CHAPTER 2

Cisco Service Control Collection Manager Overview

This chapter is an overview of the Cisco Service Control Collection Manager operating with the Cisco Service Control Engine (SCE). Cisco Service Control Collection Manager uses the hardware capabilities of the SCE platform and identifies the combination of Cisco-specific applications that create the Cisco Service Control solution.

This chapter describes the following:

- How the Cisco Service Control Collection Manager works
- The Raw Data Records (RDRs) that the SCE platforms generate and send to Collection Manager
- The components of Collection Manager software package
- The database used to store the RDRs

The chapter includes the following sections:

Data Collection Process

The Cisco SCE platform generates RDRs. An application running on the SCE platform, such as the Cisco Service Control Application for Broadband (SCA BB), specifies and defines the RDRs. The following is the data collection process:

1. The SCE platform streams RDRs by using the simple, reliable RDR-Protocol. Integrating the collection of data records with the Service Control Solution involves implementing RDR-Protocol support in the collection system.
2. Collection Manager receives the RDRs from the SCE platform. The software modules recognize and sort the various types of RDRs. The sorting uses preset categories, including type and priority. Collection Manager queues the RDRs in persistent buffers.
3. Collection Manager adapters process each RDR. Each adapter performs a specific function on an RDR:
   - Stores it in a customer adapter formatted file on a local machine
   - Sends it to an RDBMS application
   - Performs custom operations

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

Raw Data Records

The SCE platform generates RDRs. The 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.

Table 2-1 presents examples of SCP application RDRs:

Table 2-1 Examples of SCP application RDRs

Report Type	Description	RDRs contain...
Periodic Subscriber usage report	SCE platforms are subscriber-aware network devices; they can report usage records per subscriber.	• Subscriber identifiers (such as the OSS subscriber ID) • Traffic types (such as HTTP, streaming, or peer-to-peer traffic) • Usage counters (such as total upstream and downstream volume). **Note** The Periodic Subscriber usage report is necessary for usage-based billing services and for network analysis and capacity planning. **Note** The Cisco SCA BB Subscriber Usage RDRs fall under this category.

Table 2-1 Examples of SCP application RDRs (continued)

Report Type	Description	RDRs contain...
Transaction-level report	SCE platforms perform stateful tracking of each network transaction conducted on the links on which they are located. 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.	• Basic Layer 3-4 attributes (such as source IP, destination IP, and port number) • Protocol-dependant Layer 7 attributes (such as user-agent, hostname for HTTP, or email address of an SMTP mail sender) • Generic parameters (such as time of day and transaction duration). **Note** These RDRs are important for content-based billing schemes and detailed usage statistics. **Note** Cisco SCA BB application Transaction RDRs fall under this category.
SCP application activity reports	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 can produce an RDR.	• Cisco SCA BB application's Breaching RDRs Breaching RDRs are generated when the system changes its active enforcement on a subscriber (because usage exceeded a set quota). • Cisco SCA BB application's Blocking RDRs Blocking RDRs are generated when an SCE platform blocks a network transaction (according to the rules contained in the current service configuration).

Collection Manager Software Package

Collection Manager software package contains a group of processing and sorting modules. These modules include the following components:

- Raw Data Record Server, page 2-5
- Categorizer, page 2-5
- Priority Queue and Persistent Buffer, page 2-5

Raw Data Record Server

A Cisco SCE platform sends RDRs to the Collection Manager. The RDR server adds an arrival timestamp and the ID of the source SCE platform. The RDR server then sends the RDR to the categorizer.

Categorizer

A categorizer classifies each RDR according to its RDR tag. It determines the destination adapter for the RDR and the priority queue through which the RDR is sent.

An RDR can be mapped to more than one adapter. A qualified technician defines the flow in a configuration file, based on user requirements.

Priority Queue and Persistent Buffer

Each adapter has one or more priority queues; a persistent buffer is assigned to each priority queue.

A priority queue queues each RDR according to the RDR's priority level and stores it in a persistent buffer until the adapter processes it.

A persistent buffer is a nonvolatile storage area that ensures that the system processes RDRs even in cases of hardware, software, or power failures.

Adapter

Adapters are software modules that transform RDRs to match the requirements of the target system. An adapter distributes the RDRs upon requests from a database. Currently, the following adapters are shipped with the Collection Manager:

- JDBC Adapter, page 2-6
- Comma-Separated Value Adapter, page 2-6
- Custom Adapter, page 2-7
- Topper Aggregator (TA) Adapter, page 2-8
- Real-Time Aggregating Adapter, page 2-11

Some of the adapters send data to the database or write data 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 Real-Time Aggregating Adapter (RAG) configuration file, see the "The ragadapter.conf File" section on page A-5.

JDBC Adapter

The Java Database Connectivity (JDBC) adapter receives RDRs, processes them, and stores the records in a database.

The JDBC adapter is compatible with any JDBC-compliant database server, You can configure the JDBC adapter to use a database operating on a remote machine.

The JDBC adapter is preconfigured to support the following databases:

- Sybase Adaptive Server Enterprise (ASE) 12.5, 15.0, and 15.0.3
- Oracle 9.2, 10.2, and 11

Disable the recycle bin feature available in Oracle 10 and later versions. You can set the initial value of the recyclebin parameter in the text initialization file init<*SID*>.ora, for example, recyclebin=off.

- MySQL 4.1, 5.0, and 5.1

Comma-Separated Value Adapter

The comma-separated-value (CSV) adapter receives RDRs, processes them, and writes the records to the files on the disk in the CSV format. Enhanced accounting and network traffic analysis records are generated from the records. The service provider OSS can retrieves these records using standard mechanisms. A third-party billing system can also retrieve these records using standard mechanisms. FTP is an example of a standard retrieval method.

Custom Adapter

The Custom Adapter allows you to configure the RDR format. The Custom Adapter receives records with specified fields from an RDR.

You can configure only a few fields of TUR. Some of the configurable fields are Sub-ID, Upstream + Downstream, and Application. The corresponding RDR is processed with those specific fields and stored in a CSV file. The Custom Adapter also supports basic arithmetic operations (add, sub, mul) on a few of the RDR fields.

The Custom Adapter process is disabled by default. Edit the ~scmscm/cm/config/cm.conf file in the Collection Manager and restart Collection Manager to enable the Custom Adapter.

For more information on enabling Custom Adapter, see the "Enabling an Adapter" section on page 4-5.

Operating System Finger Print Reporting

The Operating System Finger Print (OSFP) reports contain information about the operating system used by all subscribers on a daily basis.

The RPT_OSFP table in the Cisco Service Control Collection Manager database is updated with the operating system usage information for each subscriber. NAT environment and non-NAT environment details are updated separately.

The value of the NAT_ENV field in the RPT_OSFP table decides whether the operating system was detected in a NAT environment. The NAT environment value is set as 1 if multiple OS is detected for the same subscriber, and the value is set to 0 if the subscriber is using a single operating system.

By default, the OSFP Reports feature is disabled.

Before enabling the OSFP in Cisco Service Control Collection Manager, enable the OSFP feature in Cisco SCE devices. For details on enabling the OSFP feature using Cisco SCA BB console, see the *Cisco Service Control Application for Broadband User Guide*.

To enable the OSFP Reports in Cisco Service Control Collection Manager:

Step 1 Enable the custom adapter.

To enable the custom adapter, uncomment the customer adapter field #adapter.5=com.cisco.scmscm.adapters.custom.CustomAdapter under the [adapter] section in ~scmscm/cm/config/cm.conf file.

Step 2 Update the value of the *active* field under the [osfp] section in ~scmscm/cm/config/customadapter.conf file to true.

Example of the [osfp] section of the Custom Adapter configuration:

```
[osfp]
active=false
acc_period=1440
```

Topper Aggregator (TA) Adapter

The TA Adapter receives subscriber usage RDRs (NURs) and aggregates the data. The TA Adapter outputs Top Reports to the database in CSV format. Top Reports are lists of top subscribers for different metrics. The top 500 volume or session consumers in the last hour is an example of a Top Report metric. The TA adapter also sends the aggregated daily statistics of all the subscribers to the database.

The TA Adapter maintains a persistent saved state (saved to disk) to minimize data loss in case of failure.

The TA Adapter uses the JDBC adapter infrastructure. The TA Adapter operates with either local or remote JDBC-compliant databases.

TA Adapter supports IP-based aggregation along with the existing service-based or package-based aggregation.The service and IP type-based aggregation is enabled by default. The TA Adapter aggregates the records and lists the top subscribers for each combination of service and IP type with a different metric.

The package-based aggregation is disabled by default. Update the agg_pkg_level property as true in the ~scmscm/cm/config/taadapter.conf file and restart Collection Manager to enable the package-based aggregation.

Note The TA Adapter information is aggregated locally on a Collection Manager server. We reccomend not to use the Collection Managers with a single database. This may affect the accuracy of TA Adapter information.

Note A TA Adapter file is in the CSV format and populated with the RDR type NUR. The TA adapter works only with subscriber usage RDRs (NURs) and not with real-time subscriber usage RDRs (SURs).

The TA Adapter has the following specifications:

- TA Adapter Cycles, page 2-8
- TA Adapter Memory Requirements, page 2-9

TA Adapter Cycles

The TA Adapter works in three cycles: short, long, and peak hours (specific hour range). Cycles are fixed intervals. This data is temporarily stored in a persistent buffer directory with a default location of ~scmscm/cm/adapters/TAAdapter. At the end of an interval, the adapter outputs its aggregated information to the database and to a CSV file. The default interval for the short cycle is 1 hour. The default interval for the long cycle is 24 hours (every day at midnight). You can configure an interval (defined in minutes) and its start and end time. Configure the interval for the peak hours cycle. At the end of the peak hours, the adapter aggregates and outputs details of the top subscribers to the database.

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 aggregated Top Reports of the cycle 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 aggregated Top Reports of the cycle to the long cycle database table
  - Saves the current state file in case of power failure
  - Creates a CSV file to contain the aggregated statistics for the long-cycle period
- Peak-hour cycle—At the end of each peak-hour cycle, the adapter:
  - Adds the aggregated Top Reports of the cycle to the peak hour cycle database table
  - Saves the current state file in case of power failure

Note The long cycle data is saved every hour. The save file has 0 to 23:00 hours of data if a power shutdown occurs between 23:00 hrs and 24:00 hrs. The CSV file contains aggregated data. The current state is saved in a nonreadable format.

TA Adapter Memory Requirements

The TA Adapter requires sufficient dedicated memory. Configure the value of the memory in the **cm.conf** file in the following location:

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

To calculate the recommended amount of memory for the TA Adapter for a 32 bit operating system, use the following formula:

Memory (MB) = (3 * TOTAL_SUBSCRIBERS * ((AVG_SUBS_ID_LENGTH + (2 * NUM_OF_PERIODS * NUM_OF_SERVICES * NUM_OF_IPTYPES) + (32 * NUM_OF_SERVICES))) /(1024 * 1024)) * (NUM_OF_IPTYPES)

To calculate the recommended amount of memory for the TA Adapter for a 64 bit operating system, use the following formula:

Memory (MB) = (5.25 * TOTAL_SUBSCRIBERS * ((AVG_SUBS_ID_LENGTH + (2 * NUM_OF_PERIODS * NUM_OF_SERVICES * NUM_OF_IPTYPES) + (32 * NUM_OF_SERVICES))) /(1024 * 1024)) * (NUM_OF_IPTYPES)

To calculate the recommended amount of memory for the TA Adapter with Package Enabled Aggregation for a 32 bit operating system, use the following formula:

Memory (MB) = (3 * TOTAL_SUBSCRIBERS * ((AVG_SUBS_ID_LENGTH + ((20 * NUM_OF_PKGS) +48) + (2 * NUM_OF_PERIODS * NUM_OF_SERVICES * NUM_OF_IPTYPES)+ (32 * NUM_OF_SERVICES)))) /(1024 * 1024) * (NUM_OF_PKGS) * (NUM_OF_IPTYPES)

To calculate the recommended amount of memory for the TA Adapter with Package Enabled Aggregation for a 64 bit operating system, use the following formula:

Memory (MB) = (5.25 * TOTAL_SUBSCRIBERS * ((AVG_SUBS_ID_LENGTH + ((20 * NUM_OF_PKGS) +48) + (2 * NUM_OF_PERIODS * NUM_OF_SERVICES * NUM_OF_IPTYPES)+ (32 * NUM_OF_SERVICES)))) /(1024 * 1024) * (NUM_OF_PKGS) * (NUM_OF_IPTYPES)

The definitions of the fields in the formula are:

- TOTAL_SUBSCRIBERS is the total number of subscribers across the network for each aggregation period.
- AVG_SUBS_ID_LENGTH is the average character length of a subscriber.
- NUM_OF_PERIODS = Total number of Aggregation cycles. Default value is 2.

- NUM_OF_IPTYPES = Total no of ip types supported +1
- NUM_OF_SERVICES = Total number of active services + 1
- NUM_OF_PKGS = Maximum number of packages for a particular subscriber +1

Note Cisco Service Control Collection Manager requires 75 percentage additional memory on a 64 bit machine when compared to a 32 bit machine. The 64 bit machine consumes 64 bit of memory to store each memory address pointers for the java application process. This is twice the memory required to store the addresses in a 32 bit machine. This doubles up the memory requirement while using the Cisco Service Control Collection Manager on 64 bit machine. The Package based aggregation and IP Type based aggregation functions of Cisco Service Control Collection Manager consumes more memory to store data and uses more object references that uses 64 bit to store each address in the memory.

Note For Solaris JRE 64 bit, you can set higher values for the configured memory. To configure the TA or RAG adapters to run with the JRE 64 bit, see the "The [adapter_mem] Section" section on page B-3.

Real-Time Aggregating Adapter

The RAG adapter processes RDRs of one or more types and aggregates the data from predesignated field positions into buckets. The contents of the buckets are written to CSV files. The RAG adapter has the following components and processes:

- RAG Adapter Aggregation Buckets, page 2-11
- Flushing a Bucket, page 2-11
- RAG Adapter Process for HTTP Transaction Usage RDR, page 2-12
- RAG Adapter Process for Video Transaction Usage RDR, page 2-12
- RAG Adapter Process for Subscriber Usage RDR, page 2-12
- Managing OSFP support for Subscriber Usage RDR, page 2-13

RAG Adapter Aggregation Buckets

A RAG adapter aggregation bucket is indexed by combining values from fields in the RDR. The indexing relation can be one-to-one or many-to-one.

The values in the bucket-identifying fields are processed using closures (equivalence classes), which are configured per type of RDR.

```
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
```

You can configure the RAG adapter to monitor the values in certain fields for change relative to the values in the first RDR that entered the bucket. For each monitored field, an action is performed when a value change is detected. The supported actions are:

- Checkpoint the bucket without aggregating this RDR into it, and start a new bucket with this RDR.
- Issue a warning to the user log.

Buckets, closures, triggers, and trigger actions are defined in an XML file. For a sample XML file, see the "ragadapter.xml File" section on page A-6.

Flushing a Bucket

When a bucket is flushed, it is written as one line to a CSV file.

The trigger for flushing a bucket (a *checkpoint*) is the earliest occurrence of any of the following:

- Time elapsed since the creation of a bucket reaches a configured duration
- Volume in an accumulated field in a bucket exceeds a configured amount
- A RAG Adapter, or the entire Collection Manager, goes down
- An RDR arrives at the bucket with some new value (relative to the bucket contents) in some field

The trigger to close a CSV file is the earliest occurrence of one of the following:

- Time elapsed since the creation of a file has reached a set duration
- Number of lines in a file has reached a set number
- A RAG Adapter, or the entire Collection Manager, goes down

RAG Adapter Process for HTTP Transaction Usage RDR

If a corresponding RDR TAG is configured under the RAG adapter section in the queue.conf file before Collection Manager is started, the RAG adapter processes the HTTP_TUR RDRs.

An XML file specific to HTTP_TUR is included in Collection Manager distribution for the RAG adapter in order to manage the HTTP_TUR RDR fields. The aggregation period for processing the RDRs is specified in the http_TURs.xml file.

Aggregation is based on the domain, package, and service for the corresponding HTTP Transaction Usage RDR. At the end of the aggregation period, the adapter adds the aggregated Top Reports for hosts to the RPT_TOP_HTTP_HOSTS database table. The adapter adds the aggregated Top Reports for domains to the RPT_TOP_HTTP_DOMAINS database table.

RAG Adapter Process for Video Transaction Usage RDR

If you configure a corresponding RDR TAG under the RAG adapter section in **queue.conf** file before you start Collection Manager, the RAG adapter processes VIDEO_TUR RDRs.

Collection Manager distribution includes an XML file specifically for VIDEO_TUR, which enables the RAG adapter to manage VIDEO_TUR RDR fields. The aggregation period for processing the RDRs is specified in the video_TURs.xml file.

Aggregation is based on the domain, package, and service for the corresponding VIDEO Transaction Usage RDR. At the end of the aggregation period, the adapter adds the aggregated Top Reports for hosts to the RPT_TOP_VIDEO_HOSTS database table. Also, the adapter adds the aggregated Top Reports for domains to the RPT_TOP_VIDEO_DOMAINS database tables.

Starting from Cisco Service Control Collection Manager Release 4.0.0, BIT_RATE is calculated from the Downstream volume and the Duration of the Video TURs. The accumulated bit rate (BIT_RATE) and the corresponding rank (BIT_RATE_RANK) is added to the RPT_VIDEO_HOSTS table and RPT_VIDEO_DOMAINS table for each service and package combination.

RAG Adapter Mobile Based Reports

Collection Manager supports both GSM and CDMA mobile-based reports. GSM reports are enabled by default.

The ~scmscm/cm/config/ragadapter.conf

file property (vsa_type=gsm or cdma) configures the mobile report type processed.

A user should to place the related XML files in ~scmscm/cm/config/ragadapter for the Collection Manager. By default all the XML files are available in the ~scmscm/cm/config/ragadapter/repository directory.

RAG Adapter Process for Subscriber Usage RDR

If a corresponding RDR TAG is configured under the RAG adapter section in the queue.conf file before Collection Manager is started, the RAG adapter processes the NUR RDRs. The Subscriber Usage RDRs are processed in two ways:

- Aggregation Based on VSA Fields, page 2-13
- Aggregation Based on Package, page 2-13

Aggregation Based on VSA Fields

VSA field aggregation supports either GSM or CDMA mobile-based reports:

Collection Manager supports one of the mobile-based reports (either GSM or CDMA). By default, GSM reports are enabled. In the ~scmscm/cm/config/ragadapter.conf file, a new property (VSA_Type = gsm or cdma) is added to configure which mobile report Collection Manager needs to process.

During restart, based on the vsa_type configuration, the Collection Manager will copy the corresponding xml config file to the ~scmscm/cm/config/ragadapter file for processing.

Aggregation Based on GSM-Specific VSA Fields

The vsa_SURs.xml file manages GSM-specific VSA fields in the Subscriber Usage RDRs. The aggregation period for processing the RDRs is specified in this file. Aggregation is based on the GSM-specific VSA fields (APN, SGSN, NETWORK_TYPE, DEVICE_TYPE, and USER_LOCATION). At the end of the aggregation period, the adapter adds the aggregated Top Reports related to the GSM-specific VSA fields to their corresponding database tables:

- RPT_TOP_APN
- RPT_TOP_SGSN
- RPT_TOP_NETWORK_TYPE
- RPT_TOP_DEVICE_TYPE
- RPT_TOP_USER_LOCATION

Aggregation Based on CDMA-Specific VSA Fields

The cdma_SURs.xml file manages the CDMA-specific VSA fields in the Subscriber Usage RDRs. The aggregation period for processing the RDRs is specified in this file. Aggregation is based on the CDMA-specific VSA fields (PCF, HOME AGENT, MEID). At the end of the aggregation period, the adapter adds the aggregated Top Reports related to the CDMA-specific VSA fields to their corresponding database tables:

- RPT_TOP_PCF
- RPT_TOP_HOME_ AGENT
- RPT_TOP_MEID

Aggregation Based on Package

The vlink_BW_per_pkg.xml file manages upstream and downstream VLINKs in the Subscriber Usage RDRs. The aggregation period for processing the RDRs is specified in this file.

Aggregation is based on the up and down VLINK fields. At the end of the aggregation period, the adapter adds the aggregated Top Reports for the up VLINKS to the RPT_UVLINK database table. Also, the adapter adds the aggregated Top Reports for the down VLINKs to the RPT_DVLINK database table.

Rag Adapter package aggregation happens based on PACKAGE_ID, UP_VLINK_ID/DOWN_VLINK_ID, and IP_TYPE.

Managing OSFP support for Subscriber Usage RDR

Operating System Finger Printing (OSFP) is supported in Collection Manager software.

The **~scmscm/script/updateVSAindex.sh** script supports backwards compatibilty.

The following properties are added in the **~scmscm/cm/config/ragadpater.conf** file:

- attr_index—Index position of attribute indicator field in NUR RDR in current the Cisco SCOS version
- attr_shift_pos—Number of fields to be shifted

Check the VSA Attribute Indicator field index value for the current Cisco SCOS version. This is based on the Cisco SCOS version update *attr_index* , *attr_shift_pos* properties in ~scmscm/cm/config/ragadpater.conf file. The "RAG Adapter Configuration File" section on page B-14 details the RAG adapter configuration file ragadapter.conf**.** Then execute the script **~scmscm/script/updateVSAindex.sh** script. The script updates the corresponding XML file (vsa_SURs.xml/cdma_SURs.xml).

Table 2-2 shows the VSA Attribute Indicator field index values.

Table 2-2 VSA Attribute Indicator Field Index Values

Cisco SCOS Version	attr_index	attr_shift_pos
4.0.0	17	0
3.8.5	17	0
3.8.0	17	0
3.7.5	17	0
3.7.0	14	3

Using Databases

Collection Manager can use either a bundled database or an external database to store the RDRs supplied by the SCE platforms. The following sections describe the procedures:

- Using the Bundled Database, page 2-14
- Using an External Database, page 2-15

Using the Bundled Database

In the bundled mode, Collection Manager uses the Sybase Adaptive Server Enterprise database. This database enables you to do the following:

- Support transaction-intensive enterprise applications
- Store and retrieve information online
- Warehouse needed information by putting it in a central depository

The Sybase database is located on the same server as the other Collection Manager components. It uses a simple schema that includes a group of small, simple tables.

1. The JDBC adapter sends the converted RDRs to the database, where the RDRs are stored in the tables of the simple schema.

2. You can access these records by 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*.)

You can maintain the database by using operating system commands and scripts. Collection Manager supports automatic purging of old records from the bundled database. By default, Collection Manager automatically purges the report table of every record that is older than two weeks and polls the records once every hour. You can configure database maintenance by using the **dbperiodic.sh** utility script. For more information on database maintenance, see the "Managing the Periodic Deletion of Old Records" section on page 5-3.

Using an External Database

With the JDBC adapter, you can use any JDBC-compliant database (for example, Oracle or MySQL). The database can be either local or remote. To use an external database, perform the following tasks:

Step 1 Configure the JDBC adapter to use this database.

Step 2 Configure a database pack to supply Collection Manager with the parameters of the database (such as its IP address and port).

Step 3 Supply a JDBC driver for the database so that the adapter can connect to the database.

For details about configuring Collection Manager to work with an external database, see Chapter 5, "Managing Databases and the Comma-Separated Value Repository".