



## CHAPTER 4

# Managing the Collection Manager

---

Revised: June 02, 2011, OL-7208-13

## Introduction

This chapter explains how to use utility scripts to view and update collection manager (CM) parameters. Any machine connected to the CM through, for example, Telnet or SSH, can use utility scripts to monitor and manage the CM. The utility scripts are located in the installation directory of the CM.

For information on managing the database and the CSV repository, see [Managing Databases and the Comma Separated Value Repository, page 5-1](#).

- [Using Utility Scripts, page 4-1](#)
- [Collection Manager Support Information, page 4-2](#)
- [Configuring the Collection Manager, page 4-2](#)
- [Configuring the Categorizer, page 4-6](#)
- [Monitoring System Health, page 4-6](#)
- [Managing Users, page 4-8](#)
- [Managing Virtual Links, page 4-9](#)
- [How to Monitor the Collection Manager, page 4-10](#)

## Using Utility Scripts

General instructions for using utility scripts include:

- To invoke any script, log in as the `scmscm` user, except where otherwise noted. An attempt to run these scripts as the root user results in an error.
- To display a description of the script, with an explanation of all flags and parameters, invoke the script with the help flag.



### Note

---

A slight variation exists in the help flag. Scripts for managing the CM use `--help`; scripts for managing the database use `-h`. Consult the specific script definition.

---

The following example shows how to display a description of the **dbperiodic.py** script.

```
>~scmscm/scripts/dbperiodic.py --help
Usage:
~scmscm/scripts/dbperiodic.py --load
load configuration from
/export/home/scmscm/db_maint/dbperiodic.conf
~scmscm/scripts/dbperiodic.py --loadfile=FILE
load configuration from FILE
~scmscm/scripts/dbperiodic.py --dump
print the current configuration in INI format to standard output
~scmscm/scripts/dbperiodic.py --help
print this help message
```


**Note**

Some of the scripts used to control and monitor the data-collector software use the Python scripting language. For more information about Python, go to <http://www.python.org>.

## Collection Manager Support Information

If you are experiencing difficulties with the CM installation, it may be necessary to provide the Cisco Technical Assistance Center (TAC) with information about the current system setup. The CM can create the required support information or files for TAC by running one of the following scripts:

- **~scmscm/unsupported/getinfo/get\_cm\_info.sh**—This script creates the CM support information.
- **~scmscm/unsupported/getinfo/get\_support\_files.sh**—This script creates a support file.

The output is a zip file with the investigation supporting files. The following example shows the output that appears when running this script:

```
bash-2.05$ ~scmscm/unsupported/getinfo/get_support_files.sh
Gathering support files into Sun_14_Oct_2007_06-05-03_PM_JST.zip, please
wait.....done.
The generated support file contains the output of
~scmscm/unsupported/getinfo/get_cm_info.sh
```

## Configuring the Collection Manager

Use utility scripts to:

- Specify which servers are to be activated at startup
- Start or stop the database
- Start or stop an adapter
- Drop a Service Control Engine (SCE) connection

Use the following scripts to configure the CM:

- **~scmscm/setup/on-boot.py**
- **~scmscm/scripts/adapterconf.py**
- **~scmscm/scripts/sceconf.sh**
- **~scmscm/scripts/dbconf.sh**

For information about scripts for managing the database and the CSV repository, see [Managing Databases and the Comma Separated Value Repository, page 5-1](#).

The following files are also used to configure the CM:

- **cm.conf**—General configuration of the CM, including which adapters will be turned on when the CM starts. See [Enabling Adapters, page 4-4](#).
- **queue.conf**—Configuration of the adapter queues, including which RDR tags will be associated with a specific adapter. See [Configuring the Categorizer, page 4-6](#).
- [Activating Servers, page 4-3](#)
- [Controlling Adapters, page 4-4](#)
- [Enabling Adapters, page 4-4](#)
- [Dropping an SCE Connection, page 4-5](#)

## Activating Servers

To set which servers (CM or Sybase) are activated at startup, use the **on-boot.py** script:

```
~scmscm/setup/on-boot.py --cm=flag --sybase=flag
```

Changes take effect the next time the system restarts.



### Note

To view the current startup status of each component, run the script with no parameters.

To restart the CM, run the following script as the scmscm user:

```
~scmscm/cm/bin/cm restart
```

**Table 4-1** *on-boot.py* Options

<code>--cm={ on   off }</code>	Activate/do not activate the CM at startup.
<code>--sybase={ on   off }</code>	Activate/do not activate the Sybase server at startup.

The following example shows how to set the CM and Sybase servers to be activated at startup. (This is the default setting of the script.)

```
>>~scmscm/setup/on-boot.py --cm=on --sybase=on
```

## Controlling Adapters

To shut down or bring up a configured adapter, or to list the currently running CM adapters, use the **adapterconf.py** script:

```
~scmscm/scripts/adapterconf.py --op=action [ --adapter=adapter name ]
```

**Table 4-2** *adapterconf.py* Options

<code>--op=start</code>	Bring up the adapter specified in the <b>adapter</b> parameter.
<code>--op=stop</code>	Shut down the adapter specified in the <b>adapter</b> parameter.
<code>--op=list</code>	List the currently running CM adapters.
<code>adapter=adapter name</code>	Identify the adapter to be operated on. Use only with <b>start</b> and <b>stop</b> actions.
<code>--help</code>	Display these options.

To shut down an adapter, as the scmscm user, run the following script:

```
~scmscm/scripts/adapterconf.py --op=stop --adapter=adapter name
```

To bring up an adapter, as the scmscm user, run the following script:

```
~scmscm/scripts/adapterconf.py --op=start --adapter=adapter name
```

## Enabling Adapters

You can define an adapter to turn on when the CM starts by removing the remark character at the start of the appropriate line in the **cm.conf** file.

The following example shows how to define the RAG adapter to turn on when the CM starts.

```
adapter.4 = com.cisco.scmscm.adapters.rag.RAGAdapter
```

The following example shows how to define the CSV adapter to remain off when the CM starts.

```
#adapter.2 = com.cisco.scmscm.adapters.CSVAdapter
```



### Note

The value of the **adapter.<number>** must match the **adapter\_id** parameter value defined in the **queue.conf** file for the corresponding adapter.

## Dropping an SCE Connection

To drop a connection to a particular SCE, use the **sceconf.sh** script:

```
~scmscm/scripts/sceconf.sh --op=drop --ip=IP address
```

This script can be used only if the HTTP adaptor of the CM is running.

This script is also used to display information about the SCE connection. (See [Checking the SCE Connection, page 4-11](#))

**Table 4-3** *sceconf.sh* Options

<b>Adapter=IP address</b>	Drop the connection at the specified IP address.
<b>--help</b>	Display these options.

To drop an SCE connection, as the scmscm user, run the following command:

```
~scmscm/scripts/sceconf.sh --op=drop --ip=IP address
```

## Configuring Databases

To configure a database, use the **dbconf.sh** script:

```
~scmscm/scripts/dbconf.sh
```

The script prompts you to choose the database type and the corresponding database configuration parameters.

The following example shows how to use the dbconf.sh script to configure the CM to work with an external Oracle database:

```
$ ~scmscm/scripts/dbconf.sh
Enter the DB type:
1 - Oracle
2 - MySQL
3 - Sybase (external not bundled)
Enter your choice: 1
Enter Oracle server host (current is localhost) :10.56.216.80
Enter Oracle server listening port (current is 1521) :
Enter Oracle server instance id (current is apricot) :
Enter CM schema user name (current is pqb_admin) :
Enter CM schema user password (current is pqb_admin) :
Do you want to test the DB connection? (yes/no): yes
PASS:db is up
DB connection succeeded.
$
```



### Note

By default, the Collection Manager handles database related transactions in uppercase table names. However, in Windows based MySQL server, table names are stored in lowercase and name comparisons are not case-sensitive; and, it automatically converts all table names to lowercase. Therefore, if you are using Windows based database server, you need to modify the *lower\_case\_table\_names* server configuration entry.

## Configuring the Categorizer

The Categorizer classifies each RDR according to its RDR tag. An RDR can be routed to a specific adapter by adding its RDR tag to the *tags* parameter (a comma-separated list of RDR tags) of the adapter. This configuration is contained in the **queue.conf** file.

The following example configures the RDR tags **4042321920** and **4042321922** to be sent to the topper/aggregator adapter.

```
# Topper/Aggregator Adapter
[topper-hi]
adapter_id=3
priority=3
warning_size=40000
maximum_size=50000
tags=4042321920,4042321922
```



### Note

The value of the **adapter\_id** parameter must match the **adapter.<number>** defined in the **cm.conf** file for the corresponding adapter.

## Monitoring System Health

The CM contains a small, expandable framework that monitors the system and issues alerts for predefined, potentially problematic conditions.

The following scripts are used to monitor the CM:

- `~scmscm/setup/monitor/setup-monitor.sh`
- `~scmscm/setup/monitor/monitor.sh`

## Installing the Periodic Checker

To make (or remove) an entry for **monitor.sh**, the periodic checker script, in the cron (periodic scheduler) subsystem, use the **setup-monitor.sh** script:

```
~scmscm/setup/monitor/setup-monitor.sh -a flag [ -I flag ]
```

**Table 4-4** *setup-monitor.sh* Options

<code>-a { install   uninstall }</code>	Make/remove an entry for <b>monitor.sh</b> in the cron.
<code>-I { 30m   1h   12h   24h }</code>	Run <b>monitor.sh</b> every 30 minutes, 1 hour, 12 hours, or 24 hours.

The following example shows how to install **monitor.sh** so that it runs one time every 30 minutes.

```
$ ./setup-monitor.sh -a install -I 30m
```

The following example shows how to uninstall **monitor.sh**.

```
$ ./setup-monitor.sh -a uninstall
```

## Periodic Checker Script

- [Periodic Checker Script, page 4-7](#)
- [Tests, page 4-8](#)

### Periodic Checker Script

The periodic checker script, **monitor.sh**, calls a series of subscripsts that monitor different aspects of a running system:

```
~scmscm/setup/monitor/monitor.sh { -a | TEST_NAME } [ -v ] [ -d ]
```

The script is not intended to be run from the command line, although you can do so. Test results are sent to the syslog subsystem and are logged in the file **/var/log/messages**.

**Table 4-5** *monitor.sh Options*

<b>-a</b>	Run all tests.
<i>TEST NAME</i>	The names of one or more tests. A test name is the test filename, without the leading digits and trailing .sh.
<b>-v</b>	Output results in verbose mode. (Log successful tests.)
<b>-d</b>	Print results to window. (By default, results are sent to syslog.)

Any test that is run returns a result in the following format:

```
STATUS: Message
```

- STATUS—PASS or FAIL
- Message—A short informative status message

For example, **FAIL: db "apricot" has only 1523 free blocks**

The following example shows how to run all available tests and print system output to the window.

```
$ ./monitor.sh -d -a
Test: 01free_db.sh. Status: PASS. Message: db apricot has 1532 free blocks
Test: 02cm_is_up.sh. Status: FAIL. Message: cm process is not running
```

The following example shows how to run one test to check that the installed database has sufficient free space.

```
$ ./monitor.sh -d free_db
Test: 01free_db.sh. Status: PASS. Message: db apricot has 1532 free blocks
```

## Tests

The following tests can be run using **monitor.sh**:

- **db\_up**—Checks that the CM database is running.
- **cm\_up**—Checks that the CM application is running.
- **free\_db**—Checks that the bundled Sybase database has at least 10 percent free space.
- **free\_log**—Check that the bundled Sybase database transaction log has at least 70 percent free space.
- **cm\_persistent\_buffers**—Checks that each CM adapter's persistent buffer contains less than 500 files.

The scripts for all these tests are located in the `~/setup/monitor/tests` directory.

When calling a test called **test\_name**, the script expects to find a file called **NNtest\_name.sh**, where NN is a number that denotes script priority. For example, the test **free\_db** is mapped to the file **01free\_db.sh**.

## Managing Users

The CM uses the **p3rpc** utility to manage users for authenticated RPC calls.

The command format is: **p3rpc OPERATION [OPTIONS]**

The following table lists **p3rpc** operations and options.

**Table 4-6** *p3rpc Operations*

Operation	Description
<b>--set-user</b> <b>--username=</b> <i>username</i> <b>--password=</b> <i>password</i>	Adds and updates the username and password.
<b>--validate-password</b> <b>--username=</b> <i>username</i> <b>--password=</b> <i>password</i>	Validates the username and password.
<b>--delete-user</b> <b>--username=</b> <i>username</i>	Delete a user configuration.
<b>--show-users</b>	Displays all configured users.

## Examples

The following example shows how to add and update the username and password:

```
bash-2.05$ p3rpc --set-user --username=lulu --password=lili
Command terminated successfully
bash-2.05$
```

The following example shows how to validate username and password. In this example, the user is successfully validated.

```
bash-2.05$ p3rpc --validate-password --username=lala --password=lala
Local machine: user lala was authenticated successfully : (auth level root)
Command terminated successfully
bash-2.05$
```



The following example shows how to validate username and password. In this example, the user validation fails.

```
bash-2.05$ p3rpc --validate-password --username=lala --password=lulu
Error - Failed to authenticate user lala
bash-2.05$
```

The following example shows how to delete a user configuration.

```
bash-2.05$ p3rpc --delete-user --username=lulu
Command terminated successfully
bash-2.05$
```

The following example shows how to display all of the configured users. In this example, only one configured user exists: clu.

```
bash-2.05$ p3rpc --show-users
clu
Command terminated successfully
bash-2.05$
```

## Managing Virtual Links

A script is included in the CM distribution to allow you to manage virtual link names and indexes that are configured for a specific SCE.

To show or set virtual links, use the **update\_vlinks.sh** script:

```
~scmscm/cm/bin/update_vlinks.sh --sce=SCE IP address [ --file=file | --show ]
```

**Table 4-7** update\_vlinks.sh Options

<code>--sce=SCE IP --file=file</code>	Update the VLINK_INI table with the data in the supplied csv formatted file for the specified SCE.
<code>--sce=SCE IP --show</code>	Query the VLINK_INI table for entries for the specified SCE.
<code>--help</code>	Display these options.

To set the virtual link details, as the scmscm user, run the following command:

```
~scmscm/cm/bin/update_vlinks.sh --sce=SCE IP address --file=file
```

The CSV file format is: link id (positive integer), link direction (0=upstream, 1=downstream), name (string).

The following validation steps are performed on the file:

- The file exists
- There are no duplicate virtual links ids for each direction
- The virtual links id is a positive value from 0 to 1024.
- The direction is either 0 (upstream) or 1 (downstream)
- No duplicate virtual links names or empty names exist for each direction
- Virtual links names can contain up to 256 characters. All printable characters with an ASCII code between 32 and 126 (inclusive) can be used; except for 34 ("), 39 ('), and 96 (^).

After the file is successfully validated, the script performs the following actions:

1. All entries containing the SCE IP address in their SCE\_IP field are deleted from the VLINK\_INI table
2. Two entries will be added to the VLINK\_INI table in the following format:
  - Timestamp, sce ip, 0, 0, "Default Virtual Link Up"
  - Timestamp, sce ip, 0, 1, "Default Virtual Link Down"
3. The CSV file is parsed and each line in the CSV file will be entered as a line entry in the VLINK\_INI table.

To show the virtual link details, as the scm scm user, run the following command:

```
~scmscm/cm/bin/update_vlinks.sh --sce=SCE IP address --show
```

## How to Monitor the Collection Manager

You can use scripts to monitor system statistics that are relevant to the CM, such as:

- Percentage of free space in the database
- Rate of RDRs entering the CM
- SCE platform connection data
- Viewing database insertion rate statistics per table
- Viewing version information

The following scripts are used to monitor the CM:

- ~scmscm/scripts/dbfree.sh
- ~scmscm/scripts/rdr-rate.py
- ~scmscm/scripts/sceconf.sh
- ~scmscm/setup/alive.sh
- ~scmscm/cm/bin/p3stats
- ~scmscm/cm/bin/cm version
- ~scmscm/cm/bin/cm dbversion

The following scripts are used to configure the CM (see [Configuring the Collection Manager, page 4-2](#)), but can also be invoked to display the relevant configuration:

- ~scmscm/setup/on-boot.py
- ~scmscm/scripts/adapterconf.py

## Checking the Database Capacity

To display the percentage of free space in the database report tables and the associated transaction log, use the **dbfree.sh** script:

```
~scmscm/scripts/dbfree.sh
```

The script can be used only with the bundled sybase installation.

---

**Step 1** As the scmscm user, run the **dbfree.sh** script

---

## Checking the RDR Rate

To display the momentary total rate of reports entering the CM, use the **rdr-rate.py** script

```
~scmscm/scripts/rdr-rate.py
```

The output is a single floating-point number representing the total rate per second of incoming RDRs (from all sources) that have entered the CM in the past 5 seconds.

This script can be used only if the HTTP adaptor of the CM is running.

---

**Step 1** As the scmscm user run the **rdr-rate.py** script

---

## Checking the SCE Connection

To display information about the SCE connections, use the **sceconf.sh** script:

```
~scmscm/scripts/sceconf.sh --op=list
```

This script can be used only if the HTTP Adaptor of the CM is running.

The script is also used to drop a connection from a particular SCE. See [Dropping an SCE Connection, page 4-5](#).

---

**Step 1** As the scmscm user, run the **sceconf.sh** script

```
~scmscm/scripts/sceconf.sh --op=list
```

### Example:

The following example shows SCE connection output:

```
>~scmscm/scripts/sceconf.sh --op=list
IP                Rate                Peak
-----
10.1.6.93         0.71798986         0.718
10.1.9.36         0.14420895         0.1442139
10.1.9.35         0.0                 0.027929332
10.1.12.11       0.0                 0.0
```

---

## Verifying Server Operation

To verify that the server is functioning correctly, use the **alive.sh** script:

```
~scmscm/setup/alive.sh
```

The script verifies that the following components are operational:

- Collection Manager.
- Database (in the bundled database installation).
- Report tables (in the bundled database installation).

If any component is down, the script issues an error message.

---

**Step 1** As the scmscm user, run the **alive.sh** script



**Note**

It takes time for the components to initialize after a startup. After a restart, wait 5 minutes before running this script.

---

## Viewing Database Statistics

To view the statistics of the CM database, use the p3stats command line utility (CLU).

```
~scmscm/cm/bin/p3stats [options]
```

Table 4-8 lists the p3stats options.

**Table 4-8** *p3stats Options*

Option	Description
<code>--show-stats</code>	Displays statistics information in the format <DB RDR tag> <RDR rate> <peak RDR rate>
<code>--reset-stats</code>	Resets the statistics counters.



**Note**

The CM must be up for this CLU to operate.

---

The following example shows how to display the statistics of the CM database:

```
$ ~scmscm/cm/bin/p3stats --show-stats
LUR { rate=0, peak=0 }
MALUR { rate=0, peak=0 }
TR { rate=0, peak=0 }
VLUR { rate=0, peak=0 }
NUR { rate=0, peak=0 }
SUR { rate=0, peak=0 }
MEDIA { rate=0, peak=0 }
PUR { rate=0, peak=0 }
Command terminated successfully
$
```

## Viewing Version Information

The Collection Manager includes two scripts to display the current version of the Collection Manager and of the database:

- `cm version`
- `cm dbversion`

The following example shows how to view the current Collection Manager version:

```
$ ~scmscm/cm/bin/cm version  
CM CD Version 3.5.5 Build 336
```

The following example shows how to view the current database version:

```
$ ~scmscm/cm/bin/cm dbversion  
Sybase ase150/
```

