



CHAPTER 2

Raw Data Records: Formats and Field Contents

Revised: November 23, 2012, OL-21066-04

Introduction

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, page 2-2](#)
- [Universal RDR Fields, page 2-2](#)
- [ADDITIONAL_INFO Field, page 2-4](#)
- [Transaction RDR, page 2-4](#)
- [Transaction Usage RDR, page 2-7](#)
- [HTTP Transaction Usage RDR, page 2-10](#)
- [Anonymized HTTP Transaction Usage RDR, page 2-14](#)
- [RTSP Transaction Usage RDR, page 2-16](#)
- [VoIP Transaction Usage RDR, page 2-19](#)
- [Video Transaction Usage RDR, page 2-23](#)
- [Generic Usage RDR, page 2-26](#)
- [Using the Generic Usage RDR to Report IPv6 Usage, page 2-29](#)
- [Subscriber Usage RDR, page 2-30](#)
- [Real-Time Subscriber Usage RDR, page 2-33](#)
- [Link Usage RDR, page 2-36](#)
- [Zone Usage RDR, page 2-39](#)
- [Package Usage RDR, page 2-41](#)
- [Virtual Links Usage RDR, page 2-43](#)
- [Blocking RDR, page 2-45](#)
- [Quota Breach RDR, page 2-47](#)
- [Quota Status RDR, page 2-50](#)

- [Quota Threshold Breach RDR, page 2-54](#)
- [Session Creation RDRs, page 2-57](#)
- [DHCP RDR, page 2-59](#)
- [RADIUS RDR, page 2-60](#)
- [Flow Start RDR, page 2-61](#)
- [Flow End RDR, page 2-63](#)
- [Ongoing Flow RDR, page 2-65](#)
- [Media Flow RDR, page 2-67](#)
- [Attack Start RDR, page 2-71](#)
- [Attack End RDR, page 2-72](#)
- [Malicious Traffic Periodic RDR, page 2-73](#)
- [Spam RDR, page 2-75](#)
- [Information About RDR Enumeration Fields, page 2-77](#)
- [RDR Tag Assignment Summary, page 2-81](#)
- [Periodic RDR Zero Adjustment Mechanism, page 2-83](#)

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 alphabetical order.

- SUBSCRIBER_ID—Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
- PACKAGE_ID—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—Layer 7 property, extracted from the transaction. For possible values see [String Fields, page 2-78](#).
- BREACH_STATE—Indicates whether the subscriber's quota was breached.
 - 0—Not breached
 - 1—Breached

- **CLIENT_IP**—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**—Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
- **CONFIGURED_DURATION**—Configured period, in seconds for periodic RDRs, 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**—ID of the flavor associated with this session. This is for protocol signatures that have flavors.
- **INFO_STRING**—Layer 7 property extracted from the transaction. For possible values see [String Fields, page 2-78](#).
- **INITIATING_SIDE**—Side of the SCE platform on which the initiator of the transaction resides.
 - 0—The subscriber side
 - 1—The network side
- **PROTOCOL_ID**—Unique ID of the protocol associated with the reported session.



Note The **PROTOCOL_ID** is the Generic IP / Generic TCP / Generic UDP protocol ID Note 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**—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**—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**—Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
- **SERVICE_ID**—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**—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. The system supports time-dependent policies, by using different rules for different time frames.
- **ZONE_ID**—ID of the zone associated with this session.



Note All volumes in RDRs are reported in Layer 3 bytes.

Related Topics

- [String Fields, page 2-78](#)

ADDITIONAL_INFO Field

This bit map field supplies additional information regarding subscriber, event or system configuration.

Table 2-1 *ADDITIONAL_INFO Field Definition*

Bit number (LSB =0)	Bit Value	Description
0	1	Anonymous subscriber
0	0	Introduced subscriber
1	1	Tariff change report
1	0	No tariff change
2-4	1	Re-authorization
2-4	2	Quota Holding Time Expired
2-4	4	Quota Validity Time Expired
5	1	More RDRs follows
5	0	No RDRs follows
6	1	Final RDR
7-10	Volume units	Number of bytes of each unit. This number is a power of 2. For example, 0 indicates bytes, 10 (2 ¹⁰) indicates KB.
11-31	0	Reserved

Transaction RDR

This section contains descriptions of transaction RDRs

- **RDR Purpose**—Analyzes a sampling of network transactions in order to estimate the network's behavior based on statistics.
- **RDR Default destination**—Sent to the CM, inserted into the database, and used by the Reporter tool for statistical reports, such as the Traffic Discovery report.
- **RDR Content**—Describes a single transaction; its connection attributes, extracted L7 attributes, duration and volume.
- **RDR Generation Logic**—Generated at the end of a session, according to a configurable sampling mechanism, you configure the number-of-transaction-RDRs-per-second which sets the number of Transaction RDRs (TRs) generated per-second.

The Transaction RDR is not generated for sessions that were blocked by a rule.

You can disable TRs, which invalidates TR-based reports.

See the Sizing Tool for the appropriate sample rate; a sample rate which is too high may cause CM sizing problems. A sample rate which is too low reduces the accuracy of TR-based reports.

- **RDR tag**— 0xf0f0f010 / 4042321936

Table 2-2 lists the Transaction RDR fields and their descriptions.

Table 2-2 Transaction RDR Fields

RDR Field Name	Type	Description	Example Value
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.	john
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and <code>maximum_number_of_packages</code> . The value <code>maximum_number_of_packages</code> is reserved for unknown subscribers.	0 [Default Package]
SERVICE_ID	INT32	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.	16 [HTTP]
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.	2 [HTTP]
SKIPPED_SESSIONS	UINT32	Number of unreported sessions since the previous RDR <i>plus one</i> . The default value is 1. A value of 2 means that <i>one</i> RDR was unreported.	10
SERVER_IP	UINT32	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.	198.133.219.25
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.	80
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.	www.cisco.com
INFO_STRING	STRING	Layer 7 property extracted from the transaction.	/en/US/partner/

Table 2-2 Transaction RDR Fields (continued)

RDR Field Name	Type	Description	Example Value
CLIENT_IP	UINT32	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.	192.118.76.130
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.	3221
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side 	0 [subscriber-initiated]
REPORT_TIME	UINT32	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.	—
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.	310
TIME_FRAME	INT8	Time frame during which the RDR was generated. The field's value is in the range 0 to 3, indicating which of the four time frames was used. The system supports time-dependent policies, by using different rules for different time frames.	0
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.	32
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.	117
SUBSCRIBER_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.	1
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.	9

Table 2-2 Transaction RDR Fields (continued)

RDR Field Name	Type	Description	Example Value
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.	0
IP_PROTOCOL	UINT8	IP protocol type.	6 [TCP]
PROTOCOL_SIGNATURE	INT32	ID of the protocol signature associated with this session.	0x3010000 [HTTP]
ZONE_ID	INT32	ID of the zone associated with this session.	0
FLAVOR_ID	INT32	ID of the flavor associated with this session.	0
FLOW_CLOSE_MODE	UINT8	Reason for the end of flow. <ul style="list-style-type: none"> 0 [TCP_NORMAL_CLOSE] 2 [The flow was closed by the aging mechanism.] 	0

Related Topics

- [Universal RDR Fields, page 2-2](#)

Transaction Usage RDR

- RDR Purpose—Log network transactions for transaction-based billing or offline data mining.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single transaction; its connection attributes, extracted L7 attributes, duration and volume.
- RDR Generation Logic—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.

- RDR tag—0xf0f0f438 / 4042323000

By default, packages and services are disabled from generating this RDR. They can be enabled for specific packages and services. You can disable generating Transaction Usage RDRs (TURs) for very short flows by setting a volume threshold. You can enable generating interim TURs for very long transactions.

The Transaction Usage RDR is designed for services and packages where specific, per-transaction RDRs are required (such as, 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. See the Sizing Tool.

**Note**

Configure the generation scheme for this RDR with extra care.

[Table 2-3](#) lists the Transaction Usage RDR fields and their descriptions.

Table 2-3 Transaction Usage RDR Fields

RDR Field Name	Type	Description	Example Value
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.	john
PACKAGE_ID	INT16	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.	0 [Default Package
SERVICE_ID	INT32	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.	16 [HTTP]
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.	2 [HTTP]
SKIPPED_SESSIONS	UINT32	Reason for RDR generation: <ul style="list-style-type: none"> • 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 	1 [SESSION_END]
SERVER_IP	UINT32	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.	198.133.219.25
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.	80

Table 2-3 Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description	Example Value
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.	www.cisco.com
INFO_STRING	STRING	Layer 7 property extracted from the transaction.	/en/US/partner/
CLIENT_IP	UNIT32	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.	192.118.76.130
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.	3221
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side 	0 [subscriber-initiated]
REPORT_TIME	UINT32	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.	—
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.	310
TIME_FRAME	INT8	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. The system supports time-dependent policies, by using different rules for different time frames.	0
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.	32
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.	117

Table 2-3 Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description	Example Value
SUBSCRIBER_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.	1
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.	9
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.	0
IP_PROTOCOL	UINT8	IP protocol type.	6 [TCP]
PROTOCOL_SIGNATURE	INT32	ID of the protocol signature associated with this session.	0x3010000 [HTTP]
ZONE_ID	INT32	ID of the zone associated with this session.	0
FLAVOR_ID	INT32	ID of the flavor associated with this session.	0
FLOW_CLOSE_MODE	UINT8	Reason for the end of flow. <ul style="list-style-type: none"> 0 [TCP_NORMAL_CLOSE] 2 [The flow was closed by the aging mechanism.] 	0

Related Topics

- [Universal RDR Fields, page 2-2](#)

HTTP Transaction Usage RDR

The *HTTP_TRANSACTION_USAGE_RDR* is a TUR specifically used for HTTP transactions.

- RDR Purpose—Log HTTP network transactions for transaction-based billing or offline data mining.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single HTTP transaction; its connection attributes, extracted L7 attributes, duration, and volume.
- RDR Generation Logic—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.

- RDR tag—0xf0f0f43C / 4042323004

By default, packages and services are disabled from generating this RDR. You can enable them for specific packages and services.

This RDR is designed for services and packages where specific, per-transaction RDRs are required (such as, 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.

**Note**

Configure the generation scheme for this RDR with extra care.

Table 2-4 lists the HTTP Transaction Usage RDR fields and their descriptions.

Table 2-4 HTTP Transaction Usage RDR Fields

RDR Field Name	Type	Description	Example Value
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.	john
PACKAGE_ID	INT16	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.	0 [Default Package]
SERVICE_ID	INT32	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.	16 [HTTP]
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.	2 [HTTP]
SKIPPED_SESSIONS	UINT32	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.	1 [SESSION_END]
SERVER_IP	UINT32	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.	198.133.219.25
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.	80
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.	www.cisco.com

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

RDR Field Name	Type	Description	Example Value
INFO_STRING	STRING	Layer 7 property extracted from the transaction.	/en/US/partner/
CLIENT_IP	UINT32	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.	192.118.76.130
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.	3221
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> 0—Subscriber side 1—Network side 	0 [subscriber-initiated]
REPORT_TIME	UINT32	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.	—
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.	310
TIME_FRAME	INT8	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. The system supports time-dependent policies, by using different rules for different time frames.	0
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.	32
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.	117
SUBSCRIBER_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.	1

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

RDR Field Name	Type	Description	Example Value
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.	9
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.	0
IP_PROTOCOL	UINT8	IP protocol type.	6 [TCP]
PROTOCOL_SIGNATURE	INT32	ID of the protocol signature associated with this session.	0x3010000 [HTTP]
ZONE_ID	INT32	ID of the zone associated with this session.	0
FLAVOR_ID	INT32	ID of the flavor associated with this session.	0
FLOW_CLOSE_MODE	UINT8	Reason for the end of flow. <ul style="list-style-type: none"> 0 [TCP_NORMAL_CLOSE] 2 [The flow was closed by the aging mechanism.] 	0
USER_AGENT	STRING	User agent field extracted from the HTTP transaction.	Moselle
HTTP_URL	STRING	URL extracted from the HTTP transaction.	/en/US/partner/
HTTP_REFERER	STRING	REFERER extracted from the HTTP transaction.	http://addition.cnn.com
HTTP_COOKIE	STRING	COOKIE extracted from the HTTP transaction.	SelectedAddition=Addition;CNNid=3459286729-09

Related Topics

- [Universal RDR Fields, page 2-2](#)

Anonymized HTTP Transaction Usage RDR

The *ANONYMIZED_HTTP_TRANSACTION_USAGE_RDR* is a TUR specifically used for HTTP transactions.

- RDR Purpose—Log HTTP network transactions for transaction-based billing or offline data mining without personal subscriber data.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single HTTP transaction; its connection attributes, extracted L7 attributes, duration, and volume.
- RDR Generation Logic—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.

- RDR tag—0xf0f0f53C / 4042323260

By default, packages and services are disabled from generating this RDR. You can enable them for specific packages and services.

This RDR is designed for services and packages where specific, per-transaction RDRs are required (such as, 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.



Note

Configure the generation scheme for this RDR with extra care.

Table 2-5 lists the RDR fields and their descriptions.

Table 2-5 Anonymized HTTP Transaction Usage RDR Fields

RDR Field Name	Type	Description
HASHED_SUBSCRIBER_ID	STRING	Subscriber identification string, may be passed through hashing algorithm.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported.
SERVICE_ID	INT32	Service classification of the reported session.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
SKIPPED_SESSIONS	UINT32	Always 1.
SERVER_IP	UINT32	HTTP server IP. If this is the subscriber IP, this field may contain the short-hash of the IP if configured.
SERVER_PORT	UINT16	Destination port number of the networking session.
HOST	STRING	Host extracted from the HTTP transaction.
URL	STRING	URL extracted from the HTTP transaction.
CLIENT_IP	UINT32	HTTP client IP. If this is the subscriber IP, this field may contain the short-hash of the IP if configured.

Table 2-5 *Anonymized HTTP Transaction Usage RDR Fields (continued)*

RDR Field Name	Type	Description
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the networking session.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0–The subscriber side • 1–The network side
REPORT_TIME	UINT32	Ending time stamp of this RDR.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	Time frame during which the RDR was generated. (0 – 3)
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	Counter to which each service is mapped. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	UINT32	ID of the protocol signature associated with this session.
ZONE_ID	UINT32	ID of the zone associated with this session.
FLAVOR_ID	UINT32	ID of the flavor associated with this session.
FLOW_CLOSE_MODE	UINT8	Reason for the end of flow.
HASHED_SUBSCRIBER_IP	STRING	Subscriber IP, may be hashed if configured.
USER_AGENT	STRING	User agent field extracted from the HTTP transaction.
HTTP_REFERER	STRING	REFERER extracted from the HTTP transaction.
HTTP_COOKIE	STRING	COOKIE extracted from the HTTP transaction.

RTSP Transaction Usage RDR

The *RTSP_TRANSACTION_USAGE_RDR* is a TUR specifically used for RTSP Transactions.

- RDR Purpose—Log RTSP network transactions for transaction-based billing or offline data mining.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single RTSP transaction; its connection attributes, extracted L7 attributes, duration, and volume.
- RDR Generation Logic—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.

- RDR tag—0xf0f0f440 / 4042323008

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 (such as, 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.



Note

Configure the generation scheme for this RDR with extra care.

Table 2-6 lists the RTSP Transaction Usage RDR fields and their descriptions.

Table 2-6 RTSP Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and <code>maximum_number_of_packages</code> . The value <code>maximum_number_of_packages</code> is reserved for unknown subscribers.
SERVICE_ID	INT32	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.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
SKIPPED_SESSIONS	UINT32	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.

Table 2-6 RTSP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
SERVER_IP	UINT32	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	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.
CLIENT_IP	UINT32	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	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side
REPORT_TIME	UINT32	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.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	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.
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	Counter to which each service is mapped. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.

Table 2-6 RTSP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.
IP_PROTOCOL	UNIT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	ID of the protocol signature associated with this session.
ZONE_ID	INT32	ID of the zone associated with this session.
FLAVOR_ID	INT32	ID of the protocol signature with flavor associated with this session.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow. <ul style="list-style-type: none"> • 0 [TCP_NORMAL_CLOSE] • 2 [The flow was closed by the aging mechanism.]
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.
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.

Related Topics

- [Universal RDR Fields, page 2-2](#)

VoIP Transaction Usage RDR

The *VOIP_TRANSACTION_USAGE_RDR* is a TUR specifically used for VoIP transactions.

- RDR Purpose—Log VOIP network transactions for transaction-based billing or offline data mining.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single RTSP transaction; its connection attributes, extracted Layer 7 attributes, duration, and volume.
- RDR Generation Logic—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.

- RDR tag—0xf0f0f46a / 4042323050

By default, packages and services are *disabled* from generating this RDR. You can enable them for specific packages and services.

The VoIP Transaction Usage RDR is enabled automatically when the Transaction Usage RDR is enabled; both RDRs are 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.



Note

Configure the generation scheme for this RDR with extra care.

Table 2-7 lists the VoIP Transaction Usage RDR fields and their descriptions.

Table 2-7 VoIP Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and <code>maximum_number_of_packages</code> . The value <code>maximum_number_of_packages</code> is reserved for unknown subscribers.
SERVICE_ID	INT32	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.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.

Table 2-7 VoIP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
SKIPPED_SESSIONS	UINT32	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	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	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.
CLIENT_IP	UINT32	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	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side
REPORT_TIME	UINT32	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.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	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. The system supports time-dependent policies, by using different rules for different time frames.
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	Counter to which each service is mapped. There are 32 subscriber usage counters.

Table 2-7 VoIP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	ID of the protocol signature associated with this session.
ZONE_ID	INT32	ID of the zone associated with this session.
FLAVOR_ID	INT32	ID of the protocol signatures with flavor associated with this session.
FLOW_CLOSE_MODE	UINT8	The ITU-U vendor ID of the application. A value of 0xFFFFFFFF indicates that this field was not found in the traffic.
APPLICATION_ID	UINT32	ITU-U vendor ID of the application. A value of 0xFFFFFFFF indicates that this field was not found in the traffic.
UPSTREAM_PACKET_LOSS (see Note, page 22)	UINT16	Average fractional upstream packet loss for the session, taken from the RTCP flow. (See 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 (see Note, page 22)	UINT16	Average fractional downstream packet loss for the session, taken from the RTCP flow. (See 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 (see Note, page 22)	UINT32	Average upstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (See 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 (see Note, page 22)	UINT32	Average downstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (See 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	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	Q931 Alias address of the session source. A value of N/A indicates that this field was not found in the traffic.

Table 2-7 VoIP Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
UPSTREAM_PAYLOAD_TYPE	UINT8	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	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	Call type (taken from H225 packet). A value of 0xFF indicates that this field is undefined (no RTP flows were opened).
MEDIA_CHANNELS	UINT8	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.

**Note****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, see RFC 1889.

Related Topics

- [Universal RDR Fields, page 2-2](#)

Video Transaction Usage RDR

The *VIDEO_TRANSACTION_USAGE_RDR* is a TUR used specifically for video transactions.

- RDR Default destination—Sent to the CM and stored in CSV format
- RDR Content—Describes a single video transaction
- RDR Generation Logic—Generated at the end of a session, for all transactions on all packages and all services if:
 - The packages and services are configured to generate the *VIDEO_TRANSACTION_USAGE_RDR*.
 - *VIDEO_TRANSACTION_USAGE_RDRs* are enabled.

This RDR is not generated for sessions that were blocked by a rule.

- RDR tag-0xf0f480 / 4042323072

By default, packages and services are disabled from generating this RDR. You can enable them for specific packages and services.

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.



Note

Configure the generation scheme for this RDR with extra care.

[Table 2-8](#) lists the *VIDEO_TRANSACTION_USAGE_RDR* fields and their descriptions.

Table 2-8 Video Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and <i>maximum_number_of_packages</i> . The value <i>maximum_number_of_packages</i> is reserved for unknown subscribers.
SERVICE_ID	INT32	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.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.

Table 2-8 Video Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
SKIPPED_SESSIONS	UINT32	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	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	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.
CLIENT_IP	UINT32	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	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side
REPORT_TIME	UINT32	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.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	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. The system supports time-dependent policies, by using different rules for different time frames.
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	Counter to which each service is mapped. There are 32 subscriber usage counters.

Table 2-8 Video Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.
IP_PROTOCOL	UNIT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	ID of the protocol signature associated with this session.
ZONE_ID	INT32	ID of the zone associated with this session.
FLAVOR_ID	INT32	ID of the protocol signatures with flavor associated with this session.
FLOW_CLOSE_MODE	UINT8	ITU-U vendor ID of the application. A value of 0xFFFFFFFF indicates that this field was not found in the traffic.
TITLE	STRING	Not supported.
DURATION	UINT32	Not supported.
ENCODING_RATE	UINT32	Not supported.
RESOLUTION	UINT32	Not supported.
REFERER	STRING	Not supported.

Generic Usage RDR

GENERIC_USAGE_RDR has a fixed structure with a unique tag, which allows the one-time creation of a database table to be used for various future RDRs.

The Generic Usage RDR is composed of universal fields like any other RDR, generic fields for all GUR RDRs, and fields for future use.

- RDR Purpose—Provides a generic template from which other Usage RDRs can be created.
- RDR Default destination—Varies depending on the specific Usage RDR created from this template
- RDR Content—Varies depending on the specific Usage RDR created from this template.
- RDR Generation Logic—Not generated, is provided as a template for creating other RDRs.
- RDR tag—0xf0f0f090 / 4042322064

Table 2-9 lists the Generic Usage RDR fields and their descriptions.

Table 2-9 **Generic Usage RDR**

Key/Data	RDR Field Name	Type	Description
Key	GUR_TYPE	INT32	Type of the GUR—defines the usage of the rest of the fields
Key	LINK_ID	INT8	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.
Key	GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR. Possible values are 0 to 3.
Key	GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters
Key	SUBSCRIBER_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.
Key	PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.
Key	SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
Key	PACKAGE_ID	INT16	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.
Key	SERVICE_ID	INT32	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.
Key	PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
Key	SIGNATURE_ID	INT32	ID of the protocol signature associated with this session.

Table 2-9 Generic Usage RDR

Key/Data	RDR Field Name	Type	Description
Key	DESTINATION_IP	UINT32	<ul style="list-style-type: none"> • SIP: Destination IP address of RTP flow. • Skype: Destination IP address of Skype flow.
Key	DESTINATION_PORT	UINT16	<ul style="list-style-type: none"> • SIP: Destination port of RTP flow. • Skype: Destination port of Skype flow.
Key	SOURCE_IP	UINT32	<ul style="list-style-type: none"> • SIP: Source IP address of RTP flow. • Skype: Source IP address of Skype flow.
Key	SOURCE_PORT	UINT16	<ul style="list-style-type: none"> • SIP: Source port of RTP flow. • Skype: Source port of Skype flow.
Key	INITIATING_SIDE	INT8	<p>Side of the SCE platform on which the initiator of the transaction resides.</p> <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side <p>For Skype, this is the initiating side of the flow (not necessarily the initiating side of the voice call).</p>
Key	ZONE_ID	INT32	ID of the zone associated with this session.
Key	FLAVOR_ID	INT32	ID of protocol signatures with flavor associated with this session.
Key	SESSION_ID	UINT32	<ul style="list-style-type: none"> • SIP: The flow-context ID of the control flow. • Skype: The flow-context ID of the flow.
Key	START_TIME	UINT32	Flow start time.
Key	END_TIME	UINT32	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.
Key	ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
Key	INFO_STRING	STRING	Layer 7 property extracted from the transaction.
Key	For future use	INT32	—
Key	For future use	INT32	—
Key	For future use	STRING	—
Key	For future use	STRING	—
Data	UPSTREAM_VOLUME	INT32	Aggregated upstream volume of all sessions, in kilobytes, for the current reporting period.
Data	DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume of all sessions, in kilobytes, for the current reporting period.
Data	TOTAL_VOLUME	INT32	Aggregated total volume of all sessions, in kilobytes, for the current reporting period.
Data	SESSIONS	INT32	Aggregated number of sessions for the reported service for the current reporting period.
Data	SECONDS	INT32	Aggregated number of session seconds for the reported service for the current reporting period.

Table 2-9 Generic Usage RDR

Key/Data	RDR Field Name	Type	Description
Data	CONCURRENT_SESSIONS	INT32	Concurrent number of sessions using the reported service at this point in time.
Data	ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers using the reported service at this point in time.
Data	TOTAL_ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers in the system at this point in time.
Data	CONFIGURED_DURATION	INT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
Data	DURATION	INT32	<ul style="list-style-type: none"> This release—Not implemented (always the same as CONFIGURED_DURATION). Future releases—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
Data	For future use	INT32	—
Data	For future use	INT32	—
Data	For future use	INT32	—
Data	For future use	INT32	—

Using the Generic Usage RDR to Report IPv6 Usage

The GUR is used to report both pure-IPv6, and tunneled IPv6. The former is reported per device, and the latter per RUC.

Both reports use the GUR type '1'.

- RDR Generation Logic— based on the user defined configuration of the Link Usage Report.

Table 2-10 describes the specific fields of the pure-IPv6 and tunneled-IPv6 reports. (Any GUR fields not listed in the table are not used.)

Table 2-10 **Generic Usage RDR Fields for IPv6 Usage**

GUR fields	Fields for pure IPv6	Fields (for tunneled IPv6
GUR_TYPE	IPV6_TYPE (0x00000001)	IPV6_TYPE (0x00000001)
LINK_ID	—	LINK_ID
GENERATOR_ID	GENERATOR_ID	GENERATOR_ID
GLOBAL_COUNTER_ID	—	GLOBAL_COUNTER_ID
END_TIME	END_TIME	END_TIME
For future use	PURE_IPV6 (0x00000001)	TUNNELED_IPV6 (0x00000002)
UPSTREAM_VOLUME	—	UPSTREAM_VOLUME
DOWNSTREAM_VOLUME	—	DOWNSTREAM_VOLUME
TOTAL_VOLUME	TOTAL_VOLUME	TOTAL_VOLUME
SESSIONS	—	SESSIONS
SECONDS	—	SECONDS
CONCURRENT_SESSIONS	—	CONCURRENT_SESSIONS
ACTIVE_SUBSCRIBERS	—	ACTIVE_SUBSCRIBERS
TOTAL_ACTIVE_SUBSCRIBERS	—	TOTAL_ACTIVE_SUBSCRIBERS
CONFIGURED_DURATION	CONFIGURED_DURATION	CONFIGURED_DURATION
DURATION	DURATION	DURATION

Subscriber Usage RDR

The SUBSCRIBER_USAGE_RDR summarizes the activity of a single subscriber on a specific service for the last user-configured number of minutes.

- RDR Purpose—Compare subscribers for the Top Subscribers report, and create daily subscriber usage summary records.
- RDR Default destination—Sent to the CM, and processed by the Topper Adapter, which stores the processing results in the database and in CSV files. The Reporter tool uses the database records for creating the Top Subscribers reports.
- RDR Content—Summary of the activity of a single subscriber on a defined service for the last user-configured number of minutes, including aggregated number of flows, total volume, and duration.
- RDR Generation Logic—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.

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.
- RDR tag—0xf0f0f000 / 4042321920

The Subscriber Usage RDRs are enabled by default. Disabling the RDRs disables Top Subscriber reports. The default interval for SUR is every 10 minutes.

The default total rate is 200 SURs per second. Consult the sizing tool for the appropriate interval and rate.

Table 2-11 lists the Subscriber Usage RDR fields and their descriptions.

Table 2-11 **Subscriber Usage RDR**

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	The subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	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.
SERVICE_USAGE_COUNTER_ID_	UINT16	Counter to which each service is mapped. There are 32 counters in the subscriber scope.
BREACH_STATE	UINT8	Indicates whether the subscriber's quota was breached. <ul style="list-style-type: none"> • 0—Not breached • 1—Breached 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: <ul style="list-style-type: none"> • 0—Period time passed • 1—Subscriber logout • 3—Wraparound • 5—Subscriber VLink change
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	Indicates the number of seconds that have passed since the previous Subscriber Usage RDR.
END_TIME	UINT32	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.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT32	Aggregated number of session seconds for the reported service, for the current reporting period.

Table 2-11 **Subscriber Usage RDR (continued)**

RDR Field Name	Type	Description
UP_VLINK	INT16	Up vlink the subscriber is mapped to. (Is valid only in CMTS-aware mode.)
DOWN_VLINK	INT16	Down vlink the subscriber is mapped to. (Is valid only in CMTS-aware mode.)

Related Topics

- [Periodic RDR Zero Adjustment Mechanism, page 2-83](#)

Real-Time Subscriber Usage RDR

The `REALTIME_SUBSCRIBER_USAGE_RDR` summarizes the activity of a single subscriber on a specific service for the last user-configured number of minutes.

- **RDR Purpose**—Create detailed subscriber-level reports of network usage per service.
- **RDR Default destination**—Sent to the CM, stored in the database, and used by the Reporter tool for subscriber usage reports such as the Subscriber Bandwidth per Service report.
- **RDR Content**—Summary of the activity of a single subscriber on a specific service for the last user-configured number of minutes, including aggregated number of flows, total volume, and duration.
- **RDR Generation Logic**—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 is 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 *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 2-83](#)). If the subscriber consumes resources associated with the service usage counter at some later time, this causes 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 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.

A zero-consumption Real-Time Subscriber Usage RDR is 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.
- RDR tag—0xf0f0f002 / 4042321922

Real-Time Subscriber Usage RDRs (RTSUR) are generated only for those subscribers with real-time monitoring enabled. By default, it is disabled for all subscribers. The default interval is RTSUR every 1 minute. The default total rate is 100 RTSURs per second. See the Sizing Tool for the appropriate interval, rate, and the number of subscribers for which you should enable it.

Table 2-12 lists the Real-Time Subscriber Usage RDR fields and their descriptions.

Table 2-12 Real-Time Subscriber Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	The subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	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_package. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_USAGE_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 counters in the subscriber scope.
AGGREGATION_OBJECT_ID	INT16	Externally assigned: <ul style="list-style-type: none"> • 0—Offline subscriber • 1—Online subscriber

Table 2-12 Real-Time Subscriber Usage RDR Fields (continued)

RDR Field Name	Type	Description
BREACH_STATE	UINT8	Indicates whether the subscriber's quota was breached. <ul style="list-style-type: none"> 0—Not breached 1—Breached 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: <ul style="list-style-type: none"> 0—Period time passed 1—Subscriber logout 3—Wraparound 5—Subscriber VLink change
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	Indicates the number of seconds that have passed since the previous Real-Time Subscriber Usage RDR. <p>Note This field is not valid for zeroing RDR, "1" with appear.</p>
END_TIME	UINT32	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.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period.
SESSIONS	INT16	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	INT16	Aggregated number of session seconds for the reported service, for the current reporting period.

Related Topics

- [Periodic RDR Zero Adjustment Mechanism, page 2-83](#)

Link Usage RDR

The LINK_USAGE_RDR summarizes the activity on one of the SCE links for a specific service for the last user-configured number of minutes.

- RDR Purpose—Create link-level reports of network usage per service.
- RDR Default destination—Sent to the CM, stored in the database, and used by the reporter for global usage reports such as the Global Bandwidth per Service report, and subscriber demographics reports, such as the Active Subscribers per Service report.
- RDR Content—Summary of the activity on one of the SCE links for a specific service for the last user-configured minutes, including aggregated number of flows, total volume, duration, and active subscribers.
- RDR Generation Logic—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 2-83](#)). If network resources associated with the service are again consumed at some later time, this causes 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.

- RDR tag—0xf0f0f005 / 4042321925

Link Usage RDRs (LUR) are enabled by default. Disabling LURs eliminates global usage reports as well as subscriber demographics reports. LURs default interval is every 5 minutes. Increasing this interval can enhance the time granularity of LUR-based reports.

Table 2-13 lists the Link Usage RDR fields and their descriptions.

Table 2-13 *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	Counter to which each service is mapped. There are 64 global usage counters.
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	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	UINT32	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.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	UINT32	Concurrent number of sessions using the reported service at this point in time. A value 0 is reported for all links except link 0. Although the values are not reported in the respective links, a cumulative value is reported in link 0.
ACTIVE_SUBSCRIBERS	UINT32	Concurrent number of subscribers using the reported service at this point in time. A value 0 is reported for all links except link 0. Although the values are not reported in the respective links, a cumulative value is reported in link 0.
TOTAL_ACTIVE_SUBSCRIBERS	UINT32	Subscribers having active bidirectional flows in the system. A value 0 is reported for all links except link 0. Although the values are not reported in the respective links, a cumulative value is reported in link 0.

Related Topics

- [Periodic RDR Zero Adjustment Mechanism, page 2-83](#)

Zone Usage RDR

The ZONE_USAGE_RDR summarizes the activity on one of the SCE zones for a specific service for the last user-configured number of minutes.

- RDR Purpose—Create zone-level reports of network usage per service.
- RDR Default destination—Sent to the CM, stored in the database, and used by the reporter for global usage reports such as the Global Bandwidth per Service report, and subscriber demographics reports such as the Active Subscribers per Service report.
- RDR Content—Summary of the activity on one of the SCE zones for a specific service for the last user-configured minutes, including aggregated number of flows, total volume, duration, and active subscribers.
- RDR Generation Logic—Generated periodically, at user-configured intervals, for each zone. 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 ZONE_USAGE_RDR generation point. Whether or not a Zone 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 Zone Usage RDR is generated.
- If network resources associated with a service usage counter were not consumed since the previous RDR generation point, no Zone Usage RDR is generated.

However, the generation logic for Zone Usage RDRs uses the zeroing methodology (as described in [Periodic RDR Zero Adjustment Mechanism, page 2-83](#)). If network resources associated with the service are again consumed at some later time, this causes the immediate generation of either one or two zero-consumption Zone Usage RDRs in addition to the eventual generation of the Zone 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 Zone 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 Zone Usage RDRs 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 Zone (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.

- RDR tag—4042321928

Zone Usage RDRs (ZUR) are enabled by default. Disabling ZURs eliminates global usage reports as well as subscriber demographics reports. The default interval for ZURs is every 5 minutes. Increasing this interval can enhance the time granularity of ZUR-based reports.

Table 2-14 lists the Zone Usage RDR fields and their descriptions.

Table 2-14 Zone Usage RDR Fields

RDR Field Name	Type	Description
ZONE_COUNTER_ID	UINT16	ID of the zone associated with this session.
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR. Possible values are in the range from 0 to 3.
SERVICE_USAGE_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	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	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.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	UINT32	Concurrent number of sessions using the reported service at this point in time. Currently not supported in Release 3.6.5, so counter always returns 0 value.
ACTIVE_SUBSCRIBERS	UINT32	Concurrent number of subscribers using the reported service at this point in time. Currently not supported in Release 3.6.5, so counter always returns 0 value.
TOTAL_ACTIVE_SUBSCRIBERS	UINT32	Concurrent number of subscribers in the system at this point in time. Currently not supported in Release 3.6.5, so counter always returns 0 value.

Related Topics

- [Periodic RDR Zero Adjustment Mechanism, page 2-83](#)

Package Usage RDR

The PACKAGE_USAGE_RDR summarizes the activity of a specific group of subscribers (belonging to the same package) for a specific service in the last user-configured number of minutes.

- RDR Purpose—Create reports about network usage per service for a group of subscribers.
- RDR Default destination—Sent to the CM, stored in the database, and used by the Reporter tool for package usage reports such as the Package Bandwidth per Service report.
- RDR Content—Summary of the activity of a specific group of subscribers (belonging to the same package) for a specific service for the last user-configured number of minutes, including aggregated number of flows, total volume, and duration.
- RDR Generation Logic—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 2-83](#)). If network resources associated with the service usage counter are again consumed by any subscriber of the package at some later time, this causes 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.

- RDR tag—0xf0f0f004 / 4042321924

Package Usage RDRs (PURs) are enabled by default. Disabling LURs eliminates package usage reports. The default interval for PURs is every 5 minutes. Increasing this interval can enhance the time granularity of PUR-based reports.

Table 2-15 lists the Package Usage RDR fields and their descriptions.

Table 2-15 Package Usage RDR Fields

RDR Field Name	Type	Description
PACKAGE_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 1024 package usage counters.
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR.
SERVICE_USAGE_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	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	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.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	UINT32	Concurrent number of sessions using the reported service in the reported package at this point in time.
ACTIVE_SUBSCRIBERS	UINT32	Concurrent number of subscribers using the reported service in the reported package at this point in time.
TOTAL_ACTIVE_SUBSCRIBERS	UINT32	Concurrent number of subscribers in the system at this point in time.

Related Topics

- [Periodic RDR Zero Adjustment Mechanism, page 2-83](#)

Virtual Links Usage RDR

The VIRTUAL_LINKS_USAGE_RDR summarizes the activity on one of the virtual links for a specific service for the last user-configured number of minutes. For information on virtual links, see *Cisco Service Control Application for Broadband User Guide*.

- RDR Purpose—Create reports relating to network usage per service for a specific virtual link.
- RDR Default destination—Sent to the CM, stored in the database, and used by the reporter for virtual link reports such as the Virtual Link Bandwidth per Service report.
- RDR Content—Summary of the activity on one of the virtual links for a specific service for the last user-configured number of minutes, including aggregated number of flows, total volume, and duration.
- RDR Generation Logic—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 2-83](#)). If network resources associated with the service usage counter are again consumed by subscribers of the virtual link at some later time, this causes 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.

- RDR tag—0xf0f0f006 / 4042321926

Virtual Link Usage RDRs (VLURs) are disabled by default. You can enable VLURs when working with virtual links to facilitate virtual link usage reports. The recommended value for intervals between VLURs is 5 minutes.

Table 2-16 lists the Virtual Links Usage RDR fields and their descriptions.

Table 2-16 Virtual Links Usage RDR Fields

RDR Field Name	Type	Description
VLINK_ID	INT16	Virtual link ID
VLINK_DIRECTION	INT8	Virtual link direction: <ul style="list-style-type: none"> • 0—Upstream • 1—Downstream
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.
SERVICE_USAGE_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 1024 global usage counters.
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	Not implemented (always the same as CONFIGURED_DURATION).
END_TIME	UINT32	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.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume on the virtual link (for a single processor) of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume on the virtual link (for a single processor) of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Reserved for future use.
SECONDS	UINT32	Reserved for future use.
CONCURRENT_SESSIONS	UINT32	Reserved for future use.
ACTIVE_SUBSCRIBERS	UINT32	Reserved for future use.
TOTAL_ACTIVE_SUBSCRIBERS	UINT32	Concurrent number of subscribers in the system at this point in time.

Related Topics

- [Periodic RDR Zero Adjustment Mechanism, page 2-83](#)

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—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—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.

Table 2-17 lists the Blocking RDR fields and their descriptions.

Table 2-17 Blocking RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	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.
SERVICE_ID	INT32	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.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
CLIENT_IP	UINT32	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	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
SERVER_IP	UINT32	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.

Table 2-17 Blocking RDR Fields (continued)

RDR Field Name	Type	Description
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.
BLOCK_REASON	UINT8	Indicates the reason why this session was blocked.
BLOCK_RDR_COUNT	INT32	Total number of blocked flows reported so far (from the beginning of the current aggregation period).
REDIRECTED	INT8	Indicates whether the flow has been redirected after being blocked. <ul style="list-style-type: none"> • 0—Not redirected • 1—Redirected
REPORT_TIME	UINT32	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.

Related Topics

- [Block Reason \(uint8\)](#), page 2-77

Quota Breach RDR

The QUOTA_BREACH_RDR is generated each time a bucket is breached.

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

The RDR tag of the QUOTA_BREACH_RDR is 0xf0f0f072 / 4,042,322,034.

[Table 2-18](#) lists the Quota Breach RDR fields and their descriptions.

Table 2-18 Quota Breach RDR Fields

RDR Field Name	Type	Description
QUOTA_MODEL_TYPE	UINT8	Quota model type: <ul style="list-style-type: none"> • 1 - Gy Quota Model • 2 - QM Quota Model • 3 - Internal Quota Model
RDR_REASON	UINT8	Reason the RDR was sent. Not in use, RESERVED - 0xfe
SUBSCRIBER_ID	STRING	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	INT16	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.
ADDITIONAL_INFO	UINT32	See ADDITIONAL_INFO Field, page 2-4 for details.
END_TIME	UINT32	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.
BUCKET_ID	UINT16	Bucket ID to report.

Table 2-18 Quota Breach RDR Fields (continued)

RDR Field Name	Type	Description
BUCKET_TYPE	UINT16	<p>Bucket type:</p> <ul style="list-style-type: none"> • 1—Volume_UP Only the upstream volume is reported in the RDR. UNIT_AMOUNT_IN is 0 and UNIT_AMOUNT_OUT indicates the upstream volume. • 2—Volume_DOWN Only the downstream volume is reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT is 0. • 3—Total Volume The sum of downstream and upstream volumes, that is, the total volume consumed, and the remaining volume, that is, bucket size – total volume is reported in the RDR. UNIT_AMOUNT_IN indicates the total volume consumed and UNIT_AMOUNT_OUT indicates the remaining volume. • 4—VolumeUpDown Both upstream and downstream volumes are reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT indicates the upstream volume. • 5—Events (sessions) UNIT_AMOUNT_IN indicates the number of sessions that has used the bucket. UNIT_AMOUNT_OUT indicates the remaining number of sessions for the bucket. • 6—Time UNIT_AMOUNT_IN indicates how long a bucket has been used. This unit is represented in seconds. The UNIT_AMOUNT_OUT field is 0. <p>Note For the following bucket types, only the UNIT_AMOUNT_IN field is valid:</p> <ul style="list-style-type: none"> - Time - Events (sessions) - Total Volume
UNIT_AMOUNT_IN	UINT32	Consumed downstream volume in volume units/ Seconds/ Sessions.

Table 2-18 *Quota Breach RDR Fields (continued)*

RDR Field Name	Type	Description
UNIT_AMOUNT_OUT	UINT32	Consumed upstream volume in volume units. For Internal/QM quota models – remaining quota as 32-bit integer value (may be negative).
BUCKET_SIZE_IN	UINT32	Original bucket size in volume units/ Seconds/ Sessions. For GY quota model – downstream bucket size in volume units.
BUCKET_SIZE_OUT	UINT32	Valid for Gy quota model only – upstream bucket size in volume units.

Quota Status RDR

The QUOTA_STAUS_RD reports consumed quota of subscriber for all associated buckets. If one RDR cannot contain all associated buckets, then two or more consecutive RDRs are sent.

The user can set a limit on the total number of these RDRs that are generated per second.

If a bucket is not in use, 0xFFFF appears in BUCKET_ID, BUCKET_TYPE, UNIT_AMOUNT_IN, and UNIT_AMOUNT_OUT fields.



Note

The QUOTA_STAUS_RDR is generated only for those subscribers whose policy requires the generation of such RDRs.

The following events trigger the sending of this RDR:

- Periodically, at user-configured intervals. The intervals are defined globally.
Applies to all quota models, including internal and QM external quota models.
- Package switch event: Indicates consumed quota before the package switch.
Applies to all quota models.
- Subscriber logout event.
Applies to all quota models.
- Quota Validity Time/Quota Holding Time expiration.
Applies to Gy quota model only.

The RDR tag of the QUOTA_STAUS_RDR is 0xf0f0f071 / 4,042,322,033.

[Table 2-19](#) lists the Quota Status RDR fields and descriptions.

Table 2-19 Quota Status RDR Fields

RDR Field Name	Type	Description
QUOTA_MODEL_TYPE	UINT8	Quota model type: <ul style="list-style-type: none"> • 1 - Gy Quota Model • 2 - QM Quota Model • 3 - Internal Quota Model
RDR_REASON	UINT8	Reason the RDR was sent: <ul style="list-style-type: none"> • 0 - Period time passed • 1 - Logout • 2 - Package Switch • 5 - Request (RAR)
SUBSCRIBER_ID	STRING	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.

Table 2-19 **Quota Status RDR Fields**

RDR Field Name	Type	Description
PACKAGE_ID	INT16	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.
ADDITIONAL_INFO	UINT32	See ADDITIONAL_INFO Field, page 2-4 for details.
END_TIME	UINT32	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.
BUCKET_ID	UINT16	Bucket ID to report.

Table 2-19 Quota Status RDR Fields

RDR Field Name	Type	Description
BUCKET_TYPE	UINT16	<p>Bucket type:</p> <ul style="list-style-type: none"> • 1—Volume_UP Only the upstream volume is reported in the RDR. UNIT_AMOUNT_IN is 0 and UNIT_AMOUNT_OUT indicates the upstream volume. • 2—Volume_DOWN Only the downstream volume is reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT is 0. • 3—Total Volume The sum of downstream and upstream volumes, that is, the total volume consumed, and the remaining volume, that is, bucket size – total volume is reported in the RDR. UNIT_AMOUNT_IN indicates the total volume consumed and UNIT_AMOUNT_OUT indicates the remaining volume. • 4—VolumeUpDown Both upstream and downstream volumes are reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT indicates the upstream volume. • 5—Events (sessions) UNIT_AMOUNT_IN indicates the number of sessions that has used the bucket. UNIT_AMOUNT_OUT indicates the remaining number of sessions for the bucket. • 6—Time UNIT_AMOUNT_IN indicates how long a bucket has been used. This unit is represented in seconds. The UNIT_AMOUNT_OUT field is 0. <p>Note For the following bucket types, only the UNIT_AMOUNT_IN field is valid:</p> <ul style="list-style-type: none"> – Time – Events (sessions) – Total Volume

Table 2-19 **Quota Status RDR Fields**

RDR Field Name	Type	Description
UNIT_AMOUNT_IN	UINT32	Consumed volume in volume units/ Seconds/ Sessions. For Gy quota model – downstream volume.
UNIT_AMOUNT_OUT	UINT32	For Gy quota model – consumed upstream volume in volume units. For QM/Internal quota models – remaining quota in 32-bit integer format (may be negative).

**Note**

The following fields report information per bucket:

- BUCKET_ID
- BUCKET_TYPE
- UNIT_AMOUNT_IN
- UNIT_AMOUNT_OUT

This section of four fields is repeated 16 times, one time for each of the 16 buckets, for a total of 64 fields. (Added to the 6 header fields results in a total of 70 fields in the RDR.)

Quota Threshold Breach RDR

The QUOTA_THRESHOLD_BREACH_RDR is generated each time a bucket exceeds the bucket threshold is defined per package.

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 0xf0f0f073 / 4,042,322,035.

[Table 2-20](#) lists the Quota Threshold Breach RDR fields and their descriptions.

Table 2-20 Quota Threshold Breach RDR Fields

RDR Field Name	Type	Description
QUOTA_MODEL_TYPE	UINT8	Quota model type: <ul style="list-style-type: none"> • 1 - Gy Quota Model • 2 - QM Quota Model • 3 - Internal Quota Model
RDR_REASON	UINT8	Reason the RDR was sent. Not in use, RESERVED - 0xfe
SUBSCRIBER_ID	STRING	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	INT16	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_package. The value maximum_number_of_packages is reserved for unknown subscribers.
ADDITIONAL_INFO	UINT32	See ADDITIONAL_INFO Field, page 2-4 for details.
END_TIME	UINT32	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.
BUCKET_ID	UINT16	Bucket ID to report.

Table 2-20 Quota Threshold Breach RDR Fields (continued)

RDR Field Name	Type	Description
BUCKET_TYPE	UINT16	<p>Bucket type:</p> <ul style="list-style-type: none"> • 1—Volume_UP Only the upstream volume is reported in the RDR. UNIT_AMOUNT_IN is 0 and UNIT_AMOUNT_OUT indicates the upstream volume. • 2—Volume_DOWN Only the downstream volume is reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT is 0. • 3—Total Volume The sum of downstream and upstream volumes, that is, the total volume consumed, and the remaining volume, that is, bucket size – total volume is reported in the RDR. UNIT_AMOUNT_IN indicates the total volume consumed and UNIT_AMOUNT_OUT indicates the remaining volume. • 4—VolumeUpDown Both upstream and downstream volumes are reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT indicates the upstream volume. • 5—Events (sessions) UNIT_AMOUNT_IN indicates the number of sessions that has used the bucket. UNIT_AMOUNT_OUT indicates the remaining number of sessions for the bucket. • 6—Time UNIT_AMOUNT_IN indicates how long a bucket has been used. This unit is represented in seconds. The UNIT_AMOUNT_OUT field is 0. <p>Note For the following bucket types, only the UNIT_AMOUNT_IN field is valid:</p> <ul style="list-style-type: none"> – Time – Events (sessions) – Total Volume
UNIT_AMOUNT_IN	UINT32	Consumed downstream volume in volume units/ Seconds/ Sessions.

Table 2-20 Quota Threshold Breach RDR Fields (continued)

RDR Field Name	Type	Description
UNIT_AMOUNT_OUT	UINT32	Consumed upstream volume in volume units. For QM/Internal quota models – remaining quota in 32-bit integer format (may be negative).
BUCKET_SIZE_IN	UINT32	Original bucket size in volume units/ Seconds/ Sessions. For GY quota model – downstream volume/total volume/sessions/seconds.
BUCKET_SIZE_OUT	UINT32	For GY quota model – original upstream volume.
THRESHOLD_SIZE_IN	UINT32	Threshold of the bucket in volume units/ Seconds/ Sessions.
THRESHOLD_SIZE_OUT	UINT32	Threshold of the bucket in volume units/ Seconds/ Sessions. Valid for Gy quota models only – upstream bucket threshold.

Session Creation RDRs

Typically the SESSION_CREATION_RDR is sent on subscriber login event. This RDR replaces the legacy QUOTA_STATE_RESTORE_RDR.

If a bucket is not in use, 0xFFFF appears in BUCKET_ID, and '0' appears in BUCKET_TYPE, UNIT_AMOUNT_IN, and UNIT_AMOUNT_OUT fields.

The following events trigger the sending of this RDR:

- Subscriber that associates with package with external quota management (Gy or Qm)
- Package switch event - transition from internal package to external one or in Gy.

The RDR tag of the SESSION_CREATION_RDR is 0xf0f0f070 / 4,042,322,032.

Table 2-21 lists the Session Creation RDR fields and their descriptions.

Table 2-21 Session Creation RDR Fields

RDR Field Name	Type	Description
QUOTA_MODEL_TYPE	UINT8	Quota model type: <ul style="list-style-type: none"> • 1 - Gy Quota Model • 2 - QM Quota Model • 3 - Internal Quota Model
RDR_REASON	UINT8	Reason the RDR was sent: <ul style="list-style-type: none"> • 2 - Package Switch • 3 - Login
SUBSCRIBER_ID	STRING	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	INT16	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.
ADDITIONAL_INFO	UINT32	See ADDITIONAL_INFO Field, page 2-4 for details.
END_TIME	UINT32	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.
BUCKET_ID	UINT16	Use for Gy quota model only. If request quota upon login is specified for the bucket, this field contains bucket id. This indicates to the server that quota should be provided to specified bucket id. '0xFFFF' – reserved.
BUCKET_TYPE	UINT16	Not used. 0

Table 2-21 *Session Creation RDR Fields (continued)*

RDR Field Name	Type	Description
UNIT_AMOUNT_IN	UINT32	Not used. 0
UNIT_AMOUNT_OUT	UINT32	Not used. 0

**Note**

The following fields report information per bucket:

- BUCKET_ID
- BUCKET_TYPE
- UNIT_AMOUNT_IN
- UNIT_AMOUNT_OUT

This section of four fields is repeated 16 times, one time for each of the 16 buckets, for a total of 64 fields. (Added to the 6 header fields results in a total of 70 fields in the RDR.)

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 is generated even if none of the fields were found.

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

[Table 2-22](#) lists the DHCP RDR fields and descriptions.

Table 2-22 *DHCP RDR Fields*

RDR Field Name	Type	Description
CPE_MAC	STRING	DHCP protocol field.
CMTS_IP	UINT32	DHCP protocol field.
ASSIGNED_IP	UINT32	DHCP protocol field.
RELEASED_IP	UINT32	DHCP protocol field.
TRANSACTION_ID	UINT32	DHCP protocol field.
MESSAGE_TYPE	UINT8	DHCP message type.
OPTION_TYPE_0 through OPTION_TYPE_7	UINT8	List of DHCP options extracted from the message.
OPTION_TYPE_0 through OPTION_TYPE_7	STRING	Values associated with the above DHCP options.
END_TIME	INT32	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.

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 is generated even if none of the fields were found.

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

Table 2-23 lists the RADIUS RDR fields and descriptions.

Table 2-23 RADIUS RDR Fields

RDR Field Name	Type	Description
SERVER_IP	UINT32	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	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
CLIENT_IP	UINT32	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	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side
RADIUS_PACKET_CODE	UINT8	Type of the RADIUS message intercepted.
RADIUS_ID	UINT8	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, as follows:

- Any flow on packages and services that are configured to generate such an RDR.
- When a SIP INVITE request for voice and video traffic is received.

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.

Table 2-24 lists the Flow Start RDR fields and their descriptions.

Table 2-24 Flow Start RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	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.
SERVICE_ID	INT32	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.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	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	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
CLIENT_IP	UINT32	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	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.

Table 2-24 Flow Start RDR Fields

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side
START_TIME	UINT32	Flow start time.
REPORT_TIME	UINT32	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.
BREACH_STATE	INT8	Indicates whether the subscriber's quota was breached. <ul style="list-style-type: none"> • 0—Not breached • 1—Breached
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	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.

Table 2-25 lists the Flow End RDR fields and their descriptions.

Table 2-25 Flow End RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	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_package. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_ID	INT32	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.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	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	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
CLIENT_IP	UINT32	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	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.

Table 2-25 Flow End RDR Fields (continued)

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side
START_TIME	UINT32	Flow start time.
REPORT_TIME	UINT32	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.
BREACH_STATE	INT8	Indicates whether the subscriber's quota was breached. <ul style="list-style-type: none"> • 0—Not breached • 1—Breached
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	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.

Table 2-26 lists the Ongoing Flow RDR fields and their descriptions.

Table 2-26 Ongoing Flow RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	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.
SERVICE_ID	INT32	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.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	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	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
CLIENT_IP	UINT32	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	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.

Table 2-26 Ongoing Flow RDR Fields (continued)

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side
START_TIME	UINT32	Flow start time.
REPORT_TIME	UINT32	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.
BREACH_STATE	INT8	Indicates whether the subscriber's quota was breached. <ul style="list-style-type: none"> • 0—Not breached • 1—Breached
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR.

Media Flow RDR

The MEDIA_FLOW_RDR is generated at the end of every SIP, Skype, or MGCP 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.
- For MGCP, this RDR is generated when a media flow is closed.



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.

Table 2-27 lists the Media Flow RDR fields and their descriptions.

Table 2-27 Media Flow RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers, this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer between 0 and maximum_number_of_packages. The maximum_number_of_packages value is reserved for unknown subscribers.
SERVICE_ID	INT32	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.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
DESTINATION_IP	UINT32	<ul style="list-style-type: none"> • SIP—Destination IP address of RTP flow. • Skype—Destination IP address of Skype flow. • MGCP—Destination IP address of RTP flow.
DESTINATION_PORT	UINT16	<ul style="list-style-type: none"> • SIP—Destination port of RTP flow. • Skype—Destination port of Skype flow. • MGCP—Destination port of RTP flow.
SOURCE_IP	UINT32	<ul style="list-style-type: none"> • SIP—Source IP address of RTP flow. • Skype—Source IP address of Skype flow. • MGCP—Source IP address of RTP flow.
SOURCE_PORT	UINT16	<ul style="list-style-type: none"> • SIP—Source port of RTP flow. • Skype—Source port of Skype flow. • MGCP—Source port of RTP flow.

Table 2-27 Media Flow RDR Fields (continued)

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides: <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side For Skype, this is the initiating side of the flow (not necessarily the initiating side of the voice call).
ZONE_ID	INT32	ID of the zone associated with this session.
FLAVOR_ID	INT32	ID of the protocol signatures with flavor associated with this session.
DOMAIN	STRING	<ul style="list-style-type: none"> • SIP—Domain name extracted from SIP header. • MGCP—Domain name extracted from MGCP header.
USER_AGENT	STRING	<ul style="list-style-type: none"> • SIP—User-Agent field extracted from SIP header. • MGCP—Not applicable.
START_TIME	UINT32	Flow start time.
REPORT_TIME	UINT32	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.
DURATION_SECONDS	INT32	<ul style="list-style-type: none"> • 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. • MGCP—The active duration of the RTP flow, not including the aging time.
UPSTREAM_VOLUME	UINT32	<ul style="list-style-type: none"> • SIP—The upstream volume of the RTP flow, in bytes. • Skype—The upstream volume between the start-of-call and end-of-call detection events. • MGCP—The upstream volume of the RTP flow, in bytes.
DOWNSTREAM_VOLUME	UINT32	<ul style="list-style-type: none"> • SIP—The downstream volume of the RTP flow, in bytes. • Skype—The downstream volume between the start-of-call and end-of-call detection events. • MGCP—The downstream volume of the RTP flow, in bytes.
IP_PROTOCOL	UINT8	IP protocol type: <ul style="list-style-type: none"> • 6—TCP • 17—UDP

Table 2-27 Media Flow RDR Fields (continued)

RDR Field Name	Type	Description
FLOW_TYPE	INT8	<ul style="list-style-type: none"> 0—All Skype flows 1—Audio (SIP/MGCP) 2—Video (SIP/MGCP)
SESSION_ID	UINT32	<ul style="list-style-type: none"> SIP—The flow-context ID of the control flow. Skype—The flow-context ID of the flow. MGCP—The flow-context ID of the control flow.
UPSTREAM_JITTER	UINT32	<ul style="list-style-type: none"> 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). MGCP—The average upstream jitter for the session, taken from the RTCP flow. N/A (0xFFFFFFFF) if RTCP flow is missing.
DOWNSTREAM_JITTER	UINT32	<ul style="list-style-type: none"> 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). MGCP—The average downstream jitter for the session, taken from the RTCP flow. N/A (0xFFFFFFFF) if RTCP flow is missing.
UPSTREAM_PACKET_LOSS	UINT16	<ul style="list-style-type: none"> 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). MGCP—The average fractional upstream packet loss for the session, taken from the RTCP flow. N/A (0xFFFF) if RTCP flow is missing.
DOWNSTREAM_PACKET_LOSS	UINT16	<ul style="list-style-type: none"> SIPvThe 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). MGCP—The average fractional downstream packet loss for the session, taken from the RTCP flow. N/A (0xFFFF) if RTCP flow is missing.

Table 2-27 Media Flow RDR Fields (continued)

RDR Field Name	Type	Description
UPSTREAM_PAYLOAD_TYPE	UINT8	<ul style="list-style-type: none"> • SIP—The upstream RTP payload type for the session. • Skype—N/A (0xFF). • MGCP—The upstream RTP payload type for the session.
DOWNSTREAM_PAYLOAD_TYPE	UINT8	<ul style="list-style-type: none"> • SIP—The downstream RTP payload type for the session. • Skype—N/A (0xFF). • MGCP—The downstream RTP payload type for the session.

**Note****Packet Loss Note**

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, see 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” section in the “Using the Service Configuration Editor: Additional Options” chapter of *Cisco Service Control Application for Broadband User Guide*.)

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

Table 2-28 lists the Attack Start RDR fields and their descriptions.

Table 2-28 Attack Start RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
ATTACK_ID	UINT32	Unique attack ID.
ATTACKING_IP	UINT32	IP address related to the attack (for example: in a DDoS, this is the IP address under attack; in a scan this is the IP address of the source of the scan).
ATTACKED_IP	UINT32	Other IP address related to the attack, if one exists; otherwise, 0xFFFFFFFF.
ATTACKED_PORT	UINT16	Attacked port: 0xFFFF if not present.
ATTACKING_SIDE	INT8	Side of the SCE ATTACKING_IP on which it resides: <ul style="list-style-type: none"> • 0—Subscriber • 1—Network
IP_PROTOCOL	UINT8	IP protocol type.
ATTACK_TYPE	UINT32	ATTACKING_IP to whom it belongs: <ul style="list-style-type: none"> • 0—Attacked • 1—Attacker
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR.
ATTACK_TIME	UINT32	Time since attack started in seconds.
REPORT_TIME	UINT32	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.

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.

Table 2-29 lists the Attack End RDR fields and their descriptions.

Table 2-29 Attack End RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
ATTACK_ID	UINT32	Unique attack ID.
ATTACKING_IP	UINT32	IP address related to the attack (for example: in a DDoS, this will be the IP address under attack; in a scan this is the IP address of the source of the scan).
ATTACKED_IP	UINT32	Other IP address related to the attack, if one exists; otherwise, 0xFFFFFFFF.
ATTACKED_PORT	UINT16	Attacked port: 0xFFFF if not present.
ATTACKING_SIDE	INT8	Side of the SCE ATTACKING_IP on which it resides: <ul style="list-style-type: none"> 0—Subscriber 1—Network
IP_PROTOCOL	UINT8	IP protocol type.
ATTACK_TYPE	UINT32	To whom ATTACKING_IP belongs: <ul style="list-style-type: none"> 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	UINT32	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.

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.

Table 2-30 lists the Malicious Traffic Periodic RDR fields and their descriptions.

Table 2-30 Malicious Traffic Periodic RDR Fields

RDR Field Name	Type	Description
ATTACK_ID	INT32	Unique attack ID.
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
ATTACK_IP	UINT32	IP address related to this attack.
OTHER_IP	UINT32	IP address other than the one displayed in ATTACK_IP. For example, in a DDoS, this is the IP address under attack; in a scan, this is the IP address of the source of the scan. If there is no attack, 0xFFFFFFFF is displayed.
PORT_NUMBER	UINT16	Displays the attacked port. If there is no attack, 0xFFFF is displayed.
ATTACK_TYPE	INT32	ATTACK_IP to whom it belongs: <ul style="list-style-type: none"> • 0—Attacked • 1—Attacker
SIDE	INT8	The IP address side: <ul style="list-style-type: none"> • 0—Subscriber • 1—Network
IP_PROTOCOL	UINT8	IP protocol type: <ul style="list-style-type: none"> • 0—Other • 1—ICMP • 6—TCP • 17—UDP
CONFIGURED_DURATION	INT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	INT32	Indicates the number of seconds that have passed since the previous MALICIOUS_TRAFFIC_RDR.
END_TIME	INT32	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.

Table 2-30 Malicious Traffic Periodic RDR Fields (continued)

RDR Field Name	Type	Description
ATTACKS	INT8	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)

Spam RDR

The SPAM_RDR is generated when mass-mailing activity is detected.

The RDR tag of the SPAM_RDR is 4042322048.

Table 2-31 lists the Spam RDR fields and their descriptions.

Table 2-31 Spam RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	UINT16	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.
SERVICE_ID	INT32	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.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
CLIENT_IP	UINT32	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	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
SERVER_IP	UINT32	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	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. <ul style="list-style-type: none"> • 0—Subscriber side • 1—Network side
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.

Table 2-31 Spam RDR Fields (continued)

RDR Field Name	Type	Description
SPAM_FOUND	UINT8	Indicates whether spam was found (1) or stopped (0).
THRESHOLD_LEVEL	UINT16	Threshold level. Reserved for future use. Currently 0.
SESSION_COUNTER	UINT32	Number of sessions found.
TIME_INTERVAL	UINT32	Time that elapsed since the beginning of the period.
DEFINED_SESSION_COUNTER	UINT32	Indicates the defined number of sessions.
DEFINED_TIME_INTERVAL	UINT32	Indicates the defined time interval.
REPORT_TIME	INT32	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.

Related Topics

- [Universal RDR Fields, page 2-2](#)

Information About RDR Enumeration Fields

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

- [Block Reason \(uint8\), page 2-77](#)
- [String Fields, page 2-78](#)
- [Aggregation Period \(uint8\), page 2-79](#)
- [Flow Close Mode \(uint8\), page 2-80](#)
- [Time Frames \(uint16\), page 2-80](#)

Block Reason (uint8)

The BLOCK_REASON field is a bit field. [Table 2-32](#) lists the meanings of the bits of this field.

Table 2-32 *Block Reason Field Bit Values*

Bits Number	Value and Description
7 (msb)	Always ON.
6	<ul style="list-style-type: none"> • 0—Action of the effective rule is block. • 1—Concurrent session limit of the effective rule was reached.
5	<ul style="list-style-type: none"> • 0—Effective rule was in pre-breach state. • 1—Effective rule was in post-breach state.
4 to 0 (lsb)	Number of the breached bucket (1 to 16).

String Fields

Table 2-33 lists the ACCESS_STRING and INFO_STRING field values.

Table 2-33 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 2-33 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 Fast Start	
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)

Table 2-34 lists the AGG_PERIOD field values.

Table 2-34 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	Quota is externally provisioned and managed by a third-party source.

Flow Close Mode (uint8)

Table 2-35 lists the FLOW_CLOSE_MODE field values.

Table 2-35 Flow Close Mode Field Values

Name	Value	Description
TCP_NORMAL_CLOSE	0	SCE observed a normal termination of the TCP connection.
FLOW_CLOSED_BY_SYSTEM	2	SCE concluded that the connection has terminated after a period of inactivity.

Time Frames (uint16)

Table 2-36 lists the TIME_FRAME field values.

Table 2-36 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

Table 2-37 summarizes RDR tag assignments.

Table 2-37 RDR Tag Assignments

RDR Name	Default Category (explained in the following table)	Tag Value (decimal)	Tag Value (hex)
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
ZONE USAGE RDR	CM-DB (1)	4,042,321,928	F0 F0 F0 08
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
VIDEO TRANSACTION USAGE RDR	CM-CSV (1)	0xf0f0f480	4042323072
BLOCKING RDR	CM-CSV (1)	4,042,321,984	F0 F0 F0 40
QUOTA BREACH RDR	QP (4)	4,042,322,034	F0 F0 F0 72
QUOTA STATUS RDR	QP (4)	4,042,322,033	F0 F0 F0 71
QUOTA THRESHOLD RDR	QP (4)	4,042,322,035	F0 F0 F0 73
SESSION CREATION RDR	QP (4)	4,042,322,032	F0 F0 F0 70
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

Table 2-37 RDR Tag Assignments (continued)

RDR Name	Default Category (explained in the following table)	Tag Value (decimal)	Tag Value (hex)
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
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 following relevant document:

- “Raw Data Formatting: The RDR Formatter and NetFlow Exporting” chapter of *Cisco SCE 2000 and SCE 1000 Software Configuration Guide*.
- “Raw Data Formatting: The RDR Formatter and NetFlow Exporting” chapter of *Cisco SCE8000 10GBE Software Configuration Guide*.
- “Raw Data Formatting: The RDR Formatter and NetFlow Exporting” chapter of *Cisco SCE8000 GBE Software Configuration Guide*.

Table 2-38 summarizes the RDR tag default categories.

Table 2-38 RDR Tag Default Categories

Default Category	Intended Destination and Use
CM-DB (1)	CM database. Used by the SCA Reporter to generate reports.
CM-CSV (1)	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.

