IBM Tivoli Identity Manager
Performance Tuning Guide
Version 4.5.1

Issue Date:
2004/02/10

Copyright Notice

Trademarks

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user. Tivoli Systems or 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785, U.S.A.

Table of Contents

1 Introduction

The IBM Tivoli Identity Manager (ITIM) product is a complex piece of software. Due to its complexity, it can be challenging to maximize the use of resources – or in other words, to tune. This tuning guide provides a system administrator with the information needed to tune the application for your environment. Other individuals (such as IBM DB2 or IBM Directory Server administrators) in your organization might offer differing advice. It has been our experience that system administrators know your environment better and their advice might be more accurate for your environment than this tuning document can be.

The ITIM product can be divided into four major pieces: J2EE application servers, the ITIM application, database servers, and directory servers. Some of these pieces can be divided further such as workflow (WF) and user interface (UI) components for the ITIM servers. We will address each of these separately in this document.

The ITIM server can be installed in three different modes: single server, clustered servers, and functional cluster servers. For performance reasons single servers and clustered servers are the same – each machine has both a WF and UI component on one machine. A clustered environment can be considered a group of single servers with regard to tuning. A functional cluster server is different than a regular cluster because in a functional cluster each component (WF/UI) is on a separate machine. In a functional cluster environment you might have any multiple of machines.

This document is a working document, as more information is gathered settings may be added, removed or changed in future releases. It is recommended that you check the IBM web site for the most recent version. To find the most recent version, go to http://www.ibm.com/support/us, select “Search technical support”, type “ITIM Tuning Guide” in the box and click submit.

ITIM configurations vary significantly by platform, operating system, middleware and hardware being used and you must tailor your configuration settings for your deployment. Configurations and settings used in this document include AIX, Windows and Solaris in single server and cluster configurations.

1.1 Vital tunings

There are several thousand different parameters that you can tune for J2EE application servers, ITIM, directory servers, and database servers. This tuning guide talks about a much smaller subset of these parameters.

If you are setting up an acceptance or production environment, read each section and do the applicable tunings for your systems. If you are setting up a test environment and want to get up and running as quickly as possible focus on these areas:

- IBM DB2 - Runstats
- IBM DB2 - Bufferpools
- WebSphere - JVM Size

The IBM DB2 – Runstats tunings is a vital part of ITIM performance when DB2 is used as the database server. See the Initial tunings section to tune this parameter in a newly deployed environment.

1.2 Initial tunings

Most of these tunings can be implemented in a newly deployed environment or an environment that is already deployed. When tuning IBM DB2 in a newly deployed environment it is important to prime your database statistics using the DB2 `runstats` command.

The `runstats` command creates information in the DB2 statistics tables based on information currently in the data tables. In a pristine environment, those tables are empty and the statistics generated from runstats on these tables are useless to DB2. Without adequate information on the distribution of data within these tables, DB2 makes poor choices on how to fulfill the queries used by the ITIM application.

The solution to this problem is to prime your database by doing a small number of transactions, running the runstats command, and then running the full transaction. The IBM DB2 – Runstats section discusses how to use the `runstats` command. Failing to prime the database in this manner can result in poor performance or transaction rollbacks.

For example: If the goal is to load a 20,000 users into ITIM using a DSML feed, load the first 50 to 100 people into ITIM using the DSML file first. After the load completes, run the DB2 `runstats` command and load the remaining users into ITIM with the DSML file.

1.3 Resource Allocation

The tuning values are difficult to manage when more than one piece of middleware is on a given system (like having ITIM, DB2, and IDS all on the same machine). Regardless of configuration, be sure that resources are not over-allocated.

1.3.1 Memory

All middleware components allow you to adjust how much memory they will use. When calculating how to allocate memory to middleware components, keep these things in mind:

- Configuring middleware memory settings too high can result in the operating system swapping memory out to disk if the physical memory is exceeded. **This will result in extremely poor performance and should be avoided.** After setting up or changing the memory values for the middleware, monitor the memory and swap space used to ensure that nothing is being swapped out to disk. If it is, adjust your memory settings to compensate.
- 32-bit processes can only allocate 2GB of RAM. If a 32-bit process is configured to allocate more than 2GB of memory the application could halt or unexpectedly fail (such as IDS).
- IDS has internal caches in addition to the IDS process. Ensure that the size of your caches plus the size of your IDS process does not exceed 2GB. The entry cache size limit determines the number of entries in the cache, not the size of the cache. The size of each cache entry will vary based on the ITIM configuration and any extensions to the base IDS schema. See the IDS – LDAP Cache Sizes section below.
- Although the bufferpools account for a large amount of the memory used by DB2, the application control heaps, the sort heaps, and the statement heaps also use memory. Do not forget to consider these sizes when computing how much memory to allocate to DB2. See the IBM DB2 section below.
- A large part of WebSphere's memory usage is the JVM size. However, the size of the JVM does not set an upper bound on the amount of memory that WebSphere may use. See the J2EE Application Servers – WebSphere section below.

1.3.2 CPU

All the components of ITIM are CPU intensive: ITIM application, J2EE application server, database server, and directory server. This document groups the J2EE application server and the ITIM application together since it is difficult to isolate their CPU usage. The following are some things to keep in mind:

- ITIM 4.4 and 4.5 are effectively single-threaded due to database locking issues. These versions are best suited for machines with fast processors rather than machines with many slower processors. ITIM 4.5.1 can take better advantage of multiple CPU machines.
- Some ITIM processes are single threaded (such as some provisioning calculations) and will be isolated to a single CPU regardless of what version of ITIM you are using.

- Both IDS and DB2 are multi-threaded applications and will perform best on a multiple CPU machine.
- Even in a well tuned environment the system bottleneck may vary between being CPU bound on ITIM, on the directory server, and on the database server. For this reason, the ITIM server, the directory server, and the database server should all be on separate machines. If that is not possible, put the database server and the directory server on the same machine instead of putting one of them on a machine with ITIM.

1.3.3 Disk Space

Each of the three middleware components use different amounts of disk space for various purposes.

- The J2EE application server and ITIM application use disk space beyond their installation size because of log files (such as the `itim.log` file) and MQSeries queues. Adjust the number of archives and size of the `itim.log` file in the `enRoleLogging.properties` file. Make sure that MQSeries has enough disk space for its processing logs (not error logs) to grow. ITIM pushes many entries onto the queues during large provisioning changes causing them to grow.
- IDS uses disk space both from the IDS process (log files like `ibmslapd.log`) and the DB2 database. IDS uses SMS tablespaces which allow the system to manage the amount of disk space used. SMS spaces do not allow the specification of an upper bound so it is important to monitor the amount of disk space used to ensure that the drive does not become full.
- In addition to the tablespaces for the database data, DB2 uses disk space for the transaction logs. When configuring the transaction logs (discussed in the IDS – Transaction logs and the IBM DB2 – Transaction logs sections) ensure that you have enough disk space for the log files.
- The ITIM DB2 database uses DMS tablespaces which require pre-allocating disk space for the database to use. If the tablespace becomes full, DB2 will be unable to continue and will return an error in the `itim.log` file. If this occurs, add more tablespace containers. See the IBM DB2 – Tablespaces section below.

2 J2EE Application Servers

Regardless of the installation type (single, regular cluster, or functional cluster), the ITIM server can be thought of as two components: the J2EE application server running the application (WebSphere Application Server or BEA WebLogic Server) and the ITIM application itself. Both need to be tuned for your environment.

2.1 WebSphere Application Server

The WebSphere Application Server allows you to tune a variety of settings for your environment. This document discusses the Java Virtual Machine (JVM) size, the minimum and maximum JDBC pool sizes, and the various timeouts.

2.1.1 Java Virtual Machine (JVM) Size

By default, WebSphere sets the JVM size to 256MB. This value is too small for ITIM to run beyond a basic sniff test and should be increased to 512MB. If your machine has adequate available RAM (such as an unused 1GB of physical RAM), consider increasing it to 1GB.

Setting the maximum memory value too high can cause large pauses in the system during a full garbage collection. Do not set the maximum memory value higher than 1GB even if your system has the available memory.

Do not set the JVM heap size to be larger than the physical RAM. WebSphere suffers significant performance degradation if the operating system swaps out the JVM to swap space.

Determining the values

> max_jvm_heap_size = The maximum size of the JVM heap in MB. Recommended value: 512 or higher.

Setting the values

1) Open the WebSphere Administration Console.
2) Expand **Servers** list in the navigation pane.
3) Select **Application Servers** in the navigation pane.
4) Select the server to manage.
5) Select **Process Definition** from the **Additional Properties** pane at the bottom.
6) Select **Java Virtual Machine** from the **Additional Properties** pane at the bottom.
7) Set the **Maximum Heap Size** to *max_jvm_heap_size*.
8) Repeat for each ITIM server.

2.1.2 JDBC pool sizes

ITIM uses JDBC connections to connect to the database backend during the processing of workflow items such as changing a password, creating a new account, or loading a DSML feed. The minimum and maximum values for the JDBC connections might need to be increased depending on the function of the ITIM installation.

Determining the values

Find the value in the table that matches your ITIM installation configuration.

	Single Server or Regular Cluster	Functional Cluster (per each component)	
		UI Component	WF Component
min_pool_size	15	5	15
max_pool_size	60	10	50

The difference in the UI and WF component values is because most transactions in the UI (such as password changes) are delegated to a WF component, which makes the actual JDBC connection. If ITIM is unable to keep your database server busy under heavy load, consider increasing the maximum pool size.

Note: Your database needs to be able to accept the total number of connections plus 5 more for maintenance purposes. For example: You have a functional cluster with 2 workflows (WF) and 2 user interfaces (UI). Your WF component will need 50 * 2 = 100 connections and your UI component will need 10 * 2 = 20 possible connections. You also need to include the additional 5 connections for administrative purposes, which means you need to set your database connections to 100 + 20 + 5 = 125.

If you have a regular cluster with 3 ITIM nodes (each ITIM node includes one WF and one UI) you need (50 + 10) * 3 + 5 = 185 database connections. Refer to the section on database connections for more information on how to set the maximum number of connections.

Setting the values

1) Open the WebSphere Administration Console.
2) Expand **Resources** list in the navigation pane.
3) Select **JDBC Providers** in the navigation pane.
4) Single server: Select **Server** and click **Apply**.
   Cluster server: Select the **Node** and **Server** (using **Browse Nodes** and **Browse Servers**) and click **Apply**.
5) Select **ITIM JDBC Provider (XA)**.
6) Select **Data Sources (Version 4)** from the **Additional Properties** pane at the bottom.
7) Select **ITIM Data Source**.
8) Select **Connection Pool** from the **Additional Properties** pane at the bottom.
9) Set the **Minimum Pool Size** to *min_pool_size*.
10) Set the **Maximum Pool Size** to *max_pool_size*.
11) Repeat for each ITIM server.

2.1.3 Timeouts

The timeouts for the ITIM application need to be adjusted from their initial settings. The following values work well for most environments. If you begin receiving a large number of timeouts, consider increasing the appropriate timeout value.

Determining the values

jdbc_connection_timeout = The number of seconds before the JDBC connection is closed from inactivity. Recommended value: 60.

jdbc_idle_timeout = The number of seconds a JDBC connection can remain idle before being freed. Recommended value: 90.

jdbc_orphan_timeout = The number of seconds an application can hold an unused JDBC connection before being freed. Recommended value: 60.

queue_connection_timeout = The number of seconds before the queue connection is closed from inactivity. Recommended value: 300.

queue_reap_time = The number of seconds between runs of the queue pool maintenance thread. Recommended value: 300.

Setting the values

First, set the values for the JDBC connections:

1) Open the WebSphere Administration Console.
2) Expand **Resources** list in the navigation pane.
3) Select **JDBC Providers** in the navigation pane.
4) Single server: Select **Server** and click **Apply**.
   Cluster server: Select the **Node** and **Server** (using **Browse Nodes** and **Browse Servers**) and click **Apply**.
5) Select **ITIM JDBC Provider (XA)**.
6) Select **Data Sources (Version 4)** from the **Additional Properties** pane at the bottom.
7) Select **ITIM Data Source**.
8) Select **Connection Pool** from the **Additional Properties** pane at the bottom.
9) Set the **Connection Timeout** to *jdbc_connection_timeout*.
10) Set the **Idle Timeout** to *jdbc_idle_timeout*.
11) Set the **Orphan Timeout** to *jdbc_orphan_timeout*.
12) Repeat for each ITIM server.

Next, set the values for the ITIM Queues.

1) Open the WebSphere Administration Console.
2) Expand **Resources** list in the navigation pane.
3) Select **WebSphere JMS Providers** in the navigation pane.
4) Single server: Select **Server** and click **Apply**.
   Cluster server: Select the **Node** and **Server** (using **Browse Nodes** and **Browse Servers**) and click **Apply**.
5) Select **WebSphere Queue Connection Factories**.
6) Select **ITIM Queue Connection Factory**.
7) Select **Connection Pool** from the **Additional Properties** pane at the bottom.
8) Set the **Connection Timeout** to *queue_connection_timeout*.
9) Set the **Reap Time** to *queue_reap_time*.
10) Repeat for each ITIM server.

2.1.4 Statement Cache Size

ITIM uses prepared statements when communicating with the database server. Adjust this value to yield a better tradeoff between performance and resource utilization.

Determining the values

jdbc_statement_cache_size = The maximum number of prepared statements to cache. Recommended value: 60.

Setting the values

1) Open the WebSphere Administration Console.
2) Expand **Resources** list in the navigation pane.
3) Select **JDBC Providers** in the navigation pane..
4) Single server: Select **Server** and click **Apply**.
   Cluster server: Select the **Node** and **Server** (using **Browse Nodes** and **Browse Servers**) and click **Apply**.
5) Select **ITIM JDBC Provider (XA)**.
6) Select **Data Sources (Version 4)** from the **Additional Properties** pane at the bottom.
7) Select **ITIM Data Source**.
8) Select **Connection Pool** from the **Additional Properties** pane at the bottom.
9) Set the **Statement Cache Size** to *jdbc_statement_cache_size*.
10) Repeat for each ITIM server.

2.2 BEA WebLogic Server

Steps to tune BEA WebLogic Server will be included here when those settings and their values are determined.

3 IBM Tivoli Identity Manager (ITIM) Application

The IBM Tivoli Identity Manager (ITIM) application includes several configuration files that provide an area for tuning various parts of the application's performance. These are in the `data/` directory under the ITIM home directory.

3.1 Intel Multi-Processor Systems

For multi-processor Intel systems it is necessary to update the enrole.messaging parameters in the `enRole.properties` file to eliminate Java context switching.

Determining the values

itim_home = The ITIM home directory, such as `/itim45`.

Setting the values

Note: The lines below are split for readability purposes, they should be all in one line in the `enRole.properties` file.

Edit the *`itim_home`*`/data/enRole.properties` file and add PRIORITY=5 to the following lines as shown in bold below:

```
enrole.messaging.adhocSyncQueue.attributes=TRANSACTED=true RECEIVE_TIMEOUT=60
MAX_THREADS=5 MIN_THREADS=5 PRIORITY=5

enrole.messaging.workflowQueue.attributes=TRANSACTED=true RECEIVE_TIMEOUT=60
MAX_THREADS=2 MIN_THREADS=2 PRIORITY=5

enrole.messaging.workflowPendingQueue.attributes=TRANSACTED=true RECEIVE_TIMEOUT=60
WAIT_TIME=0 OVERCAPACITY_WAIT_TIME=10 MAX_THREADS=1 MIN_THREADS=1 PRIORITY=5

enrole.messaging.remoteServicesQueue.attributes=TRANSACTED=false RECEIVE_TIMEOUT=60
WAIT_TIME=0 MAX_THREADS=7 MIN_THREADS=7 PRIORITY=5

enrole.messaging.mailServicesQueue.attributes=TRANSACTED=false RECEIVE_TIMEOUT=60
WAIT_TIME=0 MAX_THREADS=3 MIN_THREADS=3 PRIORITY=5
```

Restart the ITIM server for the changes to take effect.

4 Database Servers

ITIM supports three different database systems: IBM DB2, Microsoft SQL Server, and Oracle Database. Each database requires slightly different tunings.

Each database machine should have at least one processor and 1GB of RAM. It can reside on a single processor machine by itself or share a multi-processor machine with other applications, but the 1 GB/1 CPU is the minimum for the database server. Tuning the database is one of the most important tunings for ITIM.

4.1 IBM DB2

Tuning IBM DB2 to run with ITIM involves adjusting the bufferpools, modifying the number of connections, modifying internal database values, adding tablespace, adjusting logs, indexing, and running runstats.

4.1.1 Quick guide for setting the IBM DB2 tuning parameters

This section includes the steps needed to tune the IBM DB2 parameters. This uses a generic set of values and may need to be further customized for your system.

Determining the values

First, determine the following values for your system:

itim_database = The name of the ITIM database, such as itim.

mem_for_itimdb = The amount of memory in bytes to allocate to the ITIM database. This value should be small enough that it will reside in physical memory and not be swapped out to disk. Recommended value: 500,000,000 (500MB) or greater.

logs_secondary = The number of secondary logs. Recommended value: 12.

logs_size = The size (in 4k pages) of the primary and secondary logs. Recommended value: 10000.

applheap_size = The value of applheapsz. Recommended value: 2048.

appctl_size = The value of app_ctl_heap_sz. Recommended value: 1024.

Next, calculate the following values:

ibmdefaultbp_npages = (mem_for_itimdb / 4096) * 0.25
enrolebp_npages = (mem_for_itimdb / 32768) * 0.75

Setting the values

Finally, set the values on your system. As the database administrator, connect to the database and run the following commands:

```
db2 alter bufferpool ibmdefaultbp size ibmdefaultbp_npages
db2 alter bufferpool enrolebp size enrolebp_npages
db2 update database configuration for itim_database using logsecond logs_secondary
db2 update database configuration for itim_database using logfilsiz logs_size
db2 update database configuration for itim_database using applheapsz applheap_size
db2 update database configuration for itim_database using app_ctl_heap_sz appctl_size
db2set DB2_RR_TO_RS=YES
```

4.1.2 Bufferpools

The ITIM database has two bufferpools: IBMDEFAULTBP and ENROLEBP. The IBMDEFAULTBP is used as a buffer for tablespaces with small extent sizes (4k) and ENROLEBP is used as a buffer for tablespaces with large extent sizes (32k). Most of the tables within the ITIM database use the tablespace with a large extent size and thus will use the ENROLEBP bufferpool. Use a 1:3 memory ratio between the IBMDEFAULTBP and the ENROLEBP.

Determining the values

First, determine the following values for your system:

total_mem_avail = The total amount of physical memory in bytes.

mem_for_itimdb = The amount of memory in bytes to allocate to the ITIM database. This value should be small enough that it will reside in physical memory and not be swapped out to disk. Recommended value: 500,000,000 (500MB) or greater.

Next, calculate the following values:

ibmdefaultbp_npages = (mem_for_itimdb / 4096) * 0.25
enrolebp_npages = (mem_for_itimdb / 32768) * 0.75

Setting the values

As the database administrator, connect to the database and run the following commands:

```
db2 alter bufferpool ibmdefaultbp size ibmdefaultbp_npages
db2 alter bufferpool enrolebp size enrolebp_npages
```

4.1.3 Connections

The ITIM server uses JDBC connections from the J2EE application server (WebSphere or WebLogic) to communicate with the database. The number of connections from the application server to the database depends on the need of the application. Both application servers allow you to set maximum connection values. The DB2 variable MAXAPPLS needs to be set to greater than the maximum possible connections. We recommend setting MAXAPPLS to five more than the total maximum possible connections.

Determining the values

itim_database_name = The name of your ITIM database, such as itim.

num_connections = Refer to the J2EE Application Server section to calculate the maximum number of connections and add five to that value.

Setting the values

As the database administrator, connect to the database and run the following command:

```
db2 update db cfg for itim_database_name using maxappls num_connections
```

4.1.4 Tablespaces

ITIM uses a Database Managed Space (DMS) tablespace to store its data. This type of tablespace gives better performance than System Managed Space (SMS) tablespaces but requires you to set aside disk space for the database to use. The default size of the ITIM tablespace is 512MB. This is frequently not large enough and should be increased by adding more containers to the tablespace.

Add more tablespaces by using the DB2 `alter tablespace` command. It is best to add files that are on another physical drive because DB2 performs better if one tablespace has multiple containers on multiple drives.

Determining the values

database_home = The home directory of your database administrator, such as `/home/db2inst1`.

database_name = The name of the database such as `db2inst1`. This is also a subdirectory under *database_home*.

container_name = The name of the file to create to hold the tablespace container, such as `enrole_data2`.

num_pages = The number of 32k pages to add to the tablespace. To calculate the number of pages from the number of MB, divide the size in MB by 0.032768. A 512MB tablespace would be 15625 pages.

Setting the values

As the database administrator, connect to the database and run the following command:

```
db2 "ALTER TABLESPACE ENROLE_DATA ADD ( \
     FILE '/database_home/database_name/NODE0000/SQL00001/container_name' num_pages) \
     PREFETCHSIZE 16 OVERHEAD 24.1 TRANSFERRATE 0.9 BUFFERPOOL ENROLEBP"
```

4.1.5 Transaction Logs

DB2 keeps logs during transaction processing. During large transactions the default log number and sizes might be too small and cause transaction rollbacks. This is solved by increasing the size and number of log files available to DB2.

DB2 has two types of log files: primary and secondary. Primary logs are allocated when the database is started and remain allocated until the database is stopped. Secondary logs are allocated as needed after the primary logs are full and are released after they are no longer needed.

Determining the values

Increase the number of secondary logs to be prepared when large transactions occur. Similarly, the default size of the log files is 1000 4k pages, or 4MB. Increase this to 10000 4k pages, or 40MB. Note that this will increase the size of both your primary and secondary log files.

itim_database = The name of the ITIM database, such as itim.

logs_secondary = The number of secondary logs. Recommended value: 12.

logs_size = The size (in 4k pages) of the primary and secondary logs. Recommended value: 10000.

Setting the values

As the database administrator, connect to the database and run the following commands:

```
db2 update database configuration for itim_database using logsecond logs_secondary
db2 update database configuration for itim_database using logfilsiz logs_size
```

4.1.6 Lock Tuning

ITIM requires that RR_TO_RS locking be enabled. This setting allows select statements that involve rows that are deleted but not yet committed to continue without waiting for the transaction to complete.

More information on this environment setting is available at:

http://www.ibm.com/software/data/db2/udb/ad/db2irfp7/db2ir169.htm

Determining the values

N/A

Setting the values

Run the following command as the ITIM database administrator:

```
db2set DB2_RR_TO_RS=YES
```

Stop and restart the database instance for this change to take effect.

4.1.7 Indexing

Adding an index to a heavily used table can greatly increase performance. Without indexes, DB2 must scan every row of the table until it finds the specified data. With an index, it uses a more efficient search method. The following indexes have been identified as increasing database performance with ITIM.

Note: It is very important to run `runstats` on the system after adding new indexes. See the DB2 – Runstats section for more information.

Determining the values

N/A

Setting the values - ITIM 4.4 & ITIM 4.5

As the database administrator, connect to the database and run the following commands:

Note: Some of these indexes have already been added to the ITIM installation. If there are "unable to create index" errors, they can be ignored as duplicates.

```
db2 'CREATE UNIQUE INDEX ACTIVITY_ID_X ON "ENROLE.ACTIVITY" \
     ("ID" ASC) INCLUDE ("SUBTYPE")'
db2 'CREATE INDEX ACTIVITY_PID_X ON "ENROLE.ACTIVITY" ("PROCESS_ID" DESC)'
db2 'CREATE INDEX ACTIVITY_STATE_X ON "ENROLE.ACTIVITY" ("STATE" DESC)'
db2 'CREATE UNIQUE INDEX PROCESS_ID_X ON "ENROLE.PROCESS" ("ID" ASC)'
db2 'CREATE INDEX PROCESSDATA_PID_X ON "ENROLE.PROCESSDATA" \
     ("PROCESS_ID" DESC, "DEF_ID" DESC)'
db2 'CREATE INDEX WORKITEM_PID_X ON "ENROLE.WORKITEM" ("PROCESS_ID" DESC)'
db2 'CREATE INDEX PROCESS_SUB_X ON "ENROLE.PROCESS" ("SUBMITTED" DESC, "PARENT_ID" ASC)'
```

Setting the values - ITIM 4.5.1

No additional indexes are required for tuning ITIM 4.5.1.

4.1.8 Runstats

The number of rows in the tables and what indexes are available are required for DB2 to efficiently fulfill queries. Run the `runstats` command on all tables in the database before and after large DSML loads and reconciliations. If you experience high CPU usage or poor DB2 performance, run the runstats command on all the tables in the database. DB2 `reorgchk` does not update index statistics and is not a replacement for `runstats`. To update index statistics run the runstats command on each table individually. A script (`perftune_runstats.sh`) is provided in Appendix A that detects the version of DB2 and runs the runstats command against all tables for a given schema in a database.

If the runstats command is run in a working environment, allow the connected applications to continue to write to the database. Use the "shrlevel change" on DB2 7 and "allow write access" on DB2 8 to allow users to write to a database while `runstats` is running.

Use the runstats command on an idle or lightly-used database because it requires update locking on system statistics table to update the database statistics. A database with a heavy load might experience transaction rollbacks due to the system acquiring locks on the tables that are used by the database optimizer to fulfill queries.

Determining the values

Use the runstats command on every table in the ENROLE schema. As the database administrator, connect to the database and use the following command to get a listing of all tables in the schema:

```
db2 list tables for all | grep ENROLE
```

Setting the values – DB2 7

For each table in the ENROLE schema, run the following DB2 runstats command:

```
db2 runstats on table ENROLE.table_name with distribution and \
    detailed indexes all shrlevel change
```

Setting the values – DB2 8

For each table in the ENROLE schema, run the following DB2 runstats command:

```
db2 runstats on table ENROLE.table_name with distribution and \
    detailed indexes all allow write access
```

4.1.9 Database Application Heaps

Some of the queries that ITIM submits to the DB2 server result in complex SQL statements. The default sizes of the application heaps are not large enough to allow DB2 to fulfill the queries. Failing to increase these will result in errors in the `itim.log` file and might cause transaction rollback error messages. If you continue to receive errors after increasing these values continue to increase the values in increments of 256.

Determining the values

itim_database = The name of the ITIM database, such as itim.

applheap_size = The value of applheapsz in 4k pages (default is 256). Recommended value: 2048.

appctl_size = The value of app_ctl_heap_sz in 4k pages (default is 128). Recommended value: 1024.

Setting the values

As the database administrator, connect to the database and run the following commands:

```
db2 update database configuration for itim_database using applheapsz applheap_size
db2 update database configuration for itim_database using app_ctl_heap_sz appctl_size
```

Stop and restart the database instance for these changes to take effect.

4.2 Microsoft SQL Server

Steps to tune Microsoft SQL Server will be included here when those settings and their values are determined.

4.3 Oracle Database

Steps to tune Oracle Database will be included here when those settings and their values are determined.

5 Directory Servers

ITIM supports two different directory servers: IBM Directory Server (IDS) and Sun Java system Directory Server (formerly known as Sun ONE Directory Server and Netscape iPlanet Directory Server).

5.1 IBM Directory Server (IDS)

With respect to tuning, IDS can be divided into two parts: the IDS process and IBM DB2.

In a well tuned system, the IDS process and the DB2 processes consume approximately the same amount of CPU. In a poorly tuned system, DB2 can max out the CPU usage trying to fulfill queries in a poor manner.

Both IDS and DB2 have caches that speed up data retrieval. Optimizing your available memory is the key to tuning IDS. When a read request comes in to IDS, IDS checks the filter cache to see if has seen that search filter before. If it has, the results are pulled from the cache, otherwise the query goes to DB2. After the search filter is evaluated, IDS pulls the requested attributes from the entry cache that match the search filter. If the values are not in the entry cache it queries DB2. For each request to DB2, DB2 checks to see if the data resides in a bufferpool. If not, it reads the value from the disk. Ideally, all requests to the directory server register an IDS cache hit or a DB2 bufferpool hit for the quickest response. Queries that require disk access can be very slow.

5.1.1 Quick guide for setting the IDS tuning parameters

This section includes the steps needed to tune the IDS parameters. This uses a generic set of values and may need to be further customized for your system.

Determining the values

First, determine the following values for your system:

ldap_database = The name of the IDS database, such as ldapdb2.

mem_for_ldapdb2_bps = The amount of memory in bytes to allocate to the ldapdb2 bufferpools. This value should be small enough that it will reside in physical memory and not be swapped out to disk. Recommended value: 500,000,000 (500MB) or greater.

mem_for_slapd = The amount of memory in bytes to allocate to the IDS process (including caches). This value should be small enough that it will reside in physical memory and not be swapped out to disk. Recommended value: 500,000,000 (500MB) or greater.

logs_secondary = The number of secondary logs. Recommended value: 12.

logs_size = The size (in 4k pages) of the primary and secondary logs. Recommended value: 10000.

ldap_db_connections = The number of database connections. Recommended value: 30.

stmtheap_size = The value of stmtheap. Recommended value: 4096.

applheap_size = The value of applheapsz. Recommended value: 2048.

appctl_size = The value of app_ctl_heap_sz. Recommended value: 1024.

sortheap_size = The value of sortheap. Recommended value: 1024.

sortheapthres_size = The value of sheapthres. If adjusting sortheap above 1024. Ideally this value would be sortheap * number_of_database_connections however this value may be too high for your system.

Next, calculate the following values:

ibmdefaultbp_npages = (mem_for_ldapdb2_bps / 4096) * 0.75
ldapbp_npages = (mem_for_ldapdb2_bps / 32768) * 0.25
entry_cache_size = (mem_for_slapd – 128,000,000) / 7,000
max_db_connections = ldap_db_connections + 10

Setting the values

Next, set the values on your system. As the database administrator, connect to the database and run the following commands:

```
db2 alter bufferpool ibmdefaultbp size ibmdefaultbp_npages
db2 alter bufferpool ldapbp size ldapbp_npages
db2 update database configuration for ldap_database using logsecond logs_secondary
db2 update database configuration for ldap_database using logfilsiz logs_size
db2 update database configuration for ldap_database using maxappls max_db_connections
db2 update database configuration for ldap_database using stmtheap stmtheap_size
db2 update database configuration for ldap_database using applheapsz applheap_size
db2 update database configuration for ldap_database using app_ctl_heap_sz appctl_size
db2 update database configuration for ldap_database using sortheap sortheap_size
db2 update database manager configuration using sheapthres sortheapthres_size
```

Finally, stop IDS, edit `/etc/slapd32.conf` (IDS 4.1) or `/etc/ldapschema/ibmslapd.conf` (IDS 5.x) and update the following configuration options:

```
ibm-slapdDbConnections: ldap_db_connections
ibm-slapdACLCache: DISABLE
ibm-slapdFilterCacheSize: 0
ibm-slapdEntryCacheSize: entry_cache_size
```

Stop and restart DB2 and the IDS server for these changes to take effect.

5.1.2 LDAP Cache Sizes

IDS has three caches: the ACL cache, the filter cache, and the entry cache. Because the ITIM server binds as an authoritative user the ACL cache is never accessed. The filter cache is helpful for programs that issue more read requests than writes or updates because the entire filter cache is invalided at every write. Because ITIM frequently updates the directory server, the filter cache is not helpful either. Because of these reasons both the ACL cache and the filter cache should be disabled and the memory used elsewhere.

IDS allows you to control how many entries the entry cache can store but does not restrict the size of the cache. The size of each entry in the cache is based on the number and the size of attributes that a given LDAP entry has. Typically, many entries are users and their accounts, which have a fairly constant size. When setting the value for the entry cache, calculate the size of the average entry and divide that into the amount of memory used by the IDS process. Users with few attributes can generate entry sizes that are approximately 4k where users with more attributes can generate entry sizes around 9k. The IDS Tuning Document discusses the procedure for determining the size of the average entry.

Determining the values

acl_cache = Specifies if the ACL cache is used. Recommended value: FALSE (disabled).

filter_cache_size = Specifies the size of the filter cache. Recommended value: 0 (disabled).

entry_cache_size = The size of the entry cache. Recommended value: as large as the number of ITIM users times the average number of accounts plus one per user. So if you have 300 users with two accounts you would want 300 * (2+1) = 900. This value is bounded by the amount of memory allocated to the IDS process minus the size of the process itself (about 128MB).

Note: Do not set the entry cache size to be too large. If the IDS process exceeds the amount of available memory, significant performance degradation will occur because of page swapping.

Setting the values

Stop IDS, edit `/etc/slapd32.conf` (IDS 4.1) or `/etc/ldapschema/ibmslapd.conf` (IDS 5.x) and update the following configuration options:

```
        ibm-slapdACLCache: acl_cache
        ibm-slapdFilterCacheSize: filter_cache_size
        ibm-slapdEntryCacheSize: entry_cache_size
```

Restart IDS for these changes to take effect.

5.1.3 DB2 Bufferpools

DB2 bufferpools are the secondary buffer for IDS. These bufferpools should be large enough that most table searches can be read directly from memory instead of using the disk. This value can be measured by looking at the hit ratio for the bufferpools. See the DB2 Performance Monitoring section for information on calculating the bufferpool hit ratio.

The IDS database (ldapdb2 by default) has two bufferpools: IBMDEFAULTBP and LDAPBP. The IBMDEFAULTBP is used as a buffer for tablespaces with small extent sizes (4k) and LDAPBP is used as a buffer for tablespaces with large extent sizes (32k). Most of the tables in the ldapdb2 database use the tablespace with a small extent size and use IBMDEFAULTBP. Use a 3:1 memory ratio between the IBMDEFAULTBP and the LDAPBP.

Determining the values

Allocate enough memory to the DB2 bufferpools so the bufferpool hit ratio > 95% and allocate the rest of the memory to the IDS process and the caches. The following formula may help in determining the bufferpool values for your machine.

First, determine the values for your system:

ldap_database = The name of the IDS database, such as ldapdb2.

mem_for_ldapdb2_bps = The amount of memory in bytes to allocate to the ldapdb2 bufferpools This value should be small enough that it will reside in physical memory and not be swapped out to disk. Recommended value: 500,000,000 (500MB) or greater.

Next, calculate the following values:

ibmdefaultbp_npages = (mem_for_ldapdb2_bps / 4096) * 0.75
ldapbp_npages = (mem_for_ldapdb2_bps / 32768) * 0.25

Setting the values

As the database administrator, connect to the database and run the following commands:

```
        db2 alter bufferpool ibmdefaultbp size ibmdefaultbp_npages
        db2 alter bufferpool ldapbp size ldapbp_npages
```

5.1.4 Connections

There are three connection values to modify for IDS. The first value is the number of connections that IDS should make to the DB2 backend. Set this to 30.

The second value is the number of connections that the DB2 server should allow to connect to it. This value should be greater than the number of connections that IDS will make to it. Set this value to 10 more than the number specified for the IDS server. These extra connections allow for administrative functions.

The third value only applies to Windows computers. Windows allows up to 64 sockets in a select call for each thread. This limits the number of inbound connections to 64. To overcome this problem set the SLAPD_OCHANDLERS environment variable to specify the number of connection threads, each one supporting 64 connections.

Determining the values

ldap_database_name = The name of your IDS database, such as ldapdb2.

num_ldap_connections = The number of connections from IDS to the DB2 backend. Recommended value: 30 (from the IDS Tuning Guide).

num_db_connections = The number of connections the DB2 backend will accept. Recommended value: 40.

For Windows machines, also determine the following values:

max_num_LDAP_connections = The maximum number of IDS connections (enrole.connectionpool.maxpoolsize) from the `enRole.properties` file. The ITIM default is: 100.

itim_nodes = The number of ITIM nodes in your cluster. This is the sum of the WF and the UI components.

oc_handlers = (max_num_LDAP_connections * itim_nodes) / 64

Setting the values

As the database administrator, connect to the database and run the following commands:

```
db2 update db cfg for ldap_database_name using maxappls num_db_connections
```

To set the number of connections the IDS server will make to the DB2 server, stop IDS, edit `/etc/slapd32.conf` (IDS 4.1) or `/etc/ldapschema/ibmslapd.conf` (IDS 5.x) and find the stanza:

```
dn: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
```

Change the following attribute:

```
ibm-slapdDbConnections: num_ldap_connections
```

For Windows machines, also set the SLAPD_OCHANDLERS values. Stop IDS, edit `slapd32.conf` (IDS 4.1) or `ibmslapd.conf` (IDS 5.x) and find the stanza:

```
dn: cn=Front End, cn=Configuration
```

Add the following configuration option at the end of the stanza before the blank line:

```
ibm-slapdsetenv: SLAPD_OCHANDLERS=oc_handlers
```

Stop and start the IDS database and restart IDS for these changes to take effect.

5.1.5 Transaction Logs

DB2 keeps logs during transaction processing. During large transactions the default log number and sizes might be too small and cause transaction rollbacks. This is solved by increasing the size and number of log files available to DB2.

DB2 has two types of log files: primary and secondary. Primary logs are allocated when the database is started and remain allocated until the database is stopped. Secondary logs are allocated as needed after the primary logs are full and are released after they are no longer needed.

Determining the values

Increase the number of secondary logs to be prepared when large transactions occur. Similarly, the default size of the log files is 1000 4k pages, or 4MB. Increase this to 10000 4k pages, or 40MB. Note that this will increase the size of both your primary and secondary log files.

ldap_database = The name of the IDS database, such as ldapdb2.

logs_secondary = The number of secondary logs. Recommended value: 12.

logs_size = The size (in 4k pages) of the primary and secondary logs. Recommended value: 10000.

Setting the values

As the database administrator, connect to the database and run the following commands:

```
db2 update database configuration for ldap_database using logsecond logs_secondary
db2 update database configuration for ldap_database using logfilsiz logs_size
```

5.1.6 Indexing

Indexing attributes that applications search on increases IDS performance. IDS indexes are automatically translated into DB2 indexes by updating the IDS schema for those attributes.

If you extend the LDAP schema in IDS to include additional attributes, index those attributes that you will search for. Any filter in ITIM (such as with dynamic roles) is translated into a search string for the IDS server.

ITIM frequently searches against the Owner and OU. Index these attributes either through the IDS Web Administration tool or by importing the `ldap_indexes.ldif` file using the `ldapmodify` command.

After updating the LDAP schema, run DB2 `runstats` on the database to update the statistics for the newly created indexes.

Determining the values

root_dn = The root DN of the IDS server.

root_password = The password for the root DN.

Setting the values

1) To use the LDIF, create the `ldap_indexes.ldif` file from the contents in Appendix A and place it on the computer.

2) Run the following command to import the LDIF into IDS:

```
ldapmodify -D root_dn -w root_password -f ldap_indexes.ldif
```

5.1.7 Database Statement Heap

ITIM can submit very long LDAP queries to the IDS server. If the query is suitably long, the query will not fit in the DB2 statement heap (stmtheap) and IDS will return an error to ITIM. To avoid this problem, increase the size of the statement heap.

Determining the values

ldap_database = The name of the IDS database, such as ldapdb2.

stmtheap_size = The value of stmtheap in 4k pages (default is 2048). Recommended value: 4096.

Setting the values

As the database administrator, connect to the database and run the following commands:

```
db2 update database configuration for ldap_database using stmtheap stmtheap_size
```

Stop and restart the IDS database and IDS for these changes to take effect.

5.1.8 Database Application Heaps

Complex queries that ITIM submits to IDS result in complex SQL statements to the DB2 database. The default sizes of the application heaps are not large enough to allow DB2 to fulfill these queries from IDS. Failing to increase these might cause JNDI errors in the `itim.log` file. If you continue to receive errors after increasing these values continue to increase these values in increments of 256.

Determining the values

ldap_database = The name of the IDS database, such as ldapdb2.

applheap_size = The value of applheapsz in 4k pages (default is 256). Recommended value: 2048.

appctl_size = The value of app_ctl_heap_sz in 4k pages (default is 128). Recommended value: 1024.

Setting the values

As the database administrator, connect to the database and run the following commands:

```
db2 update database configuration for ldap_database using applheapsz applheap_size
db2 update database configuration for ldap_database using app_ctl_heap_sz appctl_size
```

Stop and restart the IDS database and IDS for these changes to take effect.

5.1.9 Database Sort Heaps

Several of the SQL queries from IDS require sorts in the database. To increase performance for these queries, increase the size of the sort heaps in DB2.

There are two sort heap parameters inside DB2: sortheap and sheapthres. The sortheap parameter is a database-level parameter and is allocated as needed to perform sorts. Each sort is allocated its own sortheap. The sheapthres parameter is a database-manager level parameter and provides an upper bound on the total amount of sort heap available to sorts. When increasing the size of the sortheap, ensure that sheapthres is large enough for multiple sorts to allocate sort heap.

If you experience excessive sort overflows, as identified from a database snapshot, increase the sortheap and possibly sheapthres parameters.

Determining the values

ldap_database = The name of the IDS database, such as ldapdb2.

sortheap_size = The value of sortheap in 4k pages (default is 256). Recommended value: 1024.

sortheapthres_size = The value of sheapthres in 4k pages (default is 20,000 pages – Unix; 10,000 pages – Windows). If adjusting sortheap above 1024. Ideally this value would be sortheap * number_of_database_connections however this value may be too high for your system. The sortheapthres_size should be at least twice as large as sortheap_size.

Setting the values

As the database administrator, connect to the database and run the following commands:

```
db2 update database configuration for ldap_database using sortheap sortheap_size
db2 update database manager configuration using sheapthres sortheapthres_size
```

Stop and restart the IDS database and IDS for these changes to take effect.

5.2 Sun Java System Directory Server

Steps to tune Sun Java System Directory Server (formerly known as Sun ONE Directory Server and Netscape iPlanet Directory Server) will be included here when those settings and their values are determined.

6 IBM DB2 Performance Monitoring

IBM DB2 provides several tools for troubleshooting and analyzing performance problems. This section contains a primer for how to get started monitoring with DB2. For more extensive information, refer to the *DB2 Administration Guide: Performance* book.

6.1 Enable Monitoring

To gather performance information, turn on the DB2 monitoring flags. To turn on monitoring, run the following commands as the database administrator. Note that this should be done for each database.

```
db2 update database manager configuration using DFT_MON_STMT ON
db2 update database manager configuration using DFT_MON_BUFPOOL ON
db2 update database manager configuration using DFT_MON_LOCK ON
db2 update database manager configuration using DFT_MON_SORT ON
db2 update database manager configuration using DFT_MON_TABLE ON
db2 update database manager configuration using DFT_MON_TIMESTAMP ON
db2 update database manager configuration using DFT_MON_UOW ON
```

Stop and restart the database instance for the monitoring to take effect.

6.2 Snapshots

Snapshots allow you to view the internal state of various DB2 components. The following commands access important DB2 snapshots:

```
db2 get snapshot for database on database_name
db2 get snapshot for dynamic sql on database_name
db2 get snapshot for tables on database_name
db2 get snapshot for locks on database_name
```

6.3 Bufferpool Hit Ratio

The bufferpool hit ratio gives a good indication of how many data reads are coming from the bufferpool and how many are coming from the disk. The larger the hit ratio, the less disk I/O used. Calculate the bufferpool hit ratio by enabling monitoring and taking a database snapshot. Calculate the bufferpool hit ratio with the following formula:

P = buffer pool data physical reads + buffer pool index physical reads
L = buffer pool data logical reads + buffer pool index logical reads
Hit ratio = (1-(P/L)) * 100%

7 Other Resources

You may find the following resources useful for further tuning of the ITIM middleware:

DB2 Administration Guide: Performance

http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v8pubs.d2w/en_main

IBM Directory Server Tuning Guide

http://publib.boulder.ibm.com/tividd/td/IBMDirectoryServer5.2.html

WebSphere Application Server Monitoring and Tuning Performance

http://www-306.ibm.com/software/webservers/appserv/infocenter.html

8 Appendix A – Scripts and files

The most recent version of the scripts can be obtained by going to http://www.ibm.com/support/us, select “Search technical support”, type “ITIM Tuning Guide” in the box and click submit. Most of the scripts will need to be customized for your system.

8.1 perftune_runstats.sh

```
#!/bin/sh

# perftune_runstats.sh
# Last Updated: 2003/09/11 14:11 CDT
# Desription:
#   Shell script to run the db2 runstats command on all tables for a given
#   database and schema. It autodetects the db2 version using db2level
#   and passes the appropriate string to runstats to allow writes to occur
#   during the runstats.
# Usage:
#   This script should be run as the user for the database (ie: one that
#   has connect and runstats abilities and permissions).

DATABASE=itim
SCHEMA=ENROLE
OPTIONS="with distribution and detailed indexes all"

# Find out DB2 version for runstats syntax
if db2level | grep "DB2 v7" > /dev/null; then
  ACCESS="shrlevel change"
  echo Detected DB2 major version 7
elif db2level | grep "DB2 v8" > /dev/null; then
  ACCESS="allow write access"
  echo Detected DB2 major version 8
fi

# Connect to the database
echo Connecting to $DATABASE
db2 connect to $DATABASE

# Execute runstats on all tables
echo Performing runstats on all tables for schema $SCHEMA
echo "  with options: $OPTIONS $ACCESS"
for i in `db2 connect to $DATABASE > /dev/null; db2 list tables for all | \
    grep $SCHEMA | cut -d'  -f1`
do
echo Table: $SCHEMA.$I
db2 runstats on table $SCHEMA.$i $OPTIONS $ACCESS
done
```

8.2 ldap_indexes.ldif

```
#---------- beginning of LDIF ---------

dn: cn=schema
changetype: modify
replace: attributetypes
attributetypes: ( 2.5.4.32 NAME 'owner' DESC 'Identifies the distinguished name (DN) of the
person responsible for the entry.' SUP 2.5.4.49 EQUALITY 2.5.13.1 SYNTAX
1.3.6.1.4.1.1466.115.121.1.12{1000} )
-

replace: ibmattributetypes

ibmattributetypes: ( 2.5.4.32 DBNAME ( 'OWNER'  'OWNER' ) ACCESS-CLASS NORMAL LENGTH 1000
EQUALITY ORDERING )


dn: cn=schema
changetype: modify
replace: attributetypes
attributetypes: ( 2.5.4.11 NAME ( 'ou' 'organizationalUnit' 'organizationalUnitName' ) DESC 'This
attribute contains the name of an organization (organizationName).' SUP 2.5.4.41 EQUALITY
1.3.6.1.4.1.1466.109.114.2 SUBSTR 2.5.13.4)

-
```

```
replace: ibmattributetypes
ibmattributetypes: ( 2.5.4.11 DBNAME ( 'ou' 'ou' ) ACCESS-CLASS NORMAL LENGTH 128 EQUALITY )
#---------- end of LDIF ---------
```