Managing Collection Manager

This chapter explains how to use utility scripts to view and update Collection Manager parameters.

Any machine connected to Collection Manager through, for example, Telnet or Secure Shell (SSH), can use utility scripts to monitor and manage Collection Manager. The utility scripts are located in the installation directory of Collection Manager.

For information on managing the database and the CSV repository, see Chapter 5, "Managing Databases and the Comma-Separated Value Repository."

This chapter contains the following sections:

Using Utility Scripts

General instructions for using utility scripts include:

- To invoke any script, log in as the scmscm user, except when otherwise specified. If you attempt to run these scripts as the root user, the script generates an error.

- To display a description of the script, with an explanation of all the flags and parameters, invoke the script with the help flag.

> **Note** Specify the help flag differently depending on the script. To manage Collection Manager, specify -help. To manage the database, specify -h.

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

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

Collection Manager Support Information

If you are experiencing difficulties with Collection Manager installation, you might need to provide the Cisco Technical Assistance Center (TAC) with information about the current system setup. Collection Manager 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 generates the Collection Manager 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 results from running the 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 Collection Manager

You can use both scripts and files to configure Collection Manager. The following sections provide details on these options:

- Using Scripts to Configure the Collection Manager, page 4-3
- Using Files to Configure the Collection Manager, page 4-6

Using Scripts to Configure the Collection Manager

Use utility scripts to:

- Specify which servers to activate at startup.
- Start or stop the database.
- Start or stop an adapter.
- Drop an SCE connection.

Use the following scripts to configure Collection Manager:

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

Note For information about the scripts used for managing the database and the CSV repository, see Chapter 5, "Managing Databases and the Comma-Separated Value Repository."

The below sections detail the scripts used to configure the Collection Manager.

- Activating Servers, page 4-3
- Controlling Adapters, page 4-4
- Enabling an Adapter, page 4-4
- Dropping a Cisco SCE Connection, page 4-5
- Configuring a Database, page 4-5
- Configuring the Timezone Offset, page 4-6

Activating Servers

To specify the servers (Collection Manager) that are activated at startup, use the **on-boot.sh** script:

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

Table 4-1 defines the **on-boot.sh** script parameters:

Table 4-1   on-boot.sh Parameters

`--cm={ on │ off }`	Activate or do not activate Collection Manager at startup.

The following example shows how to set Collection Manager servers to run at startup. (This setting is the default setting of the script.)

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

To view the current startup status of each component, run the `on-boot.sh` script with no parameters.

Server specification take effect when the system restarts. To restart Collection Manager, run the following script as the scmscm user:

`~scmscm/cm/bin/cm restart`

Controlling Adapters

To shut down or activate a configured adapter, use the **adapterconf.sh** script. You can use the **adapterconf.sh** script to list Collection Manager adapters that are currently running:

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

Table 4-2 shows the **adapterconf.sh** script options:

Table 4-2   adapterconf.sh Options

`--op=start`	Activate the adapter specified by the adapter parameter.
`--op=stop`	Shut down the adapter specified by the adapter parameter.
`--op=list`	List Collection Manager adapters that are currently running.
`adapter=adapter name`	Identify the adapter you want to control. Use only with the start and stop options.
`--help`	Display these options.

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

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

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

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

Enabling an Adapter

To configure an adapter to turn on when Collection Manager starts, remove 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 Collection Manager starts.

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

The following example shows how to configure the CSV adapter to remain off when Collection Manager starts:

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

Note The value of the adapter.<number> must match the adapter_id parameter value configured in the queue.conf file for the corresponding adapter.

Dropping a Cisco SCE Connection

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

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

Run this script only if the HTTP adapter of Collection Manager is running.

Also use this script to display information about the Cisco SCE connection. (See the "Checking the SCE Connection" section on page 4-17)

Table 4-3 shows the **sceconf.sh** script options:

Table 4-3   sceconf.sh Options

Adapter=*IP address*	Drop the connection at the specified IP address.
--help	Display the seconf.sh options.

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

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

Configuring a Database

To configure a database, run 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 run the dbconf.sh script to configure Collection Manager to work with an external Oracle database:

```
$ ~scmscm/scripts/dbconf.sh
Enter the DB type:
1 - Oracle
2 - MySQL
3 - Sybase
4 - Insight
5 - exit
Enter your choice: 1
Enter Oracle server host (current is localhost) :
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) :
```

Note   Restart Collection Manager for the configuration to take effect.

Configuring the Timezone Offset

You can configure the timezone offset manually or automatically.

To configure the timezone offset (from GMT) manually, issue the following command:

~scmscm/cm/bin/jselect-sce-tz.sh --offset=offset-in-minutes

Note   Enter the offset in minutes.

For example, if the SCE is located in GMT+2, issue the following command:

```
~scmscm/cm/bin/jselect-sce-tz.sh --offset=120
```

For example, if the SCE is located in GMT-10, issue the following command:

```
~scmscm/cm/bin/jselect-sce-tz.sh --offset=-600
```

To configure the timezone offset (from GMT) automatically, issue the following command:

```
~scmscm/cm/bin/updatetimezone.sh
```

The updatetimezone.sh script calculates the offset in minutes and updates the CONF_SE_TZ_OFFSET table.

Note   Run the updatetimezone.sh script every time the timezone of the Cisco SCE is changed. For example, run this script during Daylight saving start and Daylight saving end.

Using Files to Configure the Collection Manager

Use the following files to configure Collection Manager:

- **cm.conf**—General configuration of Collection Manager, including the adapters to turn on when Collection Manager starts. See the "Enabling an Adapter" section on page 4-4.

- **queue.conf**—Configuration of the adapter queues, including the RDR tags that are associated with a specific adapter. See the "Configuring the Categorizer" section on page 4-7.

Configuring the Categorizer

The Categorizer classifies each RDR according to its RDR tag. You can route an RDR 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 shows how to configure the RDR tags 4042321920 and 4042321922 to be sent to the topper/aggregator (TA) 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

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

Use the following scripts to monitor Collection Manager:

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

Installing the Periodic Checker

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

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

Table 4-4 shows the setup-monitor.sh script options.

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

-a { **install** | **uninstall** }	Insert or remove an entry for monitor.sh in the cron.
-i { **30m** | **1h** | **12h** | **24h** }	Run monitor.sh every 30 minutes, 1 hour, 12 hours, or 24 hours.

The following example shows how to install **monitor.sh** so that it runs 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

The following sections provide information on the periodic checker script:

- Periodic Checker Script, page 4-8
- Tests, page 4-9

Periodic Checker Script

The periodic checker script, **monitor.sh**, calls a series of subscripts 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. The script sends test results to the syslog subsystem and logs the results in the /var/log/messages file.

Table 4-5 shows the smonitor.sh script options.

Table 4-5   *monitor.sh Options*

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

Any test 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 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

You can run the following tests using **monitor.sh**:

- db_up—Checks whether the Collection Manager database is running.

- cm_up—Checks whether the Collection Manager application is running.

- cm_persistent_buffers—Checks whether the persistent buffer of each Collection Manager adapter 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

Collection Manager uses the p3rpc utility to manage users for authenticated RPC calls.

The command format is: p3rpc OPERATION [OPTIONS]

Table Table 4-6 lists the p3rpc operations and options.

Table 4-6   p3rpc Operations

Operation	Description
`--set-user --username`=*username* `--password`=*password*	Adds and updates the username and password.
`--validate-password --username`=*username* `--password`=*password*	Validates the username and password.
`--delete-user --username`=*username*	Deletes a user configuration.
`--show-users`	Displays all the configured users.

Examples

The following example shows how to add and update a 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 a 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 a 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, clu, exists:

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

Managing Virtual Links

A script is included in Collection Manager distribution to enable you to manage virtual link names and indexes that are configured for a specific SCE:

To show or set virtual links, run the **update_vlinks.sh** script:

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

Table 4-7 shows the **update_vlinks.sh** script options.

Table 4-7   update_vlinks.sh Options

--sce=*SCE IP* **--file**=*file*	Update the VLINK_INI table with the data in the supplied CSV formatted file for the specified SCE.
--sce=*SCE IP* **--show**	Query the VLINK_INI table for entries for the specified SCE.
--help	Display the above 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 script performs following validations on the file:

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

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

Step 1 The input CSVfile must contain three data fields (VLINK_ID, VLINK_DIRECTION, VLINK_NAME) or six data fields (VLINK_ID, VLINK_DIRECTION, VLINK_NAME, CHANNEL_ID, CHANNEL_NAME, CMTS_NAME).

Step 2 If the CSV file contains three data fields, the VLINK_INI table automatically fills the data for the rest of the columns.

Step 3 All entries in the table that match the SCE_IP_Address and CHANNEL_ID, VLINK_DIRECTION, and CHANNEL_NAME are deleted from the table.

Step 4 The script adds two entries to the VLINK_INI table in the following format:

- Timestamp, sce ip, 0, 0, "Default Virtual Link Up,0, 0, "Default Virtual Link Up"
- Timestamp, sce ip, 0, 1, "Default Virtual Link Down, 0, 1, "Default Virtual Link Down"

Step 5   The script parses the CSV file and each line in the CSV file is entered as a record entry in the VLINK_INI table.

Step 6   If the CSV file is empty, the script inserts default records into the VLINK_INI table.

Managing IMEI or MEID TAC with Device Types

Collection Manager supports either GSM or CDMA mobile-based reports. GSM reports are enabled by default. Once Collection Manager is installed, the CSV file ~scmscm/cm/config/imeitac_device_map.csv populates the IMEI_DEVICETYPE table with default values.

The MEID_DEVICETYPE table stores the CDMA Device Type mapping information in the database. You manually execute the **~scmscm/scripts/update_devicetype.sh** script to update the table. The default input mapping file is **~scmscm/cm/config/meidtac_device_map.csv**:

~scmscm/scripts/update_devicetype.sh

Table 4-8 shows the *update_devicetype.sh Options*

Table 4-8   update_devicetype.sh Options

`update_devicetype.sh --imei  --r`	Removes all the existing records from the IMEI_DEVICETYPE Table.
`update_devicetype.sh --meid  --r`	Removes all the existing records from the MEID_DEVICETYPE Table.
`update_devicetype.sh --imei  --u`	Deletes all the existing records and then updates the IMEI_DEVICETYPE table with the data given in the default CSV file.
`update_devicetype.sh --meid  --u`	Deletes all the existing records and then updates the MEID_DEVICETYPE table with the data given in the default CSV file.
`update_devicetype.sh --h`	Prints the usage messsage.

Update the IMEI_DEVICETYPE table

To update the IMEI_DEVICETYPE table, add new TAC details to the CSV file ~scmscm/cm/config/imeitac_device_map.csv in following the format:

Manufacturer <string>, Model <string>, TAC <number>.

Next, run the /update_devicetype.sh script using the --u option.

~scmscm/scripts/update_devicetype.sh --imei --u

This deletes all the existing records, then updates the IMEI_DEVICETYPE table with the data given in the default CSV file.

The update_devicetype.sh script performs the following actions:

* The script parses the CSV file ~scmscm/cm/config/imeitac_device_map.csv, and each line in the CSV file is entered as a record entry in the IMEI_DEVICETYPE table.

- If the TAC number is a positive value, records are considered valid. Otherwise, the records are ignored.
- If the script finds duplicate TAC entries, the first TAC entry is considered valid while the other entries are ignored.
- If the CSV file is empty, the script deletes all the existing records in the IMEI_DEVICETYPE table.

The default ~scmscm/cm/config/imeitac_device_map.csv file information is overwritten while restarting the Collection Manager. If multiple Collection Managers are in use and if they are configured with a centralized database, replace the updated INI file in all Collection Managers when there is a change in the default file.

Update the MEID_DEVICETYPE Table

To update the MEID_DEVICETYPE table:

Step 1   Add the new TAC details to the ~scmscm/cm/config/meidtac_device_map.csv file in the format

Manufacturer <string>, TAC <number>.

Step 2   Run the **update_devicetype.sh** script using the **--u**.

~scmscm/scripts/update_devicetype.sh --meid   --u

This deletes all the existing records then updates the MEID_DEVICETYPE table with the data given in the default CSV file.

 The update_devicetype.sh script performs the following actions:

- The script parses the CSV file **~scmscm/cm/config/meidtac_device_map.csv** and each line in the CSV file is entered as a record entry in the MEID_DEVICETYPE table.
- If the TAC number is a positive value, records are considered valid. Otherwise, the records are ignored.
- If the script finds duplicate TAC entries, the first TAC entry is considered valid while the other entries are ignored.
- If the CSV file is empty, the script deletes all the existing records in the MEID_DEVICETYPE table.

The default ~scmscm/cm/config/meidtac_device_map.csv file information is overwritten while restarting the Collection Manager. If multiple Collection Managers are in use and if they are configured with a centralized database, replace the updated INI file in all Collection Managers when there is a change in the default file.

Update the OSFP_TYPES Table

To update the OSFP_TYPES table:

Step 1   Add the new Operating System details to the ~scmscm/cm/config/osfp_types_map.csv file in the format **OS id <number>,OS name <string>**.

Step 2   Run the update_osfptypes.sh script using --update.

~scmscm/scripts/update_osfptypes.sh --update

This deletes all the existing records then updates the OSFP_TYPES table with the data given in the default CSV file.

The update_osfptypes.sh script performs the following actions:

- The script parses the CSV file ~scmscm/cm/config/osfp_types_map.csv and each line in the CSV file is entered as a record entry in the OSFP_TYPES table.

- If the CSV file is empty, the script deletes all the existing records in the OSFP_TYPES table.

The default ~scmscm/cm/config/osfp_types_map.csv file information is overwritten while restarting the Collection Manager. If multiple Collection Managers are in use and if they are configured with a centralized database, replace the updated INI file in all Collection Managers when there is a change in the default file.

Update the VIDEO_MONITOR_TYPES Table

To update the VIDEO_MONITOR_TYPES table:

Step 1 Add the new video metadata mapping details (Resolution/BitRate/Duration) to the ~scmscm/cm/config/video_monitor_type.csv file in the following format:

Mapping ID *<number>*, **Description** *<String>*,**Device Type** *<number>*,**Minimum Starting Range**<*number*>,**Maximum Ending Range**<*number*>

Step 2 Run the update_videotypes.sh script using --update.

~scmscm/scripts/update_videotypes.sh --update

This deletes all the existing records then updates the VIDEO_MONITOR_TYPES table with the data given in the default CSV file.

The update_videotypes.sh script performs the following actions:

- The script parses the CSV file ~scmscm/cm/config/video_monitor_type.csv and each line in the CSV file is entered as a record entry in the VIDEO_MONITOR_TYPES table.

- If the CSV file is empty, the script deletes all the existing records in the VIDEO_MONITOR_TYPES table.

The default ~scmscm/cm/config/video_monitor_type.csv file information is overwritten while restarting the Collection Manager. If multiple Collection Managers are in use and if they are configured with a centralized database, replace the updated INI file in all Collection Managers when there is a change in the default file.

Update the DEVICE_TYPES Table

To update the DEVICE_TYPES table:

Step 1 Add the new device details to the ~scmscm/cm/config/device_count_ini.csv file in the format: **Device Id**<*number*>,**Device Pattern**<*String*>, **Device Desc**<*String*>, **Device Type**[*0* - Mobile, *1*-PC]<**number**>

Step 2 Run the device_count_ini.sh script using --update.

~scmscm/scripts/device_count_ini.sh --update

This deletes all the existing records then updates the DEVICE_TYPES table with the data given in the default CSV file.

The device_count_ini.sh script performs the following actions:

- The script parses the CSV file ~scmscm/cm/config/device_count_ini.csv and each line in the CSV file is entered as a record entry in the DEVICE_TYPES table.
- If the CSV file is empty, the script deletes all the existing records in the DEVICE_TYPES table.

The default ~scmscm/cm/config/device_count_ini.csv file information is overwritten while restarting the Collection Manager. If multiple Collection Managers are in use and if they are configured with a centralized database, replace the updated INI file in all Collection Managers when there is a change in the default file.

Managing the Peak Hour Configuration

You need to execute the **~scmscm/scripts/updatepeakconfig.sh** script before generating the peak hour reports. This script populates the peak_hour_config table with the default peak hour details. By default, the peak_hour_config table is not created.

The updatepeakconfig.sh script then prompts you to enter the peak hour range for any days you want to change from the default. If the value is not entered for the corresponding day, the current value shown in the prompt is considered the default value.

Note Execute the ~scmscm/scripts/updatepeakconfig.sh script once with default values before generating any peak hour reports.

The following output is an example of a Peak Hours configuration:

```
-bash-2.05b$ ~scmscm/scripts/updatepeakconfig.sh
Enter the Peak Hour Range for Monday (current is 18-21) :
Enter the Peak Hour Range for Tuesday (current is 18-21) :15-16
Enter the Peak Hour Range for Wednesday (current is 18-21) :15
Enter the Peak Hour Range for Thursday (current is 18-21) :
Enter the Peak Hour Range for Friday (current is 18-21) :
Enter the Peak Hour Range for Saturday (current is 16-19) :
Enter the Peak Hour Range for Sunday (current is 16-19) :
Peak Configurations successfully updated!!!
!!Restart CM for the new config to take effect!!
```

Monitoring Collection Manager

You can use scripts to monitor system statistics that are relevant to Collection Manager, such as:

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

Use the following scripts to monitor Collection Manager:

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

Use the following scripts to configure Collection Manager (see the "Configuring Collection Manager" section on page 4-3). You can also run these scripts to display the relevant configuration:

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

Checking the Database Capacity

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

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

Use the script only with MySQL database.

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

Checking the RDR Rate

To display the momentary total rate of reports entering Collection Manager, run the **rdr-rate.sh** script

```
rdr-rate.sh
```

The output is a single floating-point number that represents the total rate per second of incoming RDRs (from all sources), which entered Collection Manager in the past 5 seconds.

Use this script only if the HTTP adapter of Collection Manager is running.

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

Checking the SCE Connection

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

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

Use this script only if the HTTP Adapter of Collection Manager is running.

Also use the script to drop a connection from a particular SCE. See the "Dropping a Cisco SCE Connection" section on 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, run the **alive.sh** script:

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

The scripts verifies that the Collection Manager is operational. If it 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 Collection Manager database, run the p3stats command line utility (CLU).

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

Table 4-9 lists the p3stats options.

Table 4-9   p3stats Options

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

Note   Collection Manager must be running for this CLU to operate.

The following example shows how to display the statistics of Collection Manager 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

Collection Manager includes two scripts to display the current version of 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 5.0.0 Build 100
```

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

```
$ ~scmscm/cm/bin/cm dbversion
CM: Database
Sybase  ase 15.0.3
```

Integrating Content Portal Authority Client

Content Portal Authority (CPA) client is used for content filtering that involves classification and control of HTTP flows based on requested URLs. CPA client queries categories for the extracted HTTP URLs from the traffic and sends it to the Cisco SCE application. Category ID is used for classifying the original HTTP flow to a service.

The CPA client in Cisco Service Control Collection Manager helps in reducing the CPU utilization of the Cisco SCE and increases the response rate of the CPA server.

The Custom Adapter in Cisco Service Control Collection Manager is configured to process a maximum CPA RDRs (Tag ID 3000000001) from one or more Cisco SCEs at a time.

By default, CPA client process is disabled in Cisco Service Control Collection Manager.

You may enable the CPA client to achieve higher processing rate or to reduce the CPU utilization of Cisco SCEs.

Enabling Content Portal Authority Client

Step 1 Confirm that the Custom Adapter is enabled.

Step 2 Add the CPA client Tag ID (3000000001) to process to the custom adapter section in the ~scmscm/cm/config/queue.conf file.

Step 3 Add the Tag ID in the property tags defined under the [bill-custom] section.

Step 4 Modify the cpa configuration section in the ~scmscm/cm/config/customadapter.conf file to control the CPA client process in Cisco Service Control Collection Manager.

```
[cpa]
active=false
sce_ip=1.1.1.1,2.2.2.2,3.3.3.3
cpa_server=4.4.4.4
sce_user_pwd=username:password,username:password
```

Note Enable either the Cisco SCE or Cisco Service Control Collection Manager CPA Client to process CPA RDRs. Do not enable CPA Client on SCE and Collection Manager simultaneously.

Video Quality of Experience Reporting

Note Effective with Release 4.1.0, Cisco Service Control Collection Manager supports Video Quality of Experience Reporting. Video QoE Reporting feature is at a nascent stage and will evolve to a complete feature in the future releases of Cisco Service Control Collection Manager. Please contact the Cisco SCE Marketing for further details and assistance on Video QoE feature.

Cisco Service Control Collection Manager, Release 4.1.0 and later, supports YouTube and DailyMotion videos.

Cisco Service Control Solution determines the Video Quality of Experience (QoE) for a user by comparing the actual duration of the video with the time taken for the video traffic to pass through the Cisco SCE. Cisco SCE determines the Quality of videos delivered over HTTP using the metadata in the video flows along with flow duration, volume, and number of sessions. If the video is delivered through several flows, all these flows are aggregated to determine the quality.

The RAGAdapter aggregates the YouTube and DailyMotion videos at package level and calculates the QoE rating on a daily basis.

Video Monitoring

Note   Effective with Release 4.1.0, Cisco Service Control Collection Manager supports Video Monitoring feature. Video Monitoring feature is at a nascent stage and will evolve to a complete feature in the future releases of Cisco Service Control Collection Manager. Please contact the Cisco SCE Marketing for further details and assistance on Video Monitoring feature.

Enhanced Video Traffic Analysis feature enables you to generate video related reports based on the resolution, length, and bit rate of video flows. Each report provides overall consumed video volumes and the number of unique videos watched by each subscriber for different ranges of resolution/bit rate/duration on a daily basis.

Cisco Service Control Collection Manager, Release 4.1.0 and later, supports only videos originating from the following Video content service providers:

- YouTube
- Dailymotion

If Video Monitoring is enabled, Cisco Service Control Collection Manager processes the incoming Video Transaction Usage RDRs. If the video originates from the supported video providers, the Collection Manager populates new tables with the title, duration, encoding rate, and resolution information. Reports are generated on a daily basis.

Cisco SCE identifies the flows that belong to each video. The Cisco Service Control Collection Manager aggregates the resolution, bit rate, and bandwidth for each video and generates statistical reports by taking a percentage distribution of the following parameters:

- Short/Medium/Long Length Videos
- High/Medium/Low Video Resolution
- High/Medium/Low Video Bit rates

Aggregating Video Monitoring Parameters

The following aggregation methods are used for video monitoring:

- Resolution Based Aggregation
- Bit Rate Based Aggregation
- Duration Based Aggregation

Resolution Based Aggregation

Resolution based aggregation reports information such as Total Volume Consumed and Unique Videos watched by the Subscriber for a defined range of resolution.

The resolutions of the video may vary across the flows of a video. Since each video has various resolutions associated with it, the resolution that corresponds to the Flow that has the highest flow duration is considered as a resolution of the video.

A set of standard resolutions that are populated in each Video flows which includes 1080p, 720p, 480p, 360p, 240p, 144pixels. This video resolution is Mapped with the ranges defined with VIDEO_MONITOR_TYPE=1 which is defined in the ~scmscm/cm/config/video_monitor_type.csv Configuration file and the Collection Manager fetches the Corresponding VIDEO_MONITOR_ID.

CM RAG Adapter uses this VIDEO_MONITOR_ID and aggregates the incoming Video TURs based on "Per Package per Service per resolution range".

Bit Rate Based Aggregation

Each video flow may have varying bit rate associated with it. An average of the valid non-zero bit rate of the flows belong to the same video is taken as video bit rate. This video Bit Rate is Mapped with the ranges defined with VIDEO_MONITOR_TYPE=2 defined in the ~scmscm/cm/config/video_monitor_type.csv Configuration file and the Collection Manager fetches the Corresponding VIDEO_MONITOR_ID.

The RAG Adapter uses this VIDEO_MONITOR_ID and aggregates the incoming Video TURs based on "Per Package per Service per Video Bit rate range".

Duration Based Aggregation

Same video flows may have slightly varying duration across the flows or may have single valid non zero duration. First non zero Duration of the flow that belongs to the same video is taken as the video duration. This video Duration is Mapped with the ranges defined with VIDEO_MONITOR_TYPE=3 defined in the ~scmscm/cm/config/video_monitor_type.csv Configuration file and the Collection Manager fetches the Corresponding VIDEO_MONITOR_ID. The RAG Adapter uses this VIDEO_MONITOR_ID and aggregates the incoming Video TURs based on "Per Package per Service per Video duration range".

Enabling Video Monitoring and Video QoE

Before you begin:

Configure the Cisco SCE using the tunables and configure the Cisco SCA BB.

Setting the Tunables

To enable video monitoring reporting, set the value of the GT_PL_INSPECT_HTTP_RESPONSE_FOR_VIDEO_REPORTING tunable to true.

```
SCE8000(config)#> tunable GT_PL_INSPECT_HTTP_RESPONSE_FOR_VIDEO_REPORTING value true
```

To include YouTube and Dailymotion services, set the value of the GT_PL_MAXIMAL_INSPECTED_HTTP_TRANSACTIONS_PER_FLOW_FOR_YOUTUBE and GT_PL_MAXIMAL_INSPECTED_HTTP_TRANSACTIONS_PER_FLOW_FOR_DAILYMOTION tunables value to 3.

```
SCE8000(config)#> tunable GT_PL_MAXIMAL_INSPECTED_HTTP_TRANSACTIONS_PER_FLOW_FOR_YOUTUBE
value 3
SCE8000(config)#> tunable
GT_PL_MAXIMAL_INSPECTED_HTTP_TRANSACTIONS_PER_FLOW_FOR_DAILYMOTION value 3
```

Configuring the Cisco SCA BB

- On the Cisco SCA BB console, enable Protocol Specific TURs:

  From **Configuration** > **Classification** > **RDR Settings** > **Transaction Usage RDRs**, select **Protocol specific TURs (include protocol specific information)**.

- On the Cisco SCA BB console, enable Video Transaction Usage RDR:

From **Configurations > Policies > System settings > Advanced Options > Advanced Service Configuration Options**, set the value of the Video Transaction Usage RDR to true. The default value is false.

Enabling Video Monitoring

By default, aggregation is disabled on Cisco Service Control Collection Manager.

To enable or disable Video Monitoring modify the enable_video_monitor parameter under the [config] section of ~scmscm/cm/config/ragadapter.conf configuration file.

Enabling Video QoE

To enable or disable Video QoE modify the enable_video_qoe parameter under the [config] section of ~scmscm/cm/config/ragadapter.conf configuration file.

If the feature is enabled, the video_monitor_qoe.xml file in the ragadapter/repository directory is copied to the ragadapter directory to process the Video TUR.

Device Counting

Device counting feature enables you to identify the mobile devices used in a network during the monitoring period and report the device count and its usage for each mobile device type configured in the Cisco Service Control Collection Manager.

The user agent field of the HTTP Transaction Usage RDR is parsed to identify the device name used by the subscriber. The Device ID mapped to the corresponding device name is used to aggregate the details.

The RPT_DEVICE_COUNT table is updated with the aggregated upstream volume, aggregated downstream volume, and unique subscribers count for each combination of packages and device ID every hour.

Enabling Device Counting Feature

Device Counting is disabled by default.

To enable device counting, change the value of the *device_counting* parameter to *true* in the config section of the RAG Adapter configuration file (~scmscm/config/ragadapter.conf) and restart the Cisco Service Control Collection Manager.

Adding Mobile Devices for Identification

To identify the mobile devices, configure the model name in the device-count.conf configuration file.

If the device name in the user agent matches with the configured device names in the device-count.conf file then the corresponding device ID is used for aggregation.

Device is categorized under *other devices*, if the device name is not available in the user agent or if the device name available in the user agent is not available in the configuration file.

Adding New Devices to the Configuration

To add a new device model:

Step 1 Configure the description and the model name in the ~scmscm/cm/config/device-count.conf configuration file in the format <Device Name to be reported>=<Device model name> | <Device model name>.

Example

```
HTC=HTC Sensation|HTC One
```

Step 2 Add the newly configured device name in the ~scmscm/cm/config/ device_count_ini.csv file in the format, Device ID, Device model name,Device description, Device type.

Where

- Device ID is a unique number representing a device model. Increment this number for each new model.

- Device model name is the device model name used in the configuration while adding it in the device-count.conf file.

- Device description is the description of the newly added device to be displayed in the report.

- Device Type is the type of the device. For mobile devices, the value is 0.

While adding the new device in the ~scmscm/cm/config/ device_count_ini.csv file make sure that the *other devices* entry remains as the last entry.

```
#Device Id,Device Pattern, Device Desc, Device Type[0 - Mobile, 1- PC]
101,Iphone,Iphone Devices,0
102,Ipad,Ipad Devices,0
103,Blackberry,Blackberry Devices,0
104,SamsungGalaxy,Samsung Galaxy Devices,0
105,Samsung,Samsung Devices,0
106,NokiaLumia,Nokia Lumia Devices,0
107,Nokia,Nokia Devices,0
108,GoogleNexus,Google Nexus Devices,0
109,HTC,HTC Devices,0
110,SonyEricsson,SonyEricsson Devices,0
111,LG,LG Devices,0
112,Macintosh,Apple iMac and MacBooks,1
113,Ipod,Apple iPod,0
114,Motorola,Motorola devices,0
115,SonyXperia, Sony Xperia Devices, 0
116,Nintendo,Nintendo Devices,0
117,Xbox,Microsoft Xbox Devices,0
118,PSP,Sony Playstation Portable,0
119,Playstation,Sony Playstation devices,0
120,Kindle,Kindle,0
121,Silk-Accelerated,Kindle Fire Silk,0
122,WindowsPC,Windows PCs,1
123,PC,Generic PC,1
200,Others,Other Devices,0
```

The values from ~/cm/config/device_count_ini.csv is populated to the DEVICE_TYPES table automatically every time Cisco Service Control Collection Manager starts. Use the ~/scripts/devices_count_ini.sh -update script to populate the values manually.

The default INI file information is overwritten while restarting the Collection Manager. If multiple Collection Managers are in use and if they are configured with a centralized database, replace the updated INI file in all Collection Managers when there is a change in the default file.