

InfiniDB[®]

Administrator's Guide

Release: 4.6
Document Version: 4.6-1

InfiniDB Administrator's Guide

July 2014

Copyright © 2014 InfiniDB Corporation. All Rights Reserved.

InfiniDB, the InfiniDB logo and any other product or service names or slogans contained in this document are trademarks of InfiniDB and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of InfiniDB or the applicable trademark holder.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of InfiniDB.

InfiniDB may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from InfiniDB, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property. The information in this document is subject to change without notice. InfiniDB shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Contents

1	Introduction	7
1.1	Audience	7
1.2	List of Documentation	7
1.3	Obtaining documentation	7
1.4	Documentation feedback	7
1.5	Additional resources	7
2	InfiniDB Console Overview	8
2.1	Logging on and off the InfiniDB Console	8
2.2	InfiniDB Console tips	9
2.2.1	Help command	9
2.2.2	Case sensitivity	9
2.2.3	Recall command history	10
2.2.4	Command repeat option	10
2.2.5	Linux and InfiniDB Console command prompts	10
2.3	Complete list of InfiniDB commands	11
3	System Maintenance	13
3.1	Upgrading InfiniDB	13
3.1.1	InfiniDB software	13
3.1.2	Upgrading InfiniDB	13
3.2	Storage Information	13
3.3	System Operations	13
3.3.1	Stopping the system	14
3.3.2	Starting the system or modules	14
3.3.3	Restarting the system	14
3.3.4	Shutting down the system	14
3.3.5	Disabling and Enabling System Modules	15
4	System Status	16
4.1	Viewing system status	16
4.2	Viewing process status	17
4.3	Viewing network configuration	18
4.4	Viewing module information	18
4.5	Monitoring module usage and thresholds	19
4.5.1	Module disk usage	20
4.5.2	Memory usage	21
4.5.3	Swap space usage	21
4.5.4	Disk thresholds	22
4.5.5	CPU thresholds	22
4.6	Monitoring alarms	23
4.6.1	Viewing active alarms	23
4.6.2	Viewing archived alarms	24
4.6.3	Viewing top CPU users	26
4.6.4	Monitoring resource usage	27
5	Configuring System Status Monitoring	28
5.1	Management console overview	28
5.1.1	InfiniDB configuration files	28
5.1.2	Linux shell	28
5.2	Configuring system	28

5.2.1	Setting module heartbeats	30
5.2.2	Configuring Disk thresholds	30
5.2.3	Configuring archived transaction log periods	31
5.3	Configuring modules	32
5.3.1	Configuring module CPU thresholds	32
5.3.2	Configuring module disk usage	32
5.3.3	Configuring swap space usage	32
5.4	Configuring alarms	33
5.4.1	Steps to configure alarms	33
5.5	Configuring logging	33
5.5.1	Enabling and disabling logging	34
6	Loading Data	35
6.1	Import overview	35
6.1.1	Estimating load sizes	35
6.1.2	Import Tasks	35
6.1.3	Import process	35
6.1.4	Directory structure	35
6.1.5	Importing as non-root user	36
6.2	Copy source files	36
6.2.1	Source file formats	36
6.2.2	Source file names	36
6.2.3	Import directory	37
6.3	Import Usage	37
6.3.1	Importing multiple tables	37
6.4	Distributed Imports	37
6.4.1	Bulk Load from a central location (Mode 1)	37
6.4.2	Single Bulk Load Command, Partitioned Source (Mode 2)	38
6.4.3	Independent Bulk Load Command, Partitioned Source (Mode 3)	39
6.4.4	Mode Availability	40
6.5	Generate job file	41
6.6	Import data	43
6.6.1	Using cpimport	43
6.6.2	STDIN Processing	47
6.6.3	Bulk Insert with Select From Table(s)	48
6.6.4	Bulk Insert with Binary Source	48
6.6.5	Enclosed By (-E) and Escape Character (-C) Usage	51
6.7	Advanced Data Mapping to Table	52
6.7.1	Data Mapping Example	52
6.8	Troubleshooting import	53
6.8.1	Location of error log and rejected data	54
6.8.2	Data mismatch	54
6.8.3	Number of error rows exceeds threshold	54
7	Transactions	55
7.1	Transaction overview	55
7.2	ACID Compliance	55
8	Performing Backup and Recovery	57
8.1	Backing up data	57
8.1.1	Suspending database writes	57
8.1.2	Backing up front-end database	58

8.1.3	Backing up InfiniDB database.....	58
8.1.4	Resuming database writes	58
8.2	Recovering from a backup.....	59
8.2.1	Restoring front-end database	59
8.2.2	Restoring InfiniDB database	59
8.3	Dropping the InfiniDB database.....	60
9	Performance Tuning.....	61
9.1	Query Tuning Guidelines - Data Access Operations.....	61
9.1.1	Operations used by InfiniDB include:	61
9.2	Gathering Performance Metrics - calgetstats().....	61
9.3	Gathering Performance Metrics.....	62
9.3.1	Using /var/log/Calpont/debug.log	62
9.3.2	Using calsettrace(1);	62
9.3.3	Using /var/log/mysqld.log.....	63
9.4	Understanding Extent Map Range Partitioning	63
9.5	Scaling	63
9.5.1	For performance	63
9.5.2	For concurrency	63
9.5.3	For data capacity	63
10	Configuring Modules.....	64
10.1	Before configuring modules	64
10.1.1	Module IDs.....	64
10.1.2	Automatic configuration	64
10.1.3	Manual configuration	65
10.2	Adding DBRoots	65
10.2.1	Creating the physical dbroot.....	65
10.2.2	Assign DBRoots to Performance Module	65
10.3	Moving DBRoots.....	67
10.4	Adding modules	67
10.5	Removing modules	69
11	Granting Access for Users.....	70
12	Cross-Engine Table Access	71
13	User Resource Allocation	72
13.1	User Priority Management.....	72
13.2	Enabling User Priorities	72
13.3	User Priority Processing	73
14	Query Performance Monitoring	74
14.1	Enablement of Query Statistics Collection	74
14.2	Query Statistics table.....	74
14.3	Column Definitions.....	75
14.4	Query Statistics Viewing	76
15	Operating Mode	78
16	Decimal to Double Math	79
16.1	Enable/Disable Decimal to Double Math	79
17	Decimal Scale	80
17.1	Enable/Disable Decimal Scale	80
17.2	Set Decimal Scale Level.....	80
18	Using cpimport for Batch Insert	81
18.1	Enable/Disable Using cpimport for Batch Insert.....	81

18.2	Changing Default Delimiter for INSERT SELECT	81
19	Using Disk Based Joins	82
20	Compression Mode.....	84
21	Local PM Query	86
21.1	Enable/Disable Local PM Query.....	86
22	Partition Management.....	87
23	System Utilities	88
23.1	configxml.sh.....	88
23.2	colxml.....	88
23.3	cpimport	88
23.4	viewtablelock	88
23.5	cleartablelock.....	89
23.6	healthCheck.....	89
23.7	redistributeDB	89
23.8	calpontSupport.....	89
23.9	databaseSizeReport	92
24	Using UTF-8 Character Sets	95
24.1	UTF-8 Character Set	95
24.1.1	my.cnf	95
24.1.2	Calpont.xml	95
24.1.3	Importing Data	95
24.2	Known Issues and Limitations	96
25	Troubleshooting.....	97
25.1	ERROR 1070 (42000): Too many key parts specified; max 0 parts allowed	97
25.2	UM memory exceeded error / distributed join performance	97
25.3	Error in Aggregation Data Overflow.....	97
25.4	Enable/Disable Decimal to Double Math	97
25.5	Circular join detected error	99
25.6	Imports executing slower with concurrent queries	100
25.7	Delay in data retrieval immediately after a CREATE TABLE	100
25.8	Lost connection when using stdin with cpimport	100
26	Appendix A - 3rd Party Licenses	101
26.1	GNU GENERAL PUBLIC LICENSE	101
26.1.1	Preamble.....	101
26.1.2	TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION	102
26.1.3	NO WARRANTY	105
26.1.4	END OF TERMS AND CONDITIONS	105
26.2	RSA Data Security.....	106
26.3	Paul E. Jones.....	107
26.4	Apache.....	107
26.5	Gary S. Brown	110
26.6	Brian M. Clapper.....	110
26.7	Net-SNMP.....	111
26.8	MySQL AB / Sun Microsystems	118
26.9	Lasse Mikkil Reinhold.....	118
27	Glossary.....	119

1 Introduction

This guide includes information for installing, configuring, maintaining, performance tuning, and troubleshooting InfiniDB.

Some of the content in this guide is only available with InfiniDB Enterprise.

1.1 Audience

This guide is intended for database administrators and IT administrators who are responsible for setting up and maintaining InfiniDB.

1.2 List of Documentation

The InfiniDB Database Platform documentation consists of several guides intended for different audiences. The documentation is described in the following table:

Document	Description
InfiniDB Apache Hadoop™ Configuration Guide	Installation and Administration of InfiniDB for Apache Hadoop system.
InfiniDB Concepts Guide	Introduction to the InfiniDB analytic database.
InfiniDB Minimum Recommended Technical Specifications	Lists the minimum recommended hardware and software specifications for implementing InfiniDB.
InfiniDB Installation Guide	Contains a summary of steps needed to perform an install of InfiniDB.
InfiniDB Multiple UM Configuration Guide	Provides information for configuring multiple User Modules.
InfiniDB SQL Syntax Guide	Provides syntax native to InfiniDB.
Performance Tuning for the InfiniDB Analytics Database	Provides help for tuning the InfiniDB analytic database for parallelization and scalability.
InfiniDB Windows Installation and Administrator's Guide	Provides information for installing and maintaining InfiniDB for Windows.

1.3 Obtaining documentation

These guides reside on our <http://www.infinidb.co> website. Contact support@infinidb.co for any additional assistance.

1.4 Documentation feedback

We encourage feedback, comments, and suggestions so that we can improve our documentation. Send comments to support@infinidb.co along with the document name, version, comments, and page numbers.

1.5 Additional resources

If you need help installing, tuning, or querying your data with InfiniDB, you can contact support@infinidb.co.

2 InfiniDB Console Overview

The InfiniDB Console allows you to configure, monitor, and manage the InfiniDB system and servers. This chapter explains how to use the InfiniDB Console and gives a complete listing of InfiniDB commands. For more detailed information, see the following chapters: Configuring InfiniDB, Monitoring InfiniDB, and Managing InfiniDB System and Servers.

2.1 Logging on and off the InfiniDB Console

You can log on to the InfiniDB Console using a SSH client. This chapter describes both processes.

If you use a SSH client to connection to a Linux shell, you run the InfiniDB Console in the Linux shell. Many of the Linux-shell features are available to the InfiniDB Console such as:

- Entering partial command names to execute commands
- Recalling previous commands using the **Up** and **Down** arrows
- Using hot keys to edit and navigate commands

The following procedures explain how to log on and off of the InfiniDB Console. You can choose to execute InfiniDB commands from a Linux prompt. See “Linux and InfiniDB Console command prompts” :

To log on to the Management Console with SSH client:

1. Connect to Linux using your SSH client.
You will be prompted to enter a user name and password.
2. Type your Linux user name and password and press **Enter**.
The server factory defaults are:
User Name: **root**
Password: Calpont1
3. **At the Linux prompt, type cmconsole**
The cmconsole prompt appears.

To log off of the InfiniDB Console:

1. At the cmconsole prompt, type **exit** or **quit**
The Linux prompt appears.

2.2 InfiniDB Console tips

The following tips and tricks help you use the InfiniDB Console.

2.2.1 Help command

The help command displays supported commands. You can view brief help definitions or verbose definitions. You can also enter partial command names with the help command to view verbose definitions.

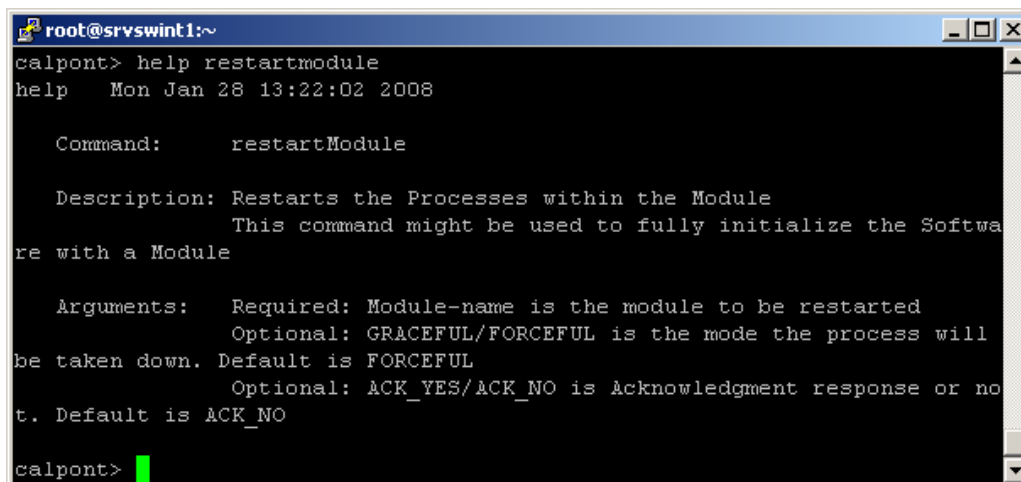
To view HELP commands:

1. At the Linux prompt, type **cmconsole**
2. Type **help**

An alphabetical list of support commands appears with brief descriptions.

You can enter the **help** command follow by a command name or partial command name to display the verbose description of the command.

For example, type **help restartModule** to get the verbose definition of the restart server command as shown in the figure below.



```
root@srvswint1:~
calpont> help restartmodule
help  Mon Jan 28 13:22:02 2008

Command:      restartModule

Description:  Restarts the Processes within the Module
              This command might be used to fully initialize the Softwa
re with a Module

Arguments:    Required: Module-name is the module to be restarted
              Optional: GRACEFUL/FORCEFUL is the mode the process will
be taken down. Default is FORCEFUL
              Optional: ACK_YES/ACK_NO is Acknowledgment response or no
t. Default is ACK_NO

calpont>
```

2.2.2 Case sensitivity

Commands are not case sensitive; however parameters and device names, like server and processes, are case sensitive. For example, the **GetAlarmLog** command can be entered as **getalarmlog**.

Below are examples of the **setsystemconfig** command followed by the **ProcessHeartbeatPeriod** parameter.

setsystemconfig	ProcessHeartbeatPeriod 5
SetSystemConfig	ProcessHeartbeatPeriod 5

2.2.3 Recall command history

To browse the history of commands entered, use the Up and Down arrow keys on your keyboard to scroll through commands.

2.2.4 Command repeat option

You can run a command continuously using the “-r” option. This is useful to check status in real-time mode.

The repeat option repeats a command every 5 seconds. You can change the repeat interval to be between 1 and 60 seconds by adding the number of seconds after the command.

For example to repeat the GetProcess Status command every 2 seconds, type:

```
GetProcessStatus -r2
```

NOTE: To exit out of a repeating command, press **Ctrl-C** to exit back to a Linux prompt, or **Ctrl-D** to return to the InfiniDB Console.

2.2.5 Linux and InfiniDB Console command prompts

For easier command execution without going back and forth between Linux and the Management Console, you can choose to execute Linux commands from the Management Console or execute InfiniDB commands from a Linux prompt.

To execute a Linux system command from InfiniDB Console:

Type **system** (*and the Linux command name*) and press **Enter**. This processes the Linux command and displays output to the console.

For example, to view the Linux manual for the copy command from the InfiniDB prompt, type:

```
system man cp
```

To execute an InfiniDB Console from the Linux prompt:

Type **cmconsole** *InfiniDB_command*

For example, to view the system status from the Linux command prompt :

```
cmconsole getSystemStatus
```

2.3 Complete list of InfiniDB commands

Below are the commands you can run from the Management Console to configure, manage, and operate the InfiniDB System.

Command	Description
?	Help
addDbroot	Add DBRoot Disk Storage to the InfiniDB System
addExternalDevice	Add External Device to Configuration file
addModule	Add a Module within the InfiniDB System
alterSystem-disableModule	Disable a Module and Alter the InfiniDB System
alterSystem-enableModule	Enable a Module and Alter the InfiniDB System
assignDbrootPmConfig	Assign currently unassigned DBroots to Performance Module.
assignElasticIPAddress	Assign Amazon Elastic IP Address to a module
disableLog	Disable the levels of process and debug logging
enableLog	Enable the levels of process and debug logging
exit	Exit from the Console tool
getActiveAlarms	Get Active Alarm list
getActiveSQLStatements	Get List Active SQL Statements within the System
getAlarmConfig	Get Alarm Configuration Information
getAlarmHistory	Get system alarms
getAlarmSummary	Get Summary counts of Active Alarm
getCalpontSoftwareInfo	Get the InfiniDB RPM detailed information
getExternalDeviceConfig	Get External Device Configuration Information
getLogConfig	Get the System log file configuration
getModuleConfig	Get Module Name Configuration Information
getModuleCpu	Get a Module CPU usage
getModuleCpuUsers	Get a Module Top Processes utilizing CPU
getModuleDisk	Get a Module Disk usage
getModuleMemory	Get a Module Memory usage
getModuleMemoryUsers	Get a Module Top Processes utilizing Memory
getModuleResourceUsage	Get a Module Resource usage
getModuleTypeConfig	Get Module Type Configuration Information
getProcessConfig	Get Process Configuration Information
getProcessStatus	Get InfiniDB Process Statuses
getStorageConfig	Get System Storage Configuration Information
getStorageStatus	Get System Storage Status.
getSystemConfig	Get System Configuration Information
getSystemCpu	Get System CPU usage on all modules
getSystemCpuUsers	Get System Top Processes utilizing CPU

getSystemDisk	Get System Disk usage on all modules
getSystemInfo	Get the Over-all System Statuses
getSystemMemory	Get System Memory usage on all modules
getSystemMemoryUsers	Get System Top Processes utilizing Memory
getSystemNetworkConfig	Get System Network Configuration Information
getSystemResourceUsage	Get System Resource usage on all modules
getSystemStatus	Get System and Modules Status
help	Get help on the Console Commands
monitorAlarms	Monitor alarms in realtime mode
movePmDbrootConfig	Move DBroots from one Performance Module to another.
quit	Exit from the Console tool
removeDbroot	Remove DBRoot Disk storage from the InfiniDB System
removeExternalDevice	Remove External Device to Configuration file
removeModule	Remove a Module within the InfiniDB System
resetAlarm	Resets an Active Alarm
restartSystem	Restarts the Processes within the InfiniDB System
resumeDatabaseWrites	Resume performing writes to the InfiniDB Database
setAlarmConfig	Set a Alarm Configuration parameter
setExternalDeviceConfig	Set a External Device Configuration parameter
setModuleTypeConfig	Set a Module Type Configuration parameter
setProcessConfig	Set a Process Configuration parameter
setSystemConfig	Set a System Configuration parameter
shutdownSystem	Shutowns the InfiniDB System
startSystem	Starts a stopped InfiniDB System
stopSystem	Stops the processing of the InfiniDB System
suspendDatabaseWrites	Suspend performing writes to the InfiniDB Database
switchParentOAMModule	Switches the Active Parent OAM Module to another Performance Module.
system	Execute a system shell command.
unassignElasticIPAddress	Unassign Amazon Elastic IP Address

3 System Maintenance

This chapter describes how to upgrade InfiniDB software and how to use the InfiniDB Console for server operations.

Using the InfiniDB Console, you can perform the following system and server operations: stop, start, restart, and shut down.

3.1 Upgrading InfiniDB

InfiniDB issues new releases and software patches as needed to fix issues or provide better performance.

3.1.1 InfiniDB software

The InfiniDB software packages that you can patch:

- **InfiniDB Platform Software** - This software configures the modules and provides the InfiniDB performance capabilities and InfiniDB Console interface.

When advised by InfiniDB Technical Support, you should apply recommended patches.

3.1.2 Upgrading InfiniDB

The InfiniDB System must be taken offline. The front-end database and InfiniDB database need to be backed up before you upgrade InfiniDB. InfiniDB keeps and uses the existing Calpont.xml file during the upgrade process.

1. Stop InfiniDB. See “Stopping the system”.
2. Back up front-end database and InfiniDB database. See “Performing Backup and Recovery”.
3. In a browser address window, go to **ftp://ftp.calpont.com** and press **Enter**.
4. Download the RPM(s).
5. Extract the files.
6. Install the upgrade. Please see the “InfiniDB Installation Guide” for installation/upgrade instruction.

3.2 Storage Information

The InfiniDB System is typically configured with RAID 0+1 or commonly called RAID10, which is a stripe of mirrors and the system is partitioned with logical unit numbers (LUN). To better understand when additional storage capacity is needed, see “Performance Tuning”.

3.3 System Operations

You can stop, start, restart or shut down the system. Additionally, you can disable and enable modules.

If you disable a module and then restart the system, the disabled module does not restart, but stays offline until you enable it.

There following options are available when stopping the system or servers:

- Graceful (Default option)- Processes perform any gradual shutdown operations they might need.
- Forceful - Processes are stopped no matter what state they are in.
- Acknowledgement Yes - The system informs you when the operation has been successfully or unsuccessfully performed.
- Acknowledgement No (Default option) - The system returns a success message when the command is received, but the operation might still be processing.

3.3.1 Stopping the system

Stopping the system stops the application processes. The platform process that supports the Management Console and System Alarms remain active. If you want to stop the system or server and immediately start the processes again (typically if the system or a server hangs), you can restart the system or servers as shown in the section “Restarting the system”.

To stop the system:

From the InfiniDB Console, type **stopSystem**

Press **y**

The system processes stop.

3.3.2 Starting the system or modules

You can start the system or module application processes with the following commands:

To start the system:

From the InfiniDB Console, type **startSystem**

The system processes start.

3.3.3 Restarting the system

When you want to stop and immediately start application processes, you can perform a restart. You restart the system or modules application processes with the following commands:

To restart the system:

From the InfiniDB Console, type **restartSystem**

The system processes stop and restart.

3.3.4 Shutting down the system

When you perform a shutdown, all InfiniDB processes are stopped. This command would be used mainly when performing software upgrades.

To shutdown the system:

From the InfiniDB Console, type **shutdownSystem**

Press **y**

The system is shutdown.

3.3.5 Disabling and Enabling System Modules

When you want to disable or enable a module either with the system ACTIVE or OFFLINE, you can do it with the following commands:

Note: Disabling a module may result in data loss if the data is local to the PM. If the data is SAN mounted, the dbroots would need to be moved to other PMs. Please refer to “Database Files (DBRoots)” of the Installation Guide for more information on DBRoots and see “Moving DBRoots” of this guide for more information on moving DBRoots.

To disable a module:

From the InfiniDB Console, type **alterSystem-disableModule *module_id***

Example: **alterSystem-disablemodule PM2, PM3**

Press **y**

The module is stopped and disabled.

To enable a module:

From the InfiniDB Console, type **alterSystem-enableModule *module_id***

Example: **alterSystem-enableModule PM2, PM3**

Press **y**

The module is enabled and started.

4 System Status

The InfiniDB Console allows you to monitor the system without using a third-party Network Monitoring System (NMS). InfiniDB recommends you use a third-party NMS for more detailed system monitoring. The system is set up with factory defaults. To configure system monitoring, see “Configuring System Status Monitoring”.

4.1 Viewing system status

The system status shows the status of the system and all equipped servers. The table below shows the available system and server statuses.

To view the system status:

Type `getSystemStatus`

The system and server status appears.

Table 1 - System and Module Statuses

Status	Definition
Active	The system, server, or Network Interface Card (NIC) is available to process database requests.
Auto Disabled	Disabled as a result of a server failure.
Auto Init	Auto initialization mode during a fault recovery.
Auto Offline	The system or server is offline due to a fault.
Busy_Init	The module/system is performing an initialization task at startup time before going to the ACTIVE state.
Degraded	The server is active, but the performance is degraded. A server is degraded when a NIC is not working.
Down	Communication failure.
Failed	A stop/start/restart request for the system or a server failed.
Initial	Initial state after a system reboot or install and before any action is taken.
Man Disabled	Disabled as a result of executing the <code>altersystem-disableModule</code> command.
Man Init	Manual initialization mode during a start or restart command.
Man Offline	The system or server was taken offline with the stop or shutdown command.
Up	Successfully communicating.

When all servers are **active**, then the system status is **active**. If one server is **man offline** and the others are **active**, the system is **man offline**. All equipped servers must be **active** before the system is shown as **active**.

4.2 Viewing process status

The Process Configuration data is stored in the ProcessConfig.xml file in the /usr/local/Calpont/etc directory.

To view process status:

Type `getProcessStatus`

The processes status appears.

The table below shows the supported process states.

Table 2 - Process Statuses

Status	Definition
Active	The process is fully functional.
Auto Init	Auto initialization mode during a fault recovery
Auto Offline	The process is offline due to a fault.
Busy Init	The process is performing an initialization task at startup time before going to the ACTIVE state.
Failed	A stop/start/restart request for a process failed.
Hot Standby	The process is functional in a standby/ready state in case a failover occurs.
Initial	State after a system reboot or install and before any action is taken
Man Init	Manual initialization mode during a start or restart command
Man Offline	The process was taken offline with the stop or shutdown command.
Standby Init	Manual initialization mode during a start or restart command of a Hot Standby process.

4.3 Viewing network configuration

The network configuration lists the servers and states of the NIC cards.

To view network system configuration:

Type `getSystemNetworkConfig`

The system and server status appears.

Table 3 - Network Configuration Statuses

Term	Definition
Device Name	The server configuration type. Example: UMs or PMs.
Device Description	The server configuration type and unique numbering scheme. Example: User Module #1.
NIC ID	This field shows the available NICs on each server.
Host Name	Host name of each server.
IP Address	IP Address of each server.
NIC State	A NIC is either UP or DOWN.

4.4 Viewing module information

The module configuration provides the server names and IP addresses for all modules.

To view the module configuration:

Type `getModuleConfig`

Module Name Configuration

Module 'uml' Configuration information

```
ModuleType = um
ModuleDesc = User Module #1
ModuleIPAdd NIC ID 1 = 10.100.7.83
ModuleHostName NIC ID 1 = qaftest7
ModuleIPAdd NIC ID 2 = 10.100.107.83
ModuleHostName NIC ID 2 = qaftest7b
```

Module 'pml' Configuration information

```
ModuleType = pm
ModuleDesc = Performance Module #1
ModuleIPAdd NIC ID 1 = 10.100.7.10
ModuleHostName NIC ID 1 = srvqaperf2
ModuleIPAdd NIC ID 2 = 10.100.107.10
ModuleHostName NIC ID 2 = srvqaperf2b
DBRootIDs assigned = 1
```

4.5 Monitoring module usage and thresholds

This section explains the module thresholds that are monitored by InfiniDB and how to view them.

To view module usage and thresholds:

Type `getmoduletype`

ModuleType 'um' Configuration information

ModuleDesc = User Module

RunType = LOADSHARE

ModuleCount = 1

ModuleHostName and ModuleIPAddr for NIC ID 1 on module 'um1' = qafest7 , 10.100.7.83

ModuleHostName and ModuleIPAddr for NIC ID 2 on module 'um1' = qafest7b ,
10.100.107.83

ModuleCPUCriticalThreshold % = 0

ModuleCPUMajorThreshold % = 0

ModuleCPUMinorThreshold % = 0

ModuleCPUMinorClearThreshold % = 0

ModuleDiskCriticalThreshold % = 90

ModuleDiskMajorThreshold % = 80

ModuleDiskMinorThreshold % = 70

ModuleMemCriticalThreshold % = 90

ModuleMemMajorThreshold % = 0

ModuleMemMinorThreshold % = 0

ModuleSwapCriticalThreshold % = 90

ModuleSwapMajorThreshold % = 80

ModuleSwapMinorThreshold % = 70

ModuleDiskMonitorFileSystem#1 = /

ModuleType 'pm' Configuration information

ModuleDesc = Performance Module

RunType = SIMPLEX

ModuleCount = 6

ModuleHostName and ModuleIPAddr for NIC ID 1 on module 'pm1' = srvqaperf2 ,
10.100.7.10

ModuleHostName and ModuleIPAddr for NIC ID 2 on module 'pm1' = srvqaperf2b ,
10.100.107.10

ModuleHostName and ModuleIPAddr for NIC ID 1 on module 'pm2' = srvswdev10 ,
10.100.7.5

ModuleHostName and ModuleIPAddr for NIC ID 2 on module 'pm2' = srvswdev10b ,
10.100.107.5

ModuleHostName and ModuleIPAddr for NIC ID 1 on module 'pm3' = srvqaperf3 ,
10.100.7.15

ModuleHostName and ModuleIPAddr for NIC ID 2 on module 'pm3' = srvqaperf3b ,
10.100.107.15

ModuleHostName and ModuleIPAddr for NIC ID 1 on module 'pm4' = srvqaperf4 ,
10.100.7.81

ModuleHostName and ModuleIPAddr for NIC ID 2 on module 'pm4' = srvqaperf4b ,
10.100.107.81

ModuleHostName and ModuleIPAddr for NIC ID 1 on module 'pm5' = srvqaperf5 ,
10.100.7.82

ModuleHostName and ModuleIPAddr for NIC ID 2 on module 'pm5' = srvqaperf5b ,
10.100.107.82

ModuleHostName and ModuleIPAddr for NIC ID 1 on module 'pm6' = srvqaperf7 ,
10.100.7.35

ModuleHostName and ModuleIPAddr for NIC ID 2 on module 'pm6' = srvqaperf7b ,
10.100.107.35

```
DBRootIDs assigned to module 'pm1' = 1
DBRootIDs assigned to module 'pm2' = 2
DBRootIDs assigned to module 'pm3' = 3
DBRootIDs assigned to module 'pm4' = 4
DBRootIDs assigned to module 'pm5' = 5
DBRootIDs assigned to module 'pm6' = 6
ModuleCPUCriticalThreshold % = 0
ModuleCPUMajorThreshold % = 0
ModuleCPUMinorThreshold % = 0
ModuleCPUMinorClearThreshold % = 0
ModuleDiskCriticalThreshold % = 90
ModuleDiskMajorThreshold % = 80
ModuleDiskMinorThreshold % = 70
ModuleMemCriticalThreshold % = 90
ModuleMemMajorThreshold % = 0
ModuleMemMinorThreshold % = 0
ModuleSwapCriticalThreshold % = 90
ModuleSwapMajorThreshold % = 80
ModuleSwapMinorThreshold % = 70
ModuleDiskMonitorFileSystem#1 = /
```

4.5.1 Module disk usage

The file systems on each local server can be monitored for the percentage used. There are three thresholds you can set for the disk usage for each server: Critical, Major, and Minor. When a threshold is crossed, an alarm is issued.

The alarms that are issued when the configured disk file system usage crosses each threshold are shown below:

- DISK_USAGE_LOW is set when usage percentage goes above threshold ModuleDiskMinorThreshold
- DISK_USAGE_MED is set when usage percentage goes above threshold ModuleDiskMajorThreshold
- DISK_USAGE_HIGH is set when usage percentage goes above threshold ModuleDiskCriticalThreshold

These same alarms are cleared as the usage percentage starts going down. The clearing is done in this manner:

- DISK_USAGE_HIGH is cleared when the usage percentage usage goes below the threshold ModuleDiskCriticalThreshold
- DISK_USAGE_MED is cleared when the usage percentage usage goes below the threshold ModuleDiskMajorThreshold
- DISK_USAGE_LOW is cleared when the usage percentage usage goes below the threshold ModuleDiskMinorThreshold

See “Configuring module disk usage” for more information.

4.5.2 Memory usage

The memory usage can be monitored for the percentage used. There are three thresholds you can set for the disk usage for each server: Critical, Major and Minor. When a threshold is crossed, an alarm is issued.

The alarms that are issued when the configured disk file system usage crosses each threshold are shown below:

- MEMORY_USAGE_LOW is set when usage percentage goes above threshold
ModuleMemMinorThreshold
- MEMORY_USAGE_MED is set when usage percentage goes above threshold
ModuleMemMajorThreshold
- MEMORY_USAGE_HIGH is set when usage percentage goes above threshold
ModuleMemCriticalThreshold

These same alarms are cleared as the usage percentage starts going down. The clearing is done in this manner:

- MEMORY_USAGE_HIGH is cleared when the usage percentage usage goes below the threshold
ModuleMemCriticalThreshold
- MEMORY_USAGE_MED is cleared when the usage percentage usage goes below the threshold
ModuleMemMajorThreshold
- MEMORY_USAGE_LOW is cleared when the usage percentage usage goes below the threshold
ModuleMemMinorThreshold

4.5.3 Swap space usage

The swap space usage can be monitored for the percentage used. There are three thresholds you can set for the swap space usage for each server: Critical, Major, and Minor. When a threshold is crossed, an alarm is issued.

The alarms that are issued when the Configured Disk File System usage crosses each threshold are shown below:

- SWAP_USAGE_LOW is set when usage percentage goes above threshold
ModuleSwapMinorThreshold
- SWAP_USAGE_MED is set when usage percentage goes above threshold
ModuleSwapMajorThreshold
- SWAP_USAGE_HIGH is set when usage percentage goes above threshold
ModuleSwapCriticalThreshold

These same alarms are cleared as the usage percentage starts going down. The clearing is done in this manner:

- SWAP_USAGE_HIGH is cleared when the usage percentage usage goes below the threshold
ModuleSwapCriticalThreshold
- SWAP_USAGE_MED is cleared when the usage percentage usage goes below the threshold
ModuleSwapMajorThreshold

- SWAP_USAGE_LOW is cleared when the usage percentage usage goes below the threshold ModuleSwapMinorThreshold.

4.5.4 Disk thresholds

You can set thresholds to alert you when a specified percentage of a file system is being used on a server.

There are three thresholds you can set for the disk usage for each server: Critical, Major, and Minor. When a threshold is crossed, an alarm is issued.

The alarms that are issued when the Disk File System usage crosses each threshold are shown below:

- DISK_USAGE_LOW is set when usage percentage goes above threshold RAIDMinorThreshold.
- DISK_USAGE_MED is set when usage percentage goes above threshold RAIDMajorThreshold.
- DISK_USAGE_HIGH is set when usage percentage goes above threshold RAIDCriticalThreshold.

These same alarms are cleared as the usage percentage starts going down as shown below:

- DISK_USAGE_HIGH is cleared when the usage percentage usage goes below the threshold RAIDCriticalThreshold.
- DISK_USAGE_MED is cleared when the usage percentage usage goes below the threshold RAIDMajorThreshold.
- DISK_USAGE_LOW is cleared when the usage percentage usage goes below the threshold RAIDMinorThreshold.

4.5.5 CPU thresholds

You can set thresholds to alert you when the CPU usage on a server reaches a specific percentage.

There are four thresholds for the server CPU usage: Critical, Major, Minor and MinorClear. When a threshold is crossed, an alarm is issued. To disable thresholds, set the thresholds to 0.

The alarms that are issued when the Server total CPU usage crosses each threshold are shown below:

- CPU_USAGE_LOW is set when CPU goes above threshold ModuleCPUMinorThreshold
- CPU_USAGE_MED is set when CPU goes above threshold ModuleCPUMajorThreshold
- CPU_USAGE_HIGH is set when CPU goes above threshold ModuleCPUCriticalThreshold

These same alarms are cleared as the CPU usage starts going down. To prevent alarm thrashing, the clearing is done in this manner:

- CPU_USAGE_HIGH is cleared when the CPU usage goes below the threshold ModuleCPUMajorThreshold
- CPU_USAGE_MED is cleared when the CPU usage goes below the threshold ModuleCPUMinorThreshold
- CPU_USAGE_LOW is cleared when the CPU usage goes below the threshold ModuleCPUMinorClearThreshold

4.6 Monitoring alarms

Alarms are triggered when system, servers, processes, or hardware are down, power on test has warning, process fail or time out, or when threshold percentages are met. These thresholds are set for the following:

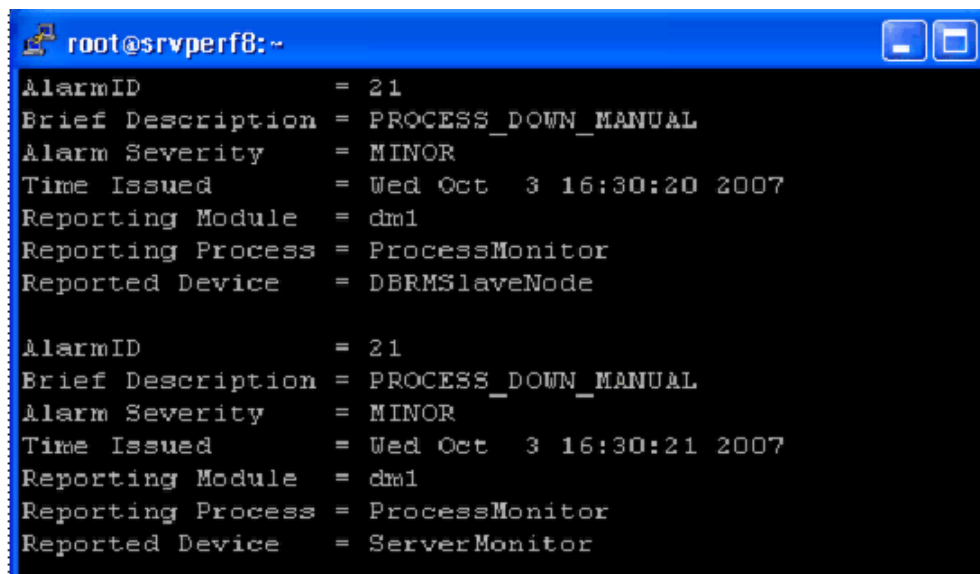
- Module disk usage
- Memory usage
- Swap space usage
- RAID usage
- CPU usage
- This section explains how to view alarms, and describes the alarm definitions. To change default thresholds, see “Configuring System Status Monitoring”.

4.6.1 Viewing active alarms

To view active alarms:

Type `getActiveAlarms`

Below is a sample alarm file.



```
root@srvperf8:~  
AlarmID           = 21  
Brief Description  = PROCESS_DOWN_MANUAL  
Alarm Severity    = MINOR  
Time Issued       = Wed Oct  3 16:30:20 2007  
Reporting Module  = dml  
Reporting Process = ProcessMonitor  
Reported Device   = DBRMSlaveNode  
  
AlarmID           = 21  
Brief Description  = PROCESS_DOWN_MANUAL  
Alarm Severity    = MINOR  
Time Issued       = Wed Oct  3 16:30:21 2007  
Reporting Module  = dml  
Reporting Process = ProcessMonitor  
Reported Device   = ServerMonitor
```

4.6.2 Viewing archived alarms

Alarms are archived each day at midnight. The InfiniDB system saves seven days of alarm history.

To view historical alarms:

Type **getAlarmHistory (yyyymmdd)**

In place of the date, you can type **today** to get the alarm log for the current day.

Example statements:

```
getAlarmHistory today
getAlarmHistory 20070923
```

Table 4 - Alarm Definitions

Description	Definition
Alarm ID#1 CPU_USAGE_HIGH	The usage on the indicated CPU has exceeded its high threshold
Alarm ID#2 CPU_USAGE_MED	The usage on the indicated CPU has exceeded its medium threshold
Alarm ID#3 CPU_USAGE_LOW	The usage on the indicated CPU has exceeded its low threshold
Alarm ID#4 DISK_USAGE_HIGH	The usage on the indicated disk drive has exceeded its high threshold
Alarm ID#5 DISK_USAGE_MED	The usage on the indicated disk drive has exceeded its medium threshold
Alarm ID#6 DISK_USAGE_LOW	The usage on the indicated disk drive has exceeded its low threshold
Alarm ID#7 MEMORY_USAGE_HIGH	The usage on the indicated memory has exceeded its high threshold
Alarm ID#8 MEMORY_USAGE_MED	The usage on the indicated memory has exceeded its medium threshold
Alarm ID#9 MEMORY_USAGE_LOW	The usage on the indicated memory has exceeded its low threshold
Alarm ID #10 SWAP_USAGE_HIGH	The usage on the indicated swap has exceeded its high threshold
Alarm ID #11 SWAP_USAGE_MED	The usage on the indicated swap has exceeded its medium threshold
Alarm ID #12 SWAP_USAGE_LOW	The usage on the indicated swap has exceeded its low threshold
Alarm ID #13 PROCESS_DOWN_AUTO	Process is auto out-of-service
Alarm ID #14 MODULE_DOWN_AUTO	Module is auto out-of-service
Alarm ID #15	System is auto out-of-service

SYSTEM_DOWN_AUTO	
Alarm ID #16 POWERON_TEST_SEVERE	A Module Power-On Severe Warning error occurred. Test failures indicate that the system is running in a degraded state
Alarm ID #17 POWERON_TEST_WARNING	Warning error occurred. Test failures indicate that the system is running with little or no degradation in performance
Alarm ID #18 HARDWARE_HIGH	A hardware device's resource has exceeded its high threshold
Alarm ID #19 HARDWARE_MED	A hardware device's resource has exceeded its medium threshold
Alarm ID #20 HARDWARE_LOW	A hardware device's resource has exceeded its low threshold
Alarm ID #21 PROCESS_DOWN_MANUAL	Process is manually out-of-service
Alarm ID #22 MODULE_DOWN_MANUAL	Module is manually out-of-service
Alarm ID #23 SYSTEM_DOWN_MANUAL	System is manually out-of-service
Alarm ID #24 EXT_DEVICE_DOWN_AUTO	External Device is auto out-of-service
Alarm ID #25 PROCESS_INIT_FAILURE	Process Initialization failed or Timed-out
Alarm ID #26 NIC_DOWN_AUTO	NIC Interface is auto out-of-service
Alarm ID #27 DBRM_LOAD_DATA_ERROR	Error occurred during loading DBRM data
Alarm ID #28 INVALID_SW_VERSION	Invalid software version found on a module which doesn't match the version running on the User Module
Alarm ID #29 STARTUP_DIAGNOSTICS_FAILURE	A failure occurred when the system startup diagnostics was executed on a module. Log files can be viewed for details on the failure
Alarm ID #30 CONN_FAILURE	A connection failure occurred that could affect the performance of system processing.
Alarm ID #31 DBRM_READ_ONLY	The DBRM is now read-only and updates to the database are not possible.
ALARM ID #33 MODULE_SWITCH_ACTIVE	An outage of Parent OAM Module or the execution of the switchParentOAMModule command initiated a change of the Parent OAM Module.

Table 5 - Alarm Severity

Severity	Description
Critical	A condition occurred that requires immediate attention and could result in loss of data.
Major	A condition occurred that impairs service and needs attention.
Minor	A condition occurred that hasn't impaired service, but needs attention so that it doesn't impair service in the future.

4.6.3 Viewing top CPU users

You can view the top users on a single module or the top users on all modules in InfiniDB. The default is to display the top five users, but you can specify to display a range from 1-10 processes. Processes that use less than 1% CPU are not displayed.

To monitor top users on a single module:

Type `getmodulecpuusers module_ID`

To monitor to the top CPU users on User Module 1, the command is:

`getmodulecpuusers um1`

To monitor top users on InfiniDB:

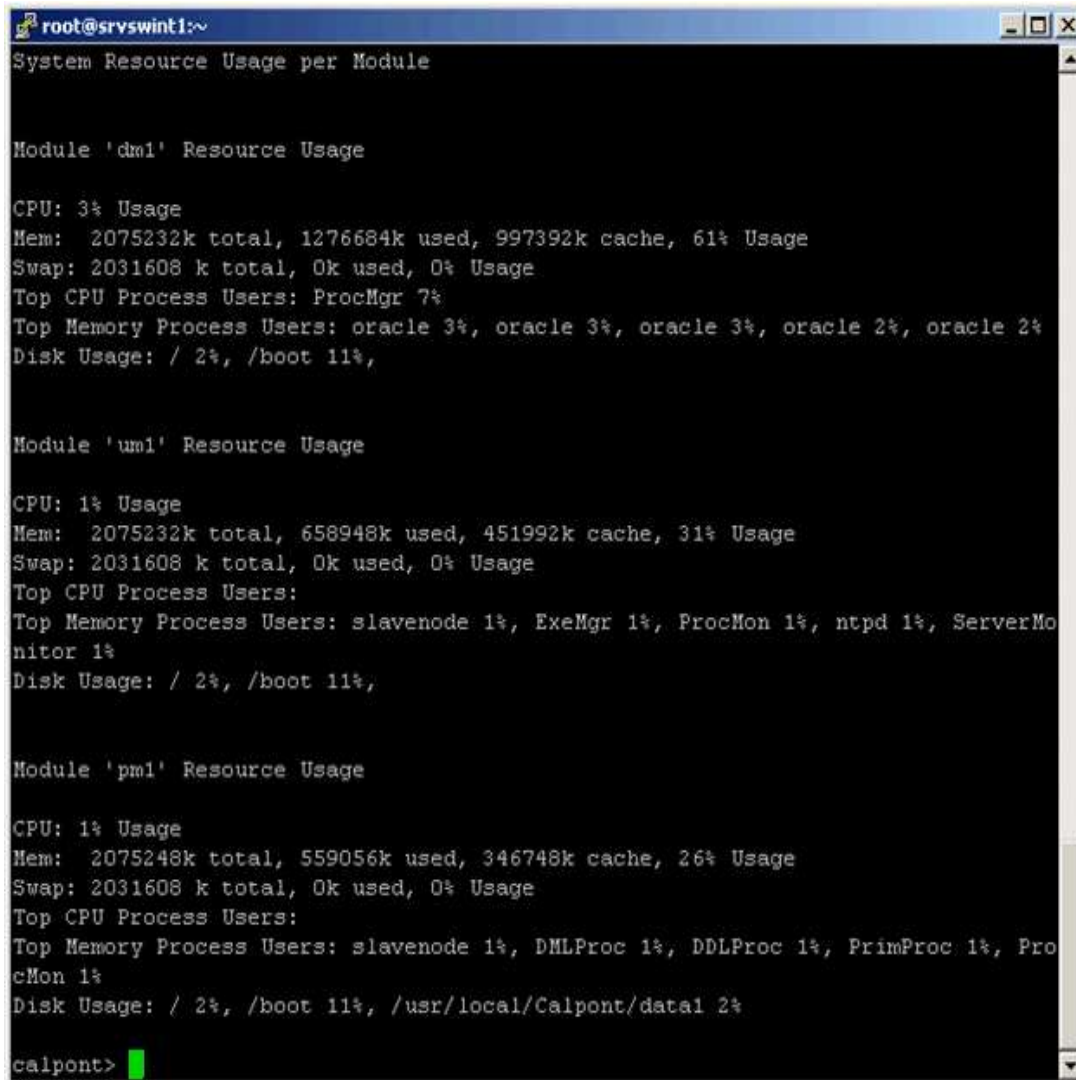
`getsystemcpuusers`

4.6.4 Monitoring resource usage

You can monitor the system CPU usage, memory, cache and swap space for the system by module.

To monitor system resources:

`getsystemresourceusage`



```
root@srvswint1:~  
System Resource Usage per Module  
  
Module 'dm1' Resource Usage  
  
CPU: 3% Usage  
Mem: 2075232k total, 1276684k used, 997392k cache, 61% Usage  
Swap: 2031608 k total, 0k used, 0% Usage  
Top CPU Process Users: ProcMgr 7%  
Top Memory Process Users: oracle 3%, oracle 3%, oracle 3%, oracle 2%, oracle 2%  
Disk Usage: / 2%, /boot 11%,  
  
Module 'um1' Resource Usage  
  
CPU: 1% Usage  
Mem: 2075232k total, 658948k used, 451992k cache, 31% Usage  
Swap: 2031608 k total, 0k used, 0% Usage  
Top CPU Process Users:  
Top Memory Process Users: slavenode 1%, ExeMgr 1%, ProcMon 1%, ntpd 1%, ServerMonitor 1%  
Disk Usage: / 2%, /boot 11%,  
  
Module 'pm1' Resource Usage  
  
CPU: 1% Usage  
Mem: 2075248k total, 559056k used, 346748k cache, 26% Usage  
Swap: 2031608 k total, 0k used, 0% Usage  
Top CPU Process Users:  
Top Memory Process Users: slavenode 1%, DMLProc 1%, DDLProc 1%, PrimProc 1%, ProcMon 1%  
Disk Usage: / 2%, /boot 11%, /usr/local/Calpont/data1 2%  
  
calpont>
```

5 Configuring System Status Monitoring

This chapter describes how to configure the system, processes, servers, CPU usage, disk usage, RAID, memory usage, and swap space.

For more information about monitoring the system, see “System Status” and for a complete list of InfiniDB Commands, see “Complete list of InfiniDB commands”.

5.1 *Management console overview*

This chapter explains how to configure the following:

- System
- Modules
- Processes
- Alarms
- Logging

5.1.1 InfiniDB configuration files

When you configure the system using the Management Console, the changes are stored in XML files that reside in the “etc” directory. Some of these configuration items are updated by InfiniDB Processes and the InfiniDB **postConfig** script at installation time.

- **AlarmConfig.xml** - Alarm configuration file.
- **Calpont.xml** - System and module configuration file.
- **ConsoleCmds.xml** - Management Console tool help file.
- **ProcessConfig.xml** - Process configuration file.

5.1.2 Linux shell

You need a SSH client to connect to the Linux shell. Once you are connected to the Linux shell, you enter the command to run the Management Console and it runs in the Linux shell. Many of the Linux-shell features are available to the Management Console such as:

- Entering partial command names to execute commands
- Recalling previous commands using the **Up** and **Down** arrows
- Using hot keys to edit and navigate commands
- For tips on using the Management Console, “ InfiniDB Console tips”.

5.2 *Configuring system*

You can modify the following system settings: the heartbeat period and counts, RAID thresholds, monitoring, alarm, and logging configurations. The system configuration file also displays the current software and hardware versions used with the InfiniDB Database Platform. This can be useful troubleshooting information.

Note: Most of the data in the configuration file is stored in the **Calpont.xml** file located in the **/usr/local/Calpont/etc** directory. The master copy of the configuration file is kept on the primary Performance Module. Upon startup, the system reads the information from the configuration file and allocates the necessary resources (e.g. data caches, etc.) needed for system activity. Any changes to this file in a multi-server configuration should be done on this primary Performance Module so the changes can get distributed to all modules.

Some data is extracted from other sources when the **getSystemConfig** command is executed.

To view the system configuration:

At the Linux prompt, type **cmconsole**.

Type **getSystemConfig** and press **Enter**.

```
getsystemconfig    Wed Mar 28 10:17:49 2012

System Configuration

SystemName = sn-qaperf-1
SoftwareVersion = 3.0
SoftwareRelease = 0
ParentOAMModuleName = pm1
StandbyOAMModuleName = pm2
NMSIPAddr = 0.0.0.0
ModuleHeartbeatPeriod = 1
ModuleHeartbeatCount = 3
DBRootCount = 6
DBRoot1 = /usr/local/Calpont/data1
DBRoot2 = /usr/local/Calpont/data2
DBRoot3 = /usr/local/Calpont/data3
DBRoot4 = /usr/local/Calpont/data4
DBRoot5 = /usr/local/Calpont/data5
DBRoot6 = /usr/local/Calpont/data6
DBRMRoot = /usr/local/Calpont/data1/systemFiles/dbrm/BRM_saves
ExternalCriticalThreshold = 90
ExternalMajorThreshold = 80
ExternalMinorThreshold = 70
MaxConcurrentTransactions = 1000
SharedMemoryTmpFile = /tmp/CalpontShm
VersionBufferFileSize = 1
OIDBitmapFile = /usr/local/Calpont/data1/systemFiles/dbrm/oidbitmap
FirstOID = 3000
TransactionArchivePeriod = 10
```

To view the storage configuration, use the **getstorageconfig** command to see what dbroots are assigned to which pms along with the system storage configuration.

To view the storage configuration:

At the Linux prompt, type **cmconsole**.

Type **getStorageConfig** and press **Enter**.

```
getstorageconfig    Wed Mar 28 10:40:34 2012

System Storage Configuration
```

```
Storage Type = internal
System DBRoot count = 6
DBRoot IDs assigned to 'pm1' = 1
DBRoot IDs assigned to 'pm2' = 2
DBRoot IDs assigned to 'pm3' = 3
DBRoot IDs assigned to 'pm4' = 4
DBRoot IDs assigned to 'pm5' = 5
DBRoot IDs assigned to 'pm6' = 6
```

5.2.1 Setting module heartbeats

Heartbeat monitoring occurs between modules in the InfiniDB Database Platform. The module heartbeat settings are the same for all modules in the InfiniDB Database Platform.

The module heartbeat period refers to how often the heartbeat test is performed. For example, if you set the period to 5, then the heartbeat test is performed every 5 seconds.

The module heartbeat count refers to how many failures in a row must take place before a fault is processed. To disable heartbeat monitoring, set the period to -1.

To set the server heartbeat period and count:

1. At the Linux prompt, type **cmconsole**.
2. Type **setSystemConfig ModuleHeartbeatPeriod n** (where *n*= number of seconds) and press **Enter**.

A message appears when the period is successfully set.

3. Type **setSystemConfig ModuleHeartbeatCount n** (where *n*= number of failures) and press **Enter**.

A message appears when the count is successfully set.

5.2.2 Configuring Disk thresholds

You can set thresholds to alert you when a specified percentage of a file system is being used on a server.

There are three thresholds you can set for the disk usage for each server: Critical, Major and Minor. When a threshold is crossed, an alarm is issued. To disable thresholds, set the threshold to 0.

The alarms that are issued when the Disk File System usage crosses each threshold are shown below:

- DISK_USAGE_LOW is set when usage percentage goes above threshold EXTERNALMinorThreshold.
- DISK_USAGE_MED is set when usage percentage goes above threshold EXTERNALMajorThreshold.
- DISK_USAGE_HIGH is set when usage percentage goes above threshold EXTERNALCriticalThreshold.

These same alarms are cleared as the usage percentage starts going down as shown below:

- DISK_USAGE_HIGH is cleared when the usage percentage usage goes below the threshold EXTERNALCriticalThreshold.

- DISK_USAGE_MED is cleared when the usage percentage usage goes below the threshold EXTERNALMajorThreshold.
- DISK_USAGE_LOW is cleared when the usage percentage usage goes below the threshold EXTERNALMinorThreshold.

The system configuration parameters are:

- **EXTERNALMinorThreshold**
- **EXTERNALMajorThreshold**
- **EXTERNALCriticalThreshold**

To configure Disk thresholds:

1. At the Linux prompt, type **cmconsole**.
2. Type **setSystemConfig EXTERNAL(Minor/Major/Critical) Threshold n** (where *n*= percentage of file system used) and press **Enter**.

A message appears when the threshold is successfully set.

For example, to modules to issue a disk usage medium alarm when the file system is 25% full, enter the following command:

```
setSystemConfig EXTERNALMajorThreshold 25
```

TIP: When setting thresholds, minor should be less than major, and major should be less than critical.

5.2.3 Configuring archived transaction log periods

You can change how often the transaction log file is archived by configuring the systemconfiguration file. The default period is set to 10 minutes. You can select between 0 and 60 minutes.

To modify archived transaction log period:

Connect to Linux using your SSH client.

You will be prompted to enter a user name and password.

Type your Linux user name and password and press **Enter**.

A Linux prompt appears.

At the Linux prompt, type **cmconsole**.

The cmconsole prompt appears.

Type **setsystemconfig TransactionArchivePeriod (log period)** and press **Enter**.

5.3 Configuring modules

You can configure the following module items:

- CPU thresholds
- Disk monitor file system usage
- Disk thresholds
- Module memory thresholds
- Module swap thresholds

5.3.1 Configuring module CPU thresholds

You can set thresholds to alert you when the CPU usage on a module reaches a specific percentage. For more information about CPU Thresholds, see “CPU thresholds”.

There are four thresholds for the module CPU usage: Critical, Major, Minor and MinorClear. When a threshold is crossed, an alarm is issued. To disable thresholds, set the thresholds to 0.

To configure the CPU thresholds:

At the Linux prompt, type **cmconsole**.

Type **setModuleTypeConfig (module name) ModuleCPU(Clear/Minor/Major/Critical)Threshold n** (where *n*= percentage of CPU usage) and press **Enter**.

The CPU threshold is set and a message appears.

For example, to set a module alert to Minor at 50%, type the following:

setModuleTypeConfig pml ModuleCPUMinorThreshold 50.

TIP: When setting thresholds, minor should be less than major, and major should be less than critical.

5.3.2 Configuring module disk usage

The file systems on each local module can be monitored for the percentage used. For more information about server disk usage, see “Module disk usage”.

To configure the module disk usage:

At the Linux prompt, type **cmconsole**.

Type **setModuleTypeConfig (module name) ModuleDisk(Minor/Major/Critical)Threshold n** (where *n*= percentage of disk system used) and press **Enter**.

The module disk usage threshold is set and a message appears.

TIP: When setting thresholds, minor should be less than major, and major should be less than critical.

5.3.3 Configuring swap space usage

The swap space usage can be monitored for the percentage used. There are three thresholds you can set for the swap space usage for each server: Critical, Major and Minor. When a threshold is crossed, an alarm is issued. To disable thresholds, set the thresholds to 0.

For more information about swap space usage, see “Swap space usage”.

To configure the module swap space usage:

At the Linux prompt, type **cmconsole**.

Type **setModuleTypeConfig (module name) ModuleSwap(Minor/Major/Critical) Threshold n** (where *n*= percentage of swap space used) and press **Enter**.

The server swap space usage threshold is set and a message appears.

TIP: When setting thresholds, minor should be less than major, and major should be less than critical.

5.4 Configuring alarms

In this chapter, you learned how to configure the system and servers to trigger alarms for specific events or conditions. You can also set the total number of alarms that can be triggered in a 30-minute period. The InfiniDB system comes with a pre-configured alarm file, stored in the CalpontAlarm.xml file, that has the threshold setting for all alarms at 100. This means that an alarm can be triggered a maximum of 100 times in 30 minutes before the alarm quits being triggered.

To disable a specific alarm from being reported, set the alarm threshold to 0.

5.4.1 Steps to configure alarms

To configure alarms, you use the **setalarmconfig** command with the Alarm ID#, and the parameter to be changed.

At the Linux prompt, type **cmconsole**.

Type **setAlarmConfig (alarmID#) Threshold n** (where *n*= maximum number of times an alarm can be triggered in 30 minutes), and press **Enter**.

For example, if you want to change Alarm ID#23 to have a threshold of 25 you would type the following at the InfiniDB prompt:

setAlarmConfig 23 Threshold 25.

5.5 Configuring logging

The Log Configuration can be displayed using the following command: **getLogConfig**.

There are 5 levels of logging:

- Critical
- Error
- Warning
- Info
- Debug

Each server within the InfiniDB System can have a different log level configured. Since InfiniDB logging is supported through the Syslog functionality, the Syslog service should always be running on each server.

The Log Configuration can be enabled and disabled using the `EnableLog` and `DisableLog` commands. The five logging levels can be individually enabled and disabled or you can enable and disable all five by entering "all" for the log level.

Messages are identified by a unique number. In the following syslog message, items 1-4 are generated automatically by syslog and items 5-8 come from the message object itself.

1. Date & time
2. Host of origin
3. Subsystem name
4. PID
5. Seconds and microseconds of message
6. Session ID, transaction ID & thread ID
7. Message severity level - A single character message-class prefix:
 - D - debug
 - I - info
 - W - warning
 - E - error
 - C - critical
8. Subsystem ID - A 2-digit subsystem ID.
9. Message number - A 4-digit error number. Error message ID's (and thus error message texts) can be shared across subsystem boundaries. This allows reuse of common error messages.
10. Message text - formatted error text after all positional arguments have been substituted.

These log files are stored on each server in `/var/log/Calpont` directory.

5.5.1 Enabling and disabling logging

To enable logging:

At the Linux prompt, type `cmconsole`.

Type `enableLog (module_ID) (log_level)` and press **Enter**.

For example, to enable Critical logging on server PM1, the command is: `enableLog PM1 Critical`.

To disable logging:

At the Linux prompt, type `cmconsole`.

Type `disableLog (module_ID) (log_level)` and press **Enter**.

For example, to disable Info logging on server PM1, the command is: `disableLog PM1 Info`.

6 Loading Data

This chapter describes how to import data into the InfiniDB System. This includes data migrations from other systems or periodic updates during the maintenance window.

6.1 Import overview

Before importing data, you must complete the InfiniDB Installation as documented in the *InfiniDB Installation Guide*. This includes creating a InfiniDB system catalog and creating the tables to which you want to import data.

6.1.1 Estimating load sizes

Business requirements for data latency drive the size of data loads. If access to new data is needed every 30 seconds, then data should be loaded in smaller loads so that it can be written to the database quickly and available for queries.

If access to new data is needed within 24 hours, then data can be loaded overnight in larger data loads.

6.1.2 Import Tasks

Perform the tasks below to import data:

- Copy data files to be imported into the correct directory
- Optionally generate job file depending on usage
- Import data

6.1.3 Import process

InfiniDB does the following during the import:

- Reads data
- Transforms the data into a column orientation
- Tokenizes data values
- Loads tokenized data into database

6.1.4 Directory structure

The default directory structure below is created when InfiniDB is installed. These locations and file names may be overridden using the “-f” and “-l” options described below (i.e., STDIN processing). The bulk load process may now be initiated from any server.

/usr/local/Calpont/data/bulk/

job - Contains the job description XML files that map tables and columns.

data/import - Contains the data files to be imported. This is the default unless you override or run the simpler interface.

log - Contains the log file for each executed job. The file extension is .log.

BulkRoot definition

There is an entry in the Calpont.XML file called **BulkRoot**. This entry allows the user to modify the default of `/usr/local/Calpont/data/bulk` to another directory if so desired.

6.1.5 Importing as non-root user

If `colxml` and `cpimport` are to be executed using a non-root user, the access mode for these utilities must be altered by the system administrator. For example:

- `chmod 4755 /usr/local/Calpont/bin/colxml`
- `chmod 4755 /usr/local/Calpont/bin/cpimport`

6.2 Copy source files

Source files need to be formatted, named correctly, and placed in the InfiniDB import folder if using the default location for InfiniDB to import the data. This may be overridden a number of ways (-p option, using the simpler interface.)

6.2.1 Source file formats

The default delimiter is the '|' (pipe) character.

You can configure the system to accept other formats by providing an optional delimiter parameter during the import. For most character delimiters, use -d and the delimiter character on `colxml`. You may use the -s if using `cpimport`.

Examples:

```
-d ,  
-d |
```

For special characters such as tab, single quote, or tilde, add a backwards slash "\" and then the special character. **Note:** If you use the backwards slash for special characters, it must be wrapped in a single quotes. The examples for each are as follows:

```
-d '\t'  
-d '\~'
```

6.2.1.1 Importing UTF-8 data

When using the `cpimport` utility to import UTF-8 data, the input file must be converted to UTF-8 data. The Linux program `iconv` is one utility to convert it.

```
iconv -f ISO-8859-1 -t UTF-8 < input.txt > output.txt
```

6.2.2 Source file names

For the full interface, the source files must be named with the table name that the data should be imported by default. You may use the -l option to override. For the simple interface, the source file name may be placed on the command line if different than the default table name.

If source data resides in a different directory, you can use the `ln` command to create a symbolic link “import” instead. For example:

```
ln -s /mnt/source /usr/local/Calpont/data/bulk/data/import
```

6.2.3 Import directory

If using the default, you must place source data to be imported into the following directory:

```
/usr/local/Calpont/data/bulk/data/import
```

This may be overridden by using the `-p` option and/or using the simple interface.

6.3 Import Usage

The import tool can be used 2 different ways:

- The first usage is a simple interface to import a single table and no XML job file (colxml) is required.
- The second usage is the full way of using `cpimport` in that you must first run the `colxml` utility to generate the XML job file. This way must be used when importing multiple tables at once but can be used in all scenarios as well.

6.3.1 Importing multiple tables

Multiple tables may be simultaneously imported using the import utility. With the simple interface, multiple jobs may be simultaneously submitted as long as the tables are unique per import or the PMs are unique per import. With the full interface, multiple tables may be imported by either importing all tables within a schema or listing specific tables using the `-t` option in `colxml`. Multiple jobs may be submitted with the traditional interface as long as the tables and jobids are unique.

6.4 Distributed Imports

The import utility has now been expanded to allow for multiple load methods to allow for the ease of use of a single, centralized load process, as well as the scalability to load from separate files without any single bottleneck that may limit overall load performance.

The following sections and diagrams illustrate the various modes available in InfiniDB:

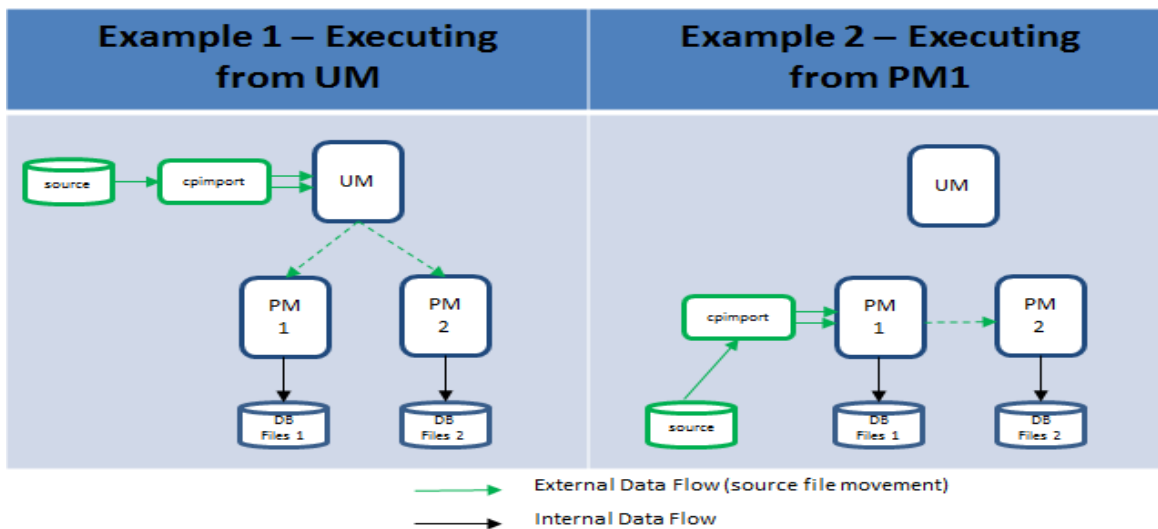
6.4.1 Bulk Load from a central location (Mode 1)

Executing Bulk Load from a central location provides the option to load a single file, with a single command, and have that data distributed properly across a suite of disk resources. Characterized by:

- Single source file
- Single bulk load command

Import Mode 1

Single Command, Central Source



cpimport with Mode 1 will read in the source file and distribute it across either all PMs or specific PMs (-P option). The source file will be split into 10,000 row increments and send them to the respective PMs for import. In a 2 PM example like above, a 100,000 row source file will result in 50,000 rows on each PM.

Note: The splitter sends 10,000 row increments to DBRoots. For example, if you have an unbalanced number of dbroots (PM1 has 1 dbroot and PM2 has 2 dbroots), PM2 will import twice the data that PM1 will.

Examples:

- import the mytable.tbl source file from the current directory into the mytest schema across all PMs:

```
cpimport -m1 mytest mytable mytable.tbl
```
- import the mytable.tbl source file from the current directory into the mytest schema for only PM1 and PM2:

```
cpimport -m1 -P1,2 mytest mytable mytable.tbl
```
- traditional use of import the tables in the mytest schema across all PMs:

```
colxml mytest -j299  
cpimport -m1 -j299
```

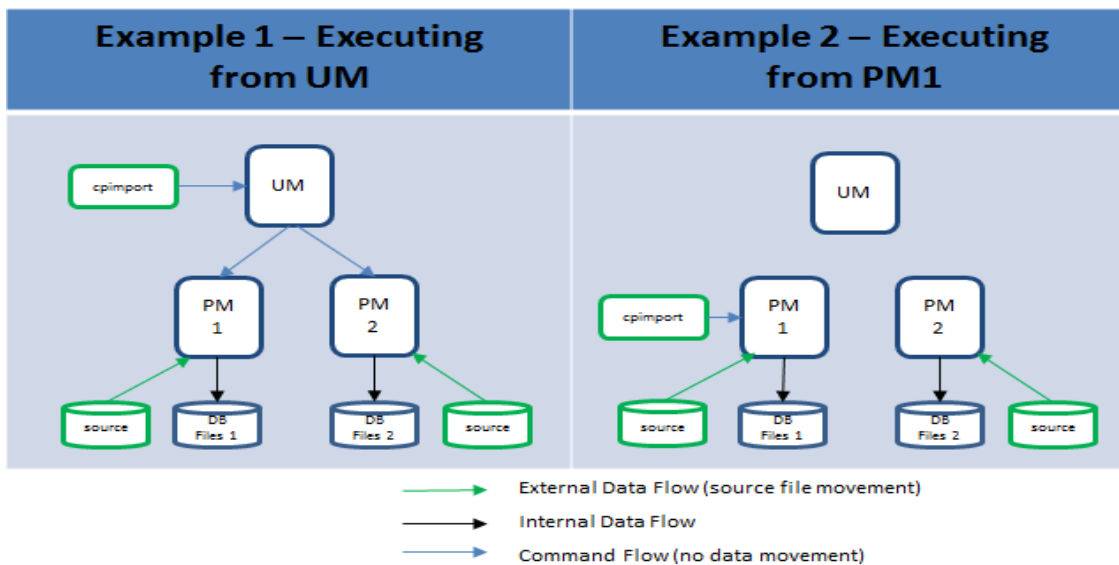
6.4.2 Single Bulk Load Command, Partitioned Source (Mode 2)

Executing Bulk Load from a central location provides the option to load already partitioned data files residing on the PMs. Characterized by:

- N Partitioned load files where N = the number of PMs
- Single bulk load command

Import Mode 2

Single Command, Distributed Source



cpimport with Mode 2 will import the source file that exists down on each PM. The source file on each PM will import into its respective PM's dbroots.

Examples:

- import the mytable.tbl source file from the /home/mydata directory on each PM into the mytest schema for that PM:

```
cpimport -m2 mytest mytable /home/mydata/mytable.tbl
```

- import the mytable.tbl source file from the /home/mydata directory on PM1 into the mytest schema for only PM2:

```
cpimport -m2 -P2 mytest mytable /home/mydata/mytable.tbl
```

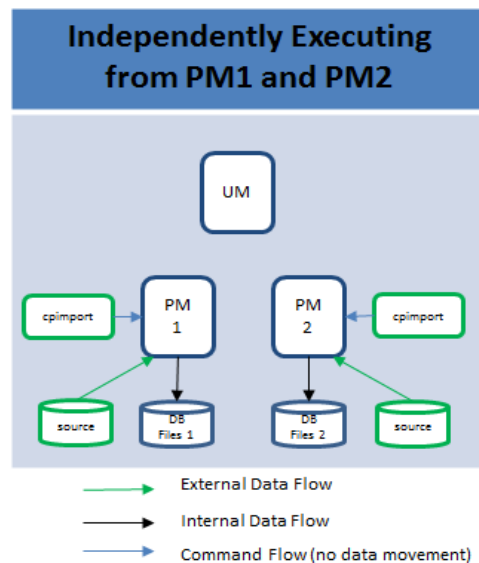
6.4.3 Independent Bulk Load Command, Partitioned Source (Mode 3)

Executing Bulk Loads individually on the PMs without any coordination from a central point. Here, one file/source can be loaded onto one PM without the other PMs knowledge. N concurrent loads into 1 table will be allowed, assuming N source files. Characterized by:

- Up to N Partitioned load files where N = the number of PMs
- Up to N bulk load commands where N=the number of PMs.

Import Mode 3

Independent/Parallel Commands, Partitioned Source



With this mode, the user will initiate the cpimport on the PM which will import the source file that exists on that PM. Concurrent imports may be executed on every PM for the same table.

Examples:

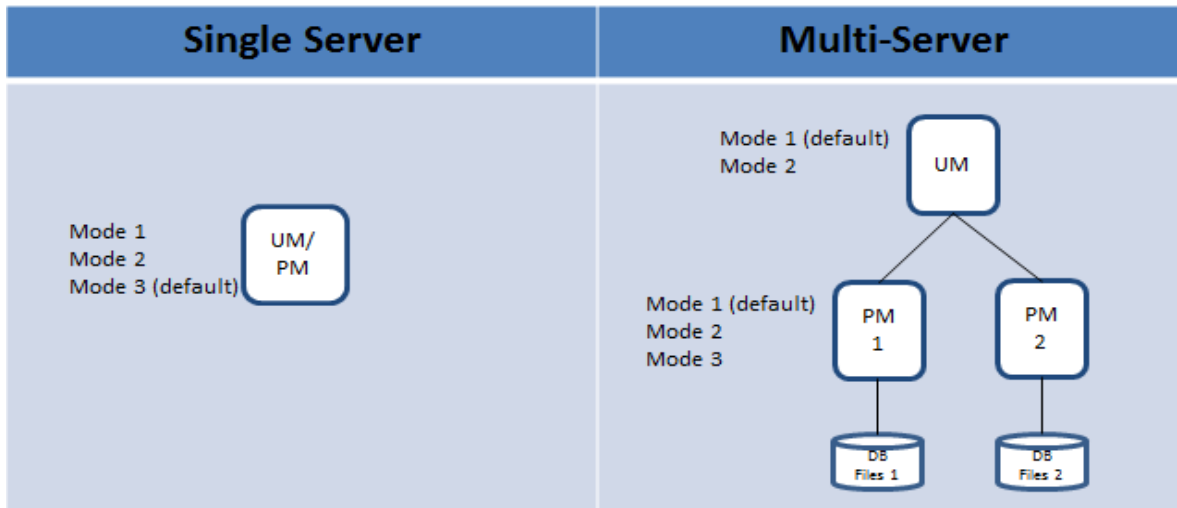
- import the mytable.tbl source file from the /home/mydata directory on PM1 into the mytest schema for PM1:

```
cpimport -m3 mytest mytable /home/mydata/mytable.tbl
```

6.4.4 Mode Availability

The following diagram illustrates what mode is available to run on each type of server and the default for each:

Mode Availability



6.5 Generate job file

Generating a job file is no longer mandatory if you are running an import on a single table and executing the simple cpimport interface. But you must create an XML job file before you can import data if you are loading multiple tables in a single job and using the full cpimport interface. If you are using the job file, you should create a new job file before each bulk load.

Colxml takes only one mandatory parameter of *schema_name*.

To generate the job file for all tables:

At the Linux prompt, type: `/usr/local/Calpont/bin/colxml schema_name`

All other parameters are optional, which are described below.

-d delimiter

The field (column) delimiter of input data files that will be imported into the database. The default is '|';

For example, to change the delimiter from the default | Pipe Bar to a tilde:

```
/usr/local/Calpont/bin/colxml schema_name -d \~
```

-e number

The maximum number of error rows above which cpimport will abort the bulkload process. The default is 10. For a distributed Mode 1 or Mode 2 import, this represents the maximum number of error rows for each PM.

-h

Print help.

-j jobid

An identifier for the job description file that will get created. colxml creates a output file with the name Job_jobid.xml. The default is 299. The maximum value is 2147483647.

-x ext

Extension of the input table data files to be loaded. The default is "tbl".

-n "name"

Logical name to describe this bulk load activity.

-p path

Location where Job_jobid.xml will be created. The default is **BulkRoot/job** directory. If the specified path begins with a '/', then it is treated as an absolute path, else it is treated as a relative path that is relative to the current working directory.

-s "description"

More detailed description of the bulk load activity.

-t table name

If mentioned then colxml will generate the job xml file for only the specified tables.

-l load file

Alternate load file(s) for a table mentioned in the -t option. The first -l specifies the file(s) corresponding to the first -t; the second -l specifies the file(s) corresponding to the second -t, etc. A -l argument can specify a list of files to be imported consecutively into the relevant database table. The -l arguments may contain a fully qualified pathname (in which it will override the -f entry) or a path relative to the -f pathname entry.

-u user

The name of the user who is running colxml. This can be the user's login id.

-r number of read buffers

cpimport reads data from input data files into read buffers. This parameter affects the number of read buffers that will be used by cpimport for each table that gets loaded. The recommended number of read buffers is 1 more than the number of parse threads. This parameter can be used to tune the bulk load throughput. The default is 5.

-c read buffer size

The size or capacity of the read buffer determines how much of the data from the input files will be read at one time. The default is 1048576 bytes.

-w i/o buffer size

This buffer is used to set the size of the internal library buffer used by setvbuf call. Currently this impacts only the file handles that are used by the read threads. This can be tuned to improve the speed of the bulkload process. Default is 10485760 bytes.

-b debug level

Set the debug level of colxml process. It ranges from 1 to 3. The default value is 0, which means no debugging messages.

-E enclosedByCharacter

If the import file(s) contain any columns which are enclosed by a designated character (such as a double quote), then the -E option should be used to specify this character. The enclosing character will be stripped from the start and end of the column value prior to being loaded into the database. See Enclosed By and Escape Character Usage example below.

-C escapeCharacter or NULL escape sequence

Escape character used in conjunction with 'enclosed by' character, or as part of NULL escape sequence ('\N').

For use as the escape character: If the -E option is enabled, then the -C option specifies an escape character to be used when the enclosedByCharacter is present in a column value. An enclosedByCharacter can be escaped by preceding it with the escapeCharacter or by repeating the enclosedByCharacter itself. The default escapeCharacter is the backslash (\). See Enclosed By and Escape Character Usage example below.

For use as the NULL escape sequence: The default NULL escape sequence is '\N' and may be overridden using this option. Example: A -CZ will override the NULL interpretation from '\N' to 'ZN'.

6.6 Import data

InfiniDB uses the **cpimport** utility for the bulk loading of data into the Infinidb database. This file is located in the following directory:

```
/usr/local/Calpont/bin
```

6.6.1 Using cpimport

As mentioned before, there are now two ways to use the cpimport utility.

6.6.1.1 Using simple cpimport interface

The first usage is a simpler interface to import a single table and no XML job file (colxml) is required. The basic command for calling the cpimport utility this way is:

```
cpimport dbName tblName [loadFile]
```

In this mode, the only required inputs are the database name and table name. Again, only 1 table can be loaded by a single cpimport job with this usage of cpimport. A third optional positional parameter specifies the name of the file to be loaded. If no load file is specified, then input is assumed to be coming from stdin (See STDIN Processing below.) The loadFile argument may contain a fully qualified pathname (in which case it will override the -f entry) or a path relative to the -f pathname entry. Otherwise the loadFile path will default to the current directory.

The full command synopsis is:

```
cpimport dbName tblName [loadFile]
[-h] [-m mode] [-f filepath] [-d DebugLevel]
[-c readBufferSize] [-b numBuffers] [-r numReaders]
[-e maxErrors] [-B libBufferSize] [-s colDelimiter] [-E EnclosedByChar]
[-C escChar] [-j jobID] [-p jobFilePath] [-w numParsers]
[-n nullOption] [-P pmList] [-i] [-S] [-q batchQty] {-j jobID#}
```

6.6.1.2 Using traditional cpimport interface

The second usage is the traditional way of using cpimport in that you must first run the colxml utility to generate the XML job file. This way must be used when importing multiple tables at once but can be used in all scenarios as well. The basic command for calling the cpimport utility this way is:

```
cpimport -j jobID#
```

The jobID is used to reference a required job description file named Job_jobid.xml (in xml format) that is generated by the colxml utility. By default, cpimport searches for the job description file in the **/BulkRoot/job** directory (See **BulkRoot** definition above). The data files for the data loading, which are specified in the job description file, should typically be in the **/BulkRoot/data/import** directory unless the -f option is used. One or more tables can be loaded by a single cpimport job with this usage of cpimport.

The full command synopsis is:

```
cpimport -j jobID
[-h] [-m mode] [-f filepath] [-d DebugLevel]
[-c readBufferSize] [-b numBuffers] [-r numReaders]
[-e maxErrors] [-B libBufferSize] [-s colDelimiter] [-E enclosedByChar]
[-C escChar] [-w numParsers] [-l loadFile]
[-n nullOption] [-P pmList] [-i] [-S] [-q batchQty]
```

6.6.1.3 Optional cpimport parameters

The following optional parameters may be used. Note that some options are only applicable depending on your choice of which usage of cpimport you use.

-d debug level

Set the debug level of the cpimport process. It ranges from 1 to 3. The default value is 0, which means no debugging messages.

-f import path

Alternate import file(s) directory path. If an alternate path of "STDIN" (must be all caps) is specified, then cpimport will read the input table data from STDIN, and in this case, only 1 table can be imported by the cpimport job.

This option is not applicable with the simple cpimport usage if the loadFile parameter is omitted.

Omitting the loadFile name implies that the input is coming from STDIN, which conflicts with the use of the -f argument.

-h

Print help.

-i

Print extended information to the console for Mode 3, else extended information will only go the log file.

-j jobid

The jobid is mandatory if executing the traditional cpimport usage. The jobid is used to lookup the job description file (Job_jobid.xml) in **/BulkRoot/bulk/job**. Job_jobid.xml is created by running colxml prior to the bulk load, which is required for the traditional cpimport usage. The jobid is also used to identify the job in any system logging. If no jobid parameter is provided when calling the simple cpimport usage, a default will be provided. The maximum value is 2147483647.

-l loadFile

Alternate names for the import file(s) being loaded. The default name is *tablename.tbl*. The first -l specifies the file(s) corresponding to the first table referenced in the job description file; the second -l specifies the file(s) for the second table, etc. A -l argument can specify a list of files (separated by comma, space, or vertical bar) to be imported consecutively into the relevant database table. The -l arguments may contain a fully qualified pathname (in which case it will override the -f entry) or a path relative to the -f pathname entry.

The same option is available when running colxml to construct the job description file. When this option is provided to cpimport, it will override any previous setting in the job description file.

This option is only applicable when calling the traditional cpimport usage.

-n NULL option

Controls the handling of any data values having a string value of NULL. Valid NULL option values are 0 and 1.

- A NULL option of 0, indicates that any data values read in as the string NULL will be treated as non-null values having the string value of NULL. This is the default mode.
- A NULL option of 1, indicates that any data values read in as the string NULL will be interpreted and stored in the database as null values.

-p jobxml path

Overrides the default path for the job description file. The default is **/BulkRoot/bulk/job**. If the specified path begins with a '/', then it is treated as an absolute path, else it is treated as a relative path that is relative to the current working directory.

This option is only applicable when calling the traditional cpimport usage.

-r number of read threads

Number of read threads to spawn that will read data from input data files. This parameter can be used to tune the bulk load throughput. The default value is 1.

-s column delimiter

Specifies the character used in delimiting the fields or columns of the input data file. The default is '|'. The same option is available when running colxml to construct the job description file. When this option is provided to cpimport, it will override any previous setting in the job description file.

-S

By default, string truncation will be treated as warnings. The use of this option will change to treat string truncations as errors. This will be treated like other errors and is subject to the max errors allowed before terminating the job. They will be reported like other errors and will be found in the .err and .bad files.

-w number of parse threads

Number of parse threads to spawn. Parse threads parse data from the buffers filled up by the read threads and load it into the column files. This parameter can be used to tune the bulk load throughput. The default value is 3.

-E enclosedByCharacter

If the import file(s) contain any columns which are enclosed by a designated character (such as a double quote), then the -E option can be used to specify this character. The enclosing character will be stripped from the start and end of the column value prior to being loaded into the database. The same option is available when running colxml to construct the job description file. When this option is provided to cpimport, it will override any previous setting in the job description file.

-C escapeCharacter or NULL escape sequence

Escape character used in conjunction with 'enclosed by' character, or as part of NULL escape sequence ('\N').

For use as the escape character: If the -E option is enabled, then the -C option specifies an escape character to be used when the enclosedByCharacter is present in a column value. An enclosedByCharacter can be escaped by preceding it with the escapeCharacter or by repeating the enclosedByCharacter itself. The default escapeCharacter is the backslash (\). The same option is available when running colxml to construct the job description file. When this option is provided to cpimport, it will override any previous setting in the job description file.

For use as the NULL escape sequence: The default NULL escape sequence is '\N' and may be overridden using this option. Example: A -CZ will override the NULL interpretation from '\N' to 'ZN'.

-b number of read buffers

cpimport reads data from input data files into read buffers. This parameter affects the number of read buffers that will be used by cpimport for each table that gets loaded. The recommended number of read buffers is 1 more than the number of parse threads. This parameter can be used to tune the bulk load throughput. The default is 5.

-c read buffer size

The size or capacity of the read buffer determines how much of the data from the input files will be read at one time. The default is 1048576 bytes.

-e number

The maximum number of error rows above which cpimport will abort the bulkload process. The default is 10. For a distributed Mode 1 or Mode 2 import, this represents the maximum number of error rows for each PM.

-B i/o buffer size

This buffer is used to set the size of the internal library buffer used by setvbuf call. Currently this impacts only the file handles that are used by the read threads. This can be tuned to improve the speed of the bulkload process. Default is 10485760 bytes.

-m mode

This represents the mode of cpimport to execute:

1. Distributed import (single command, central source)
2. Distributed import (single command, distributed source)
3. Independent import (partitioned source)

For more information on these modes and their defaults, see the “Distributed Imports” section above.

-P PM list

This represents which PM (or a list of PMs separate by comma) to run the Mode 1 or Mode 2 distributed imports. If not specified, the default is all PMs.

-q batch quantity

The number of rows that cpimport will batch up and distribute at a time. It is only applicable for a Mode 1 import. Default is 10000.

6.6.2 STDIN Processing

STDIN processing can be accomplished with either usage of cpimport.

6.6.2.1 Use of simple cpimport

STDIN can be accomplished by simply not including the loadFile parameter on the cpimport command.

```
cpimport db1 table1
```

With this option, multiple sources are not available. STDIN will simply keep reading the data source until EOF is reached.

6.6.2.2 Use of traditional cpimport

STDIN can be accomplished by using the “-f” option:

-f : alternate import path for import files (cpimport)

The user may enter “-f STDIN” for the cpimport filepath if they want to redirect the input source from STDIN.

```
ex: cpimport -f STDIN -j 299
```

When redirecting input from STDIN, cpimport will ignore any file name(s) given to colxml through the “-l” command line option. With the “-l” option, a list of files to be imported at once (ex: -l "lineitem01.tbl lineitem02.tbl lineitem03.tbl"). With the STDIN option, multiple sources are not an option. STDIN will simply keep reading the data source until EOF is reached.

6.6.3 Bulk Insert with Select From Table(s)

Standard in can also be used to directly pipe the output from an arbitrary select statement into our high speed bulk loader. Here the db2.source_table is selected from, using the -N flag, to remove non-data formatting.

Use of simple cpimport

```
idbmysql -e 'select * from source_table;' -N db2 | /usr/  
local/Calpont/bin/cpimport db table1 -s '\t'
```

Use of traditional cpimport

```
idbmysql -e 'select * from source_table;' -N db2 | /usr/  
local/Calpont/bin/cpimport -j501 -s '\t' -fSTDIN
```

6.6.3.1 Bulk Insert with Select From Table on local PM

If the Local PM Query option is set up, standard in can also be used to directly pipe the output from a select returning only data from a specific PM back into the local PM. Please see “Local PM Query Examples” section in the InfiniDB Multiple UM Configuration Guide for some examples on the use of these functions.

6.6.4 Bulk Insert with Binary Source

A binary file may also be used as input to cpimport. The following should be taken into consideration when using this method to load data:

- In general, binary data is imported using fixed length records with no column or line delimiters.
- cpimport has new command line options for binary source import:

-I1 binary mode with NULLs accepted

Any numeric field having a NULL value in the input data, will be loaded as NULL (unless the corresponding column definition has a default value)

-I2 binary mode with NULLs saturated

All numeric values (including NULL values) will be saturated depending on the column data type.

- The binary format for the various data types is as follows:

Table 6 - Binary Formats

Datatypes	Description
INTEGER/INT TINYINT SMALLINT BIGINT NUMBER	All integer types are represented in little-endian format.
DOUBLE/REAL FLOAT	Represented in the IEEE format that is native to the host computer.
CHAR VARCHAR	Fields must be padded with trailing binary zeroes ('\0'), so that each string matches the width specified in the table definition. A string that is all zeros is treated as a NULL value.
DATE	<pre>struct Date { unsigned spare : 6; unsigned day : 6; unsigned month : 4; unsigned year : 16; };</pre> <p>The spare bits in the Date struct "must" be set to 0x3E for dates that are being imported.</p>
DATETIME	<pre>struct DateTime { unsigned msecond : 20; unsigned second : 6; unsigned minute : 6; unsigned hour : 6; unsigned day : 6; unsigned month : 4; unsigned year : 16; };</pre>
DECIMAL/NUMERIC	Decimal columns are imported/stored as normalized integer data. For a column defined as DECIMAL(12,2) for example, a value of 12.75 should be represented in the import data as an integer that is the binary equivalent of 1275. The user also must be aware of what size integer is used to store a decimal column.

	<p>Decimal columns are mapped to the various integer types according to the following rules:</p> <table> <tr> <th>precision or width</th><th>corresponding integer type</th></tr> <tr> <td>-----</td><td>-----</td></tr> <tr> <td><= 2</td><td>tinyint</td></tr> <tr> <td>3</td><td>smallint</td></tr> <tr> <td>4</td><td>smallint</td></tr> <tr> <td>5</td><td>int</td></tr> <tr> <td>6</td><td>int</td></tr> <tr> <td>7</td><td>int</td></tr> <tr> <td>8</td><td>int</td></tr> <tr> <td>9</td><td>int</td></tr> <tr> <td>>= 10</td><td>bigint</td></tr> </table>	precision or width	corresponding integer type	-----	-----	<= 2	tinyint	3	smallint	4	smallint	5	int	6	int	7	int	8	int	9	int	>= 10	bigint
precision or width	corresponding integer type																						
-----	-----																						
<= 2	tinyint																						
3	smallint																						
4	smallint																						
5	int																						
6	int																						
7	int																						
8	int																						
9	int																						
>= 10	bigint																						

- If the user is importing a NULL value (in binary mode -l1), then they must import the exact designated NULL value based on the data type. The NULL values for the various numeric data types are as follows:

Table 7 - Binary NULL Formats

Datatypes	SIGNED NULL VALUE	UNSIGNED NULL VALUE
BIGINT	0x8000000000000000ULL	0xFFFFFFFFFFFFFFFFEULL
INT/NUMBER	0x80000000	0xFFFFFFFFFE
SMALLINT	0x8000	0xFFFFFE
TINYINT	0x80	0xFE
DECIMAL/NUMERIC	Use appropriate integer type value based on the DECIMAL definition. See Table 1 - Binary Formats.	Use appropriate integer type value based on the DECIMAL definition. See Table 1 - Binary Formats.
FLOAT	0xFFAAAAAA	Not Applicable.
DOUBLE/REAL	0xFFFAAAAAAAAAAAAAAULL	Not Applicable.
DATE	0xFFFFFFFFFE	Not Applicable.
DATETIME	0xFFFFFFFFFFFFFFFFEULL	Not Applicable.
CHAR/VARCHAR	All binary zeroes ('\0').	Not Applicable.

- The <IgnoreField> tag is not supported in the job xml file for binary imports.
- The following command is an example simple import method load of a binary file to load data that includes NULLs:

```
[root@server1 bin]# /usr/local/Calpont/bin/cpimport -l1 mydb mytable
```

mytablesource.bin

6.6.5 Enclosed By (-E) and Escape Character (-C) Usage

The following illustrates the use of the -E and -C options for colxml and cpimport. The use of these options are used on cpimport will override the use of these options on colxml.

With a table defined with 5 string columns, the following table will be used for import:

Row number	Data
Row 1	one two three four five
Row 2	"xyz" abc def "ghi"123 jkl
Row 3	"red ""white"" blue" uno \N NULL quatro
Row 4	"red \\"white\\" blue" uno dos
Row 5	xxx "yyyy" zzz "1 2 3 4 5 abc" def
Row 6	aaa "N" ccc "NULL" eee

- Row 1 is a simple row without any "enclosed by" characters.
- Row 2 has 2 enclosed fields. Column 4 has extra trailing characters that will be ignored.
- Row 3 has an enclosed field (column 1) that contains field delimiters and "enclosed by" characters that have been escaped.
- Row 4 is similar to Row 3, but uses a "\" instead of "" to escape embedded "enclosed by" characters.
- Row 5 has a newline character in an enclosed field (in the middle of the 4 y's in column 2). Column 4 contains is an enclosed value, and the value contains field delimiters.
- Row 6 tests the handling of enclosed null values. Column 2 will be loaded as a null value, while column 4 will not.
- The following table shows the resulting data as it will be loaded into the database. In the table, null in *italics* represents a null value. The row3/ column4 value will be null or the 4-byte character string 'NULL' depending on the setting of the cpimport -N null option.

The following cpimport command will yield the results below:

```
cpimport -E\" -C\\ -n1 -j 299
```

Table 8 - Import Enclosed By Results

Row	Column 1	Column 2	Column 3	Column 4	Column 5
1	one	two	three	four	five
2	xyz	abc	def	ghi	jkl
3	red "white" blue	uno	<i>null</i>	<i>null</i>	quatro
4	red "white" blue	uno	dos	<i>null</i>	<i>null</i>

5	xxx	yy yy	zzz	1 2 3 4 5 abc	def
6	aaa	<i>null</i>	ccc	NULL	eee

6.7 Advanced Data Mapping to Table

For the advanced user, cpimport supports the loading of import files that have a different column count or column order than that defined in the database. This is only applicable for the traditional cpimport usage where the XML job file is required. Once the colxml utility has been executed, the user will need to edit the Job xml file (typically in the **/BulkRoot/job** directory) in order to include the column mapping information. cpimport can then be executed to load the data into the appropriate database columns as spelled out in the Job xml file.

The following tags are available to support this ability:

`<IgnoreField/>` - specifies a field in the import file that is to be ignored.

`<DefaultColumn colName="col"/>` - allows the user to specify a database column that is not in the import file. All rows for the specified column will be filled with a NULL value unless the column has been defined as an autoincrement column, in which case generated autoincrement values will be used.

The `<DefaultColumn>` and `<IgnoreField>` tags **must** be listed in the same order as the columns in the import file.

6.7.1 Data Mapping Example

The following example illustrates the use of these two tags in the job XML file:

Given a table with the following table description:

```
create table orders (
  orderkey int,
  custkey int,
  orderstatus char (1),
  totalprice decimal (12,2),
  orderdate date,
  orderpriority char (15),
  clerk char (15),
  shippriority int,
  comment varchar (79)
) engine=infinidb;
```

A user wants to import a file containing the following columns in the order listed below:

```
custkey
shippriority
orderstatus
totalprice
orderkey
discount
```

```

orderpriority
orderdate
ordertime

```

The following Job XML file would be used to import the file in question.

The `<IgnoreField>` tag has been inserted for columns `discount` and `ordertime` to ignore these columns in the input files. The `<Column>` tag has been replaced with the `<DefaultColumn>` tag for the `clerk` and `comment` columns to indicate these columns have no input and will be defaulted to NULL.

```

<?xml version="1.0" encoding="UTF-8"?>
<BulkJob>
    <Id>299</Id>
    <Name/>
    <Desc/>
    <CreateDate>20110413</CreateDate>
    <CreateTime>1431</CreateTime>
    <User/>
    <Delimiter>|</Delimiter>
    <ReadBuffers count="5" size="1048576"/>
    <WriteBufferSize>10485760</WriteBufferSize>
    <Schema Name="inventory">
        <Table tblName="inventory.orders"
            loadName="orders.tbl" maxErrRow="10">
            <Column colName="custkey"/>
            <Column colName="shippriority"/>
            <Column colName="orderstatus"/>
            <Column colName="totalprice"/>
            <Column colName="orderkey"/>
            <IgnoreField/>
            <Column colName="orderpriority"/>
            <Column colName="orderdate"/>
            <IgnoreField/>
            <DefaultColumn colName="clerk"/>
            <DefaultColumn colName="comment"/>
        </Table>
    </Schema>
</BulkJob>

```

6.8 Troubleshooting import

If the import terminates abnormally, an error code and a brief description are displayed. Go to the **log** directory and open the error log file in a text editor to view more detailed information.

The error log file be logged to a file named **Job_(id#).err**. Example: **Job_299.err**.

Successful imports will be logged to a file named **Job_(id #).log**. Example: **Job_299.log**.

In both files, the job ID is the same as the job ID of the XML job file.

Below are some possible errors that you might encounter.

6.8.1 Location of error log and rejected data

The .err (error log) and .bad (rejected/errored data) can be in different locations depending on the type of import executed.

- Mode 1 executed on UM: files placed on UM in same directory as source data (i.e. server_um1/import/source)
- Mode 2 executed on UM: files placed on each PM in same directory as source (i.e. server_pm1/import/source)
- Mode 3 executed on PM: files placed on that PM in same directory as source (i.e. server_pm2/import/source)
- STDIN executed on UM: files placed on UM in current working directory (i.e. server_um1/user/user1)

6.8.2 Data mismatch

When the data to be imported does not match up correctly with the table to which it is to be imported, the rows are saved into a new file labeled with the table name and .bad. For example: nation.tbl.bad.

You can review the bad data in the files, make changes to the data to be imported, and re-run the import.

You may also re-arrange the data mapping if the import file has a different order than the table definition (See Advanced Data Mapping to Table above.)

6.8.3 Number of error rows exceeds threshold

Error message “Total number of rows exceeds the maximum error number specified in the job description file”.

The maximum errors defaults to 10. Increase the maximum errors by using the -e parameter followed by the maximum number of rows. The maximum value for this is 100,000. If a value entered is greater than 100,000, cpimport will consider the maximum value to be 100,000.

1. At the Linux prompt, type `cpimport.sh -e (number of rows)` and press **Enter**.

7 Transactions

This chapter describes how transactions are handled in the InfiniDB System.

7.1 Transaction overview

InfiniDB processes Insert, Update, and Delete statements through transactions such that changes to the database are not made visible to other queries unless a commit statement has been issued.

The commit may be applied automatically or may be explicitly issued depending on client configurations external to InfiniDB. The current setting can be viewed via the following command:

```
mysql> show variables like '%autocommit%';
```

```

-----
show variables like '%autocommit%'
-----
+-----+-----+
| Variable_name | Value |
+-----+-----+
| autocommit    | OFF   |
+-----+-----+
1 row in set (0.00 sec)

```

The current setting can be changed by one of the following commands:

```
mysql> set autocommit = on;
```

```
mysql> set autocommit = off;
```

7.2 ACID Compliance

Database systems deliver ACID behavior:

- **Atomicity:** Either all the tasks in a transaction must be done, or none of them. The transaction must be completed, or else it must be undone (rolled back).
- **Consistency:** Every transaction must preserve the integrity constraints - the declared consistency rules - of the database. It cannot place the data in a contradictory state.
- **Isolation:** Two simultaneous transactions cannot interfere with one another. Intermediate results within a transaction are not visible to other transactions.
- **Durability:** Completed transactions cannot be aborted later or their results discarded. They must persist through (for instance) restarts of the DBMS after crashes.

<http://en.wikipedia.org/wiki/Database>

The key to support ACID behavior related to Atomicity and Isolation will be accomplished for the InfiniDB Engine a custom implementation of concurrency control to support consistent read behavior in support of Isolation and rollback in support of Atomicity. InfiniDB provides the most commonly use Isolation level of Read Committed which ensures that any query is consistent as of the start of the query and won't see any partially entered data that is added after the query starts.

The most important benefit of InfiniDB's versioning is that reads are never blocked. That is, one session can read (an older) versioned block while another session is simultaneously updating that block.

Both Rollback (Atomicity) and Consistent Read (Isolation) support are implemented by retaining prior versions of data.

Datatype consistency is supported through the DDL and DML. Constraints are not supported.

Durability is supported through the transaction log and checkpointing (saves to disk at appropriate points).

8 Performing Backup and Recovery

This chapter explains how to perform a full InfiniDB data backup and how to perform a recovery in the event of a disk outage or other catastrophe. It explains how to recover data to the point of the last full backup or perform a point-in-time recovery.

This chapter will also describe dropping the InfiniDB database to start from the beginning.

8.1 Backing up data

To create a full data backup, you must back up the front-end database and the InfiniDB database from the same point in time.

To back up your data, perform the following procedures which are explained in more detail later in this chapter:

- **Suspend database writes**
This process suspends database writes to ensure data consistency during the backup process, but allows users to continue to query the database.
- **Back up front-end database**
This process backs up the users, and their database objects that connect to the InfiniDB Platform.
- **Back up InfiniDB Platform database files**
This process backs the InfiniDB Platform database files.
- **Resume database writes**
This process is done after front-end database and InfiniDB backups are complete and allows users to write to the database.

8.1.1 Suspending database writes

You must suspend database writes when backing up your data to ensure data consistency.

NOTE: If a user attempts to write to the database during a backup, InfiniDB generates an error message, but the backup process is not affected. InfiniDB allows queries during the backup process.

To suspend database writes:

1. Connect to User Module 1, or your OAM Module using a SSH client.
2. Type your Linux *user_name* and *password*.
A Linux prompt appears.
3. [root@server]# **cmconsole**
The InfiniDB Console prompt appears.
4. calpont>**suspendDatabaseWrites**

A confirmation message appears.

8.1.2 Backing up front-end database

You should back up your front-end database using the procedures and instructions for that vendor.

InfiniDB doesn't implement anything unique into the MySQL front-end and the following directory may be backed up for each User Module in lieu of previously established procedures:

/usr/local/Calpont/mysql/db

Example:

```
cp -rp /usr/local/Calpont/mysql/db /mnt/InfiniDB/backup/frontend
```

Note: The -rp options are for copying directories recursively and saving ownership information.

If selected databases are to be backed up only, then the database directories within the above directory may be backed up instead.

8.1.3 Backing up InfiniDB database

This process creates a full InfiniDB database backup by copying all the DBRM and DB (data) files. Because you can have multiple files being copied from multiple directories, you should label your backups so that you can restore them to the proper directory in the event of a failure.

Copy all files from each **Performance Module**.

To backup the InfiniDB database:

- Copy the entire directory structure of each of your DBRoots to backup storage. These directories are listed in the **/usr/local/Calpont/etc/Calpont.xml** file in the SystemConfig section. Usually they will look like **/usr/local/Calpont/datan** where n is a number starting at 1. Depending on your configuration you may also have important system data in a directory called simply **/usr/local/Calpont/ data**. You should also backup this directory if it contains any important files for your installation (e.g. bulk load scripts).
- Copying the **/usr/local/Calpont/datan** directories will include copying the DBRM files (**/usr/local/Calpont/data1/systemFiles/dbrm**).

8.1.4 Resuming database writes

When you have finished backing up your MySQL database and the InfiniDB database, resume database writes.

1. Connect to User Module 1, or your OAM Module using a SSH client.
2. Type your Linux **user_name** and **password**.

A Linux prompt appears.

3. [root@server]# **cmconsole**

The InfiniDB Console prompt appears.

```
4. calpont>resumeDatabaseWrites
```

A confirmation message appears.

8.2 Recovering from a backup

If you have a hardware failure and need to recover from a backup, perform the following procedures which are described in more detail later in this chapter:

- Restore front-end database
- Restore InfiniDB database

8.2.1 Restoring front-end database

Restore your front-end database using the recommended procedures.

InfiniDB doesn't implement anything unique into the MySQL front-end and the following directory may be restored for each User Module in lieu of previously established procedures:

/usr/local/Calpont/mysql/db

Example to remove and restore:

```
rm -rf /usr/local/Calpont/mysql/db/*
```

```
cp -rp /mnt/InfiniDB/backup/frontend/* /usr/local/Calpont/mysql/db/
```

Note 1: The -rf options are for removing directories recursively and not prompting for removal. The -rp options are for copying directories recursively and saving ownership information.

Note 2: After the copy has completed, ensure that the /usr/local/Calpont/mysql/db directory and sub-directories have mysql as the owner:

```
# chown -R mysql:mysql /usr/local/Calpont/mysql/db
```

If selected databases are to be restored only, then the database directories within the above directory may be restored instead.

8.2.2 Restoring InfiniDB database

To restore the InfiniDB database, you must copy the following files and directories from the backup directory to each Performance Module.

Note: If performing anything other than a point in time restore, you should copy off (See Saving InfiniDB Transaction Logs below) the transaction logs before performing this step.

To restore the InfiniDB database:

1. Ensure the InfiniDB system is **not** active.
2. Copy the data files and directories back to their original location.

/usr/local/Calpont/datan where *n* represents unique directories such as data1, data2, etc. Copy all the files and directories in each numbered data directory.

- Restoring the **/usr/local/Calpont/datan** directories will include copying the DBRM files (**/usr/local/Calpont/data1/systemFiles/dbrm**).

Example to remove and restore:

```
rm -rf /usr/local/Calpont/datan/*  
where n is a number starting at 1
```

```
cp -rp /mnt/InfiniDB/backup/backend/datan/*  
/usr/local/Calpont/datan  
where n is a number starting at 1
```

Note 1: The -rf options are for removing directories recursively and not prompting for removal. The -rp options are for copying directories recursively and saving ownership information.

8.3 Dropping the InfiniDB database

If you need/want to go back to "square one", that is, restore the system to the same condition as when you first installed InfiniDB, the following steps should be followed:

1. In MySQL identify all the InfiniDB tables. For each of these tables: 'drop <table> restrict;'.
Note: The 'restrict' option will speed up the drop table, but should only be used in anticipation of deleting the InfiniDB database.

2. Use calpontConsole to stopSystem.
3. Identify each of your DBRoots, possibly by going through the Calpont.xml file. For each of these roots, 'rm -rf <root>/000.dir'.
4. Locate the DBRM persistence files, usually in /mnt/OAM/dbrm and remove all of them.
5. Erase the InfiniDB shared memory, '/usr/local/Calpont/bin/clearShm'.
6. Use calpontConsole to startSystem.

When the system is fully up, at the Linux prompt build a new InfiniDB System Catalog:
'/usr/local/Calpont/bin/dbbuilder 7'.

The system is now ready for the creation of tables.

9 Performance Tuning

This chapter describes the methods of tuning InfiniDB performance.

9.1 Query Tuning Guidelines - Data Access Operations

InfiniDB does not use traditional index or table scan operations. Neither operation exists within the InfiniDB system.

9.1.1 Operations used by InfiniDB include:

- Full Column Scan - an operation that scans each entry in a column using all available threads on the Performance Modules. Speed of operation is generally related to the size of the data type and the total number of rows in the column. The closest analogy for a traditional system is an index scan operation.
- Partitioned Column Scan - an operation that uses the InfiniDB Extent Map to identify that certain portions of the column do not contain any matching values for a given set of filters. The closest analogy for a traditional row based dbms is a partitioned index scan, or partitioned table scan operation.
- Column lookup by row offset - once the set of matching filters have been applied and the minimal set of rows have been identified, additional blocks are requested using a calculation that determines exactly which block is required. The closest analogy for a traditional system is a lookup by rowid.

These operations are automatically executed together in order to execute appropriate filters and column lookup by row offset. This set of operations makes up the work issued from the User Module to the set of Performance Modules in support of a given step in a query.

9.2 Gathering Performance Metrics - *calgetstats()*

The *calgetstats()* function is used to provide information on the last query executed within a given session.

```
select calgetstats();
```

```
mysql> select count(*) from part;
+-----+
| count(*) |
+-----+
| 200000000 |
+-----+
1 row in set (0.48 sec)
```

```
mysql> select calgetstats();
+-----+
| calgestats() |
+-----+
| Query Stats: MaxMemPct-0; NumTempFiles-0; TempFileSpace-0MB; PhyI/O-0; CacheI/O-98039; |
+-----+
```

. . . continued:

```
-----+
BlocksTouched-97658; CasPartBlks-0; MsgBytesIn-2MB; MsgBytesOut-0MB| 1242146662
640516 |
-----+
```

1 row in set (0.00 sec)

The output contains information on:

- MaxMemPct - Peak memory utilization on the User Module, likely in support of a large (User Module) based hash join operation.
- NumTempFiles - Report on any temporary files created in support of query operations larger than available memory, typically for unusual join operations where the smaller table join cardinality exceeds some configurable threshold.
- TempFileSpace - Report on space used by temporary files created in support of query operations larger than available memory, typically for unusual join operations where the smaller table join cardinality exceeds some configurable threshold.
- Phyl/O - Number of 8k blocks read from disk, SSD, or other persistent storage. In most cases, the quantity of individual I/O operations are far fewer than the number of blocks; InfiniDB will read as many as 512 blocks together in one I/O operation.
- CacheI/O - Approximate number of 8k blocks processed in memory, adjusted down by the number of discrete Phyl/O calls required.
- BlocksTouched - Approximate number of 8k blocks processed in memory.
- CalPartBlks - The number of block touches eliminated via the InfiniDB Extent Map elimination behavior.
- MsgBytesIn, MsgByteOut - Message bytes in MB sent between nodes in support of the query.

9.3 Gathering Performance Metrics

9.3.1 Using /var/log/Calpont/debug.log

At the default trace level, InfiniDB records each statement being executed within the debug.log along with the start time and end time of the query. This information is stored in the /var/log/Calpont/debug.log file on the User Module executing the query.

Additional information can be emitted with calsettrace();

9.3.2 Using calsettrace(1);

The calsettrace(1) function enables more detailed information to be stored in the /var/log/Calpont/debug.log on the User module executing the statement.

Note: The increased logging with calsettrace(1) will have a performance impact.

The argument (1) enables output of this file, while (0) will turn off creation of the file. Enabling or disabling the creation of this file is at the session level. Other trace levels should only be set under instruction from InfiniDB support.

```
select calsettrace(1);
```

Note that setting `calsettrace(1);` will cause each statement to have warning information available. A sample warning might be:

```
1 row in set, 1 warning (0.06 sec)
```

9.3.3 Using `/var/log/mysqld.log`

Additional logging information related to the interface between MySQL and InfiniDB is emitted in the `/var/log/mysqld.log` file on the User Module executing the statement. The information presented here does not present a full picture of operations executed in support of a query but may be used under instruction from InfiniDB support.

9.4 Understanding Extent Map Range Partitioning

Extents are defined as 8 million continuous rows. The Extent Map stores minimum and maximum values within each Extent. For data that is loaded as it is created (real-time or near real-time), unique identifier fields and date fields generally follow ascending patterns that allow different min and max values for each extent that allows for elimination of significant blocks.

The Extent Map has the following characteristics:

1. All column files for an extent are created on the same mount point.
2. The extent is dynamically mapped to a given Performance Module at run-time.
3. The minimum value and maximum value for each column in an extent is recorded as a result of scanning the column. Scanning an extent is typically a sub-second operation, with the overhead of persisting the min and max values is less than 2%.
4. The minimum and maximum values are cleared from an extent if the data within that extent is modified, but would be re-populated by the next scan of an extent.

9.5 Scaling

9.5.1 For performance

When query response times perform slowly due to the amount and complexity of queries, you add Performance Modules to the system.

9.5.2 For concurrency

As the number of InfiniDB system users increases, you can add User Modules to maintain performance. The number of User Modules is based on the number of users and the size of intermediate and final result sets.

9.5.3 For data capacity

As data is added to the database, the system can require additional disk arrays. As disk arrays are added, the cache ratio changes and additional Performance Modules are required to efficiently access data. In some cases, additional disk controllers must be added. The total of arrays needed is highly dependent on a customer's clustering of data and the types of queries. The arrays added is probably consistent with the initial ratio.

10 Configuring Modules

This chapter describes how to add and remove InfiniDB modules. The primary reason for adding modules is to improve performance or add redundancy.

Additionally, you may want to remove modules not in use or take modules offline that need hardware or software updates.

You can configure modules using the InfiniDB Console commands. This chapter provides the instructions for both.

10.1 Before configuring modules

Before configuring modules, it is highly recommended that you shut down the database and backup up the InfiniDB System and the database. See “Performing Backup and Recovery” for instructions.

Ensure you have the root password for all modules that will be added, removed or reconfigured.

NOTE: After adding modules, update the NMS system so that new modules are monitored as part of the system.

While removing modules, the NMS may issue an alert that servers are no longer functioning on the network.

10.1.1 Module IDs

Modules in the InfiniDB System are identified by the following 2-character module types followed by a unique number.

- **umnn**— User Module
- **pmnn**— Performance Module

For example, the InfiniDB System defaults the first User Module to um1. When another User Module is added, it defaults to um2, and so forth.

InfiniDB always picks the first available module ID in the series. For example, if you have pm1, pm2, pm4 and you select to add a new Performance Module, InfiniDB names the new module pm3.

10.1.2 Automatic configuration

Automatic module configuration is less labor intensive when you accept the InfiniDB defaults. It allows you to configure more modules with fewer commands. Automatic configuration does the following:

- Creates module IDs
- Adds multiple modules of the same type with one command
- Removes multiple modules of the same type with one command

10.1.3 Manual configuration

You have the option of manually configuring modules; however, it requires entering more data with additional commands. Manual configuration allows you to do the following:

- Create your own numbering scheme adhering to the module identification format previously described in this chapter
- Add specific modules one at a time
- Remove specific modules one at a time

10.2 Adding DBRoots

To add dbroots (storage files) into your system, it requires a 2 phase approach: creating the physical dbroots and assigning them to a Performance Module.

10.2.1 Creating the physical dbroot

The OAM `addDbRoot` command is used to create the physical storage (dbroot):

```
addDbroot numRoots
```

where *numRoots* is the total number of new dbroots to be added. The command will return the dbroot id's created.

Example to add 2 additional dbroots to an already existing 2 dbroot system:

```
InfiniDB> adddbroot 2
adddbroot   Mon Aug 26 15:00:38 2013

New DBRoot IDs added = 3, 4
```

After the `addDbroot` command, dbroots 3 and 4 have been created. You can see that they have been created by using the `getSystemConfig` command. Along with other information, towards the bottom of the output you will see information that reflects the additional, unassigned dbroots.

```
InfiniDB> getsystemconfig
:
System Assigned DBRoot Count = 2
DBRoot IDs assigned to 'pm1' = 1
DBRoot IDs assigned to 'pm2' = 2

DBRoot IDs unassigned = 3, 4
:
```

10.2.2 Assign DBRoots to Performance Module

The `addDbRoot` command added the physical dbroots, but they must be assigned to a Performance Module before they can be used by using the `assignDbrootPmConfig` command.

Note: The system must be in a STOPPED state when this command is used. Use the `stopSystem` command. Do not attempt when the system is running or when the system has been ShutDown.

```
assignDbrootPmConfig dbrootid perfmod
```

where

dbrootid is the unassigned dbroot(s) to be assigned. You can assign multiple dbroots to a single Performance module with a comma separated list.

perfmod is the Performance Module being assigned the dbroots.

Example #1 of assigning the 2 new dbroots to 2 separate PMs:

```
InfiniDB> assignPmDBRootConfig 3 pm1
assigndbrootpmconfig Tue Aug 26 15:03:15 2013

DBRoot IDs assigned to 'pm1' = 1

Changes being applied

DBRoot IDs assigned to 'pm1' = 1, 3

Successfully Assigned DBRoots

REMINDER: Update the /etc/fstab on pm1 to include these dbroot mounts

InfiniDB> assignPmDBRootConfig 4 pm2
assigndbrootpmconfig Tue Aug 26 15:07:20 2013

DBRoot IDs assigned to 'pm2' = 2

Changes being applied

DBRoot IDs assigned to 'pm2' = 2, 4

Successfully Assigned DBRoots

REMINDER: Update the /etc/fstab on pm2 to include these dbroot mounts
```

Example #2 of assigning the 2 new dbroots to 1 PM:

```
InfiniDB> assignPmDBRootConfig 3,4 pm2
assigndbrootpmconfig Tue Aug 26 15:17:46 2013

DBRoot IDs assigned to 'pm2' = 2

Changes being applied

DBRoot IDs assigned to 'pm2' = 2, 3, 4

Successfully Assigned DBRoots

REMINDER: Update the /etc/fstab on pm2 to include these dbroot mounts
```

Once completed, start the system back up with the `startsystem` command.

10.3 Moving DBRoots

To move a dbroot from 1 module to another, you must first perform a stop system. Dbroots cannot be moved while the system is active.

Note: If your storage is configured as internal and once this command has been executed, you must move the data to the appropriate PMs (i.e., SCP) before the system is restarted.

To move dbroots:

At the Linux prompt, type **cmconsole**.

Type **movePmDbrootConfig** [fromPM] [DBRoot] [toPM] and press **Enter**.

Examples:

```
InfiniDB> movePmDbrootConfig pm6 6 pm5
movepmdbrootconfig    Wed Mar 28 10:44:24 2012
```

```
DBRoot IDs currently assigned to 'pm6' = 6
DBRoot IDs currently assigned to 'pm5' = 5
```

```
DBRoot IDs being moved, please wait...
```

```
DBRoot IDs newly assigned to 'pm6' =
DBRoot IDs newly assigned to 'pm5' = 5, 6
```

```
InfiniDB> movePmDbrootConfig pm5 6 pm6
movepmdbrootconfig    Wed Mar 28 10:44:33 2012
```

```
DBRoot IDs currently assigned to 'pm5' = 5, 6
DBRoot IDs currently assigned to 'pm6' =
```

```
DBRoot IDs being moved, please wait...
```

```
DBRoot IDs newly assigned to 'pm5' = 5
DBRoot IDs newly assigned to 'pm6' = 6
```

10.4 Adding modules

Adding Performance or User Modules effectively adds more processing power to the InfiniDB System. See “Performance Tuning” to learn more about which types of modules you can add to increase the performance or concurrency of your system. You can add User Modules for redundancy.

WARNING: Modules can be added with the system active or offline. Once added, they can be placed in-service with the alterSystem-Enable command.

When InfiniDB adds multiple modules of the same type with one command, it defaults the module IDs. To add modules using your own modules IDs, you must add modules one at a time.

NOTE: When adding Performance Modules with multiple NICs, you must add the host name for all NICs. If you do not, the add module process will fail with invalid parameters.

To add a module to InfiniDB System through the CalpontConsole:

1. Log on to the InfiniDB Console.
2. Do one of the following:
 - To accept default module IDs and add multiple modules, type **addModule module_type number_of_modules IP_address_or_host_name (separated by commas) root_password**
For example, to add two Performance Modules with host names SRVCPT1 and SRVCPT2, type the following:
addModule pm 2 SRVCPT1,SRVCPT2 mypwd
 - To create IDs manually one at a time, type
addModule module_ID IP_address_or_host_name root_password
For example to add one User Module number 1285 with host name SRVCPT1, type the following:
addModule UM1285 SRVCPT1 mypwd

The Calpont.xml is updated to add new modules and the appropriate files are installed to the new modules.

If the module addition fails, the InfiniDB Console displays an error message. Additional details are located in the InfiniDB Log Files on the Performance Module #1.

To add a module to the InfiniDB System on an Amazon EC2 system, perform one of the following:

- To accept default module IDs, add multiple modules and have it automatically create the Instances:
addModule module_type number_of_modules
An example to add two Performance Modules with defaulted instance names, type the following:
addModule pm 2
- To accept default module IDs, add multiple modules and have it install on existing Instances:
addModule module_type number_of_modules instance-ids
An example to add two Performance Modules with instance names id-1234567 and id-9876543, type the following:
addModule pm 2 id-1234567,id-9876543
- To create IDs manually one at a time and have it automatically create the Instance
addModule module_ID

For example to add one Performance Module with a default instance name, type the following:

addModule pm2

- To create IDs manually one at a time and have it install on existing Instance

addModule module_ID instance-i

For example to add one User Module number 2 with instance name id-1111111,
type the following:

```
addModule um2 id-1111111
```

10.5 Removing modules

You can remove modules from the system when they are no longer needed or in the event that they need to be taken offline for hardware updates. You can remove a module if it is disabled (see “Disabling and Enabling System Modules”) or if the system is stopped (see “Stopping the system”).

You can quickly remove multiple modules from the system by accepting default module IDs. This removes the last modules added to the system.

You can also choose to remove specific modules individually by module ID.

NOTE: You cannot remove the last **um** or **pm** module.

To remove a module using the InfiniDB Console:

1. Log on to the InfiniDB Console. See “Logging on and off the InfiniDB Console”.
2. Do one of the following:

- To remove the last modules added to InfiniDB, type

removeModule module_type number_of_modules

For example, to remove two Performance Modules, type the following:

RemoveModule pm 2

- To remove a specific module, type

removeModule module_ID

For example to remove one User Module with module ID UM1285, type the following:

removeModule um1285

11 Granting Access for Users

InfiniDB allows permissions to be set for user accounts. The syntax of these grants follows the standard MySQL syntax. InfiniDB comes with the full privileges for user *root*. But in order to set/restrict user accounts, privileges must be given/ restricted.

InfiniDB uses a dedicated schema called *infinidb_vtable* for creation of all temporary tables used for InfiniDB processing. The *root* user account has been given permission to this account by default, but full permission MUST be given for all user accounts to this schema:

```
grant ALL on infinidb_vtable.* to user_account;  
  
where user_account = user login, server and password characteristics
```

Further permissions/restrictions can now be placed on any existing objects (tables, functions, procedures, views) for any access/limitations wanting to be placed on users:

Example to give a user that has a password full access to all tables for a database (after the above grant has been given):

```
use mysql;  
grant ALL on my_schema.* to 'someuser'@'somehost'  
    identified by 'somepassword';  
flush privileges;
```

Example to give a user that has a password read only access to only 1 table (after the above grant has been given):

```
use mysql;  
grant SELECT on my_schema.table1 to 'someuser'@'somehost'  
    identified by 'somepassword';  
flush privileges;
```

12 Cross-Engine Table Access

InfiniDB tables may be joined with non-InfiniDB tables (i.e., MySQL tables) within a query. To enable this process, a JDBC connection must be established by defining the connection information in the Calpont.XML configuration file.

The connection string used to establish a JDBC connection needs a user name and password. The user name can be retrieved from a SQL session, but the password cannot be retrieved from MySQL.

The following is an example entry in the Calpont.XML configuration file to gain access to joined tables residing in native MySQL on the same server as InfiniDB:

```
<CrossEngineSupport>
  <Host>127.0.0.1</Host>
  <Port>3306</Port>
  <User>root</User>
  <Password>pwd</Password>
</CrossEngineSupport>
```

See “Granting Access for Users” for more information on access and permissions.

13 User Resource Allocation

InfiniDB supports the ability to give priority to resources allocated (CPU) based on a user.

Users that are designated with minimum CPU resources get first preference in an order that can be established. Effectively a particular user can be guaranteed a set amount of resources. E.g:

- User 1 gets a minimum of 40% CPU Resources
- User 2 gets a minimum of 30% CPU Resources
- If any user logs in for a query while User 1 and User 2 are running queries, these new users (i.e., User 3,4 and 5) get only the remaining 30% of the CPU Resources."

13.1 User Priority Management

Three stored procedures were created in the `infinidb_querystats` schema for the user to set, remove and view user priorities. The priority table associates a user with a priority level. A user that does not have an entry is given the low priority level by default.

`CalSetUserPriority (host varchar, user varchar, priority varchar)`

- Assigns a priority level to user@host.
- Priority is case insensitive 'high', 'medium' or 'low'.
- Host and user will be validated to exist in MySQL

`CalRemoveUserPriority(host varchar, user varchar)`

- Removes the user entry, effectively restoring the default of 'low'.
- User existence will not be validated.

`CalShowProcessList()`

- Prints a combination of mysql 'show processlist' and user priority

The MySQL user needs to be granted the execute privileges for these procedures and the select privileges for the tables in the `infinidb_querystats` schema. Or, chances are, the following should just work for a super user:

```
GRANT ALL ON infinidb_querystats.* TO 'user'@'host';  
// user will now have the privilege to use the priority procedures and view query stats.
```

13.2 Enabling User Priorities

To enable this feature, the `<UserPriority><Enabled>` element in the InfiniDB configuration file (`/usr/local/Calpont/etc/Calpont.xml`) should be set to Y (default is N).

`<UserPriority>`


```
<Enabled>N</Enabled>  
</UserPriority>
```

Cross Engine Support must also be enabled. See the "Cross-Engine Table Access" section in this guide.

13.3 User Priority Processing

The PrimProc process has one job queue for each priority level and thread assigned to each queue. The number of threads assigned to each queue is configurable using the following elements in the InfiniDB configuration file:

```
<PrimitiveServer><HighPriorityPercentage>  
<PrimitiveServer><MediumPriorityPercentage>  
<PrimitiveServer><LowPriorityPercentage>
```

The defaults are 60, 30, and 10 respectively. Each queue is given at least 1 thread so there is neither 'idle' priority configuration possible nor starvation. The number of threads started is normalized such that 100% = 2 * (the number of cores on the machine). The user can overbook or underbook their CPUs however they want.

This is an example of how threads are assigned on an 8-core system using the defaults.

- 10% of 16 = 1.6, rounds down to 1 thread for the low priority queue.
- 30% of 16 = 4.8, rounds down to 4 threads for the medium priority queue.
- The high priority queue gets the remaining 11 threads.

Technically, each thread is given a preferred queue to get work from. If a thread's preferred queue is empty, it will choose jobs from the high, then medium, then low priority queues. If there are only low priority jobs running, on an 8-core system all 16 threads will process jobs from the low priority queue. If a medium priority query starts, using the defaults, the 15 threads assigned to the high and medium queues will process the medium queue, leaving the 1 assigned to the low queue to process the low priority jobs. Then, if a high priority query starts, the 11 threads assigned to the high priority queue will begin processing the high priority jobs, the 4 assigned to the medium queue will process those jobs, and the 1 assigned to the low queue will process those jobs.

Given this algorithm, the configuration parameters could be thought of as minimum levels for each priority.

Note that this implementation only affects the processing done by PrimProc. Depending on the work distribution of a given query, a user may or may not observe overall performance proportional to their priority level.

14 Query Performance Monitoring

One task of database administrators is to monitor the query performance and accurately pinpointing SQL code that is causing a slowdown. To achieve that, statistics data about queries can be retrieved and presented for the DBA to analyze.

A tool has been provided to look at the history of queries across all sessions along with execution time, and those stats provided by `calgetstats()` today for each query. Only queries in the following InfiniDB syntax are available for statistics monitoring:

- SELECT
- INSERT
- UPDATE
- DELETE
- INSERT SELECT
- LOAD DATA INFILE

14.1 Enablement of Query Statistics Collection

To enable the collection of query stats, the `<QueryStats><Enabled>` element in the `Calpont.XML` configuration file should be set to Y (default is N).

```
<QueryStats>
<Enabled>Y</Enabled>
</QueryStats>
```

Cross Engine Support must also be enabled. See the “Cross-Engine Table Access” section in this guide.

14.2 Query Statistics table

The following table in the `infinidb_querystats` schema contains the stats information once the `QueryStats` element is turned on:

```
CREATE TABLE IF NOT EXISTS querystats
(`queryID` bigint NOT NULL AUTO_INCREMENT,
 `sessionID` bigint DEFAULT NULL,
 `host` varchar(50),
 `user` varchar(50),
 `priority` char(20),
 `queryType` char(25),
 `query` varchar(8000),
 `startTime` timestamp NOT NULL,
 `endTime` timestamp NOT NULL,
 `rows` bigint,
 `errno` int,
```

```

`phyIO` bigint,
`cacheIO` bigint,
`blocksTouched` bigint,
`CPBlocksSkipped` bigint,
`msgInUM` bigint,
`msgOutUm` bigint,
`maxMemPct` int,
`blocksChanged` bigint,
`numTempFiles` bigint,
`tempFileSpace` bigint,
PRIMARY KEY (`queryID`)
);

```

NOTE: UTF-8 must be declared for this table if the instance has been set up with a UTF-8 profile.

14.3 Column Definitions

- Session ID (sessionID)
The session number that executed the statement.
- Host (host)
The host that executed the statement.
- User ID (user)
The user that executed the statement.
- Priority (priority)
The priority the user has for this statement.
- Query Execution Times (startTime, endTime)
Calculated as end time – start time.
start time: the time that the query gets to ExeMgr, DDLProc, or DMLProc
end time: the time that the last result packet exits ExeMgr, DDLProc or DMLProc
- Rows returned or affected (rows)
The number of rows returned for SELECT queries, or the number of rows affected by DML queries.
Not valid for DDL and other query types.
- Error Number (errNo)
The IDB error number if this query failed, 0 if succeed.
- Physical I/O (phyIO)
The number of blocks that the query accessed from the disk, including the pre-fetch blocks. This statistic is only valid for the queries that processed by ExeMgr; i.e., SELECT, DML with where clause, and INSERT SELECT.
- Cache I/O (cacheIO)
The number of blocks that the query accessed from the cache. This statistic is only valid for queries that processed by ExeMgr; i.e., SELECT, DML with where clause, and INSERT SELECT.

- **Blocks Touched (blocksTouched)**
The total number of blocks that the query accessed physically and from the cache. This should be equal or less than the sum of physical I/O and cache I/O. This statistic is only valid for queries that processed by ExeMgr; i.e., SELECT, DML with where clause, and INSERT SELECT.
- **Partition Blocks Eliminated (CPBlocksSkipped)**
The number of blocks being eliminated by the extent map casual partition. This stats is only valid for queries that processed by ExeMgr; i.e., SELECT, DML with where clause, and INSERT SELECT.
- **Messages from UM to PM (msgOutUM)**
The number of messages in bytes that ExeMgr sends to the PrimProc. If a message needs to be distributed to all the PMs, the sum of all the distributed messages will be counted. This stats is only valid for queries that processed by ExeMgr; i.e., SELECT, DML with where clause, and INSERT SELECT.
- **Messages from PM to UM (msgInUM)**
The number of messages in bytes that PrimProc sends to the ExeMgr. This stats is only valid for queries that processed by ExeMgr; i.e., SELECT, DML with where clause, and INSERT SELECT.
- **Memory Utilization (maxMemPct)**
This field shows memory utilization for the User Module (UM) in support of any UM join, group by, aggregation, distinct, or other operation.
- **Blocks Changed (blocksChanged)**
Total number of blocks that query changed physically on disk. This is only for delete/update statements.
- **Temp Files (numTempFiles)**
This field shows any temporary file utilization for the User Module (UM) in support of any UM join, group by, aggregation, distinct, or other operation.
- **Temp File Space (tempFileSpace)**
This shows the size of any temporary file utilization for the User Module (UM) in support of any UM join, group by, aggregation, distinct, or other operation.

14.4 Query Statistics Viewing

Users can view the query statistics by selecting the rows from the query stats table in the *infinidb_querystats* schema. Examples listed below:

Example 1:

List execution time, rows returned for all the select queries within the past 12 hours:

```
select queryid, query, endtime-starttime, rows from querystats where  
starttime >= now() - interval 12 hour and querytype = 'SELECT';
```

Example 2:

List the three slowest running select queries of session 2 within the past 12 hours.

```
select a.* from (select endtime-starttime execTime, query from
queryStats where sessionid = 2 and querytype = 'SELECT' and starttime
>= now()-interval 12 hour order by 1 limit 3) a;
```

Example 3:

List the average, min and max running time of all the INSERT SELECT queries within the past 12 hours:

```
select min(endtime-starttime), max(endtime-starttime), avg(endtime-
starttime) from querystats where querytype='INSERT SELECT' and
starttime >= now() - interval 12 hour;
```

15 Operating Mode

InfiniDB has the ability to support full MySQL query syntax through an operating mode. This operating mode may be set as a default for the instance or set at the session level.

Please refer to the InfiniDB SQL Syntax Guide for setting the session level operating mode.

To set the operating mode at the instance level, specify `infinidb_vtable_mode` in the `my.cnf` configuration file. If not specified, the default is 1 (distributed mode).

In `my.cnf` file (`/usr/local/Calpont/mysql`):

```
[mysqld]
  infinidb_vtable_mode=value
```

where *value* is:

- 0) a generic, highly compatible row-by-row processing mode. Some WHERE clause components can be processed by InfiniDB, but joins are processed entirely by `mysqld` using a nested-loop join mechanism
- 1) (the default) query syntax is evaluated by InfiniDB for compatibility with distributed execution and incompatible queries are rejected. Queries executed in this mode take advantage of distributed execution and typically result in higher performance.
- 2) auto-switch mode: InfiniDB will attempt to process the query internally, if it cannot, it will automatically switch the query to run in row-by-row mode.

NOTE: For more information on supported InfiniDB query syntax for mode 1, please refer to the InfiniDB SQL Syntax Guide.

16 Decimal to Double Math

InfiniDB has the ability to change intermediate decimal mathematical results from decimal type to double. The decimal type has approximately 17-18 digits of precision, but a smaller maximum range whereas the double type has approximately 15-16 digits of precision, but a much larger maximum range (refer to the InfiniDB SQL Syntax Guide Datatype section for details). Therefore, the proper setting depends on the intended usage. In typical mathematical and scientific applications, the ability to avoid overflow in intermediate results with double math is likely more beneficial than the additional two digits of precision. In financial applications, however, it may be more appropriate to leave in the default decimal setting to ensure accuracy to the least significant digit.

The `infinidb_double_for_decimal_math` variable is used by InfiniDB to control the data type for intermediate decimal results. This decimal for double math may be set as a default for the instance, set at the session level, or at the statement level by toggling this variable on and off.

Please refer to the InfiniDB SQL Syntax Guide for setting at the session level.

16.1 Enable/Disable Decimal to Double Math

To enable/disable the use of the decimal to double math at the instance level, specify `infinidb_double_for_decimal_math` in the `my.cnf` configuration file. The default is 0 (disabled).

In `my.cnf` file (`/usr/local/Calpont/mysql`):

```
[mysqld]
xxxxx
infinidb_double_for_decimal_math=value
```

where *value* is 0 (disabled) or 1 (enabled).

17 Decimal Scale

InfiniDB has the ability to support varied internal precision on decimal calculations.

`infinidb_decimal_scale` is used internally by the InfiniDB engine to control how many significant digits to the right of the decimal point are carried through in suboperations on calculated columns. If, while running a query, you receive the message 'aggregate overflow', try reducing `infinidb_decimal_scale` and running the query again. Note that, as you decrease `infinidb_decimal_scale`, you may see reduced accuracy in the least significant digit(s) of a returned calculated column.

This decimal scale may be set as a default for the instance or set at the session level.

Please refer to the InfiniDB SQL Syntax Guide for setting the session level decimal scale.

17.1 Enable/Disable Decimal Scale

To enable/disable the use of the decimal scale at the instance level, specify `infinidb_use_decimal_scale` in the `my.cnf` configuration file. The default is 0 (disabled).

In `my.cnf` file (`/usr/local/Calpont/mysql`):

```
[mysqld]
xxxxx
infinidb_use_decimal_scale=value
```

where *value* is 0 (disabled) or 1 (enabled).

17.2 Set Decimal Scale Level

To set the decimal scale at the instance level, specify `infinidb_decimal_scale` in the `my.cnf` configuration file. If not specified, the default is 8.

In `my.cnf` file (`/usr/local/Calpont/mysql`):

```
[mysqld]
xxxxx
infinidb_decimal_scale=value
```

where *value* is the amount of precision desired for calculations.

18 Using cpimport for Batch Insert

InfiniDB has the ability to utilize the cpimport fast data import tool for non-transactional “LOAD DATA INFILE” and “INSERT INTO SELECT FROM” SQL statements. Using this method results in a significant increase in performance in loading data through these two SQL statements. There are two variables used to control this ability:

- The `infinidb_use_import_for_batchinsert` variable is used by InfiniDB to control if cpimport is used for these statements. This use import for batch insert may be set as a default for the instance, set at the session level, or at the statement level by toggling this variable on and off.
- The `infinidb_import_for_batchinsert_delimiter` variable is used internally by InfiniDB on a non-transactional INSERT INTO SELECT FROM statement as the default delimiter passed to the cpimport tool. With a default value `ascii 7`, there should be no need to change this value unless your data contains `ascii 7` values.

Please refer to the InfiniDB SQL Syntax Guide for setting these at the session level.

18.1 Enable/Disable Using cpimport for Batch Insert

To enable/disable the use of the use cpimport for batch insert at the instance level, specify `infinidb_use_import_for_batchinsert` in the `my.cnf` configuration file. The default is 1 (enabled).

In `my.cnf` file (`/usr/local/Calpont/mysql`):

```
[mysqld]
xxxxx
infinidb_use_import_for_batchinsert=value
```

where *value* is 0 (disabled) or 1 (enabled).

18.2 Changing Default Delimiter for INSERT SELECT

The current default of `ascii 7` should be sufficient for most cases. But if your data contains `ascii 7` values, you will need to change this default to another `ascii` value. To change this value at the instance level, specify `infinidb_import_for_batchinsert_delimiter` in the `my.cnf` configuration file.

In `my.cnf` file (`/usr/local/Calpont/mysql`):

```
[mysqld]
xxxxx
infinidb_import_for_batchinsert_delimiter=ascii_value
```

where *ascii_value* is an `ascii` value representation of the delimiter desired.

19 Using Disk Based Joins

InfiniDB performs in-memory joins on the UM node and when a join operation exceeds the memory allocated on the UM for query joins, the query is aborted with an error code. The option is available to have such queries using disk for storing intermediate join data when the memory required for the join exceeds the UM memory limit. While such query performance will be slower than the joins that completely fit in memory and still bound by the temp space availability – it increases the probability of completion of those joins that have large cardinality and memory needs.

- Currently excluded from disk based join processing are:
 - Aggregation, including count(distinct)
 - DML

There are new configuration variables that allow you to manage this capability at the instance level. These variables reside in `HashJoin` element in the `Calpont.xml` configuration file (residing in the `etc` directory for your InfiniDB installation):

- `AllowDiskBasedJoin` – Controls the option to use disk Based joins or not. Valid values are Y (enabled) or N (disabled). By default, this option is disabled.
- `TempFileCompression` – Controls whether the disk join files are compressed or non-compressed. Valid values are Y (use compressed files) or N (use non-compressed files).
- `TempFilePath` – The directory path used for the disk joins. By default, this path is the `tmp` directory for your InfiniDB installation (i.e., `/usr/local/Calpont/tmp`). Files (named `infinidb-join-data*`) in this directory will be created and cleaned on an as needed basis. The entire directory is removed and recreated by `ExeMgr` at startup.)

Note: When using disk based joins, it is strongly recommended that the `TempFilePath` reside on its own partition because the partition may fill up as queries are executed.

In addition to the above instance wide variables, a session level variable exists to limit the amount of memory used by a user before the join is switched over to a disk based join. It can be set either at the instance level or session level (refer to the InfiniDB SQL Syntax Guide for session level modification):

- `infinidb_um_mem_limit` - Memory limit in MB per user (i.e. switch to disk based join if this limit is exceeded). By default, this limit is not set (value of 0).

For modification at the instance level:

In `my.cnf` file (typically `/usr/local/Calpont/mysql`):

```
[mysqld]
xxxxx
infinidb_um_mem_limit = value
```

where *value* is the value in Mb for in memory limitation per user.

Other session level variables exist that are used internally by InfiniDB for the processing of these disk based joins. **These should not be changed unless instructed to do so by InfiniDB Support.** If instructed to do so by InfiniDB Support, they can be set either at the instance level or session level (refer to the InfiniDB SQL Syntax Guide for session level modification).

- `infinidb_diskjoin_smallsidelimit` - Max disk space allowed in MB per user for small side results.
- `infinidb_diskjoin_largesidelimit` - Max disk space allowed in MB per user for large side results.
- `infinidb_diskjoin_bucketsize` - Max bucket size in MB. Bucket will split into multiple smaller buckets when it reaches this size.

For modification at the instance level:

In `my.cnf` file (typically `/usr/local/Calpont/mysql`):

```
[mysqld]
xxxxx
infinidb_diskjoin... = value
```

where *value* is the value in Mb to use as instructed by InfiniDB Support.

20 Compression Mode

InfiniDB has the ability to allow compression through a compression mode. This compression mode may be set as a default for the instance or set at the session level.

Please refer to the InfiniDB SQL Syntax Guide for setting the session level compression mode.

To set the compression mode at the instance level, specify `infinidb_compression_type` in the `my.cnf` configuration file. **This variable must exist in the `my.cnf` file for each UM in order for compression to be defaulted at the instance level.**

In `my.cnf` file (`/usr/local/Calpont/mysql`):

```
[mysqld]
xxxxx
infinidb_compression_type=value
```

where *value* is:

- (0) compression is turned off. Any subsequent table create statements run will have compression turned off for that table unless any statement overrides have been performed. Any alter statements run to add a column will have compression turned off for that column unless any statement override has been performed.
- (1 or 2) compression is turned on. Any subsequent table create statements run will have compression turned on for that table unless any statement overrides have been performed. Any alter statements run to add a column will have compression turned on for that column unless any statement override has been performed. SEE NOTE BELOW.

Compression Value Note: Starting with InfiniDB versions 3.0.6, 3.5.1 (and for all future releases), we have retired compression type 1. You cannot compress data using that algorithm. In order to minimize the effect of this change on our customers, InfiniDB will treat a request to use compression type 1 as a request for type 2 compression. Existing data compressed with type 1 will still be readable, but if you make changes to it, it will be recompressed using type 2 compression. All of this is regardless of the `infinidb_compression_type` variable setting and the value stored in `calpontsys.syscolumn`. You can still choose to disable compression by setting `infinidb_compression_type` to 0, but you cannot select to compress new data using type 1 compression. If you try to use type 1 compression, InfiniDB will silently treat it as a request to use type 2 compression.

In other words, you can turn compression off by setting `infinidb_compression_type` to 0. You can turn compression on by setting `infinidb_compression_type` to 1 or 2, but InfiniDB will always use the type 2 algorithm to compress new data regardless of whether you set `infinidb_compression_type` to 1 or 2.

This is not fundamentally different from previous InfiniDB versions where you only had the choice of compression being on or off: you still only have the choice of compression being on or off, but InfiniDB will be using a new algorithm (which we call "type 2").

An InfiniDB system restart is required after changes to this file in order to be active.

21 Local PM Query

InfiniDB has the ability to query data from just a single PM instead of the whole database thru the UM. In order to accomplish this, the `infinidb_local_query` variable in the `my.cnf` configuration file is used and maybe set as a default for the instance or set at the session level. This variable applies only to executing a query on an individual PM and will error if executed on the UM. The PM must be set up with the local query option during installation. See the InfiniDB Multiple UM Guide for more information.

Please refer to the InfiniDB SQL Syntax Guide for setting at the session level.

21.1 Enable/Disable Local PM Query

To enable/disable the use of the local PM Query at the instance level, specify `infinidb_local_query` in the `my.cnf` configuration file. The default is 0 (disabled).

In `my.cnf` file (`/usr/local/Calpont/mysql`):

```
[mysqld]
xxxxx
infinidb_local_query=value
```

where *value* is 0 (disabled) or 1 (enabled).

22 Partition Management

InfiniDB has the ability to better manage the removal of data by managing the disablement and drop of partitions. For more information on partitions in InfiniDB, please see InfiniDB Storage Concepts in the InfiniDB Concepts Guide and Partition Management in the InfiniDB SQL Syntax Guide.

The size of a partition is defined by two Calpont.xml settings - FilesPerColumnPartition and ExtentsPerSegmentFile. The default settings for these are FilesPerColumnPartition=4 and ExtentsPerSegmentFile=2. This equates to partition sizes of $4 * 2 * 8M = 64M$ rows. This setting is key as it defines the number of rows that can be rolled off at a time when dropping or disabling a partition.

These settings cannot be changed on an existing database once rows have been imported. The upgrade process will leave the settings if there is an existing database. It's a good idea to double check these settings before importing any rows if starting with a fresh database.

23 System Utilities

There are several system utilities that are useful in the operation of InfiniDB.

23.1 *configxml.sh*

The script 'configxml.sh' reads and sets values in the Calpont.xml file. It must be run on the active parent OAM performance module (use the OAM command getsystemstatus to determine this).

To read a value:

```
#./configxml.sh getconfig SystemConfig SystemName
```

Current value of SystemConfig / SystemName is caldb01

To set a value:

WARNING: Do not make changes to the Calpont.xml file unless directed to do so by service engineers. Incorrect settings can render your system unusable and will require a service call to correct. Setting a value should be performed when the system is stopped.

Please see the stopSystem and startSystem commands for information on stopping and starting the system.

```
# ./configxml.sh setconfig SystemConfig SystemName caldb02
```

Old value of SystemConfig / SystemName is caldb01

/usr/local/Calpont/etc/Calpont.xml backed up to /usr/local/Calpont/etc/ Calpont.xml.1254929966

Old value of SystemConfig / SystemName is caldb01

SystemConfig / SystemName now set to caldb02

23.2 *colxml*

colxml creates an XML job file for your database schema before you can import data. You should create a new job file before each bulk load. For more information on colxml, please see "Import Usage".

23.3 *cpimport*

cpimport is used to import data into InfiniDB. You must first create a new XML job file using colxml. For more information on cpimport.sh, please see the "Import data".

23.4 *viewtablelock*

If an import is aborted or fails, this utility will allow you to view any current active table locks still held by the import.

23.5 *cleartablelock*

Any active table locks still held by an aborted or failed import can be cleared using this utility. The format of this is `cleartablelock lockid`.

23.6 *healthCheck*

Note: This feature is only available with InfiniDB Enterprise.

This utility will return information about the status of the InfiniDB system.

Return Codes are:

- 000 – normal system up and running
- 100 – system is offline, or module is removed
- 101 – failed PM instance (OAM module state AUTO_OFFLINE or AUTO_DISABLED)
- 102 – failed dbroot (OAM status AUTO_OFFLINE)

23.7 *redistributeDB*

Note: This feature is only available with InfiniDB Enterprise.

Data redistribution has been added to relocate data segment files between DB Roots to achieve even data distribution, in which case InfiniDB performs best. There are different use cases where segment files need be reallocated:

- new Db roots added;
- uneven disk usage due to drop partition or bulk load modes 2 and 3

The data redistribute utility will make the data distribution as even as possible [in the same PM group, if PM group is supported]. This command can be run from any InfiniDB node, and there will be only one session at a time per system.

Command

```
redistributeDB [-h|--help] [ACTION]
```

where ACTION is [START|STOP|STATUS|CLEAR], default is START.

Examples

```
redistributeDB
redistributeDB start
redistributeDB START
redistributeDB STOP
redistributeDB STATUS
redistributeDB CLEAR
```

23.8 *calpontSupport*

The script 'calpontSupport' generates a number of text reports of reports about the system and tars up the InfiniDB System Logs and puts them into a single tar file. These reports and logs are useful for InfiniDB

Customer Support to use to help diagnose issues on a system or just to retrieve information about a system.

It should be executed from the server containing the User Module Front-End database, (i.e. 'um1' on a system with separate um's and pm's or 'pm1' on a system with combined um and pm functionality.)

Command

```
Usage: calpontSupport [-h] [-a] [-hw] [-s] [-c] [-db] [-r] [-l]
                    [-bl] [-lc] [-p 'root-password']
                    [-mp 'mysql-root-password'] [-de]

-h  help
-a  Output all Reports (excluding Bulk Logs Reports)
-hw Output Hardware Reports only
-s  Output Software Reports only
-c  Output Configuration/Status Reports only
-db Output DBMS Reports only
-r  Output Resource Reports only
-l  Output InfiniDB Log/Alarms Reports only
-bl Output InfiniDB Bulk Log Reports only
-lc Output Reports for Local Server only
-p  password (multi-server systems), root-password or
    'ssh' to use 'ssh keys'
-mp mysql root user password
-de Debug Flag
-hd Output Apache Hadoop™ reports only (if applicable)
```

Report Options:

- a = This is preferred option to retrieve all reports on a system, excluding the Bulk Log Analysis Report
- bl = To get the Bulk Log Reports (this report can take a while based on the amount of Bulk Logs on a system)
- a -bl = all reports including Bulk Log Reports.

Passwords:

1. Root User Password - On a system with multiple Servers, calpontSupport will send request to other servers to retrieve data. Either a root password is needed as a command line argument OR you can specify 'ssh' as the password and calpontSupport will utilize the ssh-keys

2. MySQL Root User Password - On a system that has a MySQL Root User Password setup, this needs to be provided in one of 3 forms to the calpontSupport to make MySQL calls
 - a) Use the command line option '-mp' to pass the password on a command line when running calpontSupport
 - b) Put the password in the /usr/local/Calpont/mysql/my.cnf file in the following form and the calpontSupport will read it from there:

```
[client]
password = 'my_password'
```
 - c) Enter it when prompted by the calpontSupport script. It will detect that a password is required. If there is no command line argument and no password setup in my.cnf, the user will get a prompt asking for the MySQL Root User Password

Example

An example run on a multi-server system, the calpontSupportReport. 'system-name'.tar.gz file that is generated would be supplied to InfiniDB Customer Support. If any errors are displayed to the terminal, please report those to InfiniDB Customer Support as well.

```
# ./calpontSupport -a -p my_password
Get software report data for pm1
Get software report data for pm2
Get software report data for pm3
Get software report data for pm4
Get software report data for pm5
Get config report data for pm1
Get software report data for pm6
Get software report data for um1
Get config report data for pm2
Get config report data for pm3
Get config report data for pm4
Get config report data for pm5
Get log report data for pm1
Get config report data for pm6
Get log report data for pm2
Get config report data for um1
Get log report data for pm3
Get log report data for pm4
Get log report data for pm5
Get log report data for pm6
Get hardware report data for pm1
Get log report data for um1
Get hardware report data for pm2
Get hardware report data for pm3
Get hardware report data for pm4
Get hardware report data for pm5
```

```

Get resource report data for pm1
Get hardware report data for pm6
Get log config data for um1
Get hardware report data for um1
Get resource report data for pm2
Get resource report data for pm3
Get resource report data for pm4
Get resource report data for pm5
Get resource report data for pm6
Get dbms report data for um1
Get resource report data for um1

```

InfiniDB Support Script Successfully completed, files located in
calpontSupportReport.my_system_name.tar.gz

23.9 databaseSizeReport

Note: This feature is only available with InfiniDB Enterprise.

Reports the physical size, in gigabytes, of the data existing on the dbroots. The databaseSizeReport command may be executed from any InfiniDB node. Several options exist that allow reporting of the various levels of the database (database, schema, table, column).

Command

```
databaseSizeReport [-s schema] [-t table] [-c ] [-r] [-i] [-h]
```

Usage: databaseSizeReport [options]

```

-h                display this help
-s schemaName    display all tables in the schema
-s schemaName -t tableName  display the table only
-c              display all tables in the
                database at column level
-r              report compressed bytes without
                including unconsumed preallocated
                file space
-i              format the output to be used
                by import to the following
                table:

```

```

CREATE TABLE dbsize( tableschema varchar(128),
    tablename varchar(128),columnName varchar(128),
    dataType int, columnWidth int, dbroot int,
    partition int, segment int, filename varchar(255),
    size double ) ENGINE=InfiniDB;

```

Sizes are reported in gigabyte.

NOTES:

- UTF-8 must be declared for this table if the instance has been set up with a UTF-8 profile.
- When using the -c option to display column level information, dictionary columns will have two entries to reflect the token (pointers to the dictionary) and string (actual dictionary string values) sizes.

Examples

Displays information for the whole database:

```
[root@myserver bin]# ./databaseSizeReport
Schema      Table      Size
calpontsys  syscolumn  0.039307 GB
calpontsys  systable   0.016602 GB
my_testdb   my_table   0.000984 GB
mydb        my_detail  0.894676 GB
mydb        my_orders  0.572357 GB
Total       1.5239404 GB
```

Displays information for the mydb schema:

```
[root@myserver bin]# ./databaseSizeReport -s mydb
Schema      Table      Size
mydb        my_detail  0.894676 GB
mydb        my_orders  0.572357 GB
Total       1.467033 GB
```

Displays actual compressed bytes used (excluding pre-allocated space) for all tables in the mydb schema. Note the actual size of data in the tables vs. size on disk without the -r option in the example directly above:

```
[root@myserver bin]# ./databaseSizeReport -s mydb -r
Schema      Table      Size
mydb        my_detail  0.282402 GB
mydb        my_orders  0.071480 GB
Total       0.353882 GB
```

Displays information for the my_detail table in the mydb schema:

```
[root@myserver bin]# ./databaseSizeReport -s mydb -t my_detail
Schema      Table      Size
mydb        my_detail  0.894676 GB
```

Displays information at the column level for the my_detail table in the mydb schema:

```
[root@myserver bin]# ./databaseSizeReport -s mydb -t my_detail -c
Schema      Table      Column      Size
mydb        my_detail  detail_orderkey  0.031258 GB
mydb        my_detail  detail_partkey   0.031258 GB
mydb        my_detail  detail_supkey    0.031258 GB
mydb        my_detail  detail_linenum    0.062508 GB
```

mydb	my_detail	detail_quantity	0.062508 GB
mydb	my_detail	detail_extendedprice	0.062508 GB
mydb	my_detail	detail_discount	0.062508 GB
mydb	my_detail	detail_tax	0.062508 GB
mydb	my_detail	detail_returnflag	0.007820 GB
mydb	my_detail	detail_linestatus	0.007820 GB
mydb	my_detail	detail_shipdate	0.031258 GB
mydb	my_detail	detail_commitdate	0.031258 GB
mydb	my_detail	detail_receiptdate	0.031258 GB
mydb	my_detail	detail_shipinstruct (token)	0.062508 GB
mydb	my_detail	detail_shipinstruct (string)	0.001961 GB
mydb	my_detail	detail_shipmode (token)	0.062508 GB
mydb	my_detail	detail_shipmode (string)	0.001961 GB
mydb	my_detail	detail_comment (token)	0.062508 GB
mydb	my_detail	detail_comment (string)	0.187508 GB
Table total size:			0.894676 GB

Creates a delimited file of information at the column level for the whole database. This file may then be used with cpimport to load the dbsize table (definition above):

```
[root@myserver bin]# ./databaseSizeReport -c -i > dbsize.tbl
```

```
[root@myserver bin]# cat dbsize.tbl
```

```
mydb|my_detail|detail_orderkey|6|4|3|0|0|/usr/local/Calpont/
data3/000.dir/000.dir/040.dir/211.dir/000.dir/
FILE000.cdf|0.031258|:
:
mydb|my_detail|detail_comment (token)|12|44|3|0|0|/usr/local/
Calpont/data3/000.dir/000.dir/040.dir/226.dir/000.dir/
FILE000.cdf|0.062508|
mydb|my_detail|detail_comment (string)|12|44|3|0|0|/usr/local/
Calpont/data3/000.dir/000.dir/040.dir/229.dir/000.dir/
FILE000.cdf|0.187508|
```

```
[root@myserver bin]# /usr/local/Calpont/bin/cpimport my_testdb dbsize
dbsize.tbl
```

24 Using UTF-8 Character Sets

24.1 UTF-8 Character Set

InfiniDB has the ability to support UTF-8 character sets. This profile may be set as a default for the instance or set at the session level.

Please refer to the InfiniDB SQL Syntax Guide for the setting UTF-8 profile at the session level.

To set UTF-8 profile at the instance level, specify the following in my.cnf and the Calpont.xml files.

24.1.1 my.cnf

To configure the MySQL Locale language, modify the my.cnf file (/usr/local/Calpont/mysql):

```
[client]
default-character-set=utf8

[mysqld]
character-set-server=utf8
collation-server=value
init-connect='SET NAMES utf8'
```

where *value* is a valid unicode character set. Please see the following MySQL reference of valid values:

<http://dev.mysql.com/doc/refman/5.1/en/charset-unicode-sets.html>

If no values are specified for collation-server, the default is *xxx_general_ci*.

24.1.2 Calpont.xml

To configure the InfiniDB Locale language, modify the Calpont.xml files (/usr/local/Calpont/etc)

```
<SystemConfig>
  <SystemLang>value</SystemLang>
```

where *value* is a valid locale for the operating system. The recommended setting is *en_US.utf8*.

Once these changes have been made, InfiniDB and the InfiniDB MySQL daemon must be restarted to take effect.

24.1.3 Importing Data

When using the cpimport utility, the input file must be converted to UTF-8 data. The Linux program iconv is one utility to convert it.

```
iconv -f ISO-8859-1 -t UTF-8 < input.txt > output.txt
```

24.2 Known Issues and Limitations

- UTF-8 must be declared at the table level if the instance has been set up with a UTF-8 profile. Tables created with a non-matching character set will yield indeterminate results.
- Viewing SQL output should be done using client software that supports UTF-8 character sets.

25 Troubleshooting

25.1 **ERROR 1070 (42000): Too many key parts specified; max 0 parts allowed**

This error will be displayed during a CREATE/ALTER table statement when a key (primary, unique, etc.) is being defined. Since InfiniDB has disabled the use of keys in the MySQL front end, this error is being returned by MySQL to reflect no keys allowed.

25.2 **UM memory exceeded error / distributed join performance**

InfiniDB dynamically determines the table with the largest cardinality for use in its distributed join operation. Using the correct large side table can make a significant difference in how the query performs and, in some cases, whether the query can be run within available resources. There are cases where InfiniDB will not predict the correct large side table. For these cases, the behavior can be overridden using the `INFINIDB_ORDERED` hint which will use the first table in the from clause as the large side table. For example, the query below will use `lineitem` as the large side table.

```
select /*! INFINIDB_ORDERED */ count(*) from lineitem, supplier where  
l_suppkey = s_suppkey and l_shipdate <= '1998-12-31' and s_suppkey <= 10000;
```

25.3 **Error in Aggregation Data Overflow**

If a query is executed that receives the InfiniDB error “The error was aggregation data overflow”, an overflow in trying to aggregate a column has occurred. Modifying the `infinidb_decimal_scale` value from its default (usually 8 unless an override has been made) may assist in getting the query to execute successfully. Please see the “Decimal to Double Math

InfiniDB has the ability to change intermediate decimal mathematical results from decimal type to double. The decimal type has approximately 17-18 digits of precision, but a smaller maximum range whereas the double type has approximately 15-16 digits of precision, but a much larger maximum range (refer to the InfiniDB SQL Syntax Guide Datatype section for details). Therefore, the proper setting depends on the intended usage. In typical mathematical and scientific applications, the ability to avoid overflow in intermediate results with double math is likely more beneficial than the additional two digits of precision. In financial applications, however, it may be more appropriate to leave in the default decimal setting to ensure accuracy to the least significant digit.

The `infinidb_double_for_decimal_math` variable is used by InfiniDB to control the data type for intermediate decimal results. This decimal for double math may be set as a default for the instance, set at the session level, or at the statement level by toggling this variable on and off.

Please refer to the InfiniDB SQL Syntax Guide for setting at the session level.

25.4 **Enable/Disable Decimal to Double Math**

To enable/disable the use of the decimal to double math at the instance level, specify `infinidb_double_for_decimal_math` in the `my.cnf` configuration file. The default is 0 (disabled).

In my.cnf file (/usr/local/Calpont/mysql):

```
[mysqld]
xxxxx
infinidb_double_for_decimal_math=value
```

where *value* is 0 (disabled) or 1 (enabled).

Decimal Scale" further guidance.

25.5 Circular join detected error

If a query is executed that receives the InfiniDB not supported error with a "Circular join is not supported" error, there is likely a cross-table equal comparison involved in the query. InfiniDB cannot handle this statement by default. However, adding a "+ 0" to one side of the cross-table comparison will hint InfiniDB that the comparison is a cross-table equal comparison, rather than a join, and allow the query to be processed.

For example, this query will not run as written.

Example:

```
mysql> select count(*)
-> from customer,
->      supplier,
->      orders,
->      lineitem
-> where c_custkey = o_custkey and
->      l_orderkey = o_orderkey and
->      l_suppkey = s_suppkey and
->      c_nationkey = s_nationkey and
->      o_orderdate >= date '2012-01-01' and
->      o_orderdate < date '2012-01-01' + interval '1' year;
ERROR 138 (HY000): IDB-1003: Circular join is not supported.
```

Adding the + 0 (bolded in the statement below) to the c_nationkey = s_nationkey comparison will allow the query to run.

```
mysql> select count(*)
-> from customer,
->      supplier,
->      orders,
->      lineitem
-> where c_custkey = o_custkey and
->      l_orderkey = o_orderkey and
->      l_suppkey = s_suppkey and
->      c_nationkey = s_nationkey + 0 and
->      o_orderdate >= date '2012-01-01' and
->      o_orderdate < date '2012-01-01' + interval '1' year;
+-----+
| count(*) |
+-----+
|    36449 |
+-----+
```

25.6 Imports executing slower with concurrent queries

Imports and queries executing concurrently may reduce the speed of the imports. To correct this, users may want to modify the execution priority for these processes. Please see “Process Prioritization” in the Data Load Rates chapter of the Performance Tuning for the InfiniDB Analytics Database guide for details.

25.7 Delay in data retrieval immediately after a **CREATE TABLE**

In a multiple PM environment, the creation of tables may result in a delay selecting data back the first time. The following server setting, and changing it for each PM, may improve this delay:

/proc/sys/vm/dirty_expire_centiseconds (default 3000): In hundredths of a second, how long data can be in the page cache before it's considered expired and must be written at the next opportunity. Note that this default is very long: a full 30 seconds. That means that under normal circumstances, unless you write enough to trigger the other pdflush method, Linux won't actually commit anything you write until 30 seconds later.

To view setting:

```
# cat /proc/sys/vm/dirty_expire_centisecs
```

To temporarily modify setting from default down to 1 second:

This will be in effect until the next reboot of the server or until a subsequent command is issued to modify it.

```
# echo 100 > /proc/sys/vm/dirty_expire_centisecs
```

To permanently modify setting from default down to 1 second:

Caution is advised to consider other non-InfiniDB processes that may be running on the server as this may affect them as well. Place the following entry (or modify existing entry) in the `/etc/sysctl.conf` file:

```
vm.dirty_expire_centisecs=100
```

25.8 Lost connection when using **stdin** with **cpimport**

Partial data loading may occur when using the STDIN option with `cpimport` if an error occurs with the query portion of the command. Since this combined command is simply two commands chained together, once the query has completed (either by successful completion or by an error), `cpimport` will simply continue to load the data received.

26 Appendix A - 3rd Party Licenses

All of the code in InfiniDB that is original work of InfiniDB Corp. is copyrighted by InfiniDB Corp. and is covered under the GNU General Public License, V2. Every source file in this category contains text that identifies it as such. Other portions of InfiniDB and InfiniDB Enterprise are licensed under various terms and are copyrighted by various individuals. Their attributions follow.

26.1 GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

26.1.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

26.1.2 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program. You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or

rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

26.1.3 NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

26.1.4 END OF TERMS AND CONDITIONS

26.1.4.1 How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms. To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.  
Copyright (C) yyyy  name of author
```

```
This program is free software; you can redistribute it and/or  
modify it under the terms of the GNU General Public License  
as published by the Free Software Foundation; either version 2  
of the License, or (at your option) any later version.
```

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program. You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the [GNU Lesser General Public License](#) instead of this License.

26.2 RSA Data Security

Portions of the code are derived from copyrighted work by RSA Data Security, Inc and bear the following copyright notice:

```
// Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
// rights reserved.
```

```
//  
// License to copy and use this software is granted provided that it  
// is identified as the "RSA Data Security, Inc. MD5 Message-Digest  
// Algorithm" in all material mentioning or referencing this software  
// or this function.  
// License is also granted to make and use derivative works provided  
// that such works are identified as "derived from the RSA Data  
// Security, Inc. MD5 Message-Digest Algorithm" in all material  
// mentioning or referencing the derived work.  
// RSA Data Security, Inc. makes no representations concerning either  
// the merchantability of this software or the suitability of this  
// software for any particular purpose. It is provided "as is"  
// without express or implied warranty of any kind.  
// These notices must be retained in any copies of any part of this  
// documentation and/or software.
```

26.3 Paul E. Jones

Portions of the code are derived from copyrighted work by Paul E. Jones (<http://www.packetizer.com/security/sha1/>). They are in the public domain but retain this copyright:

- * Copyright (C) 1998, 2009
- * Paul E. Jones <paulej@packetizer.com>
- * All Rights Reserved.

26.4 Apache

Portions of the code are derived from work copyrighted by Apache Hadoop and are covered by the Apache License V2.0:

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage,

computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]
```

```
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or  
implied.
```

```
See the License for the specific language governing permissions and  
limitations under the License.
```

26.5 Gary S. Brown

Portions of the code are derived from work copyrighted by Gary S. Brown and bear the following copyright notice:

- * Copyright (C) 1986 Gary S. Brown. You may use this program, or
- * code or tables extracted from it, as desired without restriction.

26.6 Brian M. Clapper

Portions of the code are derived from copyrighted work by Brian M. Clapper and are covered under the following license:

This software is released under the following license:

Copyright (c) 1995-2002 Brian M. Clapper

All rights reserved.

Redistribution and use in source and binary forms are permitted provided that: (1) source distributions retain this entire copyright notice and comment; (2) modifications made to the software are prominently mentioned, and a copy of the original software (or a pointer to its location) are included; and (3) distributions including binaries display the following acknowledgement: "This product includes software developed by Brian M. Clapper <bmc@clapper.org>" in the documentation or other materials provided with the distribution. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Effectively, this means you can do what you want with the software except remove this notice or take advantage of the author's name. If you modify the software and redistribute your modified version, you must indicate that your version is a modification of the original, and you must provide either a pointer to or a copy of the original.

26.7 Net-SNMP

Portions of the code are derived from copyrighted work by Net-SNMP and are licensed under the following terms:

Various copyrights apply to this package, listed in various separate parts below. Please make sure that you read all the parts.

---- Part 1: CMU/UCD copyright notice: (BSD like) ----

Copyright 1989, 1991, 1992 by Carnegie Mellon University

Derivative Work - 1996, 1998-2000

Copyright 1996, 1998-2000 The Regents of the University of California

All Rights Reserved

Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU and The Regents of the University of California not be used in advertising or publicity pertaining to distribution of the software without specific written permission.

CMU AND THE REGENTS OF THE UNIVERSITY OF CALIFORNIA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL CMU OR THE REGENTS OF THE UNIVERSITY OF CALIFORNIA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM THE LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

---- Part 2: Networks Associates Technology, Inc copyright notice (BSD) ----

Copyright (c) 2001-2003, Networks Associates Technology, Inc
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Networks Associates Technology, Inc nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;

OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 3: Cambridge Broadband Ltd. copyright notice (BSD) ----

Portions of this code are copyright (c) 2001-2003, Cambridge Broadband Ltd.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * The name of Cambridge Broadband Ltd. may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 4: Sun Microsystems, Inc. copyright notice (BSD) ----

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Use is subject to license terms below.

This distribution may include materials developed by third parties.

Sun, Sun Microsystems, the Sun logo and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Sun Microsystems, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 5: Sparta, Inc copyright notice (BSD) ----

Copyright (c) 2003-2011, Sparta, Inc
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Sparta, Inc nor the names of its contributors may

be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 6: Cisco/BUPTNIC copyright notice (BSD) ----

Copyright (c) 2004, Cisco, Inc and Information Network
Center of Beijing University of Posts and Telecommunications.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Cisco, Inc, Beijing University of Posts and Telecommunications, nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF

ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 7: Fabasoft R&D Software GmbH & Co KG copyright notice (BSD) ----

Copyright (c) Fabasoft R&D Software GmbH & Co KG, 2003

oss@fabasoft.com

Author: Bernhard Penz <bernhard.penz@fabasoft.com>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * The name of Fabasoft R&D Software GmbH & Co KG or any of its subsidiaries, brand or product names may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 8: Apple Inc. copyright notice (BSD) ----

Copyright (c) 2007 Apple Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following

disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of Apple Inc. ("Apple") nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY APPLE AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APPLE OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 9: ScienceLogic, LLC copyright notice (BSD) ----

Copyright (c) 2009, ScienceLogic, LLC
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of ScienceLogic, LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS

OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

26.8 MySQL AB / Sun Microsystems

Portions of the product are compiled from code copyrighted by MySQL AB and Sun Microsystems, Inc. This code is licensed under the GPL V2 and bears the following copyright:

Copyright 2000-2008 MySQL AB, 2008 Sun Microsystems, Inc.

26.9 Lasse Mikkell Reinhold

Portions of the product are compiled from code copyrighted by Lasse Mikkell Reinhold and bear the following copyright:

Copyright (C) 2006-2011 Lasse Mikkell Reinhold

27 Glossary

ad-hoc query

A customized query that cannot be determined prior to the moment the query is issued. Typically a SQL statement.

business intelligence (BI)

The process of receiving business data that is reliable, consistent, understandable, easily manipulated and timely. With this data, business users are able to conduct analyses that yield overall understanding of where the business has been, where it is now and where it will be in the near future. Business intelligence serves two main purposes. It monitors the financial and operational health of the organization (reports, alerts, alarms, analysis tools, key performance indicators and dashboards). It also regulates the operation of the organization providing two- way integration with operational systems and information feedback analysis.

InfiniDB Console

The user interface to communicate with the InfiniDB Stack through commands entered as lines of text from the keyboard.

column-based relational database

A database that is structured by column instead of by row.

COTS (commercial off the shelf components)

Computer products that are available for sale to the general public and designed to be easily installed and compatible with other system components.

cron (Command Run ON)

A daemon that is used to execute scheduled tasks. You can use cron to automate tasks to perform periodic backups or data loads.

data definition language (DDL)

A language used by a database management system which allows users to define the database, specifying data types, structures and constraints on the data. Examples are the CREATE TABLE, CREATE INDEX, ALTER, and DROP statements.

data manipulation language (DML)

Commands that manipulate data within existing database objects. Examples are the SELECT, UPDATE, INSERT, DELETE statements.

data warehouse

A data warehouse is a database geared towards business intelligence requirements. The data warehouses are updated on a periodic basis and contain historical information that enables analysis of business performance over time.

DDL (See data definition language)

DML (See data manipulation language)

DNS (See domain name server)

domain

A set of network addresses that can directly transmit to each other on the same network without using a router.

domain name

The name given to a specific IP address so that people can recognize and remember it easier.

domain name server (DNS)

A program running on a server which automatically translates domain names into their correct TCP/IP addresses.

gateway

A network point that acts as an entrance to another network. The computers that control traffic within your company's network or at your local Internet service provider (ISP) are gateway nodes.

host name

The name of a computer that has two-way access to other computers on the Internet.

IP address

Every computer on a network has a unique IP number which is four sets of numbers divided by period with up to three numbers in each set. (Example: 10.0.0.127) Many computers also have a domain names that is easier to remember.

NIC (network interface card)

Computer hardware that provides physical access to a network. Many servers provide more than one NIC so that a server has multiple ports accessing a network.

NMS (network monitoring system)

Software that monitors the components in a network.

redundant array of independent disks (RAID)

A means of storing data on multiple hard disks that appears to the operating system to be a single logical hard disk. By placing data on multiple disks, the operations of accessing data is balanced for improved performance.

SSH (secure shell)

A set of standards that allows a secure channel between a local and a remote computer using authentication methods.

tokenization

The means of creating smaller strings of input characters to represent larger strings. The smaller strings or tokens use less memory and space.