IBM SecureWay Version 3.2.2 Directory Tuning Guide
## Contents

| Preface ........................................... | v |
| Who should read this guide ......................... | v |
| Typeface conventions ................................ | v |
| Acronyms used in this document ...................... | v |

**Chapter 1. IBM SecureWay Directory tuning general overview** ........................................... 1
- IBM quick-start tuning ...................................... 1
- Migration .................................................. 1
- SecureWay Directory tuning ................................. 1
- Specific DB2 tuning ......................................... 2
- Other DB2 tuning ........................................... 3
- Generic LDAP application tips ............................... 3

**Chapter 2. IBM SecureWay Directory tuning** .................................................. 5
- Setting front end configuration variables ............... 5
  - Possible variables ........................................ 5
- Setting back end configuration variables ................. 6
  - LDAP connections to DB2 .................................. 7
- Indexes ...................................................... 7
- Update performance and SMP systems ...................... 8
- LDAP cache ............................................... 8
  - LDAP Cache on Solaris and AIX operating systems ........ 9
  - Setting the LDAP cache .................................. 9
  - AIX considerations when defining a large LDAP cache .... 10
  - Estimating LDAP cache size ................................ 13
- Reloading directory data .................................. 14

**Chapter 3. DB2 tuning** ........................................ 17
- Database optimization, statistics, reorganization check, and reorganization. ............... 17
  - Database table organization ................................ 19
  - DB2 buffer pool tuning .................................... 20
  - Buffer pool tuning for previous versions of SecureWay Directory .......................... 20
  - Buffer pool tuning for SecureWay Directory version 3.2.2 .................................. 21
  - Guidelines for when to use LDAP Cache or buffer pool cache ................................ 21
  - Other DB2 configuration parameters ......................... 22
  - Backup and restore ....................................... 22

**Chapter 4. AIX Operating system tuning** ........................................ 25
- Enabling large files ....................................... 25
- Setting MALLOCMULTIHEAP .................................. 25
- Viewing slapd/slapdd environment variables (AIX operating system only) ................. 26

**Chapter 5. Hardware tuning** ........................................ 27
- Disk speed improvements ................................... 27

**Chapter 6. SecureWay Directory features** ........................................ 29
- Bulk loading (bulkload) .................................... 29
  - Incremental bulkload ..................................... 30
  - Replication .............................................. 31
  - Monitoring Performance .................................. 31
  - Example .................................................. 32
  - When to configure LDAP changelog ......................... 32

**Appendix** ........................................ 33
- Notices ..................................................... 33
- Trademarks ................................................ 34
Preface

Welcome to the IBM SecureWay Version 3.2.2 Directory Tuning Guide. The purpose of this document is to provide performance tuning information for IBM SecureWay Directory. It is broken down into sections dealing with SecureWay Directory, IBM Database 2 (DB2®), operating system, and hardware tuning issues.

This document can help you configure each environment for improved performance. Tuning considerations for directory sizes ranging from a few thousand entries to millions of entries are given where applicable. Some advantages and disadvantages of different settings for the tuning parameters are also given. Some of these settings might affect resource usage, speed, and functionality.

For the most current and accurate tuning information, see the Web version of the Tuning Guide on the IBM SecureWay Directory Web site:


Who should read this guide

The target audience for this guide includes:

• System installation and deployment administrators
• Network system administrators
• Information Technology architects
• Application developers

Typeface conventions

This guide uses several typeface conventions for special terms and actions. These conventions have the following meaning:

<table>
<thead>
<tr>
<th><strong>Bold</strong></th>
<th>Command names and options, keywords, and other information that you must use literally appear in bold.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Italics</strong></td>
<td>Variables and values you must provide appear in italics.</td>
</tr>
<tr>
<td><strong>Monospace</strong></td>
<td>Code examples, command lines, screen output, file names, programming keywords, message text or prompts addressed to the user, and text that the user must enter appear in monospace font.</td>
</tr>
</tbody>
</table>

Acronyms used in this document

• ACL – Access Control List
• DB2 – Database 2
• LDAP – Lightweight Directory Access Protocol
• SMP – Symmetric Multi-Processors

© Copyright IBM Corp. 2002
Chapter 1. IBM SecureWay Directory tuning general overview

The SecureWay Directory is a Lightweight Directory Access Protocol (LDAP) directory that enables users to store and retrieve data for multiple purposes. The SecureWay Directory utilizes IBM Database 2 (DB2), a powerful and scalable database product, for its data storage facility. In the most optimal LDAP environments, directory data is fairly static and the access for LDAP "cached" data is repetitive. In more robust environments, where directory data is updated frequently and the access for "non-cached" data is random, the power and scale of DB2 is used to enhance performance.

IBM quick-start tuning

This section contains IBM’s recommendations for basic tuning. Although it includes pointers to some of the more important and commonly used tunings, it is by no means a comprehensive list. We recommend that you read the entire document to learn about tuning setups for your particular system.

Basic tuning recommendations:

- Allocate more than 60% of your physical memory to the DB2 buffer pools. See “Buffer pool tuning for previous versions of SecureWay Directory” for more information.
- For AIX® operating systems on SMP machines, set MALLOCMULTIHEAP. See “Setting MALLOCMULTIHEAP” for instructions on how to set MALLOCMULTIHEAP.
- Make sure all attributes used in searches are indexed. See “Indexes” for instructions on how to define/verify indexes for SecureWay Directory.

Migration

If you follow the migration procedure in “Reloading directory data” follow the IBM tuning recommendations in “IBM quick-start tuning”.

If you are migrating from a pre 3.2.x version of SecureWay Directory, and you are moving to IBM Directory 4.1, we recommend that you do not reload your directory data in 3.2.2. See Chapter 11 in the IBM Directory Server 4.1 Installation and Configuration Guide for Multiplatforms for more information about migrating from a pre 3.2.x version of SecureWay Directory to IBM Directory 4.1.

SecureWay Directory tuning

In this release, several performance enhancements have changed the underlying DB2 table structures. If you are migrating to SecureWay Directory 3.2.2, unload and reload the data in your directory in order to take advantage of these features. See “Reloading directory data” for more details.

The SecureWay Directory allows users to set LDAP front end configuration variables in the slapd32.conf file that customize the SecureWay Directory for a specific environment. See “Setting front end configuration variables” for more information.
The most important configuration variables for search performance are related to LDAP caches, which are fast storage buffers in memory used to store LDAP information such as queries, answers, and user authentication for future use. While LDAP caches are mostly useful for applications which frequently retrieve repeated "cached" information, they can greatly improve performance by avoiding calls to the database. The negative aspect of LDAP caches is the cache invalidation that occurs on update operations. In order to take advantage of the LDAP caches, it is useful to understand how they are used and maintained in detail. See "LDAP cache" on page 8 for more information.

As a general rule, you should define larger LDAP caches for the following cases:

- If there is no or low update activity and mostly cached searches
- If there is no or low update activity and enough memory to cache the entire directory

Specific DB2 tuning

The SecureWay Directory uses the IBM DB2 relational database as the data store and Structured Query Language (SQL) query retrieval mechanism. While LDAP caches LDAP queries, answers, and authentication information, DB2 has much more sophisticated and complex caching mechanisms which cache tables, indexes, and statements.

The DB2 data caches, which are called bufferpools, are important factors that can affect DB2 performance. Bufferpools, due to the removal of LONGVARCHAR columns, generally are more effective than the LDAP cache, especially in read/write environments. Each bufferpool is a data cache between the applications and the physical database files. If there are no bufferpools, then all database activity results in disk access. If the size of each bufferpool is too small, the bufferpool hit ratio will be low and the applications will wait for disk access activity to satisfy SQL queries. If one or more bufferpools are too large, memory on the server may be wasted. If the total amount of space used by all bufferpools is larger than the physical memory available on the server, then operating system paging (disk activity) will occur. See "DB2 buffer pool tuning" on page 20 for more information.

In general, increasing the DB2 bufferpool caches can be advantageous in the following cases:

- If there is high update activity
- If most queries are "non-cached" and there is not enough memory to cache the entire directory

DB2 has many other configuration parameters that can affect either the memory or disk resources. Since disk access is usually much slower than memory access, a key database performance tuning objective is to decrease the amount of disk activity. If you are able to eliminate input/output (I/O) wait time, the database requests are CPU bound and increasing performance typically would then require faster CPUs or multiple CPUs. For more information in detail see "Other DB2 configuration parameters” on page 22.

After initially loading a directory, or after a number of updates have been performed, it may be necessary to update database statistics and table organization for DB2 to perform optimally. See “Database optimization, statistics, reorganization check, and reorganization” on page 17 for more information.
Administrators must place the DB2 log on a physical disk drive separate from the data. While there might be some performance benefit to having the DB2 log and data on the same drive, data-integrity concerns require the separation. Use the following command to set the path to the DB2 log file directory:

```
UPDATE DATABASE CONFIGURATION FOR database_alias USING NEWLOGPATH path
```

**Note:** Be sure the database instance owner has write access to the specified path or the command fails.

### Other DB2 tuning

There are numerous other DB2 methods and techniques which may be used to extend scalability and possibly improve performance. More detailed information on DB2 can be found at:

[http://www.ibm.com/software/data/db2](http://www.ibm.com/software/data/db2)

### Generic LDAP application tips

The following are some generic tips that can help improve performance:

- Perform searches on indexed attributes only. See "Indexes" on page 7 for instructions on how to define/verify indexes for SecureWay Directory.
- Open a connection only once and reuse it for many operations if possible.
- Minimize the number of searches by retrieving multiple attribute values at one time.
- Retrieve only the attributes you need, do not use ALL by default. For example, when you search for the groups a user belongs to, ask for just the Distinguished Names (DNs), and not the entire group.
- Minimize updates (add, modify, modrdn, delete) when possible.
Chapter 2. IBM SecureWay Directory tuning

This chapter discusses performance tuning tasks for the SecureWay Directory.

Topic Index:
- Setting front end configuration variables
- Setting back end configuration variables
- Indexes
- Update performance and SMP systems
- LDAP cache
- Reloading directory data

Note: The examples in this chapter referring to the slapd process use slapd, but if you selected 128 bit SSL version during installation you should use slapdd.

Setting front end configuration variables

There are severable tunable LDAP front end configuration variables that can affect performance in the slapd32.conf file.

To set front end configuration variables add this line:
ibm-slapdSetEnv: variable_name= value
to the following section:
dn: cn=Front End,cn=Configuration
objectclass: top
objectclass: ibm-slapdFrontEnd

in the slapd32.conf file.

For example, to set the RDBM cache size to 100,000 and turn concurrent read/write on, you would add the following entry to slapd32.conf:
dn: cn=Front End,cn=Configuration
objectclass: top
objectclass: ibm-slapdFrontEnd
ibm-slapdSetEnv: RDBM_CACHE_SIZE=100000
ibm-slapdSetEnv: LDAP_CONCURRERTRW=ON

Possible variables

Below are some of the front end configuration variables you might want to set.

ACLCACHE=YES|NO
Determines whether or not there will be an ACL cache on the server.
By default this variable is set to YES.

ACLCACHESIZE=<integer>
Specifies the maximum number of entries kept in the ACL cache.
By default the size is 25000.

DB2CP=<integer>
Specifies the Code Page of the directory database. “1208” is the
code page for UTF-8 databases.
By default this variable is set to 1208

LDAP_CONCURRENTRW=ON|OFF
Allows reads (searches or compares) to proceed concurrently with
update operations. When set to ON, there is the possibility that
search results may not always match the search filter requested, if
the matching entries were in the process of being updated. By default
this variable is set to ON.

RDBM_CACHE_SIZE=<integer>
Specifies the maximum number of entries to keep in the Entry cache.
By default the size is 25000.

RDBM_FCACHE_SIZE=<integer>
Specifies the maximum number of entries to keep in the Search Filter cache.
By default the size is 25000.

RDBM_CACHE_BYPASS_LIMIT=<integer>
Search filters that match more than this number of entries will not be added
to the Search Filter cache. Because the list of entry IDs that matched the
filter are included in this cache, this setting helps to limit memory use.
A value of 0 indicates no limit. The default value is
RDBM_CACHE_BYPASS_LIMIT=100

RDBM_ENTRY_CACHE_BYPASS=[any value]
When set to any value, the RDBM_CACHE_BYPASS_LIMIT applies
to the entry cache in addition to the filter cache. By default, the
RDBM_CACHE_BYPASS_LIMIT does not apply to the entry cache.

Setting back end configuration variables

Back end configuration variables are set during DB2 configuration. To set back end
configuration variables, add this line:
ibm-slapd <variable>:<value>
to the following section:
dn:cn=Directory,cn=RDBM Backends,cn=IBM SecureWay,cn=Schemas,cn=Configuration
cn:Directory
objectclass:top
objectclass:ibm-slapdRDBMBackend

In the slapd32.conf file:

The following table contains some examples of back end configuration variables:

<table>
<thead>
<tr>
<th>Back end configuration variable</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm-slapdDbInstance</td>
<td>ldapdb2</td>
</tr>
<tr>
<td>ibm-slapdDbName</td>
<td>ldapdb2</td>
</tr>
<tr>
<td>ibm-slapdDbUserId</td>
<td>ldapdb2</td>
</tr>
</tbody>
</table>

LDAP connections to DB2

The LDAP server maintains a certain number of connections to the DB2 servers.
This number is controlled by the ibm-slapdDbConnections parameter in the
slapd32.conf file. By increasing the number of DB2 connections, LDAP can increase its level of concurrency and can improve search throughput performance. On the AIX operating system, the maximum number of connections is limited to 8, unless you use local loopback to increase the number of allowable database connections. See “Using a local loopback connection to DB2” on page 10 for more information.

The following table contains the default settings by operating system:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>ibm-slapdDbConnections</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX</td>
<td>6</td>
</tr>
<tr>
<td>Linux</td>
<td>6</td>
</tr>
<tr>
<td>Solaris</td>
<td>5</td>
</tr>
<tr>
<td>Windows®</td>
<td>9</td>
</tr>
</tbody>
</table>

This parameter is set on the LDAP server machine.

Indexes

It is very important to index all attributes used in searches. The following DB2 commands can be used to verify that a particular index is defined. In the following example, the index being checked is principalName:

db2 connect to ldapdb2

db2 list tables for all | grep -i principalName

db2 describe indexes for table ldapdb2.principalName

If the second command fails or the last command does not return three entries, the index is not properly defined. The last command should return the following results:

<table>
<thead>
<tr>
<th>IndexSchema</th>
<th>Index Name</th>
<th>Unique Rule</th>
<th>Number of Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAPDB2</td>
<td>PRINCIPALNAME1</td>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td>LDAPDB2</td>
<td>PRINCIPALNAME</td>
<td>D</td>
<td>2</td>
</tr>
<tr>
<td>LDAPDB2</td>
<td>PRINCIPALNAME2</td>
<td>D</td>
<td>2</td>
</tr>
</tbody>
</table>

3 record(s) selected.

To have IBM SecureWay Directory create an index for an attribute the next time IBM SecureWay Directory is started, do one of the following:

- From the Database Management Tool:
  1. Click Schema>Attributes.
  2. Click Edit Attribute.
  3. On the IBM Extensions tab, select the EQUALITY checkbox under Indexes.
- Edit either /etc/ldapschema/V3.ibm.at or /etc/ldapschema/V3.user.at. Find the line containing the attribute of interest that looks similar to the following:

  ( 1.3.18.0.2.4.318 DBNAME ( 'principalName' 'principalName' )
  LENGTH 256 EQUALITY ORDERING SUBSTR APPROX )

Add the word "EQUALITY" to the list as in the above example.

Note: If you choose this option, you must stop and restart the server for the index to be created.

- Issue the following command:

  ldapmodify -f /ldap/etc/addindex.ldif
The ldif file should look like this:

dn: cn=schema
changetype: modify
replace: attributetypes
attributetypes: ( 1.3.18.0.2.4.318 NAME ( 'principalName' 'principal' ) DESC 'A naming attribute that may be used to identify eUser object entries.' EQUALITY 1.3.6.1.4.1.1466.109.114.2 ORDERING 2.5.13.3 SUBSTR 2.5.13.4 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 USAGE userApplications )-
replace: ibmattributetypes
ibmattributetypes: ( 1.3.18.0.2.4.318 DBNAME( 'principalName' 'principalName' ) ACCESS-CLASS normal LENGTH 256 EQUALITY ORDERING SUBSTR APPROX )

Note: Dropping the LDAP_DESC_DEID index will in many cases improve performance, especially search operations performance. To drop the LDAP_DESC_DEID index, complete the following steps:

1. Stop slapd.
2. Connect to ldapb2.
3. Run the following command:
   DROP INDEX LDAPDB2.LDAP_DESC_DEID
4. Stop and restart DB2.

Update performance and SMP systems

The SecureWay Directory serializes updates to the LDAP master server. This means update performance does not benefit from having more than one processor on the LDAP master server.

Search performance benefits from multiple processors on the LDAP server.

LDAP cache

The LDAP cache is highly efficient in terms of size and speed. An LDAP search that accesses the LDAP cache is dramatically faster than one that requires a connection to DB2, even if the information is cached in DB2.

The disadvantage to using the LDAP cache generally is the cache invalidation that occurs on update operations. The DB2 cache is not as sensitive to update operations.

The LDAP cache has three components: a filter cache, an entry cache, and an ACL cache. The filter cache consists of actual queries on the requested attribute filters and resulting entry identifiers that matched. On an update operation, all filter cache entries are invalidated.

The entry cache contains the actual entry information for all entry IDs that have been cached.

The ACL cache caches ACL information for individual entries. Even though all filter cache entries are invalidated, the entry cache and ACL cache remain and can result in improved performance. Hitting the LDAP entry cache generally is faster than going to DB2 for the entire search, but it is not typically as fast as hitting the LDAP filter cache and avoiding DB2 altogether.

Following are some LDAP cache recommendations:
• If the available memory and number of entries make it possible to cache all entries, and updates are infrequent, you should define the cache large enough to cache all entries.

• If a subset of entries is significantly more active than the entire set of entries, and there are infrequent updates, you should define the cache large enough to cache the subset of entries.

• If updates are frequent, attempt to balance the LDAP cache and DB2 cache such that the DB2 cache holds the indexes for all entries and the LDAP cache holds the entries for each user. If this combination is set correctly, it can eliminate disk accesses for authentications.

• The effect of updates can be reduced by searching to LDAP replicas and using delayed replication. The length of time it takes to reload the filter cache can vary depending upon cache sizes defined to LDAP and DB2. Refer to “Replication” on page 31 for more information on replication.

In most cases, experimentation between LDAP cache size and DB2 cache size is necessary. To allow for experimentation it is best to install large amounts of physical memory in the LDAP servers. Several gigabytes of RAM is not unreasonable when the directory contains millions of entries.

The LDAP cache may not scale well with SMP machines. If performance benefits for LDAP caching are not seen on SMP machines, it would be a good idea to experiment with different numbers of processors enabled.

On Solaris machines, processors can be enabled and disabled using the `psradm` command. Also, refer to the `psrinfo` command. For AIX machines, refer to Chapter 4, “AIX Operating system tuning” on page 25 for tuning parameters that affect SMP performance.

### LDAP Cache on Solaris and AIX operating systems

#### Setting the LDAP cache

By default the Entry and ACL cache size (Entry, Filter, and ACL) is 25000 entries. An administrator can change workload values in `slapd32.conf`. There is no optimal cache size for all LDAP servers, this must be determined on an individual basis. The LDAP monitor tool is a good tool to help administrators determine the appropriate cache sizes. The monitor search returns the size of the entry and filter cache as well as the number of cache hits and misses.

Keep in mind that in order for an entry to be added to the cache it must first be a cache miss. Analysis of this information over a period of time can help determine if the server needs a larger or smaller cache.

To prevent large uncommon searches from overwriting useful cache entries you may want to set the `RDBM_CACHE_BYPASS_LIMIT` and `RDBM_ENTRY_CACHE_BYPASS` environment variables. Queries that match a number of entries greater than the value of `RDBM_CACHE_BYPASS_LIMIT` will not be entered into the filter cache. By default the value is 100. When the `RDBM_ENTRY_CACHE_BYPASS` variable is set, the cache bypass limit applies to the entry cache in addition to the filter cache. See “Setting front end configuration variables” on page 5 for more information on setting front end configuration variables.
AIX considerations when defining a large LDAP cache

On the AIX operating system, the LDAP cache is allocated from a physical memory segment that you can set to up to 256 MB by updating the /etc/security/limits file. This must be done with care because the AIX data segment and stack segment share the same physical segment. See “Setting the slapd executable to run with large memory model” on page 12 for information on how to set the limit higher than 256 MB.

The following sections show you how to maximize the physical memory available to IBM SecureWay slapd process by:

- Using a local loopback connection to DB2 instead of the shared memory connections that are used by default. This is accomplished by completing the instructions in “Configuring TCP/IP”, “Configuring DB2 for local loopback” and “Configuring slapd32.conf file” on page 11.
- Using the large memory model for the slapd executable to avoid the single hardware segment limitation for data and stack. This involves a binary modification to the executable.

Using a local loopback connection to DB2
To use a local loopback connection to DB2 instead of the shared memory connections that are used by default, complete the following sets of instructions.

Note: Once local loopback is configured, changelog will not work.

Configuring TCP/IP:
- In /etc/services, add the two entries listed below. The names ldapbcon and ldapbint are used in this example. You can use other names as long as they are used consistently throughout the procedure.

```
ldapbcon 3700/tcp # ldapdb2 loopback servicename port
ldapbint 3701/tcp # reserved for db2 interrupts
```

Note: Make sure the port you choose is not already in service. You can pick any two sequential unused port numbers.

Configuring DB2 for local loopback:
1. Catalog the TCP/IP Node
   The hostname and servicename (e.g., ldapbcon used in the Configuring TCP/IP example above) of the target server must be cataloged in the DB2 node directory. Open a DB2 command environment and issue the following command:
   ```
   DB2 CATALOG TCPIP NODE <node_name> REMOTE <hostname> SERVER <servicename>
   ```
2. Configure the Database Manager configuration for TCP/IP
   ```
   DB2 UPDATE dbm cfg USING SVCENAME <servicename>
   ```
3. Catalog the database and node:
   ```
   DB2 CATALOG DATABASE <dbname> AS <alias> AT NODE <node_name>
   ```
4. Export the following environment variable:
   ```
   export DB2COMM=TCPPIP
   ```
5. Restart DB2:
   ```
   db2stop; db2start
   ```
The following are command examples with some suggested names.

Table 1. Suggested names

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;hostname&gt;</td>
<td>raspberry</td>
</tr>
<tr>
<td>&lt;dbname&gt;</td>
<td>ldapdb2</td>
</tr>
<tr>
<td>&lt;servicename&gt;</td>
<td>ldapdbcon</td>
</tr>
<tr>
<td>&lt;node_name&gt;</td>
<td>loopback</td>
</tr>
<tr>
<td>&lt;alias&gt;</td>
<td>ldaploop</td>
</tr>
</tbody>
</table>

>db2 catalog tcpip node loopback remote raspberry server ldapdbcon
>db2 update dbm cfg using svcname ldapdbcon
>db2 catalog database ldapdb2 as ldaploop at node loopback
>export DB2COMM=tcpip

Configuring slapd32.conf file:
1. Edit the slapd32.conf file.
2. Change the database name field (ibm-slapdDbName) to the <alias> name. In the example commands above this is ldaploop.
3. Find the configuration section under dn: cn=Front End, cn=Configuration.
   
   **Note:** If you don’t have this section, see the example below on how to add it.
4. Add a statement to set the DB2CP environment variable to the correct code page. For example, if you are using a local US code page DB2 database, add:

   ibm-slapdSetenv: DB2CP=819

   If you are using the default UTF database, also add DB2CODEPAGE. For example:

   ibm-slapdSetenv: DB2CP=1208
   ibm-slapdSetenv: DB2CODEPAGE=1208

   **Note:** If you are unsure of the proper code page value for your database, it is the database code page returned from the command

   "db2 get db cfg for <dbname>"

   The following example output shows the default code page value for the database:

   Database Configuration for Database ldapdb2

   Database configuration release level = 0x0900
   Database release level = 0x0900
   Database territory = US
   Database code page = 819
   Database code set = 1SO8859-1
   Database country code = 1
5. Set the cache size.

   ibm-slapdSetenv: RDBM_CACHE_SIZE=<number of entries to cache>

   **Example:** Below is an example of a Front End section of the slapd32.conf file that shows settings for a UTF code page and RDBM cache size. If there is no cn=Front End, cn=Configuration entry in the server’s configuration file, you can add lines like those in the example to the bottom of the file.
In this example, the RDBM cache size is set to 25000 entries. The environment variable LDAP_CONCURRENTRW is set to ON for a further boost in performance by allowing read and write operations to be handled concurrently in the server.

dn: cn=Front End, cn=Configuration
 cn: Front End
 ibm-slapdSetenv: DB2CODEPAGE=1208
 ibm-slapdSetenv: DB2CP=1208
 ibm-slapdSetenv: RDBM_CACHE_SIZE=25000
 ibm-slapdSetenv: LDAP_CONCURRENTRW=ON
 objectClass: top
 objectClass: ibm-slapdFrontEnd

**Setting the slapd executable to run with large memory model**

**Note:** You must configure local loopback before enabling large memory model. See “Using a local loopback connection to DB2” on page 10 for more information.

1. Make a copy of the slapd.
2. As root, run the setmaxmem script against the slapd executable.
   
   `/usr/ldap/sbin/setmaxmem`

**Verifying the procedure**

1. From a db2cmd window run the following command:
   
   `db2 connect to <alias> user ldapdb2`

2. Start the server.

3. Use the LDAP Monitor to view its cache size.
   
   `ldapsearch -b cn=monitor -s base objectclass=*`

   This search returns the entry and filter cache sizes and some statistics on cache hits and misses. The entry cache size should be what was set in the slapd32.conf file in this procedure.

**Unconfiguring DB2 local loopback**

If you want to return your database configuration to the original shared-memory connection, complete the following steps:

1. Stop the directory server (slapd).
2. Change the database name field (ibm-slapdDbName) in the slapd32.conf file back to the original database name. For example, using the default database configuration:
   
   `ibm-slapdDbName: ldapdb2`

3. Uncatalog the database alias used for local loopback.
   
   `db2 uncatalog database <dbalias>`

4. Uncatalog the TCP/IP node use for local loopback.
   
   `db2 uncatalog node <nodename>`

5. Undo the SVCENAME setting in the database manager configuration.
   
   `db2 update dbm cfg using svcename NULL`

6. Clear the DB2COMM environment variable (assuming it was not set for reasons other than local loopback). Unset DB2COMM

7. Restart DB2.
   
   `db2stop; db2start`

8. As root, reset slapd to use the small memory model
   
   `/usr/ldap/sbin/resetmem slapd`
Estimating LDAP cache size

The procedure documented in this section shows you how to estimate memory requirements for three LDAP caches: LDAP Entry Cache, LDAP Filter Cache and LDAP ACL Cache.

Although the largest portion of memory is allocated to the LDAP Entry Cache, the "memory per entry" number derived from this procedure represents the total memory requirements associated with each entry in all three caches.

**Note:** In SecureWay Directory 3.2.2, the amount of memory required for the LDAP entry cache size increased significantly from the among required in SecureWay Directory 3.2.1. If you are migrating from SecureWay Directory 3.2.1 to SecureWay Directory 3.2.2, and need to preserve your memory resources, we recommend that you reduce the RDBM_CACHE_SIZE value by half. For example, if you had RDBM_CACHE_SIZE=5000 in SecureWay Directory 3.2.1, you should set it to RDBM_CACHE_SIZE=2500 in SecureWay Directory 3.2.2.

To estimate memory requirements for the LDAP caches

1. Set the following front end configuration variables in the slapd32.conf file:
   - RDBM_CACHE_BYPASS_LIMIT=0
   - RDBM_CACHE_SIZE=150000
   - RDBM_FCACHE_SIZE=1000
   - RDBM_ACLCACHE_SIZE=150000

   **Note:** Set the Entry Cache and ACL Cache high to allow for unbounded growth within these caches. After you have estimated the memory requirements for the LDAP caches, you can establish a value that works best in your environment by monitoring the hit rate of the caches during normal production. You can then adjust the values based on the memory you have available and the rate of repetitive access to directory entries in your environment.

2. Start or restart the slapd server.

3. Execute the command to monitor LDAP performance and ascertain the number of entry cache misses. This will be the value for the entry_cache_miss attribute. See "[Monitoring Performance” on page 31] for more information on the entry_cache_miss attribute. After starting the server, this value should always be zero.

4. Execute some LDAP search queries (no updates) which represent the search workload for the server. This will cause the number of cache misses to grow. Continue until the number of cache misses approaches 10,000 while running some scripts to continuously monitor the server helps. Assume the value of entry_cache_miss equals $m_1$.

5. Find the process size. On UNIX systems, you can use the ps command, on NT, use the task manager or performance monitor. Assume the process size is $s_1$ KB.

6. Execute more LDAP queries until the number of cache misses grows to 50000 or more. Assume the value of entry_cache_miss equals to $m_2$ at this point.

7. Find the process size again. Assume this process size is $s_2$ KB.

8. Estimated memory requirement = $(s_2-s_1)/(m_2-m_1)$ KB/entry.

   This number represents the growth in the LDAP caches associated with a significant number of directory entries. Use this number to roughly estimate the amount of memory required to cache the desired number of directory entries. For smaller directories, it might be possible to provide enough memory to
cache the entire directory. Alternately, this number can be used to estimate the size of caches that could be accommodated by the available memory in the server.

Reloading directory data

There are several performance enhancements in this release that require underlying database table definition changes. The enhancements include:

- The elimination of Long Var Char (LVC) columns to improve update performance
- Improved performance for Large Groups
- Improved Access Control List (ACL)

If you have directories created with an earlier version of SecureWay Directory and have migrated to version 3.2.2, you will need to unload and reload the data in the directories to benefit from these enhancements.

To unload and reload data:

1. Stop the directory server.
2. Make a backup of your database. This backup can be discarded if the following steps are successful. See "Backup and restore on page 22" for help in backing up databases using DB2.

   **Note:** If you have customized any of your DB2 database configuration parameters, you will also need to update the new database you will create in step 4. You can run the following command to get the current database configuration.

   ```
db2 get database configuration for ldapdb2 > old_dbconfig
   ```

   Perform a **reorgchk** on the old database for later verification of the new database. The **reorgchk** provides information such as the table sizes and indexes defined. It may not be necessary, but in some cases can provide quick insight into any differences between the new and old database:

   ```
db2 reorgchk update statistics on table all > old_reorgchk.out
   ```

3. Create an LDIF file from your directory contents. This file will be used to reload the data into the directory.

4. Run the configuration utility **ldapxcfg** and select the option to configure a database. This will destroy the existing database and create a new one. Set the **code page** and **change log** options to meet your needs. If you are using a custom database, create a new one and use **ldapxcfg** to configure the server to use it.

5. Run **bulkload** with the file created in step 3.

   For example, if the LDIF file was named /home/ldapdb2/dir321.ldif, the command would be:

   ```
bulkload -i /home/ldapdb2/dir321.ldif
   ```

   The time required to complete loading will depend on the number of entries in your directory.

   **Notes:**
   
   a. This step must be performed as the root ID for a UNIX operating system, or as a user in the Administrator’s group on a Windows operating system.
If logged in as the root user on a UNIX® system, it is possible to switch to the ldapdb2 user as follows:

```bash
su - ldapdb2
```

To log on as the database administrator on a Windows operating system, enter the following at the command prompt window (where ldapdb2 is the defined user of the LDAP database):

```bash
runas /user:ldapdb2 db2cm
```

b. Before running the `bulkload -i` option, run `SCHEMACHECK=ONLY`. Also make sure that your LDAPIMPORT environment variable is set to file system that can store enough temporary space to build the required information to load into your database. See “Bulk loading (bulkload)” on page 29 for more information.

6. If you previously customized any other DB2 database configuration parameters, you will need to update these in the newly created database. Using information obtained in step 2, reset your DB2 configuration parameters using the following command:

```bash
db2 update database configuration for ldapdb2 using <parm name>
<parm value>
```

7. Perform a `reorgchk` (see “Database optimization, statistics, reorganization check, and reorganization” on page 17 for further information) as follows:

```bash
- db2 connect to ldapdb2
- db2 reorgchk update statistics on table all > new_reorgchk.out
```

8. Start the server and confirm the data was loaded correctly using the Database Management Tool or the command line `ldapsearch` command. If this directory server was configured as a master, and there are other replicas running SecureWay Directory version 3.2.2 that you would also like to unload and reload, perform the following additional steps:

a. Stop the master server, and use WebAdmin to make a backup of the database. See “Backup and restore” on page 22 for help in backing up databases using DB2.

b. Stop all replica servers.

c. Restart the master server.

d. Use the backup from 8a to restore one replica.

e. Restart the replica.

f. Repeat steps 8d and 8e for each replica.

**Note:** No change log data is migrated by this procedure. Make sure that any system relying on change log data is updated before performing these steps.
Chapter 3. DB2 tuning

IBM SecureWay Directory uses DB2 to store directory data. Tuning DB2 can improve LDAP performance.

Topic Index:
• Database optimization, statistics, reorganization check, and reorganization
• DB2 buffer pool tuning
• Other DB2 Configuration Parameters
• Backup and Restore

Notes:
1. In all DB2 command examples it is assumed that ldapdb2 is ibm-slapdDbInstance and ibm-slapdDbName. It is also assumed that the user is logged in as ibm-slapdDbUserId. If logged in as the root user on a UNIX system, it is possible to switch to the ldapdb2 user as follows:
   su - ldapdb2

   To log on as the database administrator on a Windows operating system, enter the following at the command prompt window (where ldapdb2 is the defined user of the LDAP database):
   runas /user:ldapdb2 db2cm

   See "Setting back end configuration variables" on page 6 for more information.

2. If you have any trouble running the DB2 commands, check to ensure the following:
   • The ID trying to run the DB2 commands is a user in the dbsysadm group (Unix) or a member of the Administrator group (Windows). Only users listed as database administrators can execute the DB2 commands. This includes the DB2 instance owner (the default is ldapdb2) and root.
   • DB2 environment variables have been established by running db2profile (if not, the db2 get and db2 update commands do not work). Script file db2profile is located in the sqllib subdirectory under the instance owner’s home directory. If you need to tailor this file, follow the comments inside the file to set your instance name, user paths, and default database name (the default path is /home/ldapdb2/sqllib/db2profile).
   • For additional stability and performance enhancements, upgrade to the latest version of DB2.

Database optimization, statistics, reorganization check, and reorganization

The SecureWay Directory Optimize button uses DB2 “runstats” to update statistical information used by the query optimizer for all the LDAP tables. In many cases, performance may be improved significantly by running statistics first. This is especially true after a large amount of data has been loaded. In 3.2.2, the command line equivalent of the Optimize button is issuing the following command for all the LDAP tables:

DB2 RUNSTATS ON TABLE table-name AND DETAILED INDEXES ALL SHRLEVEL REFERENCE
Run the following commands for more detailed list of runstats that improve performance:

```
DB2 RUNSTATS ON TABLE table-name WITH DISTRIBUTION AND DETAILED INDEXES ALL SHRLEVEL REFERENCE
DB2 RUNSTATS ON TABLE ldapdb2.objectclass WITH DISTRIBUTION AND DETAILED INDEXES ALL SHRLEVEL REFERENCE
```

Another important and often overlooked DB2 tuning command is `reorgchk`. The `reorgchk` command can improve both search and update operation performance. In addition to doing a "runstats", it also provides an indication of what results if a table gets reorganized.

After a number of updates have been performed against DB2, table indexes become sub-optimal and performance can degrade dramatically. This situation can be corrected by performing a DB2 `reorgchk` as follows:

```
db2 connect to ldapdb2
db2 reorgchk update statistics on table all
```

`reorgchk`, as shown above, does two things. It updates statistical information to the DB2 optimizer to improve performance, and reports statistics on the organization of the database tables.

`reorgchk` needs to be run periodically. For example, `reorgchk` needs to be run after a large number of updates have been performed. Note that LDAP tools such as `ldapadd`, `ldif2db` and `bulkload` can potentially do large numbers of updates that require a `reorgchk`. The performance of the database should be monitored and a `reorgchk` performed when performance starts to degrade. See "Monitoring Performance" on page 31 for more information.

`reorgchk` must be performed on all LDAP replicas, since each uses a separate database. The LDAP replication process does not include the propagation of database optimizations.

In general, reorganizing a table takes more time than running statistics. Therefore, performance may be improved significantly by running statistics first. If not, the data in the tables and indexes may not be arranged efficiently, so reorganization may help.

Due to a new performance enhancement in this version that caches prepared DB2 statements, you must stop and restart slapd in order for DB2 changes to take effect.

**Database table organization**

The organization of the data in DB2 can be tuned. Tuning organizes the data on disk in a sorted order. Sorting the data on disk is beneficial only when accesses occur in a sorted order, which is not typically the case. For this reason, organizing the table data on disk typically yields little change in performance.

The first step in the reorganization procedure is to perform a `reorgchk db2` command. The command is as follows:

```
db2 reorgchk update statistics on table all >reorgchk.out
```

The output of this command is routed to a file named `reorgchk.out` in the above example. This makes it easy to view the results after issuing the command.
This command gives organizational information about the database. The next step is an iterative process of finding the tables and indexes that need reorganizing and attempting to reorganize them. This can take a long time. The time to perform the reorganization and the `reorgchk` grows as the DB2 database size increases.

After reorganizing a group of tables or indexes, a new `reorgchk` must be run to generate new statistics. The output from `reorgchk` can then be used to determine whether the reorganization worked and whether it introduced other tables and indexes that need reorganizing.

The following explains the commands and guidelines for identifying and reorganizing the tables and indexes in this iterative process:

The `reorgchk` update statistics has two sections, the first section is the table information and the second section is the indexes. Use this command to reorganize the tables with an asterisk in the last column:

```
db2 reorg table <table name>
```

where `<table name>` is the name of the table to be reorganized, for example, `LDAPDB2.LDAP_ENTRY`.

Generally speaking, since most data in LDAP is accessed by index, reorganizing tables is usually not as beneficial as reorganizing indexes.

Use this command to reorganize the indexes with an asterisk in the last column:

```
db2 reorg table <table name> index <index name>
```

where `<table name>` is the name of the table. For example, `LDAPDB2.LDAP_ENTRY`.

And where `<index name>` is the name of the index. For example, `SYSIBM.SQL000414155358130`.

Here are some guidelines on performing a reorganization:

- If the number on the column that has an asterisk is close to the recommended values described in the header of each section and one reorganization attempt has already been done, it is probably okay to skip a reorganization on that table or index.

- In the table `LDAPDB2.LDAP_ENTRY` there exists a `LDAP_ENTRY_TRUNC` index and a `SYSIBM.SQL` index. Preference should be given to `SYSIBM.SQL` index, if attempts to reorganize them seem to alternate between one or the other needing reorganization.

- Reorganize all the attributes that you want to use in searches. In most cases you will want to reorganize to the forward index, but in cases with searches beginning with `"*"`, reorganize to the reverse index.

For example:

```
Table: LDAPDB2.SECUUID
LDAPDB2 RSECUUID <- This is a reverse index
LDAPDB2 SECUUID <- This is a forward index
LDAPDB2 SECUUIDI <- This is an update index
```
Along with the DB2 Optimization Tools (i.e., reorgchk), DB2 buffer pool is one of the most significant DB2 performance tunings. Unlike the Optimization Tools, which need to be run periodically when significant changes have occurred with the data, buffer pool tuning typically only needs to be done once.

A DB2 buffer pool is a data cache between LDAP and the physical DB2 database files for both tables and indexes. If there are no buffer pool(s), then all database activity results in disk access. If the size of each buffer pool is too small, LDAP has to wait for DB2 disk activity to satisfy DB2 SQL requests. If one or more buffer pools are too large, memory on the LDAP server may be wasted. If the total amount of space used by all buffer pools is larger than physical memory available on the server, operating system paging (disk activity) will occur.

Although it is not possible to cache an entire table containing millions of rows, it is possible that the indexes for an entire table can be cached. Cached indexes can provide a significant boost to performance, because they can make locating the data on disk very fast.

### Buffer pool tuning for previous versions of SecureWay Directory

In earlier versions of SecureWay Directory, the LDAP directory database used the default single DB2 buffer pool called IBMDEFAULTBP. The general tuning guideline suggested defining the DB2 cache from 50% to 75% of the machine’s physical memory and set it using the BUFPAGE configuration parameter. For example, in earlier versions of SecureWay Directory, you run the following commands to set the default single buffer pool to 1.2 GB:

```sql
db2 update db config for ldapdb2 using buffpage 300000
db2 alter bufferpool ibmdefaultbp size -1
```

**Note:** If you have not recreated your database using SecureWay Directory version 3.2.2, you will still need to use the above method.

In earlier versions of SecureWay Directory, you get the current settings by running the following commands:

```sql
db2 connect to ldapdb2
db2 get database configuration for ldapdb2 | grep BUFPAGE
db2 "select bpname,npages,pagesize from SYSIBM.SYSbufferpools"
```

The following example contains the expected output from the select statement:

```sql
$ db2 "select bpname,npages,pagesize from SYSIBM.SYSbufferpools"
```

<table>
<thead>
<tr>
<th>BPNAME</th>
<th>NPAGES</th>
<th>PAGESIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMDEFAULTBP</td>
<td>-1</td>
<td>4096</td>
</tr>
</tbody>
</table>

1 record(s) selected.

**Note:** If the NPAGES column is set to -1, it means that the BUFPAGE size parameter is being used to determine the Buffer Pool size.
Buffer pool tuning for SecureWay Directory version 3.2.2

In SecureWay Directory version 3.2.2, the LDAP directory database is created with an additional tablespace (LDAPSPACE) and buffer pool (LDAPBP) using a 32K page size. Since IBMDEFAULTBP uses the default 4K page size, and the LDAPBP uses the 32K page size, it is no longer possible to use the single database configuration parameter for BUFFPAGE. If you have created a new database using SecureWay Directory version 3.2.2, you need to set each buffer pool size separately using the alter buffer pool commands.

The following examples shows two buffer pools being set to a total size of 1.2 GB:

```
db2 alter bufferpool ibmdefaultbp size 7900
db2 alter bufferpool ldapbp size 3300
```

As a general guideline, a 3 to 1 ratio between memory allocated to the IBMDEFAULTBP (4K pages) and LDAPBP (32K pages) is good for performance. By default, the IBMDEFAULTBP is created with a size of 29500 (4K) pages. By default, the LDAPBP bufferpool is created with a size of 1230 (32K) pages. On a 256 MB minimum LDAP Server, this allocates roughly 60% of physical memory to the DB2 buffer pools.

To get the current DB2 buffer pool sizes, run the following commands:

```
db2 connect to ldapdb2
db2 "select bpname,npages,pagesize from SYSIBM.SYSbufferpools"
```

The following example output shows the default settings for the 256 MB example above:

```
BPNAME          NPAGES  PAGESIZE
------------------- --------- ---------
IBMDEFAULTBP     29500    4096
LDAPBP           1230     32768
```

2 record(s) selected.

**Attention:** If you are using a machine with less than 256 MB, or you have a machine with 256 MB and a read-only directory server, you need to reduce the buffer pool size by issuing the following commands:

```
db2 alter bufferpool ibmdefaultbp size 9800
db2 alter bufferpool ldapbp size 400
```

Guidelines for when to use LDAP Cache or buffer pool cache

Cached tables can provide performance benefits, but the LDAP cache is generally more efficient as a means of caching LDAP searches. On the other hand, parts of the LDAP cache get invalidated on updates and must be reloaded before performance benefits return. Some experimentation between the two caching schemes is probably appropriate.

If there are updates interspersed with authentications and searches, it may be best to allocate enough available physical memory in DB2 cache to hold the indexes and allocate the remaining available physical memory to LDAP cache.
Other DB2 configuration parameters

Performance benefits can come from setting other DB2 configuration parameters.

The current setting of parameters can be obtained by issuing the following command:

```
db2 get database configuration for ldapdb2
```

This command returns the settings of other DB2 configuration parameters as well.

The following command also shows the DB2 configuration parameters for the entire ldapdb2 instance:

```
db2 get database manager configuration
```

To set the DB2 configuration parameters use the following syntax:

```
db2 update database configuration for ldapdb2 using \ <parm name> <parm value>
db2stop
db2start
```

where `<parm name>` is the parameter to change and `<parm value>` is the value it is to be assigned.

Changes to DB2 configuration parameters do not take effect until the database is restarted with `db2stop` and `db2start`.

The following are some DB2 parameters that you might want to modify to optimize performance:

- `APPLHEAPSZ`
- `PCKCACHESZ`
- `SORTHEAP`
- `LOGFILSIZ`
- `DBHEAP`
- `APP_CTL HEAP_SZ`
- `LOCKLIST`

Backup and restore

Typically, the fastest way to backup and restore the database is with DB2 `backup` and `restore`. The LDAP alternatives, like `db2ldif` and `ldif2db`, are generally much slower in comparison. Although the IBM SecureWay Directory `bulkload` tool is a fast alternative to `ldif2db`, the performance of DB2 `backup` and `restore` is still generally better and easier to use.

The main disadvantages of using DB2 `backup` and `restore` is that the utilities may not work across dissimilar levels of SecureWay Directory or IBM Directory, or across dissimilar operating systems. For example, DB2 `backup` and `restore` will not function across SecureWay versions 3.2.1, 3.2.2, and IBM Directory 4.1, or across the AIX and Solaris operating systems.

Be aware that if you restore over an existing database, any tunings on that existing database are lost.
Check all DB2 configuration parameters after performing a restore. Also, run `reorgchk` after a restore, if it is not known whether a `reorgchk` was performed before the database was backed up. The DB2 commands to perform a backup and restore are as follows:

```
db2 force applications all
db2 backup db ldapdb2 to <directory or device>
db2 restore db ldapdb2 from <directory or device> replace \\ existing
```

where `<directory or device>` is the name of a directory or device where you want the backup to be placed or from where the restore is to come.

The most common error that occurs on a restore is a file permission error. Following are some reasons for this error:

- The DB2 instance owner does not have permission to access the specified directory and file. One way to solve this is to change directory and file ownership to the DB2 instance owner. A command similar to the following will do this:
  ```
  chown ldapdb2 <file or dir>
  ```

- The backed-up database is spread across multiple directories, and those directories do not exist on the target machine of the restore. Spreading the database across multiple directories is accomplished with a redirected restore. To solve this problem, you should either create the same directories on the target machine or do a redirected restore to specify the proper directories on the new machine. If creating the same directories, make sure the owner of the directories is `ldapdb2`.

`backup` and `restore` are required to get an LDAP replica initially synchronized with an LDAP master. `backup` and `restore` are also required anytime the master and replica get out of sync. A replica can become out of sync if it is undefined to the master. The master will not save updates on a propagation queue for a replica that is not defined.

If a newly configured master LDAP directory is to be loaded with initial data, bulk loading utilities can be used to speed up the process. This is another case in which the replica is not informed of updates and a manual backup and restore is required to get the replica synchronized with the master.
Enabling large files

The underlying AIX files that hold the contents of a large directory can grow beyond default size limits imposed by the AIX operating system. If the size limits are reached, the directory ceases to function correctly. The following steps make it possible for files to grow beyond the default limits on an AIX operating system:

1. When you create the file systems that are expected to hold the directory’s underlying files, you should create them as "Large File Enabled Journaled File Systems". The file system containing the db2 instance’s home directory (usually /home/ldapdb2), and, if bulkload is to be used, are file systems that can be created this way. The file system containing the bulkload temporary directory can be specified via the LDAPIMPORT environment variable.

2. Set the soft file size limit for the root, ldap, and the db2 instance owner (usually ldapdb2) users to -1. A soft file size limit of -1 for a user specifies the maximum file size for that user as unlimited. The soft file size limit can be changed using the "smitty chuser" command. It is necessary for each user to log off and log back in for the new soft file size limit to take effect. You will also need to restart DB2.

Setting MALLOCMULTIHEAP

The MALLOCMULTIHEAP environment variable can improve LDAP performance on SMP systems. To set it, run the following command just before starting slapd:

```
export MALLOCMULTIHEAP=1
```

The disadvantage to using MALLOCMULTIHEAP is increased memory usage.

It might take less memory, yet be less of a performance benefit, if it is set as follows:

```
export MALLOCMULTIHEAP=heaps: <numprocs>+1
```

where `<numprocs>` is the number of processors in the multiprocessor system.

More information on MALLOCMULTIHEAP can be found in the AIX documentation.
Viewing slapd/slapdd environment variables (AIX operating system only)

To view the environment settings and variables for your slapd/slapdd process, run the following command:

```bash
ps ewww <PID>
```

where `PID` is the slapd/slapdd process ID.

Example output for a PID of 20788:

```
$ ps ewww 20788

  PID  TTY STAT  TIME COMMAND
  20788  pts/0  A   20:04 /usr/bin/slapd -f /etc/slapd32.conf _=/usr/bin/slapd MA
  NPATH=/usr/dt/man:/usr/share/man:/usr/lpp/info:/usr/lpp/ssp/man DSHPATH=/usr/loc
  al/bin/setupserver:/usr/lpp/ssp/rcmd/bin:/usr/lpp/ssp/bin:/usr/lpp/ssp/kerberos/b
  in:/usr/bin:/etc:/usr/sbin:/usr/lib:/sbin:/usr/bin/X11:/sbin:/u/shared/bin:/usr/local
  /bin:/usr/local/bin:/u/shared/bin LANG=en_US LOGIN=root IMQCONFIGCL=/etc/IMNSear
  ch/dbcscl PATH=/usr/local/bin/setupserver:/usr/lpp/ssp/rcmd/bin:/usr/lpp/ssp/bin
  :/usr/bin:/etc:/usr/sbin:/usr/lib:/sbin:/u/shared/bin:/usr/local/bin:/u/shared/bin
  LANG=en_US LOGIN=root IMQCONFIGCL=/etc/IMNSearch/dbcscl PATH=/usr/local/bin/se
  tsetupserver:/usr/lpp/ssp/rcmd/bin:/usr/lpp/ssp/bin:/usr/bin:/etc:/usr/sbin:/usr/lib
  :/sbin:/u/shared/bin:/usr/local/bin:/u/shared/bin LANG=en_US LOGIN=root IMQCONFIGCL

```

IBM SecureWay Version 3.2.2 Directory Tuning Guide
Chapter 5. Hardware tuning

Disk speed improvements

With millions of entries in LDAP server, it can become impossible to cache all of them in memory. Even if a smaller directory size is cacheable, update operations must go to disk. The speed of disk operations is important. Here are some considerations for helping to improve disk drive performance:

- Use fast disk drives
- Use a hardware write cache
- Spread data across multiple disk drives
- Spread the disk drives across multiple I/O controllers
- Put log files and data on separate physical disk drives
Chapter 6. SecureWay Directory features

The sections in this chapter briefly describe additional performance related SecureWay Directory features.

Topic Index:
- Bulk loading (bulkload)
- Replication
- Monitoring Performance
- When to configure LDAP change log

Bulk loading (bulkload)

If a large number of entries are to be added at one time, or if the LDAP directory is to be initialized from a known, large set of data, using the LDAP bulkload utility should be considered.

bulkload uses an LDIF file to place information in the LDAP database. In performing this function, bulkload reads the contents of the LDIF file and creates intermediate files. After the LDIF information is stored as intermediate files, the contents of the files are put into DB2. Depending on the data stored in the LDIF files, the intermediate file creation will require storage space 2 1/2 to 5 times the size of the ldif file. bulkload by default will use the /tmp directory to hold the intermediate files. The location of the intermediate files may be changed by setting the value of the LDAPIMPORT environment variable. The main advantage of using the bulkload utility is its speed. The main disadvantage to bulkload is that it requires the directory server to be down while it is running.

Data added via bulkload is not replicated. If replicas are defined, they must be manually synchronized with the updated database. This can be accomplished by bulkloading the data on the master and then using backup and restore to synchronize the replicas. Refer to “Replication” on page 31 for more information.

After a large bulkload, you will need to do a reorgchk for optimal performance.

Before running bulkload, set the following environment variables:
- LDAPIMPORT
  Set this to a file system with enough space for bulkload to write temporary files.
- SCHEMACHECK

  Note: To set the SCHEMACHECK environment variable on a Windows NT operating system, you will need to open a DB2 command shell:
  
  db2cmd.exe

SCHEMACHECK can be set to YES, NO or ONLY.

Set SCHEMACHECK to ONLY to ensure your ldif files are parsed correctly before the data loads to your LDAP directory:

UNIX operating systems:
  export SCHEMACHECK=ONLY
Windows NT operating system:
    set SCHEMACHECK=ONLY

After setting SCHEMACHECK=ONLY, run
    bulkload -i <LDIF_File>

You should see the following message:
    11 entries passed schema checking out of 11 attempts

After you have parsed all your ldif files to load unset SCHEMACHECK,
physically load your entries to the LDAP directory by running the following
command:
    bulkload -i <LDIF_File>

You can unset the SCHEMACHECK variable by running the following
command:
    - On UNIX operating systems
      unset SCHEMACHECK
    - On Windows operating systems:
      set SCHEMACHECK=

If you have converted your bulkload file into multiple files for loading, you can
use the -c NO option to load the data without creating indexes. Run -c YES on
the final file to create the indexes.

Place the DB2 command and tools in your path by sourcing the environment
db2profile. See example below.

This example shows three separate ldif files being loaded, with indexes added on
the final load. The example assumes you are using a UNIX operating system.

```
#############
# source the db2 environment variables:
#############
./home/ldapdb2/sqllib/db2profile

#############
# confirm that your data can be read/parsed correctly before doing physical load
#############
export LDAPIMPORT=/opt/largeFileSystem
export SCHEMACHECK=ONLY
bulkload -c NO -i file1.ldif
bulkload -c NO -i file2.ldif
bulkload -c YES -i file3.ldif

#############
##Once this completes successfully, do the following
#############
export LDAPIMPORT=/opt/largeFileSystem
unset SCHEMACHECK
bulkload -c NO -i file1.ldif
bulkload -c NO -i file2.ldif
bulkload -c YES -i file3.ldif
```

**Incremental bulkload**

For performance reasons, it is recommended that when more than 250,000 entries
are to be bulk loaded, they be loaded in increments of 250,000. This results in
multiple passes through the `bulkload` utility.
Since each pass through the `bulkload` utility causes table indexes to be dropped and the index creation can take a long time, it is recommended that you run the `bulkload` utility with the `-c NO` option. `bulkload` will not attempt to recreate indexes when this option is specified. On the final `bulkload` use the `-c YES` option to create the indexes.

**Replication**

LDAP supports replication. Through replication, LDAP maintains multiple, synchronized copies of the LDAP master directory server. These copies are called LDAP replica servers. The process whereby the master server sends updates to its replicas is known as propagation.

A tuning parameter related to LDAP replication is the update interval. It specifies how long the LDAP master waits between propagations. Both immediate and delayed updates are supported.

The amount of Central Processing Unit (CPU) usage required in the replica server for propagations is the same with both immediate and delayed updates. The primary difference between the two choices is when the CPU is consumed, not how much CPU is consumed to do the updates. It may be beneficial to delay updates to reduce the cost of cache invalidation that results from updates.

The primary performance cost to replication is the increased time it takes to perform updates on the master. This increased time is the same regardless of the selection of immediate or delayed updates.

For best update performance in a replicated environment, run the LDAP master on the fastest processor available. If the environment is set up so that all searches go only to the replica servers, the LDAP master can be run on a uniprocessor.

**Monitoring Performance**

The following `ldapsearch` command can be used to monitor performance.

```
ldapsearch -h <ldap_host> -s base -b cn=monitor objectclass=*  
```

where `<ldap_host>` is the name of the LDAP host.

The monitor search returns the following attributes of the server:

- **version**: Version of the LDAP server
- **totalconnections**: Total number of connections to the server
- **currentconnections**: Total number of current connections
- **maxconnections**: Configured maximum number of connections
- **writewaiter**: Number of threads waiting to write
- **readwaiters**: Number of threads waiting to read
- **opsinitiated**: Operations initiated against the server
- **livethreads**: Number of live server threads
- **opscompleted**: Number of operations completed
- **entriessent**: Number of entries sent from the server
- **searchesrequested**: Number of searches requested
- **searchescompleted**: Number of searches completed

---

Chapter 6. SecureWay Directory features 31
filter_cache_size Configured maximum size of the filter cache
filter_cache_current Current size of the filter cache
filter_cache_hit Number of searches that have hit the filter cache
filter_cache_miss Number of searches that have missed the filter cache
entry_cache_size Configured maximum size of the entry cache
entry_cache_current Current size of the entry cache
entry_cache_hit Number of entries returned from entry cache
entry_cache_miss Number of entries returned not from entry cache
currenttime Current time of the search
starttime Start time of the server
en_currentregs Number of events currently registered
en_notificationsssent Number of event notifications sent

**Example**

The following example shows how to calculate the throughput of the server by monitoring the server statistic called `opsinitiated`, which is the number of operations initiated since the LDAP server started.

Suppose the values for the opsinitiated attribute obtained by issuing two `ldapsearch` commands to monitor the performance statistics, one at time t1 and the other at a later time t2, were opsinitiated(t1) and opsinitiated(t2). Then, the average throughput at the server during the interval between t1 and t2 can be calculated as:

\[
\frac{(\text{opsinitiated}(t_2) - \text{opsinitiated}(t_1) - 3)}{(t_2 - t_1)}
\]

(3 is subtracted to account for the number of operations performed by the `ldapsearch` command itself.)

**When to configure LDAP changelog**

SecureWay Directory 3.2.2 has a function called `change log` that results in a significantly slower LDAP update performance. The `change log` should be configured only if needed.

The `change log` function causes all updates to LDAP to be recorded in a separate change log DB2 database (that is, a different database from the one used to hold the LDAP server Directory Information Tree). The change log database can be used by other applications to query and track LDAP updates. The change log function is disabled by default.

One way to check for existence of the `change log` function is to look for the suffix CN=CHANGEOLOG. If it exists, the `change log` function is enabled.
Appendix.

Notices

This information was developed for products and services offered in the U.S.A. IBM might not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM’s suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX                     DB2                     IBM                     SecureWay

Microsoft®, MS-DOS, Windows, and Windows NT® are registered trademarks of Microsoft Corporation

Other company, product, and service names may be trademarks or service marks of others.