Solaris™ Operating Environment LDAP Capacity Planning and Performance Tuning

Steve Lopez—Enterprise Engineering

Sun BluePrints™ OnLine—May 2002

`http://www.sun.com/blueprints`

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A
650-960-1300 fax 650-969-9131

Part No. 816-4829-10
Revision 1.0, 05/10/02
Edition: May 2002

Please
Recycle

Adobe PostScript™

Solaris Operating Environment LDAP Capacity Planning and Performance Tuning

This article details methodologies for capacity planning of the Lightweight Directory Access Protocol (LDAP) naming service on the Solaris™ Operating Environment (Solaris OE). This article also presents performance tuning examples and exercises to assist experienced system administrators in increasing the performance and scalability of LDAP server deployments.

Note – Several excerpts from the Sun BluePrints™ book, *Solaris and LDAP Naming Services* by Tom Bialaski and Michael Haines are copied verbatim in this article to convey a clearer understanding of issues surrounding LDAP server tuning and capacity planning.

This article is divided into the following sections:

- Considering server sizing
- Considering directory aspects
- Allocating space for log files
- Performing directory database backups
- Determining memory size
- Estimating CPU usage
- Tuning for performance
- Indexing
- Caching for performance
- Selecting search limit parameters
- Removing unnecessary plug-ins
- Summary of Key Practices

About the Author

Steve Lopez has been an UNIX engineer for 18 years (14 with Sun Microsystems, Inc.) in positions of System Support Engineer, Systems Engineer, Educational Instructor, and Enterprise Engineer. Steve has written Sun Expert Lab Guides on Sun™ Cluster 2.2 software, Sun™ Cluster 3.0 software, and the Netscape Directory Server 4.1.2 (LDAP).

Steve is currently architecting and implementing a new paradigm shift in technology information exchange with the creation of remote laboratory environments for SunU, Sun ITT, and Sun Education.

Considering Server Sizing

You must consider several factors before determining the optimum server configuration:

- Size of directory
- Frequency of directory access
- Type of directory access
- Security requirements
- Replication strategy
- Number of concurrent connections

Knowing the impact of these factors helps determine which hardware components are required to provide optimum performance. These hardware components include:

- Physical memory
- CPU
- Disk storage
- Network adapters

Failure to properly size any one of these hardware components can cause a bottleneck that degrades the total system performance no matter how well the other components are sized.

In general, enough memory is needed to prevent swapping and you want fast disk storage devices that can handle the expected I/O throughput. Network adapters are rarely a factor because of the high data rates they can sustain, but you still must be aware of the network bandwidth consumption.

Considering Directory Aspects

Directory considerations include:

- Directory size
- Directory access
- Security requirements
- Replication strategy

Directory Size

The size of the directory affects both the amount of disk storage and physical memory required for a system. It is important to choose sufficient disk storage to accommodate the current directory size along with space for expansion.

Calculating how much space is required for the directory entries is a straightforward process. The average size of an entry is determined, then it is multiplied by the number of entries. However, in addition to the directory entries are the index databases used to increase search performance. The size of these databases can vary depending on the number of attributes indexed and the type of indexing employed.

Key Practices: Optimum directory performance is obtained when all the directory entries and the directory indexes fit into physical memory. The most important factor that determines directory performance is the availability of free memory.

Directory Access

The method, used to access directory data is accessed has a profound effect on performance. The Netscape Directory Server is optimized for search speeds. It performs well when the bulk of requests are searches. If the directory data fits into physical memory, search performance is limited only by the speed of the CPU and memory bandwidth.

Write performance is substantially less than read performance and has a greater impact on overall performance. Additionally, there is a trade-off between tuning for read and write performance. Read or search performance is greatly enhanced by indexing the directory data based on what type of searches are likely to occur. However, as the number of indexes increases, so does the time it takes to update them.

Another important factor that affects performance is how the LDAP client establishes a connection to the server. If the client is an LDAP-enabled application, then the same connection can be used for all client accesses. This technique greatly increases performance. If a new connection is created each time a client, such as a web browser, accesses the directory server, a significant load is placed on the server. These two types of connections are commonly referred to as persistent and nonpersistent.

Note – The more connections that a directory server has to maintain, the greater the load on the system resources.

Security Requirements

The level of security the directory server operates has a major effect on performance. The implementation of access control affects directory performance because more security checking is required. Transmission to and from the directory server can be in clear or encrypted text. The directory server supports encryption based on secure socket encryption (SSE), that provides greater security at the *expense* of performance.

Authentication is another aspect of security that can be performed simply by entry of a name and password or by presentation of a client certificate through the Secure Socket Layer (SSL) Protocol. The public key exchange, which is an integral part of SSL and the encryption algorithm used to protect data, are CPU-intensive and have a significant impact on performance.

Replication Strategy

Replication is a handy tool for increasing search performance by load distribution across more than one server. However, there is overhead that is associated with replication. The replication overhead consists of sending the data to two sites instead of just one site. Changes to the directory database need to be recorded in a change log, then pushed out to the replicas. Therefore, disk write speed and the network bandwidth are important due to the time it takes to send the data to two sites. In addition, CPU cycles are used during the transfer of data from the supplier to the consumer.

Allocating Space for Log Files

You need to allocate adequate disk space for log files. These log files include:

- Transaction log
- Access Log
- Error log
- Audit log
- Change log

TABLE 1 on page 6 offers log file size estimates.

Transaction Log

Transaction logs are a way of maintaining database consistency even when the server crashes. Before each write operation is performed on the database, an entry is written to the transaction log. The transaction log is relatively small but is continually updated even for search operations for efficiency and accuracy.

Key Practices: A storage device with a write cache is a benefit for transaction log activity for busy servers. The Sun StorEdge™ D1000 array is a good choice.

Key Practices: Since the write operation to the transaction log happens at about the same time as the write operation to the database, it is preferable to have the transaction log on a separate disk volume for better performance.

Access Log

The access log keeps track of who accesses the directory and what type of access is performed. The information contained in access logs is valuable in determining what type of searches are performed most often providing you with the information of about what indexes to optimize. A typical entry in the access log is about 70 bytes. Therefore, a million searches would generate a 70-Mbyte file. Based on how active the directory is and how much data you want to collect, you can determine how much space to allocate for the access log. Specifying a size limit prevents this file from growing too large.

Note – Writes to the access log are buffered, so there is little performance penalty with access logging turned on. The only consideration is the amount of space required, as long as it is not on the same volume as the directory database or transaction log. If the access log was on the same volume and there was disk failure, it would render the data unusable.

Error Log

The error log records change in status activity and errors. The information stored here is useful for troubleshooting problems but doesn't need to be kept for a long period of time; six months is long enough. Unless the directory server has a serious problem, this file does not grow very fast. Allocating 100 Mbytes is probably sufficient.

Audit Log

The audit log records certain events. The amount of data written to this log depends on what events you want to audit and how often those events occur.

Change Log

The change log on the supplier, tracks changes it needs to replicate on the consumer. Unless a lot of update activity is going on, this log file does not grow very large. TABLE 1 offers specifics on log file size estimates.

TABLE 1 Log Files Size Estimates

Purpose	Size Estimates	Comments
Access Log	100 Mbytes	Activity dependent
Error Log	10 Mbytes	Pruned often
Audit Log	0-10 Mbytes	Depends on usage
Change Log	1-10% of database size	Depends on number of updates

Directory Database

You should have enough RAM so the entire database, including indexes, can fit into memory. Ideally, everything should be cached so read speed is not important. In most environments, only a small percentage of activity will be writes, so the write speed is not that important.

Key Practices: Since the directory database must fit within a single file system, it should be large enough to accommodate future growth. If you determine that your directory database will not exceed 2 Gbytes, consider allocating a file system for your directory database that will fit closest to your memory capacity, minus system and application overhead.

Key Practices: Data availability is key here, so disk mirroring or other RAID levels that provide redundancy is recommended. Also, no other data should be kept on this volume.

Performing Directory Database Backups

Perform database backups by copying the `*.db2` files from the database directory to another directory. Performance is impacted somewhat if the two directories are on the same volume and controller. However, the bigger concern is that if the active database volume becomes inoperative, then you do not want the backup volume to be inoperative. To avoid this, the backup storage device should be on a different volume than the active database, and ideally on a different disk controller.

Key Practices: The backup storage device should be on a different volume than the active database and ideally on a different disk controller.

Determining Memory Size

The most important factor that determines directory performance is the availability of free memory. For peak performance, all directory data needs to be cached in memory. If data has to be retrieved from disk, performance suffers dramatically.

Correctly sizing the directory server's memory requires some knowledge of which components of the directory use free memory and how much is needed. These components include:

- Server executables and database image backing files
- Database entry cache

- Database index cache
- Solaris OE file system cache

Server Executables

The directory server software includes four applications executables:

1. ns-slapd—An instance of the directory server

2. ns-admin—The administration server

3. jre—Java™ runtime environment used by the Netscape Console

4. dsypserv—A Network Information Service (NIS) extension used for NIS interoperability

The ns-slapd process uses up most of the free memory. When it is running with an empty directory database, the ns-slapd process only consumes 5-6 Mbytes, which includes all of the standard plug-ins. The NIS plug-in uses another 500 Kbytes. However, what causes the use of the free memory is that the database caches reside within this process space and the caches grow as they fill up. To add to the issue of memory, the database image backing files are also mapped into this space.

The ns-admin process uses up about 8 Mbytes of free memory and does not grow in size.

The Netscape Console runs on a Java™ platform and requires the Java runtime environment (jre). The size of this process is about 27 Mbytes.

If NIS extensions are used, the dsypsev process consumes about 2.5 Mbytes of free memory.

TABLE 2 on page 9 provides a summary of memory usage for these components.

Database Entry Cache

The database entry caches all the directory data. The total size of the database entries is close to the size of the id2entry.db file. The size depends on the number of entries and the number of attributes each entry contains. For example, a relatively small entry, containing 15 attributes, uses about 1 Kbyte per entry. Therefore, a 100,000-entry database would consume 100 Mbytes of space.

You can configure the size of the database entry cache but you should make it large enough to fit all of the database entry data. The cache size grows as entries are accessed until all entries are cached. This cache memory shows up as part of the ns-slapd process.

Database Index Cache

This cache is also referred to as the database cache, but the name is misleading. What is really contained in this cache are the database indexes. These indexes speed up search time by anticipating what type of searches are most frequent and caching the results. The number and size of indexes depend on how the directory data is accessed. The size of the database indexes can approach the size of the database entries.

Key Practices: For peak performance, the database index cache should be large enough to contain all of the indexes. If configured correctly, this cache has a greater impact on performance than the database entry cache.

Solaris OE File System Cache

Data that is read from disk gets cached in the Solaris OE file system cache. Although directory data that doesn't fit into the database caches is still cached in the file system cache, performance is not as good as with the database cache.

Key Practices: It is better to make the database caches large and not rely on the file system cache. One exception to this recommendation is a directory database that is much larger than the system memory. In the case of a lot of paging activity, the file system cache is more efficient handling the paging activity than the database caches.

Summary of Memory Usage

TABLE 2 details typical memory usage for the components presented in the "Server Executables" section.

TABLE 2 Typical Memory Usage

Component	Estimated Size	Comments
`ns-admin`	8 Mbytes	Static size
`jre`	25 Mbytes	Required for console
`dsypserv`	2.5 Mbytes	NIS extension
`ns-slapd`	6 Mbytes (no data)	Main directory process
`DB Entry Cache`	1 Kbyte X # of entries	Data dependent
`DB Index Cache`	Varies	Depends on complexity of searches

Estimating CPU Usage

Estimating how much CPU power is needed, is a difficult task. While it is easy to see when a production system is overloaded, predicting at what load level this overload occurs is tricky. If you already have a directory server running that you plan to upgrade, you can measure its CPU and use the measurement as a guide. However, if you do not have any historical data, the best approach is to use benchmarks or testing results as a guide.

Industry-standard benchmarks are helpful when you are comparing two platforms but may not be representative of your environment. Homegrown or custom benchmarks are useful since you can design them to simulate your current and future environment.

Unfortunately, there are not any widely accepted industry-standard LDAP benchmarks. The DirectoryMark benchmark from Mindcraft was developed to measure the effectiveness of an LDAP server in a messaging environment. However, it has not caught on as an industry standard benchmark, such as WebBench and SPECweb have for web servers. You can find more information on DirectoryMark on the Mindcraft Web site at:

`http://www.mindcraft.com`

See Chapter 7, pages 172-178 of the Sun BluePrints book, *Solaris and LDAP Naming Services* by Tom Bialaski and Michael Haines, for examples of LDAP testing.

Tuning for Performance

Tuning the Netscape Directory Server for better performance is much like tuning a database, many of the concepts are similar. The remainder of this article deals with the following concepts to improve the performance of your Netscape Directory Server:

- Indexing
- Database caching
- Search limit parameters
- Removing unnecessary plug-ins

Indexing

An index is a lookup table of entry ID numbers that correspond to certain search criteria. The UID index can be searched to find which entries have a matching UID to a particular UID.

Indexing can help search performance, but at a cost. Indexes do consume memory, but the biggest cost in terms of resources is the increased write times. If an indexed attribute is added or modified, the associated index files also need to be updated.

Index Types

TABLE 3 lists the different types of indexes and the types of searches they perform.

TABLE 3 Types of Indexes and Types of Searches

Index Type	Type of Search	Example
Presence	Comparison against '*'	`(iscontractor=*)`
Equality	Comparison against some value	`(uid=jsmith)`
Approximate	Comparison uses the ~=sounds-like	`(sn~=steven)`
Substring	Comparison string contains *wildcards	`(cn+*smith*)`
Non-available	Always unindexed Inequality comparisons	`(cn>smith)`

Different index types are used for different types of searches.

The **Presence** index is commonly used to determine if an access control instruction (ACI) exists for a particular object. If an ACI exists, then access permissions must be checked.

The **Equality** index is used in searches that require an exact match. A common example is a mail server, which bases the search for an entry on the UID of the person logging in. In this case, only an entry that matches *exactly* is of interest.

The **Approximate** index is used to retrieve entries that sound like the value being searched for. For example, a person's first name, may have multiple spellings, such as Stephen and Steven.

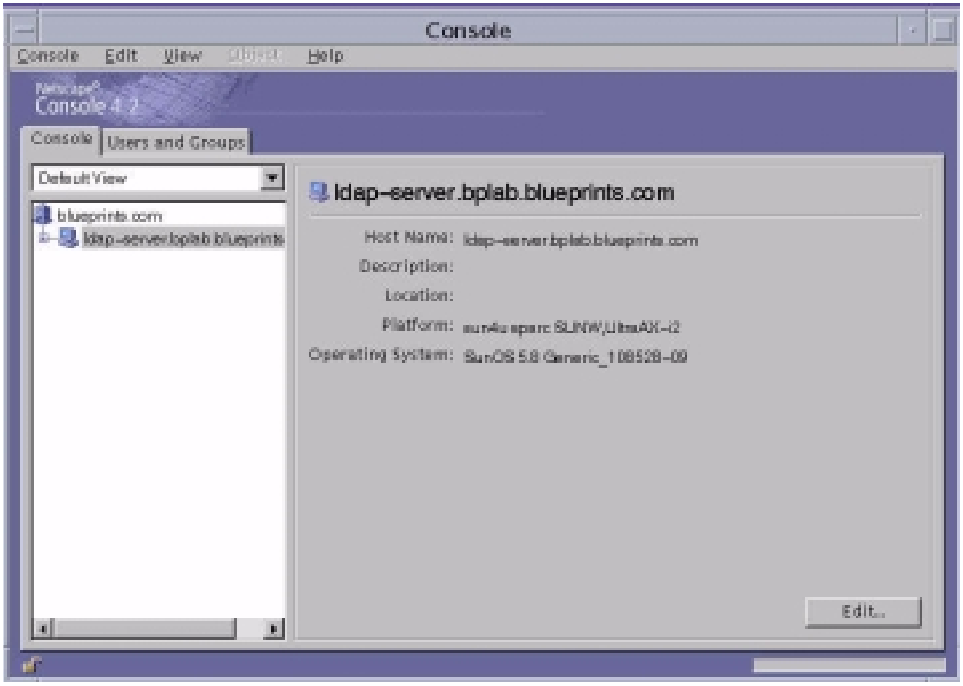
The **Substring** index is used to retrieve entries when only a portion of the total value is specified. For example, a search for a name when only a sequence of characters in the name are known.

Viewing Indexes

The best way to view directory indexes is through the Directory Server Console. To summon the Indexes screen, go to **Configuration→Database→Indexes**

Two types of indexes are displayed: System and Additional. Step 1 through Step 6 shows you how to get to the Indexes screen.

1. **Starting with the Netscape Directory Server Console Main Screen, under the Console tab, select the blueprints.com folder.**



ldap-server.bplab.blueprints.com

Host Name: ldap-server.bplab.blueprints.com

Description:

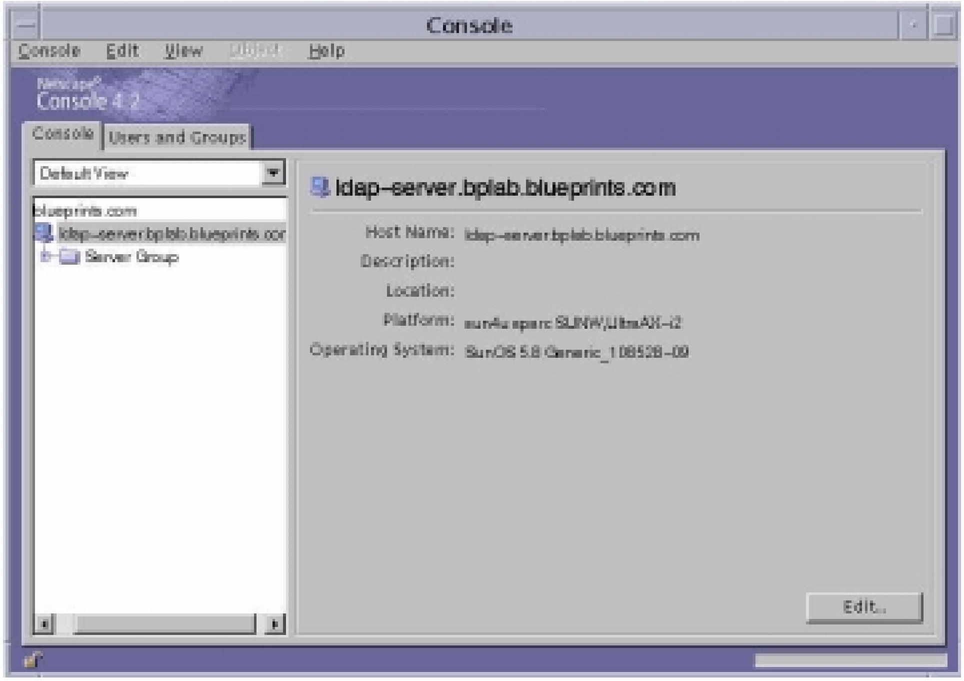
Location:

Platform: sun4u sparc SUNW,UltraAX-i2

Operating System: SunOS 5.8 Generic_108528-09

FIGURE 1 Console Tab View

2. **Under the Console tab, from the ldap-server.bplab.blueprints.com screen, select the Server Group folder.**



Host Name: ldap-server.bplab.blueprints.com

Description:

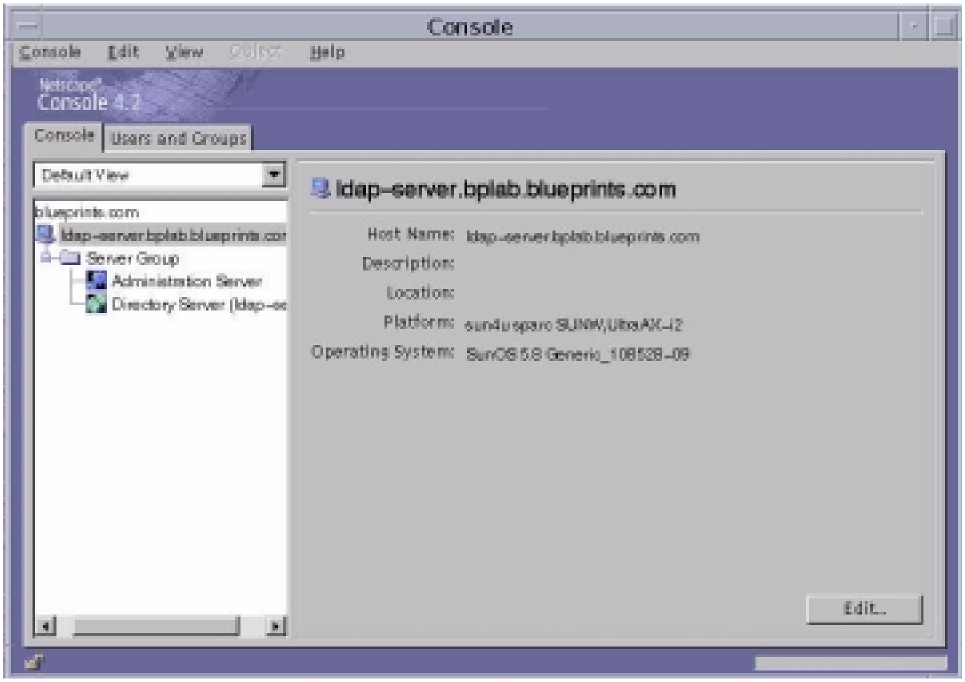
Location:

Platform: sun4u sparc SUNW,UltraAX-i2

Operating System: SunOS 5.8 Generic_108528-09

FIGURE 2 `ldap-server.bplab.blueprints.com` Screen

3. Select the Directory Server (blueprints) folder to bring up the Directory Server Console (see FIGURE 3).



ldap-server.bplab.blueprints.com

Host Name: ldap-server.bplab.blueprints.com

Description:

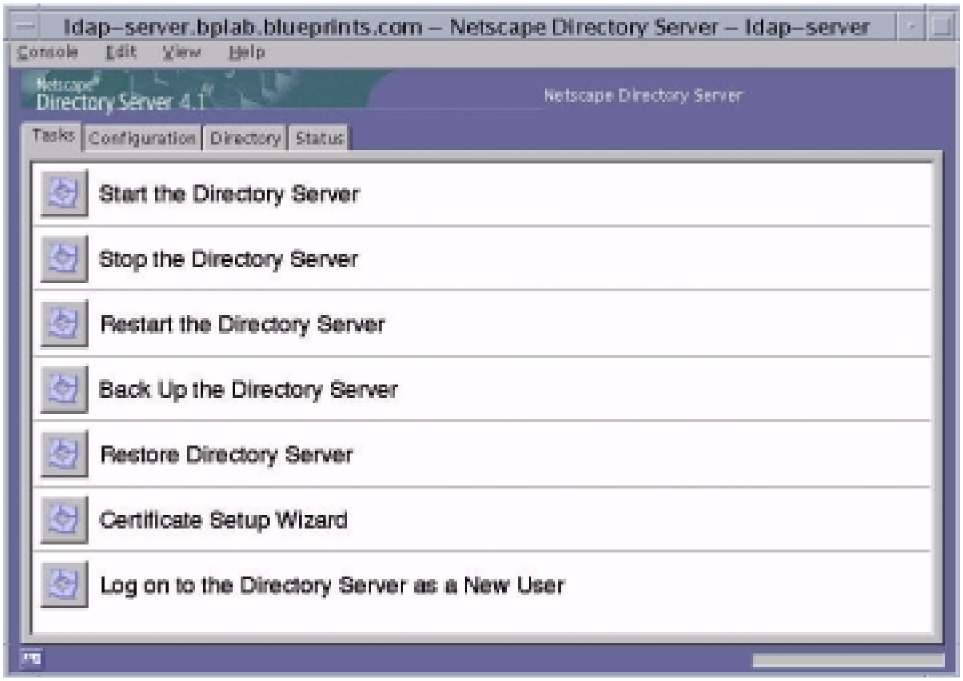
Location:

Platform: sun4u-sparc SUNW,UltraAX-i2

Operating System: SunOS 5.8 Generic_108528-09

FIGURE 3 Path to the Directory Server Console

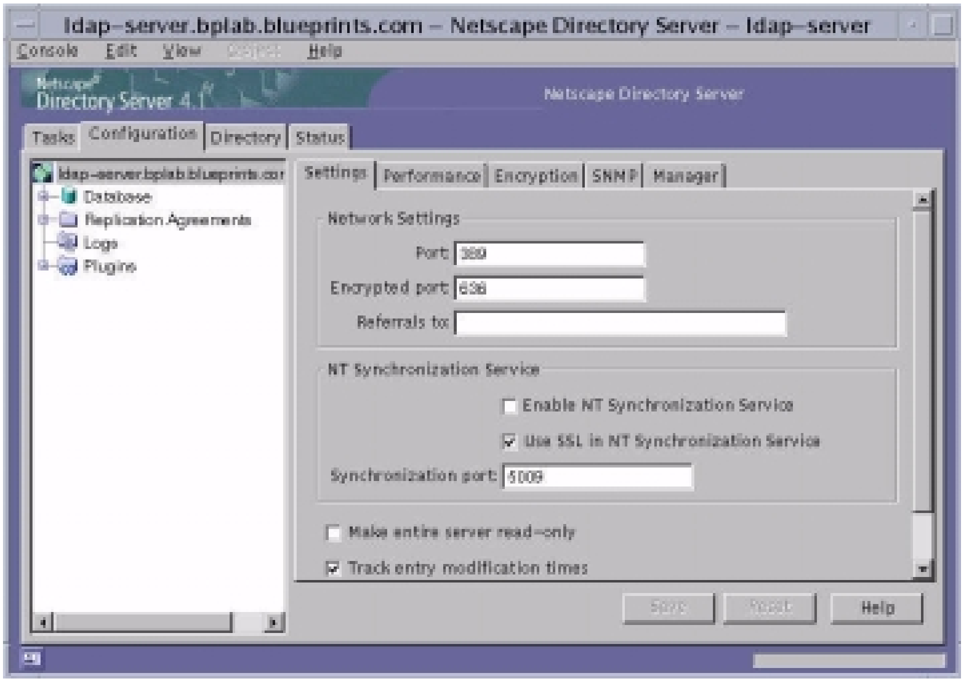
4. **You should now be viewing the Netscape Directory Server Console window.**



ldap–server.bplab.blueprints.com – Netscape Directory Server – ldap–server
Console
Edit
View
Help
Netscape Directory Server 4.1
Netscape Directory Server
Tasks
Configuration
Directory
Status
Start the Directory Server
Stop the Directory Server
Restart the Directory Server
Back Up the Directory Server
Restore Directory Server
Certificate Setup Wizard
Log on to the Directory Server as a New User

FIGURE 4 Netscape Directory Server Console Window

5. From the Directory Server Console screen, select the Configuration tab.


ldap-server.bplab.blueprints.com – Netscape Directory Server – ldap-server
Console
Edit
View
Object
Help
Netscape Directory Server 4.1
Netscape Directory Server
Tasks
Configuration
Directory
Status
ldap-server.bplab.blueprints.com
Database
Replication Agreements
Logs
Plugins
Settings
Performance
Encryption
SNMP
Manager
Save
Reset
Help

Network Settings

Port: 389

Encrypted port: 636

Referrals to:

NT Synchronization Service

□ Enable NT Synchronization Service

☑ Use SSL in NT Synchronization Service

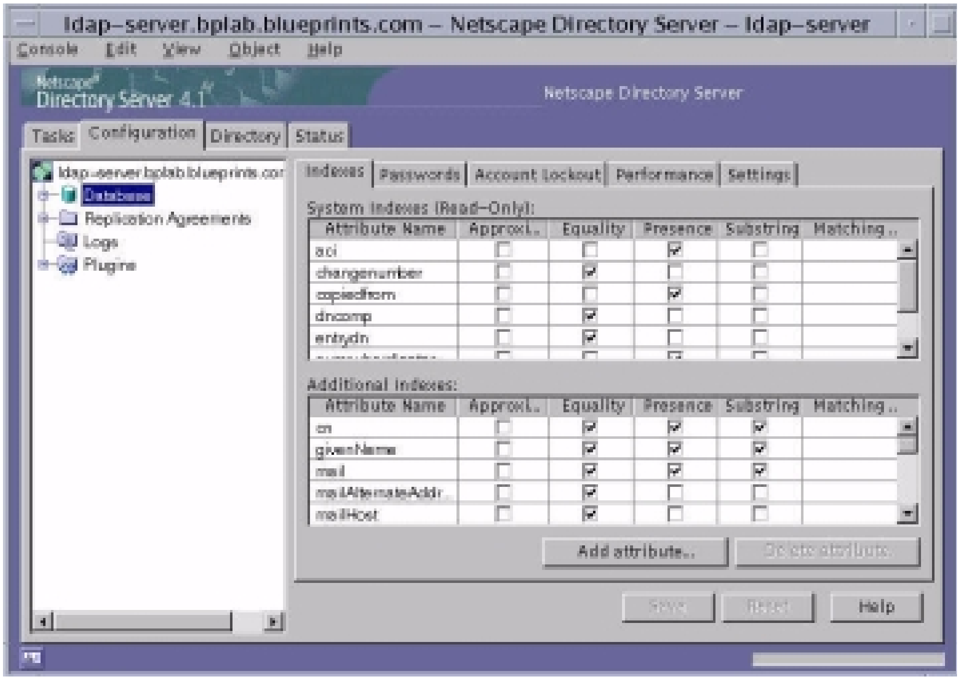
Synchronization port: 5009

□ Make entire server read-only

☑ Track entry modification times

FIGURE 5 Configuration Tab

6. **From the Configuration tab screen, select Database icon to view the Indexes screen.**



ldap–server.bplab.blueprints.com – Netscape Directory Server – ldap–server

System Indexes (Read-Only):

Attribute Name	Approxi..	Equality	Presence	Substring	Matching ..
aci	□	□	☑	□	
changenumber	□	☑	□	□	
copiedfrom	□	□	☑	□	
dncomp	□	☑	□	□	
entrydn	□	☑	□	□	

Additional indexes:

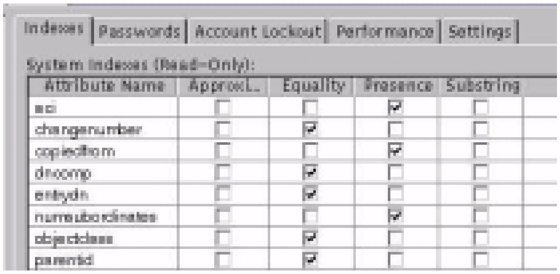
Attribute Name	Approxi..	Equality	Presence	Substring	Matching ..
cn	□	☑	☑	☑	
givenName	□	☑	☑	☑	
mail	□	☑	☑	☑	
mailAlternateAddr..	□	☑	□	□	
mailHost	□	☑	□	□	

.

FIGURE 6 Database Icon With Indexes Tab

System Indexes

Now that you have the Indexes screen displayed, look at the types of indexes available. FIGURE 7 shows the System Indexes panel.



System Indexes (Read-Only):

Attribute Name	Approxi..	Equality	Presence	Substring
aci	□	□	☑	□
changenumber	□	☑	□	□
copiedfrom	□	□	☑	□
dncomp	□	☑	□	□
entrydn	□	☑	□	□
numsubordinates	□	□	☑	□
objectclass	□	☑	□	□
parentid	□	☑	□	□

.

FIGURE 7 System Indexes Panel

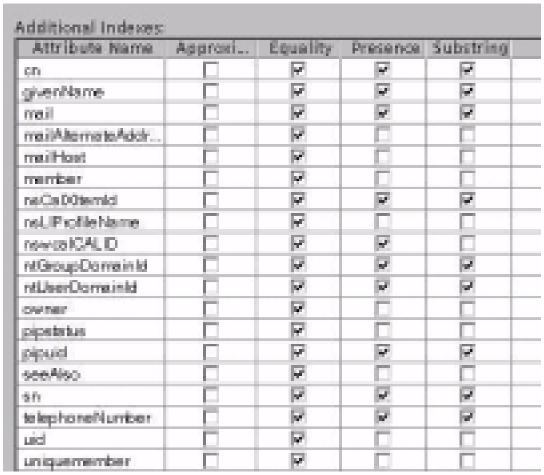
The System Indexes (Read-Only) panel cannot be modified. TABLE 4 explains the use of these indexes. Because these indexes cannot be modified, detailed information on how they work is not provided.

TABLE 4 Use of System Indexes

Attribute	Purpose
`aci`	Quickly obtains access control information
`changenumber`	Tracks the replication
`copiedfrom`	Specifies source for replication
`dncomp`	Accelerates subtree searches
`entrydn`	Speeds up entry retrieval based on DN equality searches
`numsubordinates`	Specifies source for internal operations of the Directory Server Console
`objectclass`	Accelerates subtree searches for internal administration purposes
`parentid`	Speeds up one-level searches

Additional Indexes

During the installation of the Directory Server, several additional default vendor indexes are automatically created. These additional indexes can be removed or modified.



Additional Indexes:

Attribute Name	Approxi...	Equality	Presence	Substring
cn	□	☑	☑	☑
givenName	□	☑	☑	☑
mail	□	☑	☑	☑
mailAlternateAddr...	□	☑	□	□
mailHost	□	☑	□	□
member	□	☑	□	□
nsCalXItemId	□	☑	☑	☑
nsLIProfileName	□	☑	□	□
nswcalCALID	□	☑	☑	□
ntGroupDomainId	□	☑	☑	☑
ntUserDomainId	□	☑	☑	☑
owner	□	☑	□	□
pipstatus	□	☑	□	□
pipuid	□	☑	☑	☑
seeAlso	□	☑	□	□
sn	□	☑	☑	☑
telephoneNumber	□	☑	☑	☑
uid	□	☑	□	□
uniquemember	□	☑	□	□

FIGURE 8 Additional Indexes Panel

Key Practices: Removing unused indexes reduces the amount of physical and virtual memory required, but does not significantly affect overall performance. It is a difficult task to determine which indexes are important and which are not. The use of some indexes, such as those that support the NT Synchronization service, is obvious, and if you were not running the NT Synchronization service, you could easily remove those indexes.

TABLE 5 shows the directory component that uses specific indexes.

TABLE 5 Use of Additional Indexes

Attribute	Specific Index Use
`cn`	Common Search Parameter
`givenName`	User Account
`mail`	User Account
`mailAlternative Address`	NT Synchronization Service
`mailHost`	Mail Server
`member`	Referential Integrity Plug-in
`nsCalXItemId`	Sun™ONE Calendar Server
`nsLIProfile Name`	Netscape Communicator roaming
`ntGroupDomainId`	NT Synchronization Service
`owner`	Referential Integrity Plug-in
`pipstatus`	SunONE Calendar Server
`pipuid`	Netscape Calendar Server
`seeAlso`	Referential Integrity Plug-in
`sn`	Common Search Parameter
`telephoneNumber`	Common Search Parameter
`uid`	Common Search Parameter
`uniquemember`	Referential Integrity Plug-in

Determining Which Indexes Are Important

You can determine the type of directory searches being performed by the directory server by looking in the access log (see the "Access Log" section on page 5 for additional information) for entries that contain the keyword SRCH. The following example shows what a typical SRCH record looks like.

```
ldap-server# grep -i  SRCH  /opt/netscape/slapd-ldap-server/logs/access

02/28/2001:16:09:61 -0400] conn=18863 op=18885 SRCH base="dc,blueprints,
dc=com" scope=2 filter="(cn=John Smith)"
```

This example details a search being performed on the cn attribute, which is a common search parameter. You can also see that it is an Equality index by looking at the part of the entry that reads (cn=John Smith).

Key Practices: If this type of search appears frequently in the access log then an Equality index on the cn attribute could help boost performance.

With the Netscape Directory Server, you can identify searches that reference unindexed attributes by examining the access log. Look for RESULT records that have the field notes=U. For example:

```
ldap-server# grep -i  "notes=U" /opt/netscape/slapd-ldap-server/logs/access

02/28/2001:16:10:01 -0400] conn 18863 op=18885 notes=U RESULT err=0
```

By matching the connection (conn) and operation (op) fields with the corresponding SRCH record, you can determine which searches are being performed without indexes.

Index Administration

Creating or modifying indexes is easily performed through the Directory Server Console. If an attribute is already indexed, you can modify it by checking the appropriate boxes or you can remove it completely. To add an index, use the Add Attribute button under the Additional Indexes panel to bring up the following Select Attribute window.

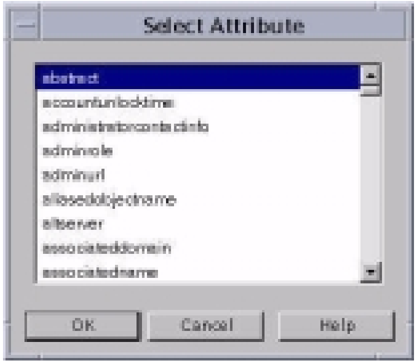


FIGURE 9 Select Attribute Window

Choose the attribute that you want to create the index for, then click the OK button. This attribute is added to the **Indexes** tab where that type of indexing is specified.

Key Practices: Indexing is a powerful tool for speeding up directory search times. However, too much indexing can have a negative effect on write performance. Before modifying or creating new indexes, you must examine the directory access log to determine what types of searches are most prevalent.

Note – Because indexes pertain only to a particular instance of a directory server, a second server could be set up with replicated data but different indexes. The first server could be optimized for write access, (minimum indexing), and the second server could be optimized for a wide variety of searches.

Caching for Performance

To provide the quickest response to LDAP queries, the Netscape Directory Server uses in-memory caches to cache database indexes and directory entries. These caches are managed to help ensure that the highest possible percentage of queries can be fulfilled from information stored in cache, instead of from information retrieved from relatively slow disks.

There are two types of caches used by the Netscape Directory Server to allow it to store information from the database in physical memory and, in turn, circumvent the need to retrieve the data from disk. These are the database cache and the entry cache.

The database cache caches pages from the database. It stores both database indexes and data. It is the lowest cache level. The size of the database cache is set by the administrator to a given amount of memory.

The entry cache caches the most recently accessed entries from the directory. It uses a least recently used algorithm to ensure that the most frequently accessed directory entries are always available in memory. Unlike the database cache size, the size of the entry cache is set by the administrator and is based on the maximum number of entries the cache should hold, not the maximum amount of memory it should consume.

The concept here is to maximize performance and to avoid having the Netscape Directory Server go to disk for data. You must cache as much directory data in physical memory as possible.

There are three important rules you must follow in maximizing read performance.

The database cache must always be large enough to hold the database's indexes.

Key Practices: Ensure that your database cache is at least big enough to hold the directory indexes. If it is not, the directory is forced to read indexes from disk for every search request. This behavior quickly brings directory throughput to a virtual halt.

The database and entry caches must always fit into available physical memory.

Key Practices: When given the choice between allocating physical memory to the database cache or the entry cache, choose the database cache.

The database cache is more important than the entry cache.

Again, to maximize directory performance, allocate as much available physical memory to the directory caches as possible. You can achieve efficient operation by allocating physical memory between the caches in the ratio of 75 percent for the database cache, 25 percent for the entry cache. For example, if the system has 500 Mbytes of free memory during normal operation, allocate 375 Mbytes to the database cache, and 125 Mbytes to the entry cache.

Caution – If the sizes of the two caches combined is bigger than the amount of available physical memory on the server, then the operating environment begins to swap the cache memory to disk. This can cause a significant amount of virtual memory *thrashing* that quickly brings the directory and the entire system to a virtual halt.

Sizing the Database and Entry Caches

Although bigger caches are usually better, performance can degrade if `dbcachesize` is set too large.

Caution – If you set the database cache too large and exceed the physical free memory available, the process begins to page-out to disk. The result is severe performance degradation. You can detect this behavior by using `vmstat` and looking for excessive page-out activity.

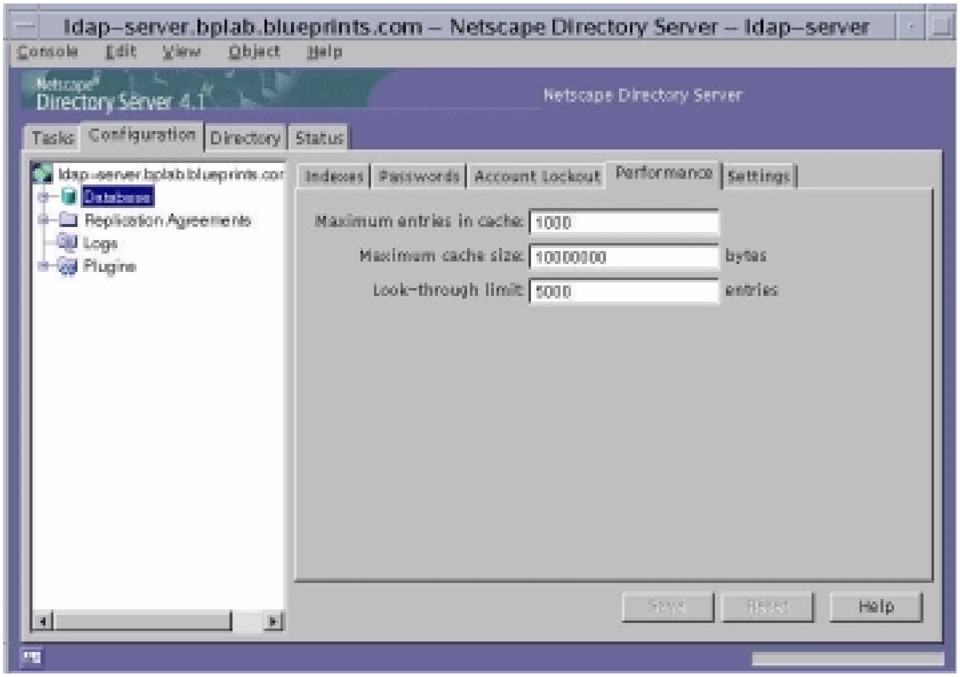
Setting the Database Cache Size

The database cache size can be set either through the Directory Server Console GUI or through a parameter in the `slapd.ldbm.conf` file by editing the configuration file. Step 1 and Step 2 detail how to set the database cache size.

1. **Set the size of the database cache through a parameter in the `slapd.ldbm.conf` file by editing the configuration file. For example:**

```
ldap-server# cd   /opt/netscape/slapd-ldap-server/config

ldap-server# vi slapd.ldbm.conf

. . .

suffix “dc=blueprints,dc=com”

suffix “o=NetscapeRoot”

directory         “/db_fs/db”

cachesize       1000

dbcachesize      100000000   <-------- set size here in bytes

lookthroughlimit     5000

. . . .

ldap-server# /opt/netscape/slapd-ldap-server/restart
```

2. **To set or change the database cache size through the GUI, go back to the Netscape Directory Server Console window→Configuration tab screen and select the Performance tab. This is the same window that you used to select the Indexes tab in the Step 6 on page 17. FIGURE 10 shows the Performance tab. Change the value in the Maximum cache size box to a value that best suits the needs of your Netscape Directory Server configuration.**



Maximum entries in cache: 1000

Maximum cache size: 10000000 bytes

Look-through limit: 5000 entries

FIGURE 10 Performance Tab

Note – Be aware that the actual amount of physical memory used by the database cache can exceed the size if you specify by up to 25 percent because of the additional memory required to manage the cache itself.

Setting Entry Cache Size

The entry cache size is set by the maximum number of entries you would want it to hold. The actual amount of physical memory this cache consumes is a function of the average entry size. For example, if the average entry size is 1 Kbyte, and you specify that the entry cache should hold a maximum of 10,000 entries, then the amount of memory the cache consumes is (1 Kbyte/entry * 10,000 entries) 10 Mbytes + 25 percent cache for cache management overhead.

The entry cache size can be set either through the Netscape Directory Server Console or by editing the `/opt/netscape/slapd-ldap-server/config/slapd.ldbm.conf` file. Step 3 and Step 4 detail both methods of setting the entry cache.

3. **Set the size of entry cache size by editing the `/opt/netscape/slapd-ldap-server/config/slapd.ldbm.conf` file. For example:**

```
ldap-server# cd  /opt/netscape/slapd-ldap-server/config

ldap-server# vi slapd.ldbm.conf

. . .

suffix “dc=blueprints,dc=com”

suffix “o=NetscapeRoot”

directory         “/db_fs/db”

cachesize       1000        <-------- set size here in # of entries

dbcachesize      100000000

lookthroughlimit     5000

. . . .

ldap-server# end codebox
```

4. **Use the Netscape Directory Server Console GUI to set the entry cache size value.**

   As you did in Step 2 to change the database cache size, go back to the **Netscape Directory Server Console window→Configuration tab** screen and select the Performance tab. This is the same window that you used to select the Indexes tab in Step 6 on page 17. FIGURE 10 on page 25 shows the Performance tab.

   Change the value in the Maximum cache size box to a value that best suits the needs of your Netscape Directory Server configuration.

Fine-tuning Cache Sizes

Step 5 through Step 8 detail the fine-tuning technique for database and entry cache sizing.

5. **Using `vmstat`, identify the amount of free memory available when your system configuration is operating normally.**

```
ldap-server# vmstat 2
 procs      memory                 page                disk           faults       cpu
 r b w    swap   free   re   mf pi po fr de sr dd f0 s0 --    in    sy    cs us sy id
 0 0 0 2939576 149184 0    0  0  0  0  0  0  1  0  0  0  124   247   109  0  0 100
 0 0 0 2939512 147048 0    3  0  0  0  0  0  0  0  0  0  125   233   103  0  0 100
 0 0 0 2939512 147048 0    0  0  0  0  0  0  0  0  0  0  125   265   135  1  0 99
 0 0 0 2939512 147048 0    0  0  0  0  0  0  0  0  0  0  126   265   112  0  0 100
```

From the `vmstat` output you can see that this particular system has approximately 147 Mbytes of free memory.

6. **To determine the amount of memory your system has, execute the `/usr/bin/dmesg` command combined with the `grep` command to obtain information on system RAM.**

```
ldap-server# /usr/bin/dmesg | grep  mem

Feb 26 09:05:44 ldap-server unix: [ID 389951 kern.info] mem = 262144K
(0x10000000)

Feb 26 09:05:44 ldap-server unix: [ID 930857 kern.info] avail mem =
251731968

Feb 28 13:28:46 ldap-server unix: [ID 389951 kern.info] mem = 262144K
(0x10000000)

Feb 28 13:28:46 ldap-server unix: [ID 930857 kern.info] avail mem =
251731968

Feb 28 13:46:01 ldap-server unix: [ID 389951 kern.info] mem = 262144K
(0x10000000)

Feb 28 13:46:01 ldapserver unix: [ID 930857 kern.info] avail mem = 251731968
```

From the `dmesg|grep mem` command output you can see that this particular system was configured with 264 Mbytes of system RAM.

7. **Use 75 percent of your free memory to allocate to your database cache. Divide that amount by 1.25 to account for the cache overhead, and use the result as your database cache size.**

If the result is larger than 1.6 Gbytes, reduce it to 1.6 Gbytes (the maximum dbcache size). If the result is smaller than the sizes of the database indexes (calculated by adding up the sizes of all *.db2 files in the directory database directory and subtracting the size of id2entry.db2), increase it to cover the sizes of the indexes. Do not exceed the available memory or 1.6 Gbytes, whichever is lower.

Here is an example from the Sun BluePrints book *Solaris and LDAP Naming Services* by Tom Bialaski and Michael Haines (ISBN 0-13-030678-9).

> Assume the following system / directory parameters:
>
> 150 Mbytes of available RAM (free memory)
>
> Estimated 1 Kbyte entry size on average
>
> **Calculate DB cache size:**
>
> 150 M * .75 = 112 Mbytes to allocate to db cache
>
> 112/1.25 = 90Mbytes for db cache

8. **Use 25 percent of your free memory to allocate to the entry cache. Divide that amount by 1.25 to account for cache overhead. Divide the result by the average entry size, and use the result as the entry cache size (maximum number of cache entries).**

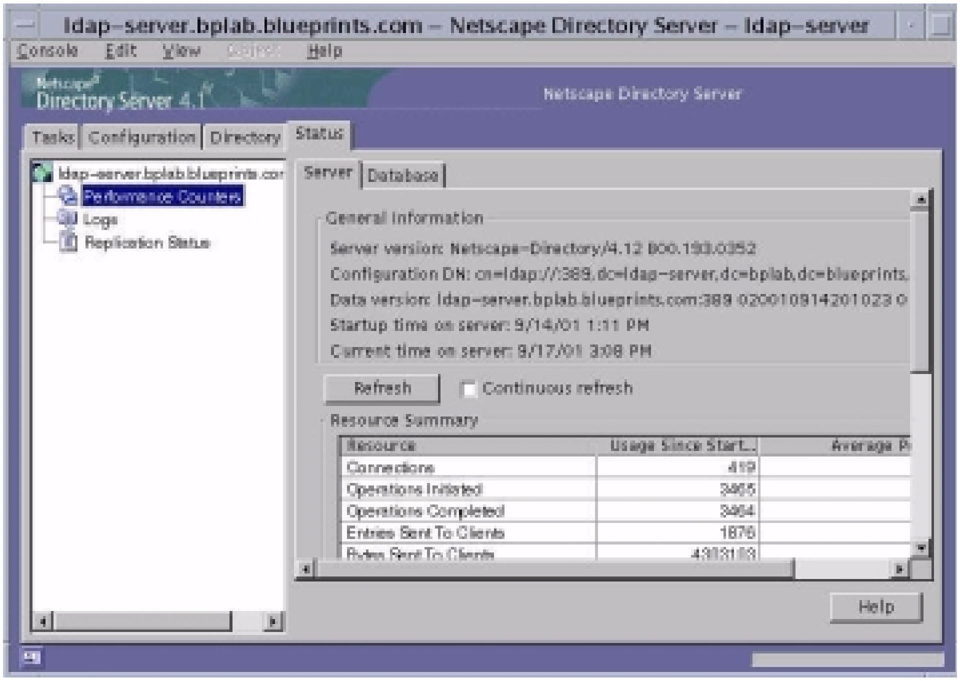
   Here is an example from the Sun BluePrints book *Solaris and LDAP Naming Services* by Tom Bialaski and Michael Haines (ISBN 0-13-030678-9).

> Assume the following system / directory parameters:
>
> 150 Mbytes of available RAM (free memory)
>
> Estimated 1 Kbyte entry size on average
>
> **Calculate entry cache size:**
>
> 150 - 112 = 38 Mbytes for entry cache
>
> 38/1.25 = 30 Mbytes available for entries
>
> 30M/1 Kbytes = max 30,000 cache entries

Monitoring Cache Utilization

After you set the cache sizes, it is necessary to monitor the cache utilization from time-to-time to assure efficient use. The Netscape Directory Server Console GUI has performance counters you can use to monitor such activity. FIGURE 11 shows Database Statistics. To view the statistics go to **Database→Status→Performance Counters**.

Note – The Performance Counters display data available since the directory server was last started. To reset the counters, restart the directory server and delete the `*.share` backing files located in the directory `/opt/netscape/slapd-ldap-server/` db.



Server version: Netscape–Directory/4.12 B00.193.0352

Configuration DN: cn=ldap://:389,dc=ldap–server,dc=bplab,dc=blueprints,

Data version: ldap–server.bplab.blueprints.com:389 020010914201023 0

Startup time on server: 9/14/01 1:11 PM

Current time on server: 9/17/01 3:08 PM

□ Continuous refresh

Resource	Usage Since Start...	Average P
Connections	419	
Operations Initiated	3465	
Operations Completed	3464	
Entries Sent To Clients	1876	
Bytes Sent To Clients	4313103	

FIGURE 11 Database Statistics

In FIGURE 11 you can see that the entry cache is set to 1000 but is only approximately 1/9th full. Also, the database cache hit ration is 99 percent which means that the indexes are essentially always cached. You can also refer to the Sun BluePrints book, *Solaris and LDAP Naming Services*, by Tom Bialaski and Michael Haines (ISBN 0-13-030678-9).

You can obtain the information on these parameters from the command line, for example:

```
ldap-server#  ldapsearch -b "cn=monitor,cn=ldbm" -s base "objectclass=*"

dn: cn=monitor,cn=ldbm

objectclass: top

objectclass: extensibleObject

cn: monitor

database: ldbm

readonly: 0

entrycachehits: 677

entrycachetries: 821

entrycachehitratio: 82

currententrycachesize: 125

maxentrycachesize: 1000

dbchehits: 84448

dbcachetries: 84710

dbcachehitratio: 99

dbcachepagein: 226

dbcachepageout: 404

.

.
```

Under optimal conditions, both the entry and database cache hit ratios will be above 95 percent. If either hit ratio is less than 95 percent and you have additional physical memory, consider increasing your cache size to increase the hit ratio.

Conversely, If you have been running your directory server for several days and find that your entry cache is not filled to the maximum level, you might consider lowering the size of your entry cache to just above the high-water mark, giving the additional free memory to the database cache.

Selecting Search Limit Parameters

In the Netscape Directory Server there are three search limit parameters that limit the amount of resources the server allocates to client requests. These parameter limits aid in administering the performance of your LDAP server.

The search limit parameters are: **Size limit**, **Time limit**, and **Lookthrough limit**.

Size limit (in entries) specifies the maximum number of entries the server returns to the client in response to a search operation. If this limit is reached, the server returns any entries it has located that match the search request, and the exceeded size limit error. The default value for this parameter is 2000.

Time limit (in seconds) specifies the maximum amount of real time the server spends performing a request. If this limit is reached during a search, the server returns any entries it has located that match the search request, and the exceeded time limit error. The default value for this parameter is 3600.

Lookthrough limit (in entries) specifies the maximum number of entries the server checks when seeking candidate entries in response to a search request. If this limit is reached, the server returns any entries it has located that match the search request, and the exceeded size limit error. The default value for this parameter is 5000.

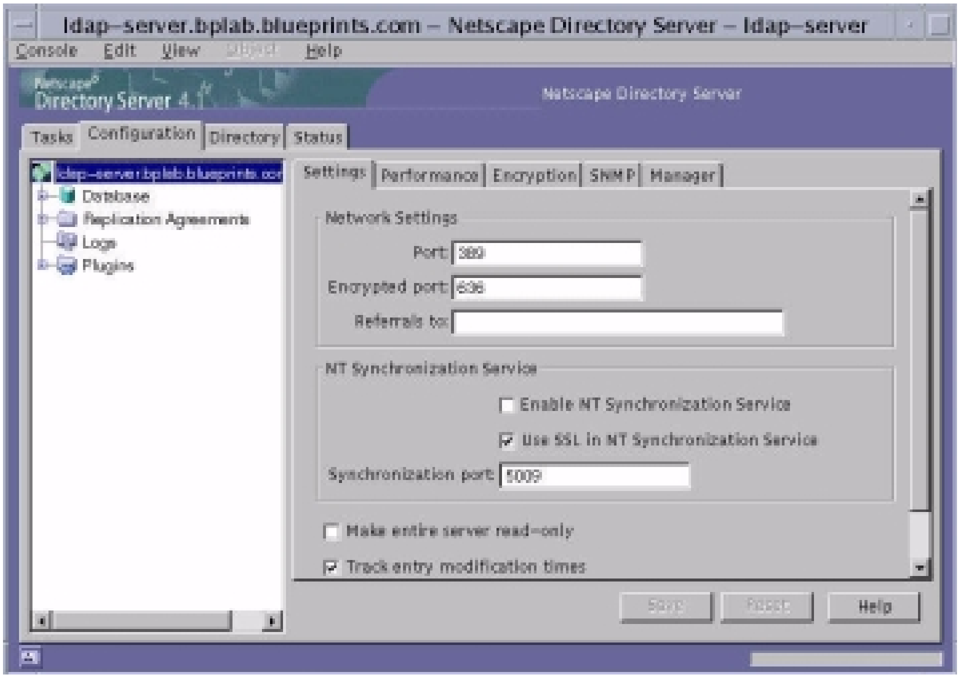
Changing Search Limit Parameters

You can set the Size limit and Time limit parameters either by using the Netscape Directory Server Console or by editing the `slapd.conf` file. Step 1 through Step 5 detail the two methods used to make these changes.

You can change the Lookthrough limit parameter either with the Netscape Directory Server Console or from the command line. Step 6 through Step 9 detail the command line method of changing the Lookthrough limit parameter.

Key Practices: Using the Netscape Directory Server Console to change the Lookthrough limit parameter causes previously hand-edited data to be overwritten, so the command line method is recommended.

1. **From the Netscape Directory Server Console, go to the Configuration tab screen and select the bplab.blueprints.com:389 file.**



Network Settings

Port: 389

Encrypted port: 636

Referrals to:

NT Synchronization Service

□ Enable NT Synchronization Service

☑ Use SSL in NT Synchronization Service

Synchronization port: 5009

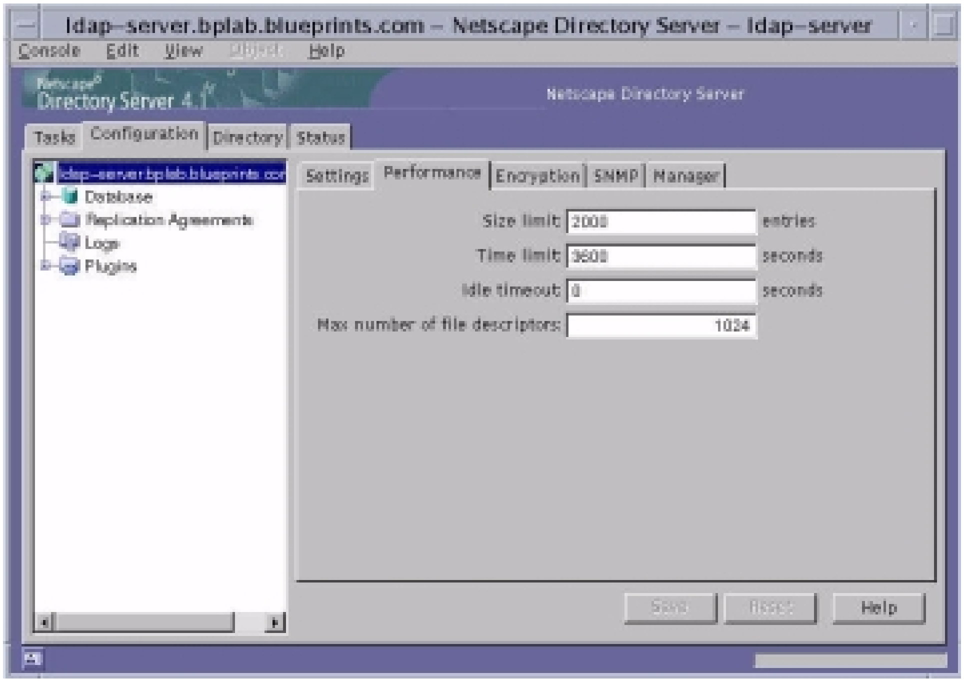
□ Make entire server read–only

☑ Track entry modification times

FIGURE 12 Configuration Tab Screen With the Settings Tab

The current selection is the **Settings** tab with information on the network settings.

2. **Select the Performance tab and the current parameters for the Size limit and Time limit are displayed.**


ldap–server.bplab.blueprints.com – Netscape Directory Server – ldap–server
Console Edit View Object Help
Netscape Directory Server 4.1
Netscape Directory Server
Tasks Configuration Directory Status
ldap–server.bplab.blueprints.com
Database
Replication Agreements
Logs
Plugins
Settings Performance Encryption SNMP Manager
Save Reset Help

Size limit: 2000 entries

Time limit: 3600 seconds

Idle timeout: 0 seconds

Max number of file descriptors: 1024

FIGURE 13 Configuration Tab Screen With the Performance Tab

Change the parameters for the Size limit and Time limit to the values you require.

Note – You should halt the Netscape Directory Server Console from the command line, then restart it, before you make changes.

---

3. **Change the size limit and time limit parameters from the command line by editing `slapd.conf` file.**

```
ldap-server#  /opt/netscape/slapd-ldap-server/stop-slapd

ldap-server# cd  /opt/netscape/slapd-ldap-server/config

...
```

4. **Edit the `slapd.conf` file, by changing the parameters in the `timelimit` and the `sizelimit` lines to the values you require.**

```
ldap-server#   vi  slapd.conf
########################################################################
# /opt/netscape/slapd-ldap-server/config/slapd.conf
# Netscape Directory Server global configuration file
# Do not modify this file while ns-slapd is running
########################################################################
.
.
.include "/opt/netscape/slapd-ldap-server/config/slapd.at.conf"
include "/opt/netscape/slapd-ldap-server/config/slapd.oc.conf"
include "/opt/netscape/slapd-ldap-server/config/ns-schema.conf"
readonly         off
timelimit        3600                <----- change this line
sizelimit        2000                  <----- change this line
lastmod on
idletimeout     0
...
```

5. **Restart the server process to activate the new parameters.**

```
ldap-server#   /opt/netscape/slapd-ldap-server/start-slapd
```

6. **Change the Lookthrough limit parameters from the command line, by editing the `slapd.ldbm.conf` file.**

```
ldap-server# /opt/netscape/slapd-ldap-server/stop-slapd
ldap-server# cd /opt/netscape/slapd-ldap-server/config
...
```

7. **Edit the `slapd.ldbm.conf` file, by changing the parameter in the `lookthroughlimit` line to the value you require.**

```
ldap-server#   vi slapd.ldbm.conf

########################################################################

# /opt/netscape/slapd-ldap-server/config/slapd.ldbm.conf

# Database configuration file. Do not modify this file

# while ns-slapd is running.

########################################################################

.

.

suffix  "dc=blueprints,dc=com"

suffix  "dc=bplab,dc=blueprints,dc=com"

suffix  "o=NetscapeRoot"

directory        "/opt/netscape/slapd-ldap-server/db"

cachesize        1000

dbcachesize      10000000

lookthroughlimit          5000       <-----change this line

readonly         off
```

8. **Restart the server process to activate the new parameters.**

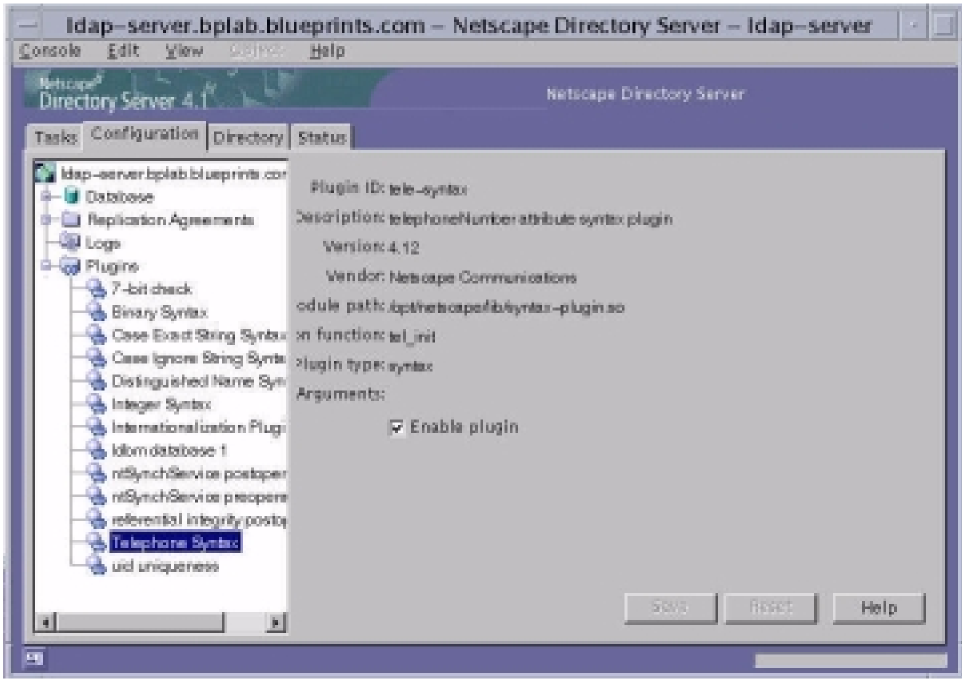
```
ldap-server# /opt/netscape/slapd-ldap-server/start-slapd
```

Removing Unnecessary Plug-ins

The Netscape Directory Server can be enhanced by the addition of features implemented as plug-ins which are contained in libraries loaded at runtime. If the features provided by the plug-ins are *not* used, there is only a small impact on performance. The memory footprint of the directory server will also be somewhat larger because of the inclusion of plug-ins.

Key Practices: It is usually safe to remove the two plug-ins associated with your service but you should keep most of the default plug-ins, since removing them could have unwanted side effects. If you are using the Windows NT Synchronization service, it is not recommended to remove *any* plug-ins.

To remove unwanted plug-ins, from the Netscape Directory Server Console, go to the Configuration tab screen and expand the Plug-ins icon as shown in FIGURE 14.



Plugin ID: tele-syntax
Description: telephoneNumber attribute syntax plugin
Version: 4.12
Vendor: Netscape Communications
odule path: /opt/netscape/lib/syntax-plugin.so
on function: tel_init
Plugin type: syntax
Arguments:
☑ Enable plugin

FIGURE 14 Configuration Tab Screen With Expanded Plug-ins Icon

You can now select the plug-ins you want to delete, then uncheck the Enable plug-in toggle.

Note – Since this operation modifies the `slapd.conf` file, you should halt the server, then restart it after the change has been made.

Solaris and LDAP Naming Services

In the Sun BluePrints book, *Solaris and LDAP Naming Services*, by Tom Bialaski and Michael Haines, there are other optional tuning methodologies that you can consider. The top factor in the rank of tuning your Netscape Directory Server lies within the tuning and configuration of physical memory. Cache sizing and monitoring is a must for LDAP administration.

Summary of Key Practices

The following is a summary list of the Key Practices presented in this article.

- Maximize free memory.
- Utilize storage devices with write cache capabilities.
- Store logs on disks separate from those that contain the directory database, and preferably on separate controllers.
- Utilize RAID mirroring to maximize data availability and reduce downtime.
- Keep backup disks on separate controllers.
- Size your database index cache large enough to contain all of the indexes.
- Enable the system cache to manage the system paging instead of allowing your database cache to do it. Shrinking your database cache size would be more beneficial.
- Remove indexes that are not used to increase performance.
- Create indexes for frequent searches found in the access log files.
- Examine the directory access log to determine what types of searches are more prevalent, before modifying or creating new indexes. Too much indexing can have a negative effect on write performance.
- Ensure that your database cache is at least big enough to hold the directory indexes.
- Select the database cache over the entry cache, when given the choice between allocating memory.
- Edit the Lookthrough limit parameter using the command line.
- Remove unnecessary plug-ins.