



CHAPTER 2

Collection Manager Overview

This module provides detailed information about the functionality of collection manager components.

This module describes how the Cisco Service Control Management Suite (SCMS) Collection Manager (CM) works. It describes the Raw Data Records (RDRs) that the Service Control Engine (SCE) platforms produce and send to the Collection Manager, and provides an overview of the components of the CM software package. It also gives an overview of the database used to store the RDRs.

- [Data Collection Process, page 2-1](#)
- [Raw Data Records, page 2-1](#)
- [Collection Manager Software Package, page 2-3](#)
- [Adapter Description, page 2-3](#)
- [Using Databases, page 2-7](#)

Data Collection Process

SCE platform, such as the Cisco Service Control Application for Broadband (SCA BB).

1. RDRs are streamed from the SCE platform using the simple, reliable *RDR-Protocol*
2. After the CM receives the RDRs from the SCE platforms, CM software modules recognize and sort the various types of RDR, based on preset categories and according to type and priority, and queue them in persistent buffers.
3. One or more of the CM adapters processes each RDR. Each adapter performs a specific function on RDRs (stores it in a comma separated value (CSV) formatted file on a local machine, sends it to an RDBMS application, or performs custom operations).

You can use preinstalled utility scripts to customize many of the parameters that influence the behavior of the CM.

Raw Data Records

Raw Data Records (RDRs) are reports produced by SCE platforms. The list of RDRs, their fields, and their semantics depend on the specific service control protocol (SCP) application. Each RDR type has a unique ID known as an RDR tag.

contains examples of RDRs produced by SCP applications:

Table 2-1 Example RDRs Produced by SCP Applications

Report Type	Description	RDRs contain...
	<p>devices; they can report usage records per subscriber.</p>	<p>These RDRs typically contain a subscriber identifier (such as the OSS subscriber ID), the traffic type (such as HTTP, streaming, or peer-to-peer traffic), and usage counters (such as total upstream and downstream volume). These types of usage reports are necessary for usage-based billing services, and for network analysis and capacity planning.</p> <p>The SCA BB application Subscriber Usage RDRs are in this category.</p>
Transaction level report	<p>SCE platforms perform stateful tracking of each network transaction conducted on the links on which they are situated. Using this statefulness, the SCP tracks several OSI Layer 7 protocols (such as HTTP, RTSP, SIP, or Gnutella) to report on various application level attributes.</p>	<p>These RDRs typically contain transaction-level parameters ranging from basic Layer 3-4 attributes (such as source IP, destination IP, and port number) to protocol-dependant Layer 7 attributes (such as user-agent, hostname for HTTP, or e-mail address of an SMTP mail sender), and also generic parameters (such as time of day and transaction duration). These RDRs are important for content-based billing schemes and for detailed usage statistics.</p> <p>The SCA BB application Transaction RDRs are in this category.</p>
SCP application activity reports	<p>The SCP application can program the SCE platform to perform various actions on network traffic. These actions include blocking transactions, shaping traffic to certain rates and limits, and performing application-level redirections. When such an operation is performed, the SCP application may produce an RDR.</p>	<p>The SCA BB application Breaching RDRs and Blocking RDRs are in this category. Breaching RDRs are generated when the system changes its active enforcement on a subscriber (because usage exceeded a certain quota). Blocking RDRs are generated when an SCE platform blocks a network transaction (according to rules contained in the current service configuration).</p>

Collection Manager Software Package

-
-
- [Priority Queues and Persistent Buffers, page 2-3](#)

Raw Data Record Server

Categorizer

Priority Queues and Persistent Buffers

Adapter Description

Adapters are software modules that transform RDRs to match the target system's requirements, and distribute the RDRs upon request. At this time, the following adapters are shipped with the system:

- [JDBC Adapter, page 2-4](#)
- [Comma Separated Value Adapter, page 2-4](#)
- [Topper/Aggregator Adapter, page 2-4](#)
- [Real-Time Aggregating Adapter, page 2-6](#)
- [HTTTPC Adapter, page 2-7](#)

Some of the adapters send data to the database or write it to CSV files. The structures of the database tables, and the location and structures of these CSV files are described in the [Cisco Service Control Application for Broadband Reference Guide](#)

Each adapter has its own configuration file; all the configuration files are similar in structure. For a sample RAG adapter configuration file, see [ragadapter.conf File, page A-4](#).

JDBC Adapter

- Sybase Adaptive Server Enterprise (ASE) 12.5 and 15.0
- Oracle 9.2 and 10


Note

The *recycle bin* feature available in Oracle 10 should be disabled. You can set the initial value of the `recyclebin` parameter in the `init<SID>.ora` file:

```
recyclebin=off
```

Comma Separated Value Adapter

Topper/Aggregator Adapter

and outputs 'Top Reports' to the database and aggregated daily statistics of all subscribers (not just the top consumers) to CSV files. Top Reports are lists of the top subscribers for different metrics (for example, the top 50 volume or session consumers in the last hour).

This adapter maintains a persistent saved state (saved to disk) to minimize any data loss in case of failure.

The TA adapter, which uses the JDBC adapter infrastructure, can be configured to use any JDBC-compliant database, either locally or remotely.

- [TA Adapter Cycles, page 2-4](#)
- [TA Adapter Memory Requirements, page 2-5](#)

TA Adapter Cycles

The TA Adapter works in two cycles: short and long. Cycles are fixed intervals at the end of which the adapter can output its aggregated information to the database and to a CSV file. The default interval for the short cycle is 1 hour; for the long cycle it is 24 hours (every day at midnight). The intervals (defined in minutes) and their start and end times are configurable.


Note

The long-cycle interval must be a multiple of the short-cycle interval.

The activities in each cycle differ slightly, as follows:

Short Cycle—At the end of each short cycle, the adapter:

- Adds the cycle’s aggregated Top Reports to the short cycle database table
- Saves the current state file in case of power failure

Long Cycle—At the end of each long cycle, the adapter:

- Adds the cycle’s aggregated Top Reports to the long cycle database table
- Saves the current state file in case of power failure
- Creates a CSV file containing the aggregated statistics for the long-cycle period

TA Adapter Memory Requirements

To work correctly, you must dedicate a sufficient amount of memory to the TA adapter. Configure the value in the **cm.conf**

```
[adapter_mem]
com.cisco.scm.scm.adapters.topper.TAAdapter=<Memory for TA Adapter>
```

Memory (Bytes) = 2.5 * NUM_SUBSCRIBERS * (AVG_SUBS_ID_LENGTH + 64*NUM_SERVICES + 12*NUM_TOP_ENTRIES)

Where:

NUM_SUBSCRIBERS is the number of new subscribers that will be introduced in one day (on all SCEs sending reports to this CM).

This is usually a high number; especially when working in anonymous subscriber mode.

To display an estimate of the number of subscribers that are known to the CM, use the following command:

```
~/setup/mbean.py --getattr=Subscribers DCAdapters
```

AVG_SUBS_ID_LENGTH is the average character length of a subscriber.

In most cases, this is approximately 20.

NUM_SERVICES is the number of subscriber usage counters and is configured in the **taadapter.conf**

```
num_top_entries
```

taadapter.conf



not

Real-Time Aggregating Adapter

-
-

RAG Adapter Aggregation Buckets

Example:

```
Bucket-identifying field = field number 3
Closures: 4 = 4,5,6; 10 = 8,10,11
Value in field 3 = 4, 5, or 6; field reported as 4
Value in field 3 = 8, 10, or 11; field reported as 10
```

Buckets, closures, triggers, and trigger actions are defined in an XML file. For a sample XML file, see [ragadapter.xml File, page A-5](#).

Flushing a Bucket

checkpoint

-
-
-
-
-
-
-
-

HTTPC Adapter

-
-
-

Using Databases

-
- [, page 2-8](#)

Using the Bundled Database

In bundled mode, the CM uses the Sybase Adaptive Server Enterprise database, which supports transaction-intensive enterprise applications, allows you to store and retrieve information online, and can warehouse information as needed.

The Sybase database is located on the same server as the other CM components. It uses a simple schema consisting of a group of small, simple tables.

1. The JDBC adapter sends converted RDRs to the database to be stored in these tables.
2. Records can then be accessed using standard database query and reporting tools. (Cisco provides a template-based reporting tool that can generate reports on subscriber usage, network resource analysis, and traffic analysis; for information about the Service Control reporting tool, see the [Cisco Service Control Application Reporter User Guide](#).)

Database maintenance is performed using operating system commands and scripts. The CM supports automatic purging of old records from the bundled database. By default, the report tables are automatically purged of every record that is more than two weeks old. The records are polled once every hour. Database maintenance can be configured using the [utility script](#). For more information, see [Managing the Periodic Deletion of Old Records, page 5-2](#).

Using an External Database

Any JDBC-compliant database (for example, Oracle or MySQL) may be used with the CM in conjunction with the JDBC adapter. In this case, the database can be local or remote. You should:

- Configure the JDBC adapter to use this database
- Configure a database pack to supply the CM with the parameters of the database (such as its IP address and port).
- Supply a JDBC driver for the database, to be used by the adapter when connecting to it.

For details about configuring the CM to work with an external database, see [Managing Databases and the Comma Separated Value Repository, page 5-1](#).