CHAPTER 2

Raw Data Records: Formats and Field Contents

This chapter contains a list of the Raw Data Records (RDRs) produced by the SCE platform and a full description of the fields contained in each RDR.

The chapter also contains field-content information for those fields that are generated by Service Control components.

Raw Data Records Overview

RDRs are the collection of fields that are sent by the Service Control Engine (SCE) platforms to the Cisco Service Control Management Suite (SCMS) Collection Manager (CM).

Fields that are common to many of the RDRs are described in the next section, before the individual RDRs are described.

Universal RDR Fields

This section contains descriptions of fields that are common to many RDRs. The first two fields, SUBSCRIBER_ID and PACKAGE_ID, appear in almost all the RDRs. The other fields are listed in alphabetic order.

- SUBSCRIBER_ID—The subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 40 characters. For unknown subscribers this field may contain an empty string.
- PACKAGE_ID—The ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
- ACCESS_STRING—A Layer 7 property, extracted from the transaction. For possible values, see String Fields, page 2-47.
- BREACH_STATE—This field indicates whether the subscriber's quota was breached.
  - 0—Not breached
  - 1—Breached
- CLIENT_IP—The IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
- CLIENT_PORT—For TCP/UDP-based sessions, the port number of the client side (initiator) of the networking session. For non-TCP/UDP sessions, this field has the value zero.
- CONFIGURED_DURATION—For periodic RDRs, the configured period, in seconds, between successive RDRs.
- END_TIME—Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
- FLAVOR_ID—For protocol signatures that have flavors, this field contains the ID of the flavor associated with this session.
- INFO_STRING—A Layer 7 property extracted from the transaction. For possible values, see String Fields, page 2-47.

- INITIATING_SIDE—On which side of the SCE platform the initiator of the transaction resides.
  - 0—The subscriber side
  - 1—The network side
- PROTOCOL_ID—This field contains the unique ID of the protocol associated with the reported session.

Note The PROTOCOL_ID will be the Generic IP / Generic TCP / Generic UDP protocol ID value, according to the specific transport protocol of the transaction, unless a more specific protocol definition (such as a signature-based protocol or a port-based protocol), which matches the reported session, is assigned to a service.

- PROTOCOL_SIGNATURE—This field contains the ID of the protocol signature associated with this session.
- REPORT_TIME—Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
- SERVER_IP—Contains the destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
- SERVER_PORT—For TCP/UDP-based sessions, this field contains the destination port number of the networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
- SERVICE_ID—This field indicates the service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
- TIME_FRAME—The system supports time-dependent policies, by using different rules for different time frames. This field indicates the time frame during which the RDR was generated. The field's value can be in the range 0 to 3, indicating which of the four time frames was used.
- ZONE_ID—This field contains the ID of the zone associated with this session.

Note All volumes in RDRs are reported in L3 bytes.

Transaction RDR

The TRANSACTION_RDR may be generated at the end of a session, according to a user-configurable sampling mechanism—configuring number-of-transaction-RDRs-per-second sets the number of Transaction RDRs generated per-second. This RDR is not generated for sessions that were blocked by a rule.

The RDR tag of the TRANSACTION_RDR is **0xf0f0f010 / 4042321936.**

The following table lists the RDR fields and their descriptions.

Table 2-1 Transaction RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	INT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
PROTOCOL_ID	INT16	See Universal RDR Fields, page 2-2.
SKIPPED_SESSIONS	INT32	The number of unreported sessions since the previous RDR *plus one.* The default value is 1. A value of 2 means that *one* RDR was unreported.
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
ACCESS_STRING	STRING	See Universal RDR Fields, page 2-2.
INFO_STRING	STRING	See Universal RDR Fields, page 2-2.
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.

Table 2-1 Transaction RDR Fields (continued)

RDR Field Name	Type	Description
TIME_FRAME	INT8	See Universal RDR Fields, page 2-2.
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	See Universal RDR Fields, page 2-2.
ZONE_ID	INT32	See Universal RDR Fields, page 2-2.
FLAVOR_ID	INT32	See Universal RDR Fields, page 2-2.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow.

Transaction Usage RDR

The TRANSACTION_USAGE_RDR is generated at the end of a session, for all transactions on packages and services that are configured to generate such an RDR. This RDR is not generated for sessions that were blocked by a rule.

Note By default, packages and services are *disabled* from generating this RDR.

This RDR is designed for services and packages where specific, per-transaction RDRs are required (for example, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate. *Configure the generation scheme for this RDR with extra care.*

The RDR tag of the TRANSACTION_USAGE_RDR is **0xf0f0f438 / 4042323000.**

The following table lists the RDR fields and their descriptions.

Table 2-2 Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
PROTOCOL_ID	INT16	See Universal RDR Fields, page 2-2.
SKIPPED_SESSIONS	INT32	Reason for RDR generation: • 0 (INTERIM)—Interim Transaction Usage RDR • 1 (SESSION_END)—Normal Transaction Usage RDR for a flow that had no interim Transaction Usage RDRs • 2 (LAST_TUR)—The last Transaction Usage RDR for a flow that had interim Transaction Usage RDRs
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
ACCESS_STRING	STRING	See Universal RDR Fields, page 2-2.
INFO_STRING	STRING	See Universal RDR Fields, page 2-2.

Table 2-2 Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	See Universal RDR Fields, page 2-2.
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated stream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	See Universal RDR Fields, page 2-2.
ZONE_ID	INT32	See Universal RDR Fields, page 2-2.
FLAVOR_ID	INT32	See Universal RDR Fields, page 2-2.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow.

HTTP Transaction Usage RDR

The HTTP_TRANSACTION_USAGE_RDR is generated at the end of an HTTP session, for all transactions on packages and services that are configured to generate a Transaction Usage RDR. This RDR is not generated for sessions that were blocked by a rule.

Note By default, packages and services are *disabled* from generating this RDR.

This RDR is designed for services and packages where specific, per-transaction RDRs are required (for example, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate. *Configure the generation scheme for this RDR with extra care.*

The RDR tag of the HTTP_TRANSACTION_USAGE_RDR is **0xf0f0f43C / 4042323004.**

The following table lists the RDR fields and their descriptions.

Table 2-3 HTTP Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
PROTOCOL_ID	INT16	See Universal RDR Fields, page 2-2.
SKIPPED_SESSIONS	INT32	Number of unreported sessions since the previous RDR. Since an HTTP Transaction Usage RDR is generated only at the end of a flow, this field always has the value 1.
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
ACCESS_STRING	STRING	See Universal RDR Fields, page 2-2.
INFO_STRING	STRING	See Universal RDR Fields, page 2-2.
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.

Table 2-3 HTTP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	See Universal RDR Fields, page 2-2.
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated stream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	See Universal RDR Fields, page 2-2.
ZONE_ID	INT32	See Universal RDR Fields, page 2-2.
FLAVOR_ID	INT32	See Universal RDR Fields, page 2-2.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow.
USER_AGENT	STRING	The user agent field extracted from the HTTP transaction.
HTTP_URL	STRING	The URL extracted from the HTTP transaction.

RTSP Transaction Usage RDR

The RTSP_TRANSACTION_USAGE_RDR is generated at the end of a session, for all RTSP transactions on packages and services that are configured to generate a Transaction Usage RDR. This RDR is not generated for sessions that were blocked by a rule.

Note By default, packages and services are *disabled* from generating this RDR.

This RDR is designed for services and packages where specific, per-transaction RDRs are required (for example, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate. *Configure the generation scheme for this RDR with extra care.*

The RDR tag of the RTSP_TRANSACTION_USAGE_RDR is **0xf0f0f440 / 4042323008.**

The following table lists the RDR fields and their descriptions.

Table 2-4 RTSP Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
PROTOCOL_ID	INT16	See Universal RDR Fields, page 2-2.
SKIPPED_SESSIONS	INT32	Number of unreported sessions since the previous RDR. Since an RTSP Transaction Usage RDR is generated only at the end of a flow, this field always has the value 1.
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
ACCESS_STRING	STRING	See Universal RDR Fields, page 2-2.
INFO_STRING	STRING	See Universal RDR Fields, page 2-2.
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.

Table 2-4 RTSP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	See Universal RDR Fields, page 2-2.
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated stream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	See Universal RDR Fields, page 2-2.
ZONE_ID	INT32	See Universal RDR Fields, page 2-2.
FLAVOR_ID	INT32	See Universal RDR Fields, page 2-2.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow.
RTSP_SESSION_ID	STRING	RTSP session ID as seen on an RTSP SETUP request.
RTSP_URL	STRING	RTSP URL.
RESPONSE_DATE	STRING	RTSP DESCRIBE date.
TOTAL_ENCODING_RATE	UINT32	Sum of encoding rates of data flows.
NUMBER_OF_VIDEO_ STREAMS	UINT8	Number of video streams for this RTSP session.

Table 2-4 RTSP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
NUMBER_OF_AUDIO_ STREAMS	UINT8	Number of audio streams for this RTSP session.
SESSION_TITLE	STRING	Title for this RTSP stream.
SERVER_NAME	STRING	Name of the RTSP server.

VoIP Transaction Usage RDR

The VOIP_TRANSACTION_USAGE_RDR is generated at the end of a session, for all transactions on packages and services that are configured to generate such an RDR. This RDR is not generated for sessions that were blocked by a rule.

Note By default, packages and services are *disabled* from generating this RDR.

The VoIP Transaction Usage RDR is enabled automatically when the Transaction Usage RDR is enabled; both RDRs will be generated when the session ends. Currently, the VoIP Transaction Usage RDR is generated for H323, Skinny, SIP, and MGCP sessions.

This RDR is designed for services and packages where specific, per-transaction RDRs are required (for example, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate. *Configure the generation scheme for this RDR with extra care.*

The RDR tag of the VOIP_TRANSACTION_USAGE_RDR is **0xf0f0f46a / 4042323050.**

The following table lists the RDR fields and their descriptions.

Table 2-5 VoIP Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
PROTOCOL_ID	INT16	See Universal RDR Fields, page 2-2.
SKIPPED_SESSIONS	INT32	Number of unreported sessions since the previous RDR. Since a VoIP Transaction Usage RDR is generated only at the end of a flow, this field always has the value 1.
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
ACCESS_STRING	STRING	See Universal RDR Fields, page 2-2.
INFO_STRING	STRING	See Universal RDR Fields, page 2-2.
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.

Table 2-5 VoIP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	See Universal RDR Fields, page 2-2.
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	See Universal RDR Fields, page 2-2.
ZONE_ID	INT32	See Universal RDR Fields, page 2-2.
FLAVOR_ID	INT32	See Universal RDR Fields, page 2-2.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow.
APPLICATION_ID	UINT32	The ITU-U vendor ID of the application. A value of 0xFFFFFFFF indicates that this field was not found in the traffic.

Table 2-5 VoIP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
UPSTREAM_PACKET_LOSS	UINT16	The average fractional upstream packet loss for the session, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.) A value of 0xFFFF indicates that this field is undefined (no RTCP flows were opened).
DOWNSTREAM_PACKET_ LOSS	UINT16	The average fractional downstream packet loss for the session, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.) A value of 0xFFFF indicates that this field is undefined (no RTCP flows were opened).
UPSTREAM_AVERAGE_ JITTER	UINT32	The average upstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.) A value of 0xFFFFFFFF indicates that this field is undefined (no RTCP flows were opened).
DOWNSTREAM_AVERAGE_ JITTER	UINT32	The average downstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.) A value of 0xFFFFFFFF indicates that this field is undefined (no RTCP flows were opened).
CALL_DESTINATION	STRING	The Q931 Alias address of the session destination. A value of N/A indicates that this field was not found in the traffic.
CALL_SOURCE	STRING	The Q931 Alias address of the session source. A value of N/A indicates that this field was not found in the traffic.

Table 2-5 VoIP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
UPSTREAM_PAYLOAD_ TYPE	UINT8	The upstream RTP payload type for the session. A value of 0xFF indicates that this field was not available (no RTP flows were opened).
DOWNSTREAM_PAYLOAD_ TYPE	UINT8	The downstream RTP payload type for the session. A value of 0xFF indicates that this field is undefined (no RTP flows were opened).
CALL_TYPE	UINT8	The call type (taken from H225 packet). A value of 0xFF indicates that this field is undefined (no RTP flows were opened).
MEDIA_CHANNELS	UINT8	The number of data flows that were opened during the session.

Note

Packet Loss
This field is taken from the RTCP field “fraction lost”. It is the average value of all RTCP packets seen during the flow life for the specified direction. The value is the numerator of a fraction whose denominator is 256. To get the packet loss value as percentage, divide this value by 2.56.

Average Jitter
This field is taken from the RTCP field “interval jitter”. The reported value is the average value of all RTCP packets seen during the flow life for the specified direction. This value is multiplied by the NTP time-stamp delta (middle 32 bits) and divided by the RTCP time-stamp delta to convert it to normal time units. These two time stamps are also taken from the RTCP packet. The reported value is the average jitter in units of 1/65536 second. To convert to milliseconds divide by 65.536.

For more information about the RCP/RTCP standard, refer to RFC 1889.

Subscriber Usage RDR

The SUBSCRIBER_USAGE_RDR is generated periodically, at user-configured intervals, for each subscriber. A separate RDR is generated for each service usage counter. The RDR is generated only if the subscriber consumed resources associated with the service usage counter during the current reporting period.

At fixed, user-configurable intervals (for example, every 30 minutes), there is a periodic SUBSCRIBER_USAGE_RDR generation point. Whether or not a Subscriber Usage RDR *for a particular subscriber* is actually generated depends on the following:

- If the subscriber consumed resources associated with a service usage counter since the previous RDR generation point, a Subscriber Usage RDR is generated.
- If the subscriber did *not* consume resources associated with a service usage counter since the previous RDR generation point, *no* Subscriber Usage RDR is generated.

Note Unlike other Usage RDRs, the generation logic for Subscriber Usage RDRs does NOT use the zeroing methodology (as described in Periodic RDR Zero Adjustment Mechanism, page 2-52).

Subscriber Usage RDRs may also be generated in the following situation:

- The subscriber performed a logout in a subscriber-integrated installation or was un-introduced from the SCE platform:
  - If the subscriber consumed resources associated with a service usage counter since the previous Subscriber Usage RDR, a Subscriber Usage RDR is generated.
  - If the subscriber did not consume resources since the previous RDR, no RDR is generated for that service usage counter.

The RDR tag of the SUBSCRIBER_USAGE_RDR is **0xf0f0f000 / 4042321920.**

The following table lists the RDR fields and their descriptions.

Table 2-6 Subscriber Usage RDR

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	INT16	See Universal RDR Fields, page 2-2.
SERVICE_USAGE_ COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 counters in the subscriber scope.
BREACH_STATE	UINT8	See Universal RDR Fields, page 2-2. Holds the breach state of a service. However, this RDR reports usage counters, which cannot be breached, so the value is always zero.

Table 2-6 Subscriber Usage RDR (continued)

RDR Field Name	Type	Description
REASON	UINT8	Reason for RDR generation: • 0—Period time passed • 1—Subscriber logout • 2—Package switch • 3—Wraparound • 4—End of aggregation period
CONFIGURED_DURATION	INT32	See Universal RDR Fields, page 2-2.
DURATION	INT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	INT32	See Universal RDR Fields, page 2-2.
UPSTREAM_VOLUME	INT32	Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT16	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT16	Aggregated number of session seconds for the reported service, for the current reporting period.

Real-Time Subscriber Usage RDR

The REALTIME_SUBSCRIBER_USAGE_RDR is generated periodically, at user-configured intervals, for each subscriber that has real-time monitoring enabled. A separate RDR is generated for each service usage counter. The RDR is generated only if the subscriber consumed resources associated with the service usage counter during the current reporting period.

Note A Real-Time Subscriber Usage RDR will be generated only for those subscribers with real-time monitoring enabled. For information about enabling real-time monitoring, see the "Additional Management Tools and Interfaces" chapter of the *Cisco Service Control Application for Broadband User Guide.*

At fixed, user-configurable intervals (for example, every 5 minutes), there is a periodic REALTIME_SUBSCRIBER_USAGE_RDR generation point. The REALTIME_SUBSCRIBER_USAGE_RDR reports the same usage information as the SUBSCRIBER_USAGE_RDR, but is generated more frequently to provide a more detailed picture of subscriber activity. It is used by the Cisco Service Control Application Reporter to generate reports on the activities of single subscribers over time.

Whether or not a Real-Time Subscriber Usage RDR *for a particular subscriber* is actually generated depends on the following:

- If the subscriber consumed resources associated with a service usage counter since the previous RDR generation point, a Real-Time Subscriber Usage RDR is generated.
- If the subscriber did *not* consume resources associated with a service usage counter since the previous RDR generation point, *no* Real-Time Subscriber Usage RDR is generated now.

  However, the generation logic for Subscriber Usage RDRs uses the zeroing methodology (as described in Periodic RDR Zero Adjustment Mechanism, page 2-52; if the subscriber consumes resources associated with the service usage counter at some later time, this will cause the *immediate* generation of either one or two zero-consumption Real-Time Subscriber Usage RDRs. (In addition to the eventual generation of the Real-Time Subscriber Usage RDR associated with this latest consumption of resources).

  - If there was only one interval (for example, 0805–0810) for which there was no subscriber consumption of resources, only one zero-consumption Real-Time Subscriber Usage RDR is generated.
  - If there were multiple consecutive intervals (for example, 0805–0810, 0810–0815, 0815–0820, 0820–0825) for which there was no subscriber consumption of resources, two zero-consumption Real-Time Subscriber Usage RDRs are generated: one for the first such time interval (0805–0810) and one for the last (0820–0825).

Real-Time Subscriber Usage RDRs may also be generated in the following situation:

- The subscriber performed a logout in a subscriber-integrated installation or was un-introduced from the SCE platform:
  - If the subscriber consumed resources associated with a service usage counter since the previous Real-Time Subscriber Usage RDR, a Real-Time Subscriber Usage RDR is generated and then a zero-consumption Real-Time Subscriber Usage RDR is generated.
  - If the subscriber did not consume resources since the previous RDR, no RDR is generated for that service usage counter.

A zero-consumption Real-Time Subscriber Usage RDR will also be generated for a subscriber in the following situation:

- The subscriber performed a login in a subscriber-integrated installation or was introduced from the SCE platform:
  - Before the first Real-Time Subscriber Usage RDRs reporting actual consumption are generated, a zero-consumption Real-Time Subscriber Usage RDR is generated.

The RDR tag of the REALTIME_SUBSCRIBER_USAGE_RDR is **0xf0f0f002 / 4042321922.**

The following table lists the RDR fields and their descriptions.

Table 2-7 Real-Time Subscriber Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	INT16	See Universal RDR Fields, page 2-2.
SERVICE_USAGE_ COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 counters in the subscriber scope.
AGGREGATION_OBJECT_ID	INT16	Externally assigned: • 0—Offline subscriber • 1—Online subscriber
BREACH_STATE	UINT8	See Universal RDR Fields, page 2-2. Holds the breach state of a service. However, this RDR reports usage counters, which cannot be breached, so the value is always zero.
REASON	UINT8	Reason for RDR generation: • 0—Period time passed • 1—Subscriber logout • 2—Package switch • 3—Wraparound • 4—End of aggregation period
CONFIGURED_DURATION	INT32	See Universal RDR Fields, page 2-2.

Table 2-7 Real-Time Subscriber Usage RDR Fields (continued)

RDR Field Name	Type	Description
DURATION	INT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	INT32	See Universal RDR Fields, page 2-2.
UPSTREAM_VOLUME	INT32	Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT16	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT16	Aggregated number of session seconds for the reported service, for the current reporting period.

Link Usage RDR

The LINK_USAGE_RDR is generated periodically, at user-configured intervals, for each link. A separate RDR is generated for each service usage counter. The RDR is generated only if resources associated with the service usage counter were consumed during the current reporting period.

At fixed, user-configurable intervals (for example, every 30 minutes), there is a periodic LINK_USAGE_RDR generation point. Whether or not a Link Usage RDR is actually generated depends on the following:

- If network resources associated with a service usage counter were consumed since the previous RDR generation point, a Link Usage RDR is generated.
- If network resources associated with a service usage counter were *not* consumed since the previous RDR generation point, *no* Link Usage RDR is generated.

  However, the generation logic for Link Usage RDRs uses the zeroing methodology (as described in Periodic RDR Zero Adjustment Mechanism, page 2-52); if network resources associated with the service are again consumed at some later time, this will cause the immediate generation of either one or two zero-consumption Link Usage RDRs. (In addition to the eventual generation of the Link Usage RDR associated with this latest consumption of network resources).

  – If there was only one interval (for example, 0830–0900) for which there was no consumption of network resources, only one zero-consumption Link Usage RDR is generated.

  – If there were multiple consecutive intervals (for example, 0830–0900, 0900–0930, 0930–1000, 1000–1030) for which there was no consumption of network resources, two zero-consumption Link Usage RDR are generated: one for the first such time interval (0830–0900) and one for the last (1000–1030).

Note A separate RDR is generated for each link (on a single traffic processor) in the SCE platform, where each RDR represents the total traffic processed and analyzed by that processor (for the specified service usage counter). To compute the total traffic in any given time frame, take the sum of traffic of the RDRs of all the processors.

The RDR tag of the LINK_USAGE_RDR is **0xf0f0f005 / 4042321925.**

The following table lists the RDR fields and their descriptions.

Table 2-8 Link Usage RDR Fields

RDR Field Name	Type	Description
LINK_ID	INT8	A numeric value associated with the reported network link. Possible values are 0 and 1 (referring to physical links 1 and 2 respectively). For future use.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR. Possible values are 0 to 3.
SERVICE_USAGE_ COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.

Table 2-8 Link Usage RDR Fields (continued)

RDR Field Name	Type	Description
CONFIGURED_DURATION	INT32	See Universal RDR Fields, page 2-2.
DURATION	INT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	INT32	See Universal RDR Fields, page 2-2.
UPSTREAM_VOLUME	INT32	Aggregated upstream volume of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume of all sessions, in kilobytes, for the current reporting period.
SESSIONS	INT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	INT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	INT32	Concurrent number of sessions using the reported service at this point in time.
ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers using the reported service at this point in time.
TOTAL_ACTIVE_ SUBSCRIBERS	INT32	Concurrent number of subscribers in the system at this point in time.

Package Usage RDR

The PACKAGE_USAGE_RDR is generated periodically, at user-configured intervals, for each package usage counter. A separate RDR is generated for each service usage counter. The RDR is generated only if resources associated with the service usage counter were consumed during the current reporting period. The RDR contains aggregated network usage information for all subscribers to the package or group of packages represented by the package usage counter.

At fixed, user-configurable intervals (for example, every 5 minutes), there is a periodic PACKAGE_USAGE_RDR generation point. Whether or not a Package Usage RDR is actually generated depends on the following:

- If network resources associated with a service usage counter were consumed by a subscriber of the Package since the previous RDR generation point, a Package Usage RDR is generated.
- If a subscriber of the Package has not consumed network resources associated with a service usage counter since the previous RDR generation point, no Package Usage RDR is generated.

  However, the generation logic for Package Usage RDRs uses the zeroing methodology (as described in Periodic RDR Zero Adjustment Mechanism, page 2-52; if network resources associated with the service usage counter are again consumed by any subscriber of the package at some later time, this will cause the immediate generation of either one or two zero-consumption Package Usage RDRs. (In addition to the eventual generation of the Package Usage RDR associated with this latest consumption of network resources).

  – If there was only one interval (for example, 0805–0810) for which there was no consumption of network resources by any subscriber of the package, only one zero-consumption Package Usage RDR is generated.

  – If there were multiple consecutive intervals (for example, 0805–0810, 0810–0815, 0815–0820, 0820–0825) for which there was no consumption of network resources by any subscriber of the package, two zero-consumption Package Usage RDR are generated: one for the first such time interval (0805–0810) and one for the last (0820–0825).

Note Each traffic processor in the SCE platform generates a separate RDR, where each RDR represents the total traffic processed and analyzed by that processor (for the specified service usage counter). To compute the total traffic (for a package) in any given time frame, take the sum of the traffic of the RDRs of all the processors.

The RDR tag of the PACKAGE_USAGE_RDR is **0xf0f0f004 / 4042321924.**

The following table lists the RDR fields and their descriptions.

Table 2-9 Package Usage RDR Fields

RDR Field Name	Type	Description
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.
SERVICE_USAGE_ COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.

Table 2-9 Package Usage RDR Fields (continued)

RDR Field Name	Type	Description
CONFIGURED_DURATION	INT32	See Universal RDR Fields, page 2-2.
DURATION	INT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	INT32	See Universal RDR Fields, page 2-2.
UPSTREAM_VOLUME	INT32	Aggregated upstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period.
SESSIONS	INT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	INT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	INT32	Concurrent number of sessions using the reported service in the reported package at this point in time.
ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers using the reported service in the reported package at this point in time.
TOTAL_ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers in the system at this point in time.

Virtual Links Usage RDR

The VIRTUAL_LINKS_USAGE_RDR is generated periodically, at user-configured intervals, for each service usage counter. A separate RDR is generated for each virtual link. The RDR is generated only if resources associated with the virtual link were consumed during the current reporting period. The RDR contains aggregated network usage information for all subscribers to the same virtual link.

At fixed, user-configurable intervals (for example, every 5 minutes), there is a periodic VIRTUAL_LINKS_USAGE_RDR generation point. Whether or not a Virtual Links Usage RDR is actually generated depends on the following:

- If network resources associated with the service usage counter were consumed by any subscriber of the virtual link since the previous RDR generation point, a Virtual Links Usage RDR is generated.
- If no subscriber of the virtual link has consumed network resources associated with the service usage counter since the previous RDR generation point, no Virtual Links Usage RDR is generated.

  However, the generation logic for Virtual Links Usage RDRs uses the zeroing methodology (as described in Periodic RDR Zero Adjustment Mechanism, page 2-52); if network resources associated with the service usage counter are again consumed by subscribers of the virtual link at some later time, this will cause the immediate generation of either one or two zero-consumption Virtual Links Usage RDRs. (In addition to the eventual generation of the Virtual Links Usage RDR associated with this latest consumption of network resources by subscribers of the virtual link.)

  - If there was only one interval (for example, 0805–0810) for which there was no consumption of network resources by any subscriber of the virtual link, only one zero-consumption Virtual Links Usage RDR is generated.
  - If there were multiple consecutive intervals (for example, 0805–0810, 0810–0815, 0815–0820, 0820–0825) for which there was no consumption of network resources by any subscriber of the virtual link, two zero-consumption Virtual Links Usage RDR are generated: one for the first such time interval (0805–0810) and one for the last (0820–0825).

Note Each traffic processor in the SCE platform generates a separate RDR, where each RDR represents the total traffic processed and analyzed by that processor (for the specified service usage counter and the specified virtual link). To compute the total traffic (for a virtual link) in any given time frame, take the sum of the traffic of the RDRs of all the processors.

The RDR tag of the VIRTUAL_LINKS_USAGE_RDR is **0xf0f0f006 / 4042321926.**

The following table lists the RDR fields and their descriptions.

Table 2-10 Virtual Links Usage RDR Fields

RDR Field Name	Type	Description
VLINK_ID	INT16	The virtual link ID
VLINK_DIRECTION	INT8	The virtual link direction: • 0—Upstream • 1—Downstream
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.

Table 2-10 Virtual Links Usage RDR Fields (continued)

RDR Field Name	Type	Description
SERVICE_USAGE_ COUNTER_ID	UINT16	Each service is mapped to a counter. There are 1024 global usage counters.
CONFIGURED_DURATION	INT32	See Universal RDR Fields, page 2-2.
DURATION	INT32	Not implemented (always the same as CONFIGURED_DURATION).
END_TIME	INT32	See Universal RDR Fields, page 2-2.
UPSTREAM_VOLUME	INT32	Aggregated upstream volume on the virtual link (for a single processor) of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume on the virtual link (for a single processor) of all sessions, in kilobytes, for the current reporting period.
SESSIONS	INT32	Reserved for future use.
SECONDS	INT32	Reserved for future use.
CONCURRENT_SESSIONS	INT32	Reserved for future use.
ACTIVE_SUBSCRIBERS	INT32	Reserved for future use.
TOTAL_ACTIVE_ SUBSCRIBERS	INT32	Concurrent number of subscribers in the system at this point in time.

Blocking RDR

The SERVICE_BLOCK_RDR is generated each time a transaction is blocked, and the profile and the rate/quota limitations indicate that this RDR should be generated.

- A Blocking RDR is generated when a session is blocked. A session may be blocked for various reasons; for example, access is blocked or concurrent session limit is reached.
- Generation of Blocking RDRs is subject to two limitations:
  - Quota—The maximum number of Blocking RDRs that SCA BB can generate for a subscriber in a specific aggregation period (day, week, month, and so forth). The quota is package-dependent; its value is set according to the package assigned to the subscriber.
  - Rate—The global, maximum number of Blocking RDRs that an SCE platform can generate per second. The rate is a global value that sets an upper limit for the total number of RDRs that are generated for all subscribers.

The RDR tag of the SERVICE_BLOCK_RDR is **0xf0f0f040 / 4042321984.**

The following table lists the RDR fields and their descriptions.

Table 2-11 Blocking RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
PROTOCOL_ID	INT16	See Universal RDR Fields, page 2-2.
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
ACCESS_STRING	STRING	See Universal RDR Fields, page 2-2.
INFO_STRING	STRING	See Universal RDR Fields, page 2-2.

Table 2-11 Blocking RDR Fields (continued)

RDR Field Name	Type	Description
BLOCK_REASON	UINT8	Indicates the reason why this session was blocked. For possible values and their interpretation, see Block Reason (uint8), page 2-46.
BLOCK_RDR_COUNT	INT32	Total number of blocked flows reported so far (from the beginning of the current time frame).
REDIRECTED	INT8	Indicates whether the flow has been redirected after being blocked. • 0—Not redirected • 1—Redirected Redirection is performed only for HTTP and RTSP flows that were mapped to a rule ordering them to be blocked and redirected.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.

Quota Breach RDR

The QUOTA_BREACH_RDR is generated each time a bucket is breached for the first time in a session.

This RDR does not have a rate limit; it is generated whenever a quota breach occurs, provided that the RDR is enabled.

This RDR is generated subject to the following conditions:

- One of the Subscriber's buckets was depleted.
- Quota Breach RDRs are enabled.
- This is the first time this subscriber has breached this bucket.

The RDR tag of the QUOTA_BREACH_RDR is **0xf0f0f022 / 4042321954.**

The following table lists the RDR fields and their descriptions.

Table 2-12 Quota Breach RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
BUCKET_ID	UINT8	1 to 16, according to the number of the breached bucket.
END_TIME	INT32	See Universal RDR Fields, page 2-2.
BUCKET_QUOTA	INT32	The remaining quota in the indicated bucket: • Volume bucket—Kilobytes • Number of sessions bucket—Integer
AGGREGATION_PERIOD_TYPE	UINT8	Defines how often the bucket is refilled. Gor possible values and their interpretations, see Periodic RDR Zero Adjustment Mechanism, page 2-52.

Remaining Quota RDR

The REMAINING_QUOTA_RDR is generated periodically, at user-configured intervals, if the RDR is enabled.

Note A Remaining Quota RDR will be generated only for those subscribers *whose policy requires the generation of such an RDR.*

At fixed, user-configurable intervals (for example, every 30 minutes), there is a periodic REMAINING_QUOTA_RDR generation point. If REMAINING_QUOTA_RDRs are enabled, they will be generated at the specified times.

You can set total limit enforcement on the number of these RDRs that are generated per second.

This RDR is also generated after a subscriber performs a logout in a subscriber-integrated installation or is un-introduced from the SCE platform, or when the subscriber's package-ID is changed.

The RDR tag of the REMAINING_QUOTA_RDR is **0xf0f0f030 / 4042321968.**

The following table lists the RDR fields and descriptions.

Table 2-13 Remaining Quota RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
RDR_REASON	UINT8	• 0—Period time passed • 1—Logout • 2—Package switch • 3—Wraparound • 4—End of aggregation period
END_TIME	INT32	See Universal RDR Fields, page 2-2.
REMAINING_QUOTA_1 through REMAINING_QUOTA_16	INT32	The remaining quota in the bucket that was breached, in kilobytes. There are sixteen Remaining Quota fields, one for each bucket.
TOTAL_VOLUME_USAGE	UINT32	Total Volume Usage for all services that are not quota provisioned, in kilobytes, for the current reporting period.

Quota Threshold Breach RDR

The QUOTA_THRESHOLD_BREACH_RDR is generated each time a bucket exceeds the global threshold.

This RDR does not have a rate limit; it is generated whenever a threshold is exceeded, provided that the RDR is enabled.

The RDR tag of the QUOTA_THRESHOLD_BREACH_RDR is **0xf0f0f031 / 4042321969.**

The following table lists the RDR fields and their descriptions.

Table 2-14 Quota Threshold Breach RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
BUCKET_ID	UINT8	1 to 16, according to the number of the breached bucket.
GLOBAL_THRESHOLD	UINT32	The globally configured threshold in kilobytes.
END_TIME	INT32	See Universal RDR Fields, page 2-2.
BUCKET_QUOTA	INT32	The remaining quota in the indicated bucket in kilobytes.

Quota State Restore RDRs

The QUOTA_STATE_RESTORE_RDR is generated each time a subscriber is introduced.

The RDR tag of the QUOTA_STATE_RESTORE_RDR is **0xF0F0F032 / 4042321970.**

The following table lists the RDR fields and their descriptions.

Table 2-15 Quota State Restore RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
RDR_REASON	UINT8	The reason that the RDR was sent: • 0—Subscriber introduced (currently, the only available value)
END_TIME	INT32	See Universal RDR Fields, page 2-2.

DHCP RDR

The DHCP_RDR is generated each time a DHCP message of a specified type is intercepted.

Note DHCP RDRs are generated only if activated by a subscriber integration system, such as the SCMS Subscriber Manager (SM) DHCP LEG.

For each message read, the Cisco Service Control Application for Broadband (SCA BB) extracts several option fields. You can configure which fields to extract. An RDR will be generated even if none of the fields were found.

The RDR tag of the DHCP_RDR is **0xf0f0f042 / 4042321986.**

The following table lists the RDR fields and descriptions.

Table 2-16 DHCP RDR Fields

RDR Field Name	Type	Description
CPE_MAC	STRING	A DHCP protocol field.
CMTS_IP	UINT32	A DHCP protocol field.
ASSIGNED_IP	UINT32	A DHCP protocol field.
RELEASED_IP	UINT32	A DHCP protocol field.
TRANSACTION_ID	UINT32	A DHCP protocol field.
MESSAGE_TYPE	UINT8	DHCP message type.
OPTION_TYPE_0 through OPTION_TYPE_7	UINT8	A list of DHCP options extracted from the message.
OPTION_TYPE_0 through OPTION_TYPE_7	STRING	The values associated with the above DHCP options.
END_TIME	INT32	See Universal RDR Fields, page 2-2.

RADIUS RDR

The RADIUS_RDR is generated each time a RADIUS message of a specified type is intercepted.

Note RADIUS RDRs are generated only if activated by a subscriber integration system, such as the SCMS-SM RADIUS LEG.

For each message read, SCA BB extracts several option fields. You can configure which fields to extract. An RDR will be generated even if none of the fields were found.

The RDR tag of the RADIUS_RDR is **0xf0f0f043 / 4042321987.**

The following table lists the RDR fields and descriptions.

Table 2-17 RADIUS RDR Fields

RDR Field Name	Type	Description
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
RADIUS_PACKET_CODE	UINT8	The type of the RADIUS message intercepted.
RADIUS_ID	UINT8	The RADIUS transaction ID.
ATTRIBUTE_VALUE_1 through ATTRIBUTE_VALUE_20	STRING	Attributes extracted from the message. Sent as string format TLV. The last attribute field filled takes the value 0.

Flow Start RDR

The FLOW_START_RDR is generated when a flow starts, for any flow on packages and services that are configured to generate such an RDR.

This RDR is designed for services and packages where specific, per-transaction RDRs are required (for example, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate. *Configure the generation scheme for this RDR with extra care.*

The RDR tag of the FLOW_START_RDR is **0xf0f0f016 / 4042321942.**

The following table lists the RDR fields and their descriptions.

Table 2-18 Flow Start RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
START_TIME	UINT32	Flow start time.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.
BREACH_STATE	INT8	See Universal RDR Fields, page 2-2.
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.

Flow End RDR

The FLOW_END_RDR is generated when a flow stops, for any flow that generated a FLOW_START_RDR.

This RDR is designed for services and packages where specific, per-transaction RDRs are required (for example, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate. *Configure the generation scheme for this RDR with extra care.*

The RDR tag of the FLOW_END_RDR is **0xf0f0f018 / 4042321944.**

The following table lists the RDR fields and their descriptions.

Table 2-19 Flow End RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
START_TIME	UINT32	Flow start time.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.
BREACH_STATE	INT8	See Universal RDR Fields, page 2-2.
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.

Ongoing Flow RDR

The FLOW_ONGOING_RDR is generated at set time intervals during the life of a flow, for any flow that generated a FLOW_START_RDR, if the system is configured to issue such RDR.

This RDR is designed for services and packages where specific, per-transaction RDRs are required (for example, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate. *Configure the generation scheme for this RDR with extra care.*

The RDR tag of the FLOW_ONGOING_RDR is **0xf0f0f017 / 4042321943.**

The following table lists the RDR fields and their descriptions.

Table 2-20 Ongoing Flow RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
PACKAGE_ID	UINT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	See Universal RDR Fields, page 2-2.
SERVER_PORT	UINT16	See Universal RDR Fields, page 2-2.
CLIENT_IP	UINT32	See Universal RDR Fields, page 2-2.
CLIENT_PORT	UINT16	See Universal RDR Fields, page 2-2.
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2.
START_TIME	UINT32	Flow start time.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.
BREACH_STATE	INT8	See Universal RDR Fields, page 2-2.
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.

Media Flow RDR

The MEDIA_FLOW_RDR is generated at the end of every SIP or Skype media flow:

- For SIP, this RDR is generated when a media channel is closed.
- For Skype, this RDR is generated when an end-of-call is detected.

Note SIP includes all SIP based applications (such as Vonage and Yahoo Messenger VoIP).

The RDR tag of the MEDIA_FLOW_RDR is **0xF0F0F46C / 4042323052.**

The following table lists the RDR fields and their descriptions.

Table 2-21 Media Flow RDR Fields

Field name	Type	Description
SUBSCRIBER_ID	String	See Universal RDR Fields, page 2-2.
PACKAGE_ID	INT16	See Universal RDR Fields, page 2-2.
SERVICE_ID	INT32	See Universal RDR Fields, page 2-2.
PROTOCOL_ID	INT16	See Universal RDR Fields, page 2-2.
DESTINATION_IP	UINT32	SIP: Destination IP address of RTP flow. Skype: Destination IP address of Skype flow.
DESTINATION_PORT	UINT16	SIP: Destination port of RTP flow. Skype: Destination port of Skype flow.
SOURCE_IP	UINT32	SIP: Source IP address of RTP flow. Skype: Source IP address of Skype flow.
SOURCE_PORT	UINT16	SIP: Source port of RTP flow. Skype: Source port of Skype flow.
INITIATING_SIDE	INT8	See Universal RDR Fields, page 2-2. For Skype, this is the initiating side of the flow (not necessarily the initiating side of the voice call).

Table 2-21 Media Flow RDR Fields (continued)

Field name	Type	Description
ZONE_ID	Int32	See Universal RDR Fields, page 2-2.
FLAVOR_ID	Int32	See Universal RDR Fields, page 2-2.
SIP_DOMAIN	String	SIP: Domain name extracted from SIP header.
SIP_USER_AGENT	String	SIP: User-Agent field extracted from SIP header.
START_TIME	UINT32	Flow start time.
REPORT_TIME	UINT32	See Universal RDR Fields, page 2-2.
DURATION_SECONDS	INT32	SIP: The active duration of the RTP flow, not including aging time. Skype: The time between the start-of-call and end-of-call detection events.
UPSTREAM_VOLUME	UINT32	SIP: The upstream volume of the RTP flow. Skype: The upstream volume between the start-of-call and end-of-call detection events.
DOWNSTREAM_VOLUME	UINT32	SIP: The downstream volume of the RTP flow. Skype: The downstream volume between the start-of-call and end-of-call detection events.
IP_PROTOCOL	UINT8	IP protocol type: • 6—TCP • 17—UDP
FLOW_TYPE	INT8	• 0—All Skype flows • 1—Audio (SIP) • 2—Video (SIP)
SESSION_ID	UINT32	SIP: The flow-context ID of the control flow. Skype: The flow-context ID of the flow.

Table 2-21 Media Flow RDR Fields (continued)

Field name	Type	Description
UPSTREAM_JITTER	UINT32	SIP: The average upstream jitter for the session, taken from the RTCP flow: N/A (0xFFFFFFFF) if RTCP flow is missing. Skype: N/A (0xFFFFFFFF).
DOWNSTREAM_JITTER	UINT32	SIP: The average downstream jitter for the session, taken from the RTCP flow: N/A (0xFFFFFFFF) if RTCP flow is missing. Skype: N/A (0xFFFFFFFF).
UPSTREAM_PACKET_LOSS	UINT16	SIP: The average fractional upstream packet loss for the session, taken from the RTCP flow: N/A (0xFFFF) if RTCP flow is missing. Skype: N/A (0xFFFF).
DOWNSTREAM_PACKET_LOSS	UINT16	SIP: The average fractional downstream packet loss for the session, taken from the RTCP flow: N/A (0xFFFF) if RTCP flow is missing. Skype: N/A (0xFFFF).
UPSTREAM_PAYLOAD_TYPE	UINT8	SIP: The upstream RTP payload type for the session. Skype: N/A (0xFF).
DOWNSTREAM_PAYLOAD_TYPE	UINT8	SIP: The downstream RTP payload type for the session. Skype: N/A (0xFF).

Note

Packet Loss
This field is taken from the RTCP field “fraction lost”. It is the average value of all RTCP packets seen during the flow life for the specified direction. The value is the numerator of a fraction whose denominator is 256. To get the packet loss value as percentage, divide this value by 2.56.

Average Jitter
This field is taken from the RTCP field “interval jitter”. The reported value is the average value of all RTCP packets seen during the flow life for the specified direction. This value is multiplied by the NTP time-stamp delta (middle 32 bits) and divided by the RTCP time-stamp delta to convert it to normal time units. These two time stamps are also taken from the RTCP packet. The reported value is the average jitter in units of 1/65536 second. To convert to milliseconds divide by 65.536.

For more information about the RCP/RTCP standard, refer to RFC 1889.

Attack Start RDR

The ATTACK_START_RDR is generated at the beginning of an attack for all attack types that are configured to generate such an RDR. (To enable and configure the generation of these RDRs, see "The Service Security Dashboard" in the "Using the Service Configuration Editor: Additional Options" chapter of the *Cisco Service Control Application for Broadband User Guide*.)

The RDR tag of the ATTACK_START_RDR is **0xf0f0f019 / 4042321945.**

The following table lists the RDR fields and their descriptions.

Table 2-22 Attack Start RDR Fields

RDR Field Name	Type	Description
ATTACK_ID	UINT32	Unique attack ID.
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
ATTACKING_IP	UINT32	The IP address related to the attack (for example: in a DDoS, this will be the IP address under attack; in a scan this will be the IP address of the source of the scan).
ATTACKED_IP	UINT32	The other IP address related to the attack, if one exists; otherwise, 0xFFFFFFFF.
ATTACKED_PORT	UINT16	Attacked port: 0xFFFF if not present.
ATTACKING_SIDE	INT8	On which side of the SCE ATTACKING_IP resides: • 0—Subscriber • 1—Network
IP_PROTOCOL	UINT8	IP protocol type.
ATTACK_TYPE	UINT32	To whom ATTACKING_IP belongs: • 0—Attacked • 1—Attacker
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.
ATTACK_TIME	UINT32	Time since attack started in seconds.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.

Attack End RDR

The ATTACK_END_RDR is generated at the end of an attack for any attack that caused the generation of an ATTACK_START_RDR.

The RDR tag of the ATTACK_END_RDR is **0xf0f0f01a / 4042321946.**

The following table lists the RDR fields and their descriptions.

Table 2-23 Attack End RDR Fields

RDR Field Name	Type	Description
ATTACK_ID	UINT32	Unique attack ID.
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
ATTACKING_IP	UINT32	The IP address related to the attack (for example: in a DDoS, this will be the IP address under attack; in a scan this will be the IP address of the source of the scan).
ATTACKED_IP	UINT32	The other IP address related to the attack, if one exists; otherwise, 0xFFFFFFFF.
ATTACKED_PORT	UINT16	Attacked port: 0xFFFF if not present.
ATTACKING_SIDE	INT8	On which side of the SCE ATTACKING_IP resides: • 0—Subscriber • 1—Network
IP_PROTOCOL	UINT8	IP protocol type.
ATTACK_TYPE	UINT32	To whom ATTACKING_IP belongs: • 0—Attacked • 1—Attacker
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.
ATTACK_TIME	UINT32	Time since attack started in seconds.
REPORT_TIME	INT32	See Universal RDR Fields, page 2-2.

Malicious Traffic Periodic RDR

The MALICIOUS_TRAFFIC_PERIODIC_RDR is generated when an attack is detected, periodically, at user-configured intervals, for the duration of the attack, and at the end of the attack. The MALICIOUS_TRAFFIC_PERIODIC_RDR reports the details of the attack or malicious traffic.

The RDR tag of the MALICIOUS_TRAFFIC_PERIODIC_RDR is **0xf0f0f050 / 4042322000.**

The following table lists the RDR fields and their descriptions.

Table 2-24 Malicious Traffic Periodic RDR Fields

RDR Field Name	Type	Description
ATTACK_ID	INT32	Unique attack ID.
SUBSCRIBER_ID	STRING	See Universal RDR Fields, page 2-2.
ATTACK_IP	UINT32	The IP address related to this attack.
OTHER_IP	UINT32	The other IP address related to this attack, if such exists (if this is a DOS attack), or -1 otherwise.
PORT_NUMBER	UINT16	The port number related to this attack, if such exists (if this is an IP scan, for example), or -1 otherwise.
ATTACK_TYPE	INT32	Who ATTACK_IP belongs to: • 0—Attacked • 1—Attacker
SIDE	INT8	The IP address side: • 0—Subscriber • 1—Network
IP_PROTOCOL	UINT8	IP protocol type: • 0—Other • 1—ICMP • 6—TCP • 17—UDP
CONFIGURED_DURATION	INT32	See Universal RDR Fields, page 2-2.
DURATION	INT32	Indicates the number of seconds that have passed since the previous MALICIOUS_TRAFFIC_RDR.
END_TIME	INT32	See Universal RDR Fields, page 2-2.

Table 2-24 Malicious Traffic Periodic RDR Fields (continued)

RDR Field Name	Type	Description
ATTACKS	INT8	The number of attacks in the current reporting period. Since this report is generated per attack, the value is 0 or 1.
MALICIOUS_SESSIONS	UINT32	Aggregated number of sessions for the reported attack, for the current reporting period. If the SCE platform blocks the attack, this field takes the value -1.

Note You can identify the type of attack (scan, DDOS, or DOS) from Malicious Traffic Periodic RDR data:

Scan—OTHER_IP=-1 and ATTACK_TYPE=1 (the RDR contains the source (attacker) IP address)

DDOS attack—OTHER_IP=-1 and ATTACK_TYPE=0 (the RDR contains the destination (attacked) IP address)

DOS attack—OTHER_IP contains an IP address (the RDR contains two IP addresses)

Information About RDR Enumeration Fields

The following sections list possible values for the RDR enumeration fields.

Block Reason (uint8)

The BLOCK_REASON field is a bit field. The following table lists the meanings of the bits of this field.

Table 2-25 Block Reason Field Bit Values

Bits Number	Value and Description
7 (msb)	Always ON.
6	• 0—The action of the effective rule is block. • 1—The concurrent session limit of the effective rule was reached.
5	• 0—The effective rule was in pre-breach state. • 1—The effective rule was in post-breach state.
4 to 0 (lsb)	The number of the breached bucket (1 to 16).

String Fields

The following table lists the ACCESS_STRING and INFO_STRING field values.

Table 2-26 String Field Values

Name	TR ACCESS_STRING	TR INFO_STRING	Description
PROTOCOL_TCP_ GENERIC	Null	Null	
PROTOCOL_UDP_ GENERIC	Null	Null	
PROTOCOL_HTTP_ BROWSING	Host name	URL	
PROTOCOL_FTP	Null	Null	
PROTOCOL_RTSP	Host name	Null	
PROTOCOL_MMS	Null	Null	
PROTOCOL_SMTP	Server IP	Sender	
PROTOCOL_POP3	Server name	Login name	
PROTOCOL_IP_ GENERIC	Null	Null	Non–TCP/UDP transaction
PROTOCOL_ GNUTELLA_ NETWORKING	Null	Null	Peer to peer
PROTOCOL_ GNUTELLA_FILE_ TRANSFER	Null	Null	Peer to peer
PROTOCOL_ FASTTRACK_ NETWORKING	Null	Null	Peer to peer
PROTOCOL_NNTP	Null	Group name	
PROTOCOL_NAP_ WINMX_TRANSFER	Null	Null	Peer to peer
PROTOCOL_WINNY	Null	Null	Peer to peer
PROTOCOL_ EDONKEY	Null	Null	Peer to peer
PROTOCOL_ DIRECT_CONNECT	Null	Null	Peer to peer
PROTOCOL_ HOTLINE	Null	Null	Peer to peer
PROTOCOL_ DYNAMIC_ SIGNATURE	Null	Null	
PROTOCOL_ MANOLITO	Null	Null	Peer to peer
PROTOCOL_SIP	SIP Method	SIP Domain	

Table 2-26 String Field Values (continued)

Name	TR ACCESS_STRING	TR INFO_STRING	Description
PROTOCOL_ BITTORRENT	Null	Null	Peer to peer
PROTOCOL_SKYPE	Null	Null	Peer to peer
PROTOCOL_VONAGE	SIP Method	SIP Subscriber ID	
PROTOCOL_SHARE	Null	Null	Peer to peer
PROTOCOL_H323	Null	Is FastStart	
PROTOCOL_ SOULSEEK	Null	Null	Peer to peer
PROTOCOL_ITUNES	Null	Null	Peer to peer
PROTOCOL_ FILETOPIA	Null	Null	Peer to peer
PROTOCOL_ NAPSTER	Null	Null	Peer to peer
PROTOCOL_DHCP	Null	Null	
PROTOCOL_MUTE	Null	Null	Peer to peer
PROTOCOL_ NODEZILLA	Null	Null	Peer to peer
PROTOCOL_WASTE	Null	Null	Peer to peer
PROTOCOL_NEONET	Null	Null	Peer to peer
PROTOCOL_MGCP	Null	Null	
PROTOCOL_WAREZ	Null	Null	Peer to peer

Aggregation Period (uint8)

The following table lists the AGG_PERIOD field values.

Table 2-27 AGG_PERIOD Field Values

Name	Value	Description
AGGREGATE_HOURLY	0	Hourly aggregate—Every hour, on the hour.
AGGREGATE_DAILY	1	Daily aggregate—Every day at midnight.
AGGREGATE_WEEKLY	2	Deprecated in 3.0.
AGGREGATE_MONTHLY	3	Deprecated in 3.0.
EXTERNAL_QUOTA_ PROVISION	4	The quota is externally provisioned and managed by a third-party source.

Time Frames (uint16)

The following table lists the TIME_FRAME field values.

Table 2-28 Time Frame Field Values

Name	Value	Description
TIME_FRAME_0 through TIME_FRAME_3	0–3	ID of active time frame. A number from 0 to 3 that indicates the time frame internal index.

RDR Tag Assignment Summary

The following table summarizes RDR tag assignments.

Table 2-29 RDR Tag Assignments

RDR Name	Default Category (explained in the following table)	Tag Value (decimal)	Tag Value (hexa)
SUBSCRIBER USAGE RDR (NUR)	CM-DB (1)	4,042,321,920	F0 F0 F0 00
REALTIME SUBSCRIBER USAGE RDR (SUR)	CM-DB (1)	4,042,321,922	F0 F0 F0 02
PACKAGE USAGE RDR	CM-DB (1)	4,042,321,924	F0 F0 F0 04
LINK USAGE RDR	CM-DB (1)	4,042,321,925	F0 F0 F0 05
VIRTUAL LINK RDR	CM-DB (1)	4,042,321,926	F0 F0 F0 06
TRANSACTION RDR	CM-DB (1)	4,042,321,936	F0 F0 F0 10
TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,000	F0 F0 F4 38
HTTP TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,004	F0 F0 F4 3C
RTSP TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,008	F0 F0 F4 40
VOIP TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,050	F0 F0 F4 6A
BLOCKING RDR	CM-CSV (1)	4,042,321,984	F0 F0 F0 40
QUOTA BREACH RDR	QP (4)	4,042,321,954	F0 F0 F0 22
REMAINING QUOTA RDR	QP (4)	4,042,321,968	F0 F0 F0 30
QUOTA THRESHOLD RDR	QP (4)	4,042,321,969	F0 F0 F0 31
QUOTA STATE RESTORE RDR	QP (4)	4,042,321,970	F0 F0 F0 32
RADIUS RDR	SM (3)	4,042,321,987	F0 F0 F0 43
DHCP RDR	SM (3)	4,042,321,986	F0 F0 F0 42
FLOW START RDR	RT (2)	4,042,321,942	F0 F0 F0 16
FLOW END RDR	RT (2)	4,042,321,944	F0 F0 F0 18
MEDIA FLOW RDR	CM-DB (1)	4,042,323,052	F0 F0 F4 6C
FLOW ONGOING RDR	RT (2)	4,042,321,943	F0 F0 F0 17
ATTACK_START RDR	RT (2)	4,042,321,945	F0 F0 F0 19

Table 2-29 RDR Tag Assignments (continued)

RDR Name	Default Category (explained in the following table)	Tag Value (decimal)	Tag Value (hexa)
ATTACK_END RDR	RT (2)	4,042,321,946	F0 F0 F0 1A
MALICIOUS TRAFFIC RDR	DC-DB (1)	4,042,322,000	F0 F0 F0 50

RDR categories are the mechanism by which different types of RDRs can be sent to different collectors. You can configure the RDR categories using the SCE CLI. For more information, see the "Raw Data Formatting: The RDR Formatter and NetFlow Exporting" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide.*

Table 2-30 RDR Tag Default Categories

Default Category	Intended Destination and Use
CM-DB (1)	The CM database. Used by the SCA Reporter to generate reports.
CM-CSV (1)	The CM. Stored as CSV files.
RT (2)	Other network devices. Typically used for functionality that requires a real-time response, such as QoS, provisioning, and deletion.
SM (3)	SM's DHCP and RADIUS legs.
QP (4)	External quota provisioning systems. Used as notifications of the SCE Subscribers API.

Periodic RDR Zero Adjustment Mechanism

The Periodic RDRs (or Network Usage RDRs) include the Link Usage, Package Usage, and Real-Time Subscriber Usage RDRs. When there is traffic for a particular service or package, the appropriate Usage RDRs are generated periodically, according to user-configured intervals. The RDR includes a time stamp of the end of the interval during which the traffic was recorded.

When there is *no* traffic (and therefore no consumed resources) for a particular service or package during a given period of time, the SCA BB application uses the Periodic RDR Zero Adjustment Mechanism, also called the zeroing methodology, to reduce the number of Usage RDRs generated for that service or package. This technique also simplifies collection for external systems by reducing the number of RDRs that they need to handle.

Note Unlike other Usage RDRs, the generation logic for Subscriber Usage RDRs does *not* use the zeroing methodology.

The zeroing methodology algorithm works as follows: for any number of consecutive time intervals having no traffic for a particular service or package, zero-consumption RDRs are generated for the first and last zero-consumption time intervals, but not for the intermediate time intervals. These two zero-consumption RDRs are generated when the next traffic arrives.

Example 1

The Real-Time Subscriber Usage RDR (for a given subscriber) has a generation period of 30 minutes. There is subscriber traffic during the interval 1200–1230, no subscriber traffic during the following five intervals (1230–1300, 1300–1330, 1330–1400, 1400–1430, 1430–1500), and the next subscriber traffic occurs at 1522. The following Real-Time Subscriber Usage RDRs are generated:

- At 1230, one RDR with the values of the consumed resources for the interval 1200–1230, and with the time stamp 1230.
- At 1522, one zero-consumption RDR having the time stamp (1300) of the end of the first interval (1230–1300) with no traffic for that subscriber.
- At 1522, one zero-consumption RDR having the time stamp (1500) of the end of the last interval (1430–1500) with no traffic for that subscriber.

  No RDR is generated for the three intermediate zero-consumption intervals (1300–1330, 1330–1400, and 1400–1430).
- At 1530, one RDR with the values of the consumed resources for the interval 1500–1530, and with the time stamp 1530.

Example 2

The Real-Time Subscriber Usage RDR (for a given subscriber) has a generation period of 30 minutes. There is subscriber traffic during the interval 1200–1230, no subscriber traffic during the following interval 1230–1300, and the next subscriber traffic occurs at 1322. The following Real-Time Subscriber Usage RDRs are generated:

- At 1230, one RDR with the values of the consumed resources for the interval 1200–1230, and with the time stamp 1230.
- At 1322, one zero-consumption RDR having the time stamp (1300) of the single interval (1230–1300) with no traffic for that subscriber.
- At 1330, one RDR with the values of the consumed resources for the interval 1300–1330, and with the time stamp 1330.