.
      - Leave all other parameters on the page at their default settings.
      - Press **F10**.

c. When the Repair Directory menu appears

- Read the warning indicating that you have selected to lock the DB (DIB) database while the repair operating is running and that users will be prevented from logging in.
- Select **Yes** to continue.

d. Wait while the repair operation proceeds.

e. When prompted that the repair is complete:

- In the Total Errors field, note the number of errors. (It should be 0.)

NOTE **If errors were encountered, you may want to continue running Repair Local DS Database until no errors are displayed.**

- Press **Enter** to continue.

f. When the View the Current Log File menu appears, select **No**.

g. When the Repair Local Database Options window appears

- If errors were encountered in step 8e, press **F10** to repeat the repair process.
- If errors no were encountered in step 8e, exit `DSREPAIR`.

9. Turn off `DSTRACE`. At the server console prompt, enter these commands:

```
Set DSTRACE=nodebug
Set DSTRACE=+min
Set DSTRACE=off
```

Step 2: eDirectory Import/Export Wizard

Once your network is ready to accept eDirectory 8.6 objects, you can take advantage of Novell's new eDirectory Import/Export Wizard to create large batches of objects with the touch of a single button. The wizard uses the Novell Import/Conversion Export (ICE) engine installed with ConsoleOne. This engine allows you to convert LDAP Data Interchange Format (LDIF) files into eDirectory objects.

In this second eDirectory implementation lesson, you will learn how to use the eDirectory Import/Export Wizard to manage LDIF files. But, first, let's review the basics of LDAP and LDIF.

The NetWare 6 installation program copies two versions of the Novell Import/Conversion Export engine to your server automatically: a Win32 version (`ICE.EXE`) and a NetWare version (`ICE.NLM`). On Linux, Solaris, and Tru64 UNIX systems, ICE is included in the "NDSadmut1" package.

LDAP and LDIF Basics

LDAP and LDIF combine to create the directory access file format used by the ICE engine to create large groups of eDirectory objects with the touch of a single button.

LDAP is an Internet communications protocol based on the X.500 Directory Access Protocol (DAP). Fundamentally, LDAP allows client applications to access directory information running on a NetWare server. This is accomplished using an eDirectory service called LDAP Services for eDirectory, which is provided by `NLDAP.NLM`.

LDIF is a standard that defines an ASCII text file format that is used to exchange data between LDAP-compliant directories. LDIF files are commonly used to initially build a directory database or to add a large number of entries to a directory all at once. In this case, we are using LDIF files with the ICE engine to add a large number of network object entries to eDirectory with the touch of a single button.

So how do they work? LDIF files consist of one or more entries separated by a blank line. Each LDIF entry has an optional entry ID, a required distinguished name, one or more object classes, and multiple attribute definitions. You can specify object classes and attributes in any order.

Table 3.2 describes the LDIF fields used in the following example. This example accomplishes two tasks: it creates an Organization object named ACME, and then it creates a user named AEinstein in the ACME container.

```
dn: o=ACME
changetype: add
o: ACME
objectClass: organization
objectClass: ndsLoginProperties
objectClass: ndsContainerLoginProperties
objectClass: top
ACL: 2#entry#o=ACME#loginScript
ACL: 2#entry#o=ACME#printJobConfiguration

dn: cn=aeinstein,o=ACME
changetype: add
uid: aeinstein
otherGUID:: bsaWkLmDlk+Sdcy8z17PpA==
givenName: Albert
fullName: Albert Einstein
Language: ENGLISH
Title: Chief Scientist
sn: Einstein
ou: LABS
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: ndsLoginProperties
objectClass: top
ou: NORAD
cn: aeinstein
ACL: 2#subtree#cn=aeinstein,o=ACME#[All Attributes Rights]
ACL: 6#entry#cn=aeinstein,o=ACME#loginScript
ACL: 2#entry#[Public]#messageServer
ACL: 2#entry#[Root]#groupMembership
ACL: 6#entry#cn=aeinstein,o=ACME#printJobConfiguration
ACL: 2#entry#[Root]#networkAddress
```

LDIF Field Formats TABLE 3.2

PARAMETER	DESCRIPTION
Dn	Specifies the distinguished name for the entry.
changetype	Valid `changetype` values are `add`, `modify`, `moddn`, and `delete`.
objectClass	Specifies an object class to use with this entry. Each object class defines the types of attributes allowed or required for the entry.
attribute type	Specifies an attribute to define for the entry.
attribute value	Specifies a value to be assigned to the attribute type.

TIP

LDAP and eDirectory share a similar naming syntax. There are, however, two important differences when specifying object names in LDAP:

- **LDAP uses commas (,) as naming separators instead of periods (.)**
- **LDAP names always uses typeful full distinguished names**

Using the eDirectory Import/Export Wizard

The eDirectory Import/Export Wizard is a snap-in utility built into ConsoleOne. The wizard uses ICE as an import/export engine to manage a collection of handlers that read from or write to LDIF files. For example, to import LDIF data into an LDAP directory, ICE uses an LDIF source handler to read the LDIF file and an LDAP destination handler to send the data to the correct LDAP directory server.

NOTE

ICE replaces BULKLOAD and UIMPORT that were included with previous versions of eDirectory. ICE supports a command-line interface in addition to the Import/Export Wizard.

As you can see in Figure 3.3, the ConsoleOne Import/Export Wizard supports three different tasks:

- Import data from LDIF files to an LDAP directory
- Export data from an LDAP directory to an LDIF file
- Migrate data between LDAP servers

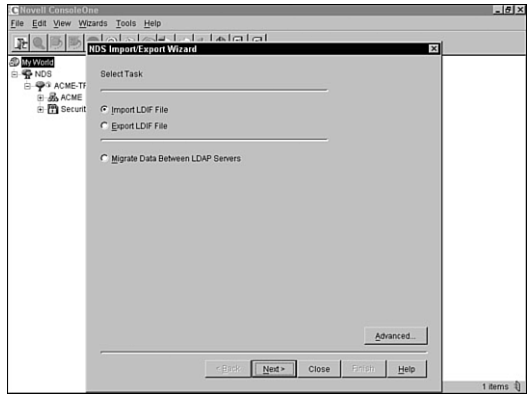


FIGURE 3.3
Using the eDirectory Import/Export Wizard in ConsoleOne.

Whether you are importing, exporting, or migrating LDIF data, the steps are nearly identical. Following is a step-by-step description of all three tasks and how to accomplish them by using the eDirectory Import/Export Wizard:

1. In ConsoleOne, select **Wizards**, and then select **NDS Import/Export.**
2. In the Select Task screen shown in Figure 3.3, choose **Import**, **Export**, or **Migrate**, depending on the task you want to accomplish.
3. Based on the task you chose in option 2, perform one of the following:
   a. *Import*—Enter the name of the LDIF file containing the data you want to import, select **Next**, and then specify the LDAP-complaint server where the data will be imported.
   b. *Export*—Specify the LDAP-compliant server holding the entries you want to export. Enter a DNS name or IP address.
   c. *Migrate*—Specify the LDAP-complaint server holding the entries you want to migrate. Enter a DNS name or IP address.
4. Regardless of the task you select, the wizard will ask you to fill out a form full of import/export options. Follow along in Table 3.3 as you complete the appropriate form. Select **Next** when you are done.
5. Based on the option you chose in step 2 above, perform the appropriate task below:
   a. *Import*—Click **Finish** to begin the LDIF import.
   b. *Export*—Specify the *search criteria* for the entries you want to export. These criteria include Base DN, Scope, Filter, and search Attributes. After you have specified the search criteria, select

Next and enter the name of the LDIF file that will store the exported information. Finally, select **Next** and **Finish** to begin the LDIF export.

c. *Migrate*—Specify the search criteria for the entries you want to migrate, and then select **Next** and choose an *LDAP server* where the data will be migrated. Finally, select **Next** and **Finish** to migrate the LDIF data.

eDirectory Import/Export Configuration Options **TABLE 3.3**

OPTION	DESCRIPTION
Server DNS Name/IP Address	Enter the DNS name or IP address of the source or destination LDAP server.
Server DNS Name/IP Address	Enter the DNS name or IP address of the source or destination LDAP server.
Port	Enter the integer port number of the source or destination LDAP server. By default, you can use the number "389" for clear-text or "636" for secure transmissions.
Login Method Guidelines	Select "Authenticated Login" or "Anonymous" for the entry specified in the User DN field.
User DN	If using Authenticated Login, enter the distinguished name of the entry that should be used when binding to the server.
Password	If using Authenticated Login, enter the password for the entry specified in the User DN field.
DER file	(optional) Enter the name of the DER file containing a server key used for SSL authentication. This field is required if you use Port 636 for secure communications. Of course, you can always use the default "RootCert.der" file created during installation in the SYS:\PUBLIC directory.

Using the LBURP Protocol

In addition to the standard synchronous protocol that ICE uses, you can also take advantage of the LDAP Bulk Update/Replication Protocol (LBURP). Excuse me.

LBURP allows ICE to send several update operations in a single request and to receive a response for all update operations in a single response. This asynchronous update processing guarantees that import/export requests are processed in the order specified and adds a tremendous amount of network efficiency to the overall system. LBURP lets ICE present data to the server as fast as the network connection will allow. In fact, if the network connection is fast enough, LBURP will keep the server busy processing update operations 100 percent of the time.

LBURP is enabled by default but you can disable it during an LDIF import by using the Advanced Options screen shown in Figure 3.4. To enable or disable LBURP during an LDIF import, select or deselect the **Use LBURP** option in Figure 3.4. You can find the Advanced Options screen by selecting the **Advanced** tab on the LDAP Server Selection screen.

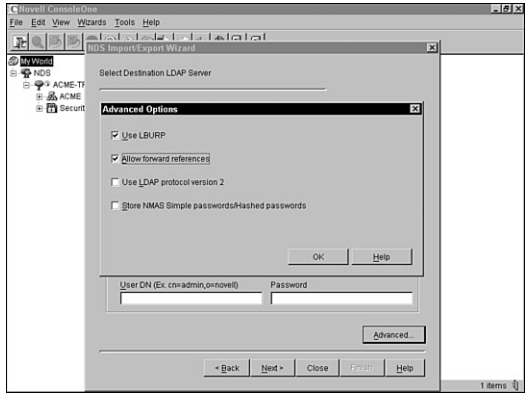


FIGURE 3.4
eDirectory Import/Export Wizard Advanced Options.

TIP

Because LBURP is relatively new, eDirectory servers prior to version 8.5 and most non-eDirectory LDAP servers do not support it. If you are using the eDirectory Import/Export Wizard to import an LDIF file to one of these servers, you must disable the LBURP option in order for the import to work.

This completes our comprehensive lesson in eDirectory 8.6 implementation. In this two-step process, you learned how to integrate eDirectory 8.6 into an existing network and import large groups of eDirectory objects using the eDirectory Import/Export Wizard. In step 1, you learned there are three important preintegration tasks that you must accomplish in order to prepare your network for eDirectory 8.6. In addition, you learned how to perform a variety of eDirectory health check procedures after your network has been updated. These procedures included a time synchronization check, server-to-server synchronization check, replica check, and schema check.

Once eDirectory 8.6 was in place, we shifted our attention to the eDirectory Import/Export Wizard. This wizard uses an import/export engine called ICE to manage directory entries in LDIF format. You learned how to use the eDirectory Import/Export Wizard to import data from LDIF files to an LDAP directory, export data from an LDAP directory, and perform a data migration between two LDAP servers.

Congratulations, you are now an eDirectory 8.6 pro! Now it's time to build a comprehensive maintenance plan. At this point, your attention shifts from building it to keeping it running.

Lab Exercise 3.2: Import Users with eDirectory Import/Export Wizard

In this lab exercise, you will learn to use the ConsoleOne eDirectory Import/Export Wizard to import LDIF files that are located on the Sams Publishing web site. You will then use these files to create two Organizational Unit containers in the ACME container and to add users to these containers by using the information in Table 3.4.

In this lab exercise, you will need the following servers:

- LABS-SRV1 server created in Lab Exercise 2.1.
- WHITE-SRV1 server created in Lab Exercise 2.2.

TABLE 3.4 LDIF Import File Information

FILE	RELATED INFORMATION
First LDIF file	Organizational Unit: Administrators
First LDIF file	Log File: `ADM-ICE.LOG`
First LDIF file	LDIF File: `ADM-LDIF.LDF`
Second LDIF file	Organizational Unit: Contractors
Second LDIF file	Log File: `CON-ICE.LOG`
Second LDIF file	LDIF file: `CON-LDIF.LDF`

Complete the following tasks:

1. At the WHITE-SRV1 server console prompt, execute **ConsoleOne**. If necessary, authenticate as admin.
2. Import the `ADM-LDIF.LDF` file.
   a. In ConsoleOne, browse to the ACME Organization object.
   b. Select **Wizards**, **NDS Import/Export**.
   c. When the Select Task dialog box appears:
      - Verify that Import LDIF File is selected.
      - Select **Advanced**.

d. When the Advanced Options dialog box appears

- In the Log File field, change the name of the log file to **ADM-ICE.LOG**.
- Select **Overwrite Existing Log File**.
- Select **OK**.

e. When the Select Task dialog box reappears, select **Next**.

f. When the Select Source LDIF file dialog box appears

- Browse to and select the `ADM-LDIF.LDF` file (that you downloaded from the Hungry Minds web site).
- Select **Advanced**.

g. When the Advanced Options dialog box appears, deselect **Exit on Error**; then select **OK**.

h. When the Select Source LDIF File dialog box reappears, select **Next**.

i. When the Select Destination LDAP Server dialog box appears, select **New**.

j. When the Add Server dialog box appears

- In the Description field, enter **`ACME Import`**.
- In the Server DNS Name/IP Address field, enter the *IP address of your server*. (If you're using the IP address listed in this book, enter 192.168.1.100.)
- In the Port field, enter **`389`**.
- In the User DN field, enter **`cn=admin,o=ACME`**.

Make sure you use a comma (`,`) after `cn=admin` instead of a period (`.`) because the use of a comma is an LDAP syntax rule. TIP

- Select **OK**.

k. When the Select Destination LDAP Server screen appears

- Select **ACME Import**.
- In the Password field, enter **acme**.
- Select **Advanced**.

l. When the Advanced Options dialog box appears, select **Allow Forward References**, and then select **OK**.

TIP

When working with LDIF, you may encounter a situation where an operation to add an entry precedes an operation to add its parent. If this occurs, an error is generated because the parent does not exist. This problem can be solved through the use of forward references. Under such a scenario, when an entry is created before its parent, a forward reference is created, which allows the entry to be created. If a subsequent operation creates the parent, the forward reference is converted to a normal entry.

m. When the Select Destination LDAP Server screen reappears, select **Next**.

n. When the summary window appears, select **Finish**.

o. You see text similar to the following:

```
Source Handler: ICE LDIF handler for Novell
eDirectory 8.6.0 version: 10110.05
Destination Handler: ICE LDAP handler for Novell
eDirectory 8.6.0 version: 10110.05
ICE log file: ADM-ICE.LOG
Start time: Friday, January 25, 2001 5:40:32 am
operation in progress .
Total entries processed: 24
Total number of errors: 0
End time: Friday, January 25, 2001 5:40:33 am Total
Time: 0:00:01.107
Time per entry: 00:00-044
```

p. Select **Close**.

q. Refresh your tree view by selecting **View**, **Refresh**.

3. Repeat step 2 using the `CON-LDIF.LDF` file. Also, make the following changes to the process:

   a. Log File: `CON-ICE.LOG`.

   b. Source File: `CON-LDIF.LDF`.

4. After both LDIF files are imported, make sure the following Organizational Unit containers appear in the ACME container:

   - Administrators
   - Contractors

Maintaining eDirectory 8.6

Test Objectives Covered:

4. Use Index Manager to improve directory performance.
5. Configure replica filters.
6. Maintain eDirectory 8.6.

As you learned at the beginning of the chapter, eDirectory is a distributed, replicated database. Therefore, it's imperative that you develop an eDirectory maintenance plan to help maintain the health of the directory. Your maintenance plan should accomplish two important goals: performance and reliability.

In this final eDirectory management lesson, you will learn some critical strategies for maintaining a healthy, high-performing eDirectory:

- *eDirectory indexes*—eDirectory gives you the ability to create trees with a very large number of objects. To maintain a high level of performance with a large tree, eDirectory records frequently requested information and stores it in indexes. In this lesson, you will learn how to use the Novell Index Manager to create, delete, and associate eDirectory indexes.
- *eDirectory filtered replicas*—eDirectory partitioning has many advantages because it enables you to separate the tree into smaller segments. You can also increase network fault tolerance by placing copies of other partitions on local servers. This process is known as *replication*. eDirectory 8.6 lets you configure filtered replicas of a partition so that you can limit synchronization across the network and reduce the overall size of the directory database. In this lesson, you learn what filtered replicas are and learn how to configure them with the Filtered Replica Configuration Wizard.
- *eDirectory cache*—The most significant setting that affects eDirectory performance is the *cache*. eDirectory 8.6 provides two default cache settings for controlling cache memory consumption. In this lesson, you learn how to configure cache limits by using `DSTRACE`.
- *eDirectory health checks*—You should run regular health checks to keep eDirectory running smoothly and to make upgrades and troubleshooting much easier. You should adjust the frequency of health checks as your environment changes. Some of the factors that influence the timing of your health checks include the number of

partitions and replicas in your Directory; the stability of replica holding servers; the amount of information in an eDirectory partition; the eDirectory object size; and the number of errors in previous `DSREPAIR` operations. For more information regarding specific eDirectory health check procedures, refer to the "eDirectory Integration" section in the first lesson of this chapter.

Now, let's create an eDirectory maintenance plan by using indexes, filtered replicas, and cache.

eDirectory Indexes

eDirectory relies on distributed indexes to maintain its scalability and high level of performance. An index is an attribute of a Server object and is stored in the Directory. As such, each index is unique to one server and is not shared by other servers in the eDirectory tree.

For example, suppose you have a tree with a single partition and three replicas. If you want to index the Telephone Number attribute of User objects in the tree, you would have to create an index on each server that holds a replica. Although indexes improve search performance, additional indexes can also add overhead to server operations. Therefore, avoid putting duplicate indexes on the same replica.

In this first component of the eDirectory maintenance plan, you will learn how to use eDirectory indexes and Predicate Statistics Data to increase the search performance of large directory trees.

Understanding Indexes and Predicate Statistics

eDirectory 8.6 includes a ConsoleOne utility called Index Manager that enables you to create, configure, and maintain eDirectory indexes. With Index Manager, you can view a variety of properties of each index, including the index name, state, type, rule, and specific attribute indexed.

The following index types are available in eDirectory 8.6:

- *User*—This type of index is user-defined and is the only type that can be added using Index Manager. It can be edited and deleted as well.
- *Auto Added*—eDirectory adds this type of index when certain attributes are created. It can also be edited and deleted. (See Table 3.5 for more information.)

- *Operational*—These indexes must be present to run the system. They cannot be edited or deleted. (See Table 3.5 for more information.)
- *System*—These indexes must be present to run the system. They also cannot be edited or deleted.

Table 3.5 shows the default indexes created during NetWare 6 installation. As you can see from the table, eDirectory automatically creates several Operational and Auto Added index types. As a NetWare 6 CNE, you should consider creating a number of User-type indexes to ensure an acceptable level of directory searching performance.

Default eDirectory Indexes **TABLE 3.5**

INDEX NAME	ATTRIBUTE INDEXED	TYPE
CN	cn	Auto Added
CN_SS	cn	Auto Added
dc	dc	Auto Added
Given Name	Given Name	Auto Added
Surname	Surname	Auto Added
uniqueID	uniqueID	Auto Added
uniqueID_SS	uniqueID	Auto Added
Aliased Object	Aliased Object Name	Operational
Obituary	Obituary	Operational
Member	Member	Operational
Reference	Reference	Operational
Equivalent to Me	Equivalent to Me	Operational

TIP

In earlier versions of NDS, administrators were advised to keep partitions limited to between 3,500 and 5,000 objects to maintain an acceptable level of searching performance. Because eDirectory has no such limit, indexing was implemented to provide fast access to a database that could potentially become very large.

eDirectory 8.6 includes an assessment tool to identify the types and quantities of Directory Services queries that are being requested by your server. This information is called Predicate Statistics Data. By using this information, you can create indexes for the most frequently requested information.

Now let's take a look at the step-by-step procedures for gathering Predicate Statistics Data, analyzing that data, and creating appropriate eDirectory indexes.

Configuring eDirectory Indexes

Creating an eDirectory index with the help of Predicate Statistics Data is a three-step process:

Step 1: Enable Predicate Statistics Data Gathering

Step 2: Check Predicate Statistics Data

Step 3: Create an eDirectory Index

Step 1: Enable Predicate Statistics Data Gathering

The index assessment tool built into eDirectory is based on the ndsPredicateStats object. Before you can analyze the Predicate Statistics Data and create an eDirectory index, you need to enable data gathering at the server level. In the property of your Server object, there is a Predicate Data tab that displays the frequency of Directory Service queries as they are recorded by the ndsPredicateStats object. See Figure 3.5 for more information.

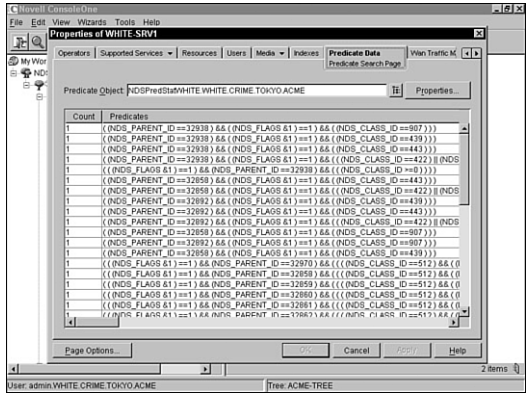


FIGURE 3.5
Viewing Predicate Data information at the Server Object.

To create an ndsPredicateStats object and associate it with a Server object, complete the following steps:

1. Open ConsoleOne and browse to the container where your Server object resides. Right-click the container and select **New**, **Object**.
2. In the New Object field, select **ndsPredicateStats**, and click **OK**.

3. In the Name field, enter an appropriate name. Make sure to incorporate the name of the server so you can track which object hosts the predicate statistics.
4. In the Server field, browse to and select the Server object you want to gather statistics for.
5. Select **Advanced** and mark the following three options: **Enable**, **Display Value Text**, and **Write to Disk**. Select **OK** twice and predicate statistics data gathering has been enabled.

Step 2: Check Predicate Statistics Data

Once the ndsPredicateStats object has been created and associated with your server, allow 24 hours to pass so that a reasonable amount of data can be recorded. During this time, the ndsPredicateStats assessment tool will gather enough directory services data to represent your typical network demands.

To check Predicate Statistics Data for your server, complete the following steps:

1. Activate ConsoleOne and right-click your Server object.
2. Select **Properties** and select the **Predicate Data** tab.
3. The Predicate Data statistics that the server gathered for the past 24 hours are displayed. (This is similar to the screen shown in Figure 3.5.) Identify the Object Class attributes that are queried most often on your network. These will become the foundation of your new eDirectory indexes.

When you have finished checking your Predicate Statistics Data, make sure to disassociate the ndsPredicateStats object from the server and to disable it. Otherwise, server performance will be impacted.

Step 3: Create an eDirectory Index

After you have gathered your server's Predicate Statistics Data and identified the most frequently queried attributes, it is time to create several eDirectory indexes to increase search performance.

To create an eDirectory index, complete these steps:

1. Activate ConsoleOne and right-click your Server object.
2. Select **Properties** and the **Indexes** tab. A complete list of already defined indexes for this server will appear. Initially, this list should match the list of default indexes shown in Table 3.5. Also note that the state of each index is Online.

3. To add a new index for an attribute that you identified using the ndsPredicateStats object and the Predicate Data tab in ConsoleOne, select **Add.**
4. The Create Index dialog box should appear (as shown in Figure 3.6). This dialog requires three pieces of information:
   a. *Index Name*—Usually matches the attribute.
   b. *Attribute*—Matches the attribute that you identified as having a high frequency in the Predicate Statistics Data step.
   c. *Rule*—Applies one of the following rules for the index: ***value*** (reports the presence of an attribute and its value), ***presence*** (reports the presence only of an attribute), or ***substring*** (matches a part of the larger, stored attribute string).

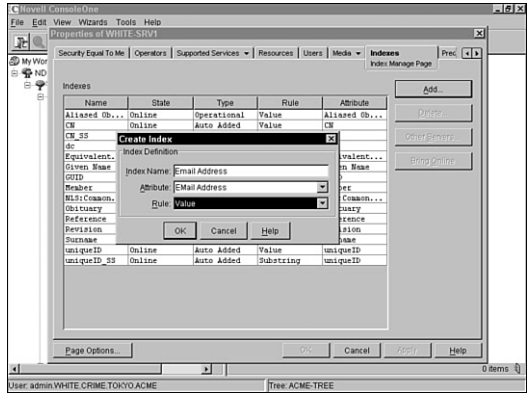


FIGURE 3.6
The Create Index dialog box in ConsoleOne.

5. After you have entered all the required information in the Create Index dialog box, select **OK** and your new index will be created. ConsoleOne will respond with a new Indexes list (as shown in Figure 3.7). Notice in the figure that the new index is italicized and has a state of New.

In this section, you have learned the three-step process for identifying frequently accessed attributes and creating an index for each of them. Remember that balance is the best policy when it comes to creating eDirectory indexes. It is possible to have too much of a good thing and your server performance will suffer.

Now let's shift our focus away from searching performance to eDirectory partitioning and replication.

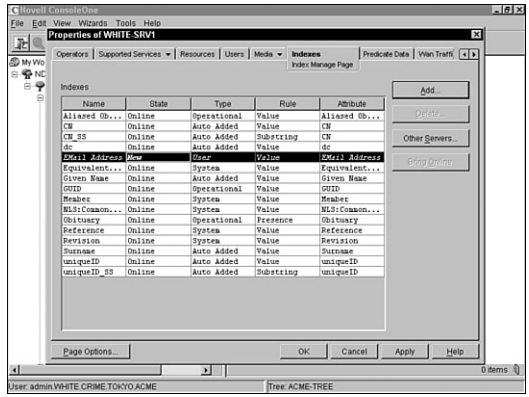


FIGURE 3.7
The Indexes tab in ConsoleOne.

Lab Exercise 3.3: Create Novell eDirectory Indexes

As you learned in this lesson, creating an eDirectory index is a three-step process:

Step 1: Enable Predicate Statistics Data gathering

Step 2: Check Predicate Statistics Data

Step 3: Create an eDirectory index

As the ACME administrator, you have decided that you want to create some eDirectory indexes on the WHITE-SRV1 server. In the first step, you have already created the ndsPredicateStats object and associated it with the WHITE-SRV1 server. After 24 hours, you analyzed the data and determined that the most actively used attribute is the Internet E-mail Address. Now it's time to create an eDirectory index for the Internet E-mail Address attribute.

In this exercise, you will learn how to use eDirectory data to populate your users' Netscape address books with names and e-mail addresses. To increase performance, you will use ConsoleOne to create an index on the WHITE-SRV1 server for the Internet E-mail Address attribute.

In this lab exercise, you'll need the WHITE-SRV1 server created in Lab Exercise 2.2.

Complete the following tasks:

1. At the WHITE-SRV1 server console prompt, execute **ConsoleOne**. If necessary, authenticate as admin.
2. When the ConsoleOne screen appears, navigate to the WHITE-SRV1 Server object. Right-click **WHITE-SRV1**, and then select **Properties**.
3. When the Properties of WHITE-SRV1 screen appears, navigate to the **Indexes** tab, and then select it.
4. When the Index Manage page appears, select **Add**.
5. When the Create Index dialog box appears
   - In the Index Name field, enter **E-mail Index**.
   - In the Attribute drop-down list, select **Internet EMail Address**.
   - In the Rule drop-down list, select **Value**.
   - Select **OK**.

6. When the Index Manage page reappears, note that the status of the E-mail Index is either New or Building.
7. To refresh the view, select the **Media** tab, and then reselect the **Indexes** tab.
8. When the Index Manage page reappears, verify that the status of the E-mail Index has changed to "Online," and then select **OK**.

eDirectory Filtered Replicas

eDirectory 8.6 includes a segmentation strategy known as partitioning. *Partitioning* breaks up your eDirectory tree into two or more logical divisions that can be separated and distributed. Copies of partitions can be distributed on multiple file servers in a strategy known as ***replication***. eDirectory replicas increase network performance (by decreasing the size of database files and by placing them closest to the users that need them) and increase fault tolerance (because extra copies of the database are distributed throughout the network).

eDirectory supports four types of full replicas:

- *Master*—The Master replica is the original read/write copy of a partition. The Master replica is created by default when you define the partition and contains a complete copy of the object data for the partition. Only 1 Master per partition, please.
- *Read/Write*—A Read/Write replica is a read/write copy of a partition. The Read/Write replica contains a complete copy of the object data for the partition. Each partition can have multiple Read/Write replicas.
- *Read-Only*—A Read-Only replica is a read-only copy of a partition. The Read-Only replica contains a complete copy of the object data for the partition. These replicas are only used for searching the eDirectory tree and viewing objects.
- *Subordinate References*—Subordinate References are a special type of replica that are created and maintained by eDirectory. They do not contain object data; they simply point to replicas that do.

Because eDirectory is a distributed, replicated database, NetWare servers continually share information and synchronize changes with each other. When you modify objects in a Read/Write or Master replica, those changes are propagated to all other replicas of the same partition. This process, known as ***replica synchronization***, creates background traffic over network communication lines. The amount of time required for a change to be replicated and synchronized depends on the type of change, the size of the partition, and the number of servers the partition is replicated on.

Fortunately, eDirectory 8.6 allows you to create ***filtered replicas*** that limit background synchronization traffic and improve overall network performance. In this lesson, we will explore the fundamentals of filtered replicas and explain how to create them using the Filtered Replica Configuration Wizard.

Understanding Filtered Replicas

Filtered Replicas are partial versions of full Read/Write or Read-Only replicas that limit synchronization to a specific set of attributes and/or values of selected objects or object classes. The filtered replica is a property of each Server object and can be customized at the server level. A filtered replica operates just like a traditional one, except for two important differences: 1) Filtered replicas are associated with Server objects, not stored on physical server machines, and 2) Filtered replicas can filter the data stored and updated in a server replica. A filtered replica can be changed back to a full replica at any time.

Filtered replicas offer three benefits. They reduce

- Synchronization traffic to the server by reducing the amount of data that must be replicated from other servers.
- The number of events that must be filtered by directory applications such as DirXML.
- The size of the directory database on your local server.

eDirectory synchronization is accomplished within a group of servers known as a replica ring. A *replica ring* is an internal system group that includes all servers that contain replicas of a given partition. Because filtered replicas support all versions of eDirectory, you can receive synchronization from any server in the replica ring. The benefits provided by filtered replicas, however, behave very differently, depending on which version of eDirectory your server is running. Following is a timeline of the two most popular eDirectory versions and how they synchronize filtered replicas:

- Filtered Replica synchronization before eDirectory 8.5
- Filtered Replica synchronization after eDirectory 8.5

Filtered Replica Synchronization Before eDirectory 8.5

Before eDirectory 8.5, a full replica of a given partition was stored by all servers in the replica ring. One server stored the Master replica while the other servers stored a Read/Write or Read-Only version of the Master. Changes made to Master or Read/Write replicas were broadcast to all other servers in the replica ring.

A server running a version of eDirectory before 8.5 will not recognize filtered replicas. It simply sends all changes in an update to the target server it is synchronizing with. If the target server (running eDirectory 8.5 or later)

has a filtered replica, only the filtered changes are made to the server's replica database. However, all data in the update is broadcast over the network. In fact, it defeats the network bandwidth efficiency that would normally be achieved using filtered replicas.

Filtered Replica Synchronization After eDirectory 8.5

Replica synchronization became a lot more sophisticated in eDirectory 8.5 with the dynamic recognition of filtered replicas. With eDirectory 8.5 or 8.6, the sending server reads the target server's replica filter and stores the filter information in memory. Then, the sending server uses this filter to send only the data allowed by the target server's replica filter. This way no unnecessary information is broadcast over the network. If you change the filtered replica on a target server, the target sends notification to the sending server and the sending server updates its filter information before sending the changes to the target. This is the dynamic aspect of eDirectory 8.5 and 8.6 synchronization.

Now that you understand the basic fundamentals of filtered replicas, it's time to learn how to configure them using the Filtered Replica Configuration Wizard.

TIP

Following are two examples of how you may apply Filtered Replicas to your eDirectory network:

- ***Remote sales office*—Suppose your company has a remote sales office with a single server. Salespeople from all over the company pass through the office and log in while they are there. Rather than placing full replicas of the entire eDirectory database on this server, you can create a set of filtered replicas on the server that contain only User objects from various partitions in the tree.**
- ***LDAP-compliant applications*—Novell's LDAP implementation requires that a replica containing each user be present on the local server to be available through LDAP. Rather than place a full replica of every partition containing users on the server running `NLDAP.NLM`, you could place filtered replicas containing only User objects.**

Using the Filtered Replica Configuration Wizard

ConsoleOne includes a Filtered Replica Configuration Wizard that enables you to define filters for specific attributes on specific servers for a set of eDirectory replicas. Before we explore the step-by-step procedure for setting up a server's replica filter and partition scope, let's discuss some important configuration guidelines:

- *Master partition replica*—The Master partition replica of a filtered replica must be hosted on a server running eDirectory 8.5 or later.
- *Scopes and filters*—The descriptions of a given server's partition, scope, and data filters are stored as attributes of the Server object and are managed using ConsoleOne.
- *Attribute subsets*—In addition to filtering by Object Class, you can also include only a subset of a specific object's attributes. For example, you could filter by Given Name, Surname, and Telephone Number attributes of the User object.
- *Filter modification*—A server's filter can be modified at any time, however, the operation generates a resynchronization of the replica and can consume server resources and take time to complete.
- *Filter limitation*—A server filter contains the set of eDirectory classes and attributes you want the server to host. Remember that you can only set up one filter per server. This means that any filter defined for a server applies to all filtered replicas on that server. However, it does not apply to full replicas.

Follow these steps to set up a server's replication filter and partition scope using the Filtered Replica Configuration Wizard:

1. Activate ConsoleOne and select **Wizards**. Next, choose **Filtered Replica Configuration.** The Filtered Replica Configuration Wizard will appear.
2. Select the *Server object* that will host the filtered replicas and choose **Next**.
3. To define the replication filter for this server, select **Define the Filter Set**. Next, choose **Edit Filter** and a list of available classes and attributes will appear. Refer to Figure 3.8.
4. Add the *classes and attributes* you want in the replica using the Select Filter checkboxes shown in Figure 3.8. Notice in the figure that object classes are listed in the left window with their corresponding attributes appearing in the right window. After you have configured the specific object classes and attributes for this replica, select **OK**.
5. Select **OK, Next**, and the Configuration Wizard will ask you to define the partition scope. The *partition scope* is the set of partitions that you want replicas of placed on this server. The Partition Scope screen in ConsoleOne provides an expandable view of the eDirectory partitions. You can select individual partitions, a set of partitions of a given

branch, or all partitions in the tree. To define the partition scope for this server, select **Define the Partition Scope** and click **OK** when you are finished.

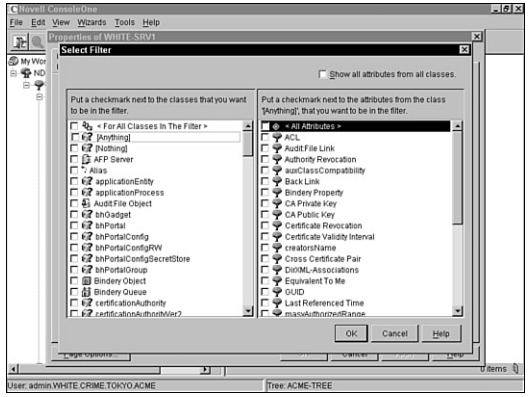


FIGURE 3.8
Editing filtered replicas in ConsoleOne.

6. After you have configured the filtered replica and defined the partition scope, you must change the replica's type. First, select the replica from the list and select **Change Replica Type.** Choose one of two types in the Select Replica Type dialog box: **Filtered Read/Write** *or* **Filtered Read-Only**. Select **OK** when you are finished.
7. Select **OK**, **Next**, and then **Finish** to complete the Filtered Replica Configuration Wizard. The replica you have just created now appears in the Partition and Replica view of ConsoleOne. See Figure 3.9.

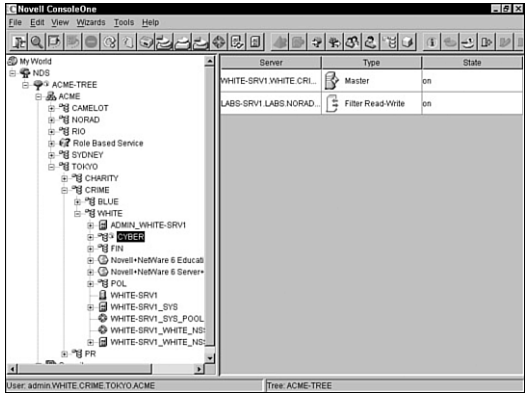


FIGURE 3.9
Viewing the new filtered replica in ConsoleOne.

You can appreciate the sophistication of this eDirectory tool as you work hard to optimize eDirectory synchronization and performance. Now let's complete our eDirectory 8.6 maintenance plan with a quick lesson in eDirectory caching.

Lab Exercise 3.4: Configure Filtered Replicas

In this exercise, you learn to use ConsoleOne to create partitions of the Organizational Unit containers you imported in Exercise 3.2. You then create filtered replicas of these partitions on the WHITE-SRV1 server.

In this lab exercise, you need the following servers:

- LABS-SRV1 server created in Lab Exercise 2.1.
- WHITE-SRV1 server created in Lab Exercise 2.2.

Complete the following tasks:

1. At the WHITE-SRV1 server console prompt, execute **ConsoleOne**. If necessary, authenticate as admin.
2. Create a partition for the Administrators container in ACME by performing the following tasks:
   a. Right-click the Administrators container; then select **Views**, **Partition and Replica View**.
   b. Select **Edit**, **Create Partition**.
   c. When the Create Partition window appears, select **OK**.
   d. When the Creating Partition window appears, select **Close**.
   e. Select **View**, **Refresh**. (You might need to select Refresh twice to view the results.)
3. Create a partition for the Contractors container in ACME by performing the following tasks:
   a. Right-click the Contractors container and then select **Views**, **Partition and Replica View**.
   b. Select **Edit**, **Create Partition**.
   c. When the Create Partition window appears, select **OK**.
   d. When the Creating Partition window appears, select **Close**.
   e. Select **View**, **Refresh**. (You might need to select Refresh twice to view the results.)
4. Use the Filtered Replica Configuration Wizard to create a filtered replica of the partition you created that contains administrator objects and associated attributes:

a. Select **Wizards**, **Filtered Replica Configuration**.

b. When the first Filtered Replica Configuration Wizard window appears, browse to and select the LABS-**SRV1** server (which should hold a read/write replica of each partition you created in steps 2 and 3), and then select **Next**.

c. When the next Filtered Replica Configuration Wizard window reappears, select **Define the Filter Set**.

d. When the Properties of LABS-SRV1 window appears, select **Edit Filter**.

e. When the Select Filter dialog box appears

- In the left column, select the **User** object class.
- In the Attributes column, select **All Attributes**.
- Select **OK**.

f. When the Properties of LABS-SRV1 window reappears, verify that Enable Local Login is selected, then select **OK**.

g. When the Filtered Replica Configuration Wizard window reappears, select **Next**.

h. When the next Filtered Replica Configuration Wizard window appears, select **Define the Partition Scope**.

i. When the Properties of LABS-SRV1 window appears

- In the ACME-TREE folder, select the **Administrators replica** you created in step 2.
- Select Change Replica Type.

j. Next, choose **Filtered Read/Write**; then select **OK**.

k. When the Properties of LABS-SRV1 window reappears, select **OK**.

l. When the Filtered Replica Configuration Wizard window reappears, select **Next**.

m. When the next Filtered Replica Configuration Window appears, select **Finish** to complete the operation of configuring the filtered replicas on LABS-SRV1.

5. Verify that the filtered replicas were created on server LAB-SRV1:

a. Select the **Administrators partition** you created in step 2.

b. Confirm that a filtered replica of the partition exists.

eDirectory Cache

The most significant setting that improves eDirectory performance is the *cache*. eDirectory 8.6 caches physical blocks from the server disk into file server memory using one of following two types of cache:

- *Block cache*—Block cache caches only physical blocks from the hard disk without any organization of the information contained in the block. This is the older cache type used by earlier versions of NDS. Block cache is most useful for update operations. It can also improve query performance by speeding up index searching.
- *Entry cache*—Entry cache is a new feature in eDirectory 8.6 that caches the logical directory structure of the eDirectory tree. By caching the logical structure of containers and objects, eDirectory can use this cache to quickly find and retrieve entries from memory. Entry cache is most useful for operations that browse the eDirectory tree by reading through entries (such as name resolution). Finally, entry cache can improve query performance by speeding up the retrieval of entries referenced from an index.

Although there is some redundancy between these two eDirectory cache types, each cache boosts performance for different types of operations. Earlier versions of eDirectory created multiple versions of blocks and entries in its cache for transaction integrity. eDirectory 8.5 (and prior) did *not* remove these blocks and entries from the cache when they were no longer needed. This caused considerable overhead in cache memory consumption. Fortunately, eDirectory 8.6 includes a background process which periodically browses the cache and cleans out older versions. The default browsing interval is 15 seconds.

The total available memory for eDirectory caching is shared between these two cache types. The default is an equal division. The more blocks and entries that are cached, the better your overall Directory performance. The ideal strategy is to cache the entire database in both the entry and block caches (although this is impossible for large trees). In general, you should try to achieve a 1:1 ratio of block cache to eDirectory database size and a 1:2 or 1:4 ratio for entry cache.

eDirectory 8.6 provides two default cache settings for controlling cache memory consumption:

- *Dynamically Adjusting Limit*—The dynamically adjusting limit causes eDirectory to periodically adjust its memory consumption in response to the needs of the network. In this option, you specify the limit as a percentage of available physical memory and eDirectory recalculates a new memory limit at fixed intervals. The new memory limit is the percentage of physical memory available at the time. Along with the percentage, you can set a maximum and minimum threshold as either the number of bytes to use or the number of bytes to leave available. The minimum threshold default is 16MB (8MB for entry cache and 8MB for block cache). The maximum threshold default is 4GB. With the dynamically adjusting limit, you can also specify the interval length (15 seconds by default). The shorter the interval, the more memory consumption is based on current conditions. However, shorter intervals are not necessarily better because the percentage recalculation will create more memory allocation and freeing.
- *Fixed Memory Limit*—The fixed memory limit is the method used by earlier versions of NDS. In this cache memory consumption method, you can set a fixed memory limit in one of the following ways: ***fixed number of bytes*** (this is a set number of bytes assigned to the memory limit), ***percentage of physical memory*** (the percentage of physical memory at the interval becomes a fixed number of bytes), or ***percentage of available physical memory*** (the percentage of available physical memory at the interval becomes a fixed number of bytes).

You can use either of these two methods for controlling cache memory consumption but you cannot use them at the same time because they are mutually exclusive. The last method used always replaces prior settings. If the server you are installing into the tree does not have a replica, the default fixed memory limit is 16MB (with an even split between block and entry cache). However, if your server does contain a replica, the default memory is a dynamically adjusting limit of 51 percent of available server memory with a minimum threshold of 8MB per cache and a maximum threshold of keeping 24MB server memory available for other processes.

TIP

If the minimum and maximum threshold limits are incompatible when using the dynamically adjusting limit method, the minimum threshold limit is followed.

When using eDirectory on NetWare, you can use the `DSTRACE` utility at the server console to configure dynamically adjusting or fixed memory limits. You do not need to restart the server for the changes to take effect.

To set a fixed hard limit, enter the following DSTRACE command at the server console:

```
SET DSTRACE=!MB[memory in bytes]
```

An example of setting a fixed hard limit of 8MB:

```
SET DSTRACE=!MB8388608
```

To set a calculated limit, enter the following DSTRACE command at the server console:

```
SET DSTRACE=!MHARD,%:
[percent], MIN:[bytes],MAX:[bytes],LEAVE:[bytes],NOSAVE
```

An example of a calculated limit of 75 percent of total physical memory with a minimum of 16MB and an option indicating that these options should not be saved to the startup file is

```
SET DSTRACE=!MHARD,%:75,MIN:16777216,NOSAVE
```

To set a dynamically adjusting limit, enter the following DSTRACE command at the server console:

```
SET DSTRACE=!MDYN,%:
[percent], MIN:[bytes],MAX:[bytes],LEAVE:[bytes],NOSAVE
```

An example of a dynamically adjusted limit of 75 percent of available memory with a minimum of 8MB is

```
SET DSTRACE=!MDYN,%:75,MIN:8388608
```

This completes our detailed lesson in configuring the eDirectory cache to tune database performance. As you have learned, this is the most significant setting that can be configured to improve eDirectory performance. In this lesson, you learned about the benefits of block and entry cache and as well as how to distribute memory between these two cache types. In addition, you learned about two different default cache settings (dynamically adjusting and fixed memory) for controlling cache memory consumption. Now that's what I call eDirectory fun.

Congratulations! Your eDirectory database has been integrated, maintained, and optimized. This completes our lesson in eDirectory 8.6 management. In this chapter, you learned four procedures for eDirectory integration and used the eDirectory Import/Export Wizard to add large batches of objects to

the eDirectory tree using LDIF files. We then turned our attention to eDirectory maintenance and learned how to improve eDirectory performance using indexes, filtered replicas, and cache.

Now it's time to move beyond the NetWare 6 directory into a chapter full of NetWare 6 advanced administration tasks, including: managing NetWare 6 remotely, configuring NetWare 6 DNS/DHCP, configuring NetWare 6 to use SLPv2, and using NetWare 6 multitasking, multithreading, and multiprocessing.

Ready, set, take off!

Lab Exercise 3.5: Understanding Novell's Newest eDirectory (Word Search Puzzle)

Circle the 20 Novell eDirectory terms hidden in this word search puzzle using the hints provided.

S	T	R	E	A	M	S	F	D	U	I	Z	G	T	Q
M	C	T	U	A	T	L	D	I	F	Q	W	J	M	A
E	X	H	S	Y	S	T	E	M	L	B	U	R	P	U
N	D	T	E	S	G	I	C	H	Q	T	S	R	X	T
T	E	R	R	M	B	J	Y	I	C	D	E	N	O	O
R	T	H	B	J	A	N	X	P	K	A	O	R	C	A
Y	E	U	G	B	L	O	C	K	C	A	C	H	E	D
C	X	P	A	R	T	I	T	I	O	N	I	N	G	D
A	S	F	L	K	L	O	S	E	S	E	S	W	K	E
C	Y	X	P	I	T	K	F	F	U	V	O	Q	K	D
H	Y	T	H	R	C	Q	N	Y	V	N	S	V	D	M
E	B	P	T	S	V	A	E	W	O	C	F	S	K	V
K	W	R	T	P	O	J	T	A	G	S	Q	E	J	D
Q	R	H	B	V	L	G	Q	I	D	Q	O	G	U	R
S	E	C	S	D	S	O	D	M	O	F	N	H	J	P
J	U	W	D	U	R	T	F	V	B	N	N	R	F	U

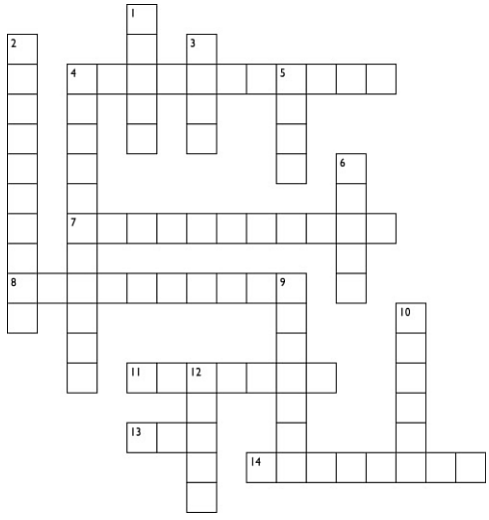
Hints

1. Index automatically created by eDirectory when certain attributes are created.
2. Older type of Directory cache that caches only physical blocks from the hard disk without any organization of the information contained in the block.
3. Component that has the most significant impact on eDirectory performance.
4. New type of cache found in eDirectory 8.6 that caches the logical directory structure of the eDirectory tree.
5. Type of replica that can be used to limit background synchronization traffic and improve overall network performance.

6. Allows ICE to send several update operations in a single request and receive a response for all update operations in a single response.
7. Text file format used to exchange data between LDAP-compliant directories.
8. Type of replica that is automatically created when a partition is defined.
9. Segmentation strategy that allows you to break up your eDirectory tree into two or more logical divisions that can be separated and distributed.
10. Public key encryption system used by Novell Certificate Server.
11. Increases network fault tolerance by placing copies of eDirectory partitions on local servers.
12. Defines the eDirectory naming structure for all network resources.
13. eDirectory files that hold information such as print job configurations and login scripts.
14. Type of required index that cannot be edited or deleted.
15. Only type of index that can be added using Index Manager.

See Appendix C for answers.

Lab Exercise 3.6: Novell eDirectory Management (Crossword Puzzle)



Across

4. eDirectory can support billions of objects

7. Servers holding replicas of a partition

8. NDS is distributed and _____

11. Predecessor to NDS

13. Predecessor to eDirectory

14. Least popular replica type

Down

1. eDirectory's highly-scalable, indexed database core

2. Secure, high-performance NetWare 6 Directory service

3. Programming language supported by eDirectory

4. Collection of patches and fixes

5. Directory access protocol supported by eDirectory

6. Main NDS file
9. Used to configure cache limits
10. NDS database core
12. Main eDirectory control file

See Appendix C for answers.