Cisco Service Control Application for Broadband Reference Guide

Release 3.1
May 2007

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Customer Order Number:
Text Part Number: OL-8410-04

Cisco Service Control Application for Broadband Reference Guide

CONTENTS

Document Revision History ix

Audience x

Organization xi

Document Conventions xii

Related Documentation xii

Obtaining Documentation, Obtaining Support, and Security Guidelines xiii

CHAPTER 1 **Default Service Configuration Reference Tables** 1-1

Filter Rules 1-1

Information About Protocols 1-4

Generic Protocols 1-5

Signature-Based Protocols 1-5

IP Protocols 1-9

Port-Based Protocols 1-12

Protocols Identified on Unidirectional Flows 1-29

Services 1-31

RDR Settings 1-33

Rules 1-34

System Mode 1-34

CHAPTER 2 **Raw Data Records: Formats and Field Contents** 2-1

Raw Data Records Overview 2-2

Universal RDR Fields 2-2

Transaction RDR 2-3

Transaction Usage RDR 2-5

HTTP Transaction Usage RDR 2-6

RTSP Transaction Usage RDR 2-8

VoIP Transaction Usage RDR 2-10

Subscriber Usage RDR 2-13

Real-Time Subscriber Usage RDR 2-15

Link Usage RDR 2-17

Package Usage RDR 2-19

Virtual Links Usage RDR 2-21
Blocking RDR 2-23
Quota Breach RDR 2-24
Remaining Quota RDR 2-25
Quota Threshold Breach RDR 2-26
Quota State Restore RDRs 2-26
DHCP RDR 2-27
RADIUS RDR 2-28
Flow Start RDR 2-28
Flow End RDR 2-29
Ongoing Flow RDR 2-30
Media Flow RDR 2-31
Attack Start RDR 2-34
Attack End RDR 2-35
Malicious Traffic Periodic RDR 2-35
Information About RDR Enumeration Fields 2-37
Block Reason (uint8) 2-37
String Fields 2-38
Aggregation Period (uint8) 2-39
Time Frames (uint16) 2-40
RDR Tag Assignment Summary 2-40
Periodic RDR Zero Adjustment Mechanism 2-41

CHAPTER 3 **NetFlow Records: Formats and Field Contents 3-1**
NetFlow 3-1
NetFlow Field Types 3-2

CHAPTER 4 **Database Tables: Formats and Field Contents 4-1**
Database Tables Overview 4-1
Table RPT_NUR 4-2
Table RPT_SUR 4-2
Table RPT_PUR 4-3
Table RPT_LUR 4-4
Table RPT_TR 4-4
Table RPT_MEDIA 4-5
Table RPT_MALUR 4-6

Table RPT_TOPS_PERIOD0 4-7
Table RPT_TOPS_PERIOD1 4-8
Table INI_VALUES 4-9
Table VLINK_INI 4-11
Table CONF_SE_TZ_OFFSET 4-11

CHAPTER 5 **CSV File Formats 5-1**
Information About Service Configuration Entities CSV File Formats 5-1
Service CSV Files 5-1
Protocol CSV Files 5-2
Zone CSV Files 5-2
Information About Flavor CSV Files 5-2
HTTP URL 5-2
HTTP User Agent 5-3
RTSP User Agent 5-3
RTSP Host Name 5-3
RTSP Composite 5-3
SIP Destination Domain 5-3
SIP Source Domain 5-3
SIP Composite 5-3
SMTP Host Name 5-3
Information About Subscriber CSV File Formats 5-4
Import/Export File: Format of the mappings Field 5-4
SCE Subscriber CSV Files 5-4
SCMS SM Subscriber CSV Files 5-4
SCE Anonymous Group CSV Files 5-5
SCE Subscriber Template CSV File 5-5
Information About Collection Manager CSV File Formats 5-5
CSV Adapter CSV Files 5-5
TA Adapter CSV Files 5-6
RAG Adapter CSV Files 5-6

CHAPTER 6 **SCA BB Proprietary MIB Reference 6-1**
Information About SNMP Configuration and Management 6-1
Configuring the SNMP Interface on the SCE platform 6-1
Related Info 6-1
Required MIB Files 6-2
Service Control Enterprise MIB 6-2

Information About The CISCO-SCAS-BB MIB 6-3
Using this Reference 6-3
pcubeEngageObjs (pcubeWorkgroup 2) 6-3
pcubeEngageObjs Objects 6-3
pcubeEngageObjs Structure 6-4
Service Group: serviceGrp (pcubeEngageObjs 1) 6-4
serviceTable (serviceGrp 1) 6-4
Link Group: linkGrp (pcubeEngageObjs 2) 6-5
linkServiceUsageTable (linkGrp 1) 6-5
linkServiceUsageEntry (linkServiceUsageTable 1) 6-5
linkServiceUsageUpVolume (linkServiceUsageEntry 1) 6-5
linkServiceUsageDownVolume (linkServiceUsageEntry 2) 6-6
linkServiceUsageNumSessions (linkServiceUsageEntry 3) 6-6
linkServiceUsageDuration (linkServiceUsageEntry 4) 6-6
linkServiceUsageConcurrentSessions (linkServiceUsageEntry 5) 6-6
linkServiceUsageActiveSubscribers (linkServiceUsageEntry 6) 6-7
linkServiceUpDroppedPackets (linkServiceUsageEntry 7) 6-7
linkServiceDownDroppedPackets (linkServiceUsageEntry 8) 6-7
linkServiceUpDroppedBytes (linkServiceUsageEntry 9) 6-8
linkServiceDownDroppedBytes (linkServiceUsageEntry 10) 6-8
Package Group: packageGrp (pcubeEngageObjs 3) 6-8
packageCounterEntry (packageCounterTable 1) 6-9
packageCounterIndex (packageCounterEntry 1) 6-9
packageCounterStatus (packageCounterEntry 2) 6-9
packageCounterName (packageCounterEntry 3) 6-9
packageCounterActiveSubscribers (packageCounterEntry 4) 6-9
packageServiceUsageTable (packageGrp 2) 6-10
packageServiceUsageEntry (packageServiceUsageTable 1) 6-10
packageServiceUsageUpVolume (packageServiceUsageEntry 1) 6-10
packageServiceUsageDownVolume (packageServiceUsageEntry 2) 6-11
packageServiceUsageNumSessions (packageServiceUsageEntry 3) 6-11
packageServiceUsageDuration (packageServiceUsageEntry 4) 6-11
packageServiceUsageConcurrentSessions (packageServiceUsageEntry 5) 6-11
packageServiceUsageActiveSubscribers (packageServiceUsageEntry 6) 6-12
packageServiceUpDroppedPackets (packageServiceUsageEntry 7) 6-12
packageServiceDownDroppedPackets (packageServiceUsageEntry 8) 6-12
packageServiceUpDroppedBytes (packageServiceUsageEntry 9) 6-13
packageServiceDownDroppedBytes (packageServiceUsageEntry 10) 6-13
Subscriber Group: subscriberGrp (pcubeEngageObjs 4) 6-13

subscribersTable (subscriberGrp 1) 6-14
subscribersEntry (subscribersTable 1) 6-14
subscriberPackageIndex (subscribersEntry 1) 6-14
subscriberServiceUsageTable (subscriberGrp 2) 6-14
subscriberServiceUsageEntry (subscriberServiceUsageTable 1) 6-14
subscriberServiceUsageUpVolume (subscriberServiceUsageEntry 1) 6-15
subscriberServiceUsageDownVolume (subscriberServiceUsageEntry 2) 6-15
subscriberServiceUsageNumSessions (subscriberServiceUsageEntry 3) 6-15
subscriberServiceUsageDuration (subscriberServiceUsageEntry 4) 6-16
Service Counter Group: serviceCounterGrp (pcubeEngageObjs 5) 6-16
globalScopeServiceCounterTable (serviceCounterGrp 1) 6-16
globalScopeServiceCounterEntry (globalScopeServiceCounterTable 1) 6-16
globalScopeServiceCounterIndex (globalScopeServiceCounterEntry 1) 6-17
globalScopeServiceCounterStatus (globalScopeServiceCounterEntry 2) 6-17
globalScopeServiceCounterName (globalScopeServiceCounterEntry 3) 6-17
subscriberScopeServiceCounterTable (serviceCounterGrp 2) 6-17
subscriberScopeServiceCounterEntry (subscriberScopeServiceCounterTable 1) 6-17
subscriberScopeServiceCounterIndex (subscriberScopeServiceCounterEntry 1) 6-18
subscriberScopeServiceCounterStatus (subscriberScopeServiceCounterEntry 2) 6-18
subscriberScopeServiceCounterName (subscriberScopeServiceCounterEntry 3) 6-18
Accessing Subscriber InformationGuidelines for Using the CISCO-SCAS-BB MIB 6-18
globalScopeServiceCounterTable and subscriberScopeServiceCounterTable 6-19
packageCounterTable 6-19
Accessing Subscriber Information (the spvIndex) 6-20

About This Guide

Revised: May 30, 2007, OL-8410-04

This preface describes who should read the *Cisco Service Control Application for Broadband Reference Guide*, how it is organized, its document conventions, and how to obtain documentation and technical assistance.

This guide assumes a basic familiarity with the concept of the Cisco Service Control solution, the Service Control Engine (SCE) platforms, and related components.

This introduction provides information about the following topics:

- Document Revision History
- Audience
- Organization
- Document Conventions
- Related Documentation
- Obtaining Documentation, Obtaining Support, and Security Guidelines

Document Revision History

Cisco Service Control Release	Part Number	Publication Date
Release 3.1.0	OL-8410-04	May, 2007

Description of Changes

Added the following new features:

- NetFlow
- Virtual Links Usage RDR
- Asymmetric routing classification mode (see Protocols Identified on Unidirectional Flows)

Cisco Service Control Release	Part Number	Publication Date
Release 3.0.5	OL-8410-03	November, 2006

Description of Changes

Added the following new feature:

- Quota State Restore RDRs

Cisco Service Control Release	Part Number	Publication Date
Release 3.0	OL-8410-03	May, 2006

Description of Changes

Added the following new feature:

- Media Flow RDR

Added the following section to the document:

- SCE Subscriber Template CSV File

Cisco Service Control Release	Part Number	Publication Date
Release 3.0	OL-8410-03	December, 2005

Description of Changes

Created the *Cisco Service Control Application for Broadband Reference Guide* .

Chapters 1, 2, 3 of this document are based on Appendixes B, C, D of the Release 2.5.5 *Cisco Service Control Application for Broadband User Guide* .

Audience

This guide provides information about the data structures created and used by SCA BB. It is intended for:

- The administrator who is responsible for daily operation of the Cisco Service Control solution.
- Integrators who are developing applications on top of SCA BB.

Organization

The major sections of this guide are as follows:

Table 0-1

Chapter	Title	Description
Chapter 1	Default Service Configuration Reference Tables	Describes the default service configuration provided with the Cisco Service Control Application for Broadband (SCA BB).
Chapter 2	Raw Data Records: Formats and Field Contents	Lists the various RDRs produced by the Service Control Engine (SCE) platform and gives their structure, describes the columns and fields of each RDR, and states under what conditions each kind of RDR is generated. Also provides field-content information for fields generated by Service Control components (such as tags), and a description of the Periodic RDR Zero Adjustment Mechanism.
Chapter 3	NetFlow Records: Formats and Field Contents	Lists the RDRs whose data can be generated as NetFlow records and describes the fields that may be contained in a NetFlow record."
Chapter 4	Database Tables: Formats and Field Contents	Presents the different database tables used for storing RDRs (after their conversion by an adapter), and a description of the table columns (field names and types).
Chapter 5	CSV File Formats	Describes the location and structure of CSV files pertaining to service configuration, subscriber management, and data collection management.
Chapter 6	SCA BB Proprietary MIB Reference	Describes that part of the Cisco SCE proprietary MIB that provides configuration and runtime status for SCA BB.

Document Conventions

This guide uses the following conventions:

- **Bold** is used for commands, keywords, and buttons.
- *Italics* are used for command input for which you supply values.
- `Screen` font is used for examples of information that are displayed on the screen.
- **`Bold screen`** font is used for examples of information that you enter.
- Vertical bars (|) indicate separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate optional elements.
- Braces ({}) indicate a required choice.
- Braces within square brackets ([{}]) indicate a required choice within an optional element.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the guide.

Timesaver Means the *described action saves time*. You can save time by performing the action described in the paragraph.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Warning **Means *danger*. You are in a situation that could cause bodily injury. Before you work on any equipment, you must be aware of the hazards involved with electrical circuitry and familiar with standard practices for preventing accidents. To see translated versions of warnings, refer to the *Regulatory Compliance and Safety Information* document that accompanied the device.**

Related Documentation

Your SCE 2000 platform and the software running on it contain extensive features and functionality, which are documented in the following resources:

- Cisco CLI software:
  - *Cisco Service Control Engine (SCE) Software Configuration Guide*
  - *Cisco Service Control Engine (SCE) CLI Command Reference*

Note Note You can access Cisco software configuration and hardware installation and maintenance documentation on the World Wide Web at Cisco Website URL. Translated documentation is available at the following URL: International Cisco Website

- For initial installation and startup information, refer to the *SCE 2000 4xGBE Quick Start Guide* .

- For international agency compliance, safety, and statutory information for wide-area network (WAN) interfaces for the SCE 2000 platform, refer to the *Regulatory Compliance and Safety Information for Cisco Service Control Engine (SCE)* .
- For installation and configuration of the other components of the Service Control Management Suite refer to:
  - *Cisco SCMS Subscriber Management User Guide*
  - *Cisco SCMS Collection Manager User Guide*
  - *Cisco Service Control Application for Broadband User Guide*
  - *Cisco Service Control Application Reporter User Guide*
- To view Cisco documentation or obtain general information about the documentation, refer to the following sources:
  - Obtaining Documentation
  - The Cisco Information Packet that shipped with your SCE 2000 platform.

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

CHAPTER 1

Default Service Configuration Reference Tables

This chapter describes the default service configuration provided with the Cisco Service Control Application for Broadband (SCA BB). The default service configuration serves as a starting point for creating a service configuration tailored to customers' needs.

- Filter Rules
- Information About Protocols

Filter Rules

Filter rules allow you to instruct the Service Control Engine (SCE) platform to ignore some types of flow based on the flow's Layer 3 and Layer 4 properties, and transmit the flows unchanged.

The following table lists the filter rules defined in the default service configuration.

Table 1-1 Filter Rules

Flow Filter Name	Default State	Description
ICMP Filter	Active	Applies to ICMP packets, packets bypass the policy engine and are mapped to CoS BE
DNS (to network)	Active	Applies to UDP packets, network-side port is equal to 53, packets bypass the policy engine and are mapped to CoS BE
DNS (to subscriber)	Active	Applies to UDP packets, subscriber-side port is equal to 53, packets bypass the policy engine and are mapped to CoS BE
net-bios (to network)	Active	Applies to UDP packets, network-side port is equal to 137, packets bypass the policy engine and are mapped to CoS BE

Table 1-1 Filter Rules (continued)

Flow Filter Name	Default State	Description
net-bios (to subscriber)	Active	Applies to UDP packets, subscriber-side port is equal to 137, packets bypass the policy engine and are mapped to CoS BE
eDonkey UDP (to network)	Active	Applies to UDP packets, network-side ports in the range 4661 to 4665, packets bypass the policy engine and are mapped to CoS BE
eDonkey UDP (to subscriber)	Active	Applies to UDP packets, subscriber-side ports in the range 4661 to 4665, packets bypass the policy engine and are mapped to CoS BE
eMule UDP (to network)	Active	Applies to UDP packets, network-side ports in the range 4670 to 4674, packets bypass the policy engine and are mapped to CoS BE
eMule UDP (to subscriber)	Active	Applies to UDP packets, subscriber-side ports in the range 4670 to 4674, packets bypass the policy engine and are mapped to CoS BE
eMule UDP 2 (to network)	Active	Applies to UDP packets, network-side ports in the range 5670 to 5674, packets bypass the policy engine and are mapped to CoS BE
eMule UDP 2 (to subscriber)	Active	Applies to UDP packets, subscriber-side ports in the range 5670 to 5674, packets bypass the policy engine and are mapped to CoS BE
eMule UDP 3 (to network)	Active	Applies to UDP packets, network-side ports in the range 5780 to 5784, packets bypass the policy engine and are mapped to CoS BE
eMule UDP 3 (to subscriber)	Active	Applies to UDP packets, subscriber-side ports in the range 5780 to 5784, packets bypass the policy engine and are mapped to CoS BE

Table 1-1 Filter Rules (continued)

Flow Filter Name	Default State	Description
BGP Filter	Inactive	Applies to TCP packets, network-side port is equal to 179, packets bypass the policy engine and are mapped to CoS BE
DHCP Filter	Inactive	Applies to UDP packets, network-side ports in the range 67 to 68, packets bypass the policy engine and are mapped to CoS BE
OSPF Filter	Inactive	Applies to OSPFIGP packets, packets bypass the policy engine and are mapped to CoS BE
IS-IS Filter	Inactive	Applies to ISIS packets, packets bypass the policy engine and are mapped to CoS BE
IGRP Filter	Inactive	Applies to IGP packets, packets bypass the policy engine and are mapped to CoS BE
EIGRP Filter	Inactive	Applies to EIGRP packets, packets bypass the policy engine and are mapped to CoS BE
HSRP Filter 1	Inactive	Applies to UDP packets, network-side IP is equal to 224.0.0.2, packets bypass the policy engine and are mapped to CoS BE
HSRP Filter 2	Inactive	Applies to UDP packets, network-side port is equal to 1985, packets bypass the policy engine and are mapped to CoS BE
HSRP Filter 3	Inactive	Applies to UDP packets, subscriber-side port is equal to 1985, packets bypass the policy engine and are mapped to CoS BE
RIP Filter 1	Inactive	Applies to UDP packets, network-side IP is equal to 224.0.0.9, packets bypass the policy engine and are mapped to CoS BE

Table 1-1 Filter Rules (continued)

Flow Filter Name	Default State	Description
RIP Filter 2	Inactive	Applies to UDP packets, network-side port is equal to 520, packets bypass the policy engine and are mapped to CoS BE
RIP Filter 3	Inactive	Applies to UDP packets, subscriber-side port is equal to 520, packets bypass the policy engine and are mapped to CoS BE
RADIUS Filter	Inactive	Applies to UDP packets, network-side port is equal to 1812, packets bypass the policy engine and are mapped to CoS BE
RADIUS Filter (early deployment)	Inactive	Applies to UDP packets, network-side ports in the range 1645 to 1646, packets bypass the policy engine and are mapped to CoS BE

Information About Protocols

Protocols are divided into four groups:

- Generic Protocols - These protocols are used for transactions that were not mapped to a service by one of the more specific protocol types.
- Signature-Based Protocols - Protocols classified according to a Layer 7 application signature. This group includes the most common protocols, such as HTTP and FTP, and a large group of popular P2P protocols.
- IP Protocols - Protocols (such as ICMP), other than TCP and UDP protocols, identified according to the IP protocol number of the transaction.
- Port-Based Protocols - TCP and UDP protocols that are classified according to their well-known ports. The default configuration includes more than 600 common port-based protocols.

  You may add new protocols (for example, to classify a new gaming protocol that uses a specific port) and edit or remove existing ones.

The tables in the following sections list the protocols defined in the default service configuration.

- Generic Protocols
- Signature-Based Protocols
- IP Protocols
- Port-Based Protocols
- Protocols Identified on Unidirectional Flows
- Services

- RDR Settings
- Rules
- System Mode

Generic Protocols

The three generic protocols (IP, TCP, and UDP) serve as default containers for classifying transactions of the relevant type (IP, TCP, or UDP) that were not classified as belonging to a more specific protocol.

A transaction is classified as belonging to one of the generic protocols if it meets *both* the following conditions:

- It was not classified as belonging to a signature-based protocol.
- It was not classified as belonging to an IP or port-based protocol that is specifically mapped to a service.

Table 1-2 Generic Protocols

Protocol Name	ID	Description
Generic IP	10	Any non-TCP/UDP transaction where the related IP protocol is not specifically mapped to a service.
Generic TCP	0	Any TCP transaction that does not match any signature-based protocol, and where the related port-based protocol (if it exists) is not specifically mapped to a service1.
Generic UDP	1	Any UDP transaction that does not match any signature-based protocol, and where the related port-based protocol (if it exists) is not specifically mapped to a service.

Signature-Based Protocols

A transaction is classified as belonging to one of the signature-based protocols if it is carried on the protocol's well-known port or matches the protocol's signature.

Table 1-3 Signature-Based Protocols

Protocol Name	ID	TCP Ports	UDP Ports
Behavioral P2P	1044		
Behavioral Upload/Download (See note following table)	127		

Table 1-3 Signature-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
CUWorld	117		
DHCP Sniff	33		
DingoTel	42		
DNS	933		
Flash	1033		
Flash YouTube	1034		
Flash MySpace	1035		
Flash Yahoo	1036		
FTP	4	21	
Generic Non-Established TCP (See note following table)	126		
GoogleEarth	118		
Hopster	115		
HTTP Browsing	2	80, 8080	
HTTP Tunnel	55		
ICQ	119		
IRC	62		
Jabber	116		
MMS	6	1755	
Mobile MMS	46		
NNTP	15	119	
NTP	54		
POP3	9	110	
QQ	52		
RTSP Streaming	5	554, 1554, 7070	
Sling	112		
SMTP	8	25	
STUN	114		
Thunder	50		
UC	48		
Yahoo Messenger	40	5000-5001	5000-5001
imap	59	143	143
radius	738		
tftp	60	69	69

- Behavioral Upload/Download—Transactions that have download packet flow characteristics and do not match a more specific signature are classified to this protocol. This protocol applies to downloads both from the network side and from the subscriber side.
- Generic Non-Established TCP—TCP flows that are not established properly (syn-ack is missing) are mapped to this protocol.

Table 1-4 Signature-Based P2P Protocols

Protocol Name	ID	TCP Ports	UDP Ports
AntsP2P	113		
BaiBao	43		
BitTorrent	24	6881-6889	
DHT	106		
Dijjer	120		
DirectConnect	19	411-413	
Entropy	125		
Exosee	121		
FastTrack KaZaA File Transfer	14		
FastTrack KaZaA Networking	13	1214	
Filetopia	31		
Freenet	107		
Furthur	123		
Gnutella File Transfer	12		
Gnutella Networking	11	6346-6349	
Hotline	20		
Konspire2b	1031	6085	6085
Kontiki	124		
LOCO	5123		
Manolito	22		
Mute	34		
Napster	32		
NeoNet	37		
NodeZilla	35		
PeerEnabler	122		
Poco	51		
PPLive	44		
PPStream	49		
QQ-Live	1032		
Rodi	111		

Table 1-4 Signature-Based P2P Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
Share	27		
Soulseek	29		
SSDP	53		
TVAnts	109		
Warez/FileCroc	39		
Waste	36		
WinMX/OpenNap	16	6257, 6699	6257
Winny	17	7742-7745, 7773	
eDonkey	18	4661-4665, 4672-4673, 4711, 5662, 5773, 5783	4661-4665, 4672-4673, 4711, 5662, 5773, 5783
eMuleEncrypted	105		
guruguru	66		
iTunes	30		
kuro	67		
soribada	69		
v-share	71		

Table 1-5 Signature-Based VoIP Protocols

Protocol Name	ID	TCP Ports	UDP Ports
H323	28	1720	
ICQ VoIP	110		
MGCP	38		2427, 2727
Primus	108		
PTT Winphoria	61		
RTP	57		
SIP	23	5060-5061	5060-5061
Skinny	41		
Skype	25		
Yahoo Messenger VoIP	45	33033	
Yahoo VoIP over SIP	1039		

IP Protocols

This section lists the IP protocols supported by SCA BB.

Table 1-6 IP Protocols

IP Protocol Number	Protocol Name	Protocol ID
0	HOPOPT	756
1	ICMP	757
2	IGMP	758
3	GGP	759
4	IP	760
5	ST	761
6	Generic TCP	0
7	CBT	762
8	EGP	763
9	IGP	764
10	BBN-RCC-MON	765
11	NVP-II	766
12	PUP	767
13	ARGUS	768
14	EMCON	769
15	XNET	770
16	CHAOS	771
17	Generic UDP	1
18	MUX	772
19	DCN-MEAS	773
20	HMP	774
21	PRM	775
22	XNS-IDP	776
23	TRUNK-1	777
24	TRUNK-2	778
25	LEAF-1	779
26	LEAF-2	780
27	RDP	781
28	IRTP	782
29	ISO-TP4	783
30	NETBLT	784
31	MFE-NSP	785
32	MERIT-INP	786

Table 1-6 IP Protocols (continued)

IP Protocol Number	Protocol Name	Protocol ID
33	SEP	787
34	3PC	788
35	IDPR	789
36	XTP	790
37	DDP	791
38	IDPR-CMTP	792
39	TP++	793
40	IL	794
41	IPv6-Over-IPv4	795
42	SDRP	796
43	IPv6-Route	797
44	IPv6-Frag	798
45	IDRP	799
46	RSVP	800
47	GRE	801
48	MHRP	802
49	BNA	803
50	ESP	804
51	AH	805
52	I-NLSP	806
53	SWIPE	807
54	NARP	808
55	MOBILE	809
56	TLSP	810
57	SKIP	811
58	IPv6-ICMP	812
59	IPv6-NoNxt	813
60	IPv6-Opts	814
61	any host internal protocol	815
62	CFTP	816
63	any local network	817
64	SAT-EXPAK	818
65	KRYPTOLAN	819
66	RVD	820
67	IPPC	821
68	any distributed file system	822

Table 1-6 IP Protocols (continued)

IP Protocol Number	Protocol Name	Protocol ID
69	SAT-MON	823
70	VISA	824
71	IPCV	825
72	CPNX	826
73	CPHB	827
74	WSN	828
75	PVP	829
76	BR-SAT-MON	830
77	SUN-ND	831
78	WB-MON	832
79	WB-EXPAK	833
80	ISO-IP	834
81	VMTP	835
82	SECURE-VMTP	836
83	VINES	837
84	TTP	838
85	NSFNET-IGP	839
86	DGP	840
87	TCF	841
88	EIGRP	842
89	OSPFIGP	843
90	Sprite-RPC	844
91	LARP	845
92	MTP	846
93	AX.25	847
94	IPIP	848
95	MICP	849
96	SCC-SP	850
97	ETHERIP	851
98	ENCAP	852
99	any private encryption scheme	853
100	GMTP	854
101	IFMP	855
102	PNNI	856
103	PIM	857
104	ARIS	858

Table 1-6 IP Protocols (continued)

IP Protocol Number	Protocol Name	Protocol ID
105	SCPS	859
106	QNX	860
107	A/N	861
108	IPComp	862
109	SNP	863
110	Compaq-Peer	864
111	IPX-in-IP	865
112	VRRP	866
113	PGM	867
114	any 0-hop protocol	868
115	L2TP	869
116	DDX	870
117	IATP	871
118	STP	872
119	SRP	873
120	UTI	874
121	SMP	875
122	SM	876
123	PTP	877
124	ISIS	878
125	FIRE	879
126	CRTP	880

Port-Based Protocols

This section lists the TCP/UDP port-based protocols defined in the SCA BB default service configuration.

Table 1-7 Port-Based Protocols (Ports 1 to 400)

IP Protocol Number	Protocol Name	Protocol ID
0	HOPOPT	756
1	ICMP	757
2	IGMP	758
3	GGP	759
4	IP	760
5	ST	761
6	Generic TCP	0

Table 1-7 Port-Based Protocols (Ports 1 to 400) (continued)

IP Protocol Number	Protocol Name	Protocol ID
7	CBT	762
8	EGP	763
9	IGP	764
10	BBN-RCC-MON	765
11	NVP-II	766
12	PUP	767
13	ARGUS	768
14	EMCON	769
15	XNET	770
16	CHAOS	771
17	Generic UDP	1
18	MUX	772
19	DCN-MEAS	773
20	HMP	774
21	PRM	775
22	XNS-IDP	776
23	TRUNK-1	777
24	TRUNK-2	778
25	LEAF-1	779
26	LEAF-2	780
27	RDP	781
28	IRTP	782
29	ISO-TP4	783
30	NETBLT	784
31	MFE-NSP	785
32	MERIT-INP	786
33	SEP	787
34	3PC	788
35	IDPR	789
36	XTP	790
37	DDP	791
38	IDPR-CMTP	792
39	TP++	793
40	IL	794
41	IPv6-Over-IPv4	795
42	SDRP	796

Table 1-7 Port-Based Protocols (Ports 1 to 400) (continued)

IP Protocol Number	Protocol Name	Protocol ID
43	IPv6-Route	797
44	IPv6-Frag	798
45	IDRP	799
46	RSVP	800
47	GRE	801
48	MHRP	802
49	BNA	803
50	ESP	804
51	AH	805
52	I-NLSP	806
53	SWIPE	807
54	NARP	808
55	MOBILE	809
56	TLSP	810
57	SKIP	811
58	IPv6-ICMP	812
59	IPv6-NoNxt	813
60	IPv6-Opts	814
61	any host internal protocol	815
62	CFTP	816
63	any local network	817
64	SAT-EXPAK	818
65	KRYPTOLAN	819
66	RVD	820
67	IPPC	821
68	any distributed file system	822
69	SAT-MON	823
70	VISA	824
71	IPCV	825
72	CPNX	826
73	CPHB	827
74	WSN	828
75	PVP	829
76	BR-SAT-MON	830
77	SUN-ND	831
78	WB-MON	832

Table 1-7 Port-Based Protocols (Ports 1 to 400) (continued)

IP Protocol Number	Protocol Name	Protocol ID
79	WB-EXPAK	833
80	ISO-IP	834
81	VMTP	835
82	SECURE-VMTP	836
83	VINES	837
84	TTP	838
85	NSFNET-IGP	839
86	DGP	840
87	TCF	841
88	EIGRP	842
89	OSPFIGP	843
90	Sprite-RPC	844
91	LARP	845
92	MTP	846
93	AX.25	847
94	IPIP	848
95	MICP	849
96	SCC-SP	850
97	ETHERIP	851
98	ENCAP	852
99	any private encryption scheme	853
100	GMTP	854
101	IFMP	855
102	PNNI	856
103	PIM	857
104	ARIS	858
105	SCPS	859
106	QNX	860
107	A/N	861
108	IPComp	862
109	SNP	863
110	Compaq-Peer	864
111	IPX-in-IP	865
112	VRRP	866
113	PGM	867
114	any 0-hop protocol	868

Table 1-7 Port-Based Protocols (Ports 1 to 400) (continued)

IP Protocol Number	Protocol Name	Protocol ID
115	L2TP	869
116	DDX	870
117	IATP	871
118	STP	872
119	SRP	873
120	UTI	874
121	SMP	875
122	SM	876
123	PTP	877
124	ISIS	878
125	FIRE	879
126	CRTP	880

Table 1-8 Port-Based Protocols (Ports 401 to 700)

Protocol Name	ID	TCP Ports	UDP Ports
ups	316	401	401
genie	317	402	402
decap	318	403	403
nced	319	404	404
ncld	320	405	405
imsp	321	406	406
timbuktu	322	407	407
prm-sm	323	408	408
prm-nm	324	409	409
decladebug	325	410	410
rmt	326		411
synoptics-trap	327		412
smsp	328		413
infoseek	329	414	414
bnet	330	415	415
silverplatter	331	416	416
onmux	332	417	417
hyper-g	333	418	418
ariel1	334	419	419
smpte	335	420	420
ariel2	336	421	421

Table 1-8 Port-Based Protocols (Ports 401 to 700) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
ariel3	337	422	422
opc-job-start	338	423	423
opc-job-track	339	424	424
icad-el	340	425	425
smartsdp	341	426	426
svrloc	342	427	427
ocs_cmu	343	428	428
ocs_amu	344	429	429
utmpsd	345	430	430
utmpcd	346	431	431
iasd	347	432	432
nnsp	348	433	433
mobileip-agent	349	434	434
mobilip-mn	350	435	435
dna-cml	351	436	436
comscm	352	437	437
dsfgw	353	438	438
dasp	354	439	439
sgcp	355	440	440
decvms-sysmgt	356	441	441
cvc_hostd	357	442	442
https	358	443	
snpp	359	444	444
microsoft-ds	360	445	445
ddm-rdb	361	446	446
ddm-dfm	362	447	447
ddm-ssl	363	448	448
as-servermap	364	449	449
tserver	365	450	450
sfs-smp-net	366	451	451
sfs-config	367	452	452
creativeserver	368	453	453
contentserver	369	454	454
creativepartnr	370	455	455
scohelp	371	457	457
appleqtc	372	458	458

Table 1-8 Port-Based Protocols (Ports 401 to 700) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
ampr-rcmd	373	459	459
skronk	374	460	460
datasurfsrv	375	461	461
datasurfsrvsec	376	462	462
alpes	377	463	463
kpasswd	378	464	464
url-rendezvous	379	465	465
digital-vrc	380	466	466
mylex-mapd	381	467	467
photuris	382	468	468
rcp	383	469	469
scx-proxy	384	470	470
mondex	385	471	471
ljk-login	386	472	472
hybrid-pop	387	473	473
tn-tl-w1	388	474	
tn-tl-w2	389		474
tn-tl-fd1	390	476	476
ss7ns	391	477	477
spsc	392	478	478
iafserver	393	479	479
iafdbase	394	480	480
ph	395	481	481
bgs-nsi	396	482	482
ulpnet	397	483	483
integra-sme	398	484	484
powerburst	399	485	485
avian	400	486	486
saft	401	487	487
gss-http	402	488	488
nest-protocol	403	489	489
micom-pfs	404	490	490
go-login	405	491	491
ticf-1	406	492	492
ticf-2	407	493	493
pov-ray	408	494	494

Table 1-8 Port-Based Protocols (Ports 401 to 700) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
intecourier	409	495	495
pim-rp-disc	410	496	496
dantz	411	497	497
siam	412	498	498
iso-ill	413	499	499
isakmp	414	500	500
stmf	415	501	501
asa-appl-proto	416	502	502
intrinsa	417	503	503
citadel	418	504	504
mailbox-lm	419	505	505
ohimsrv	420	506	506
crs	421	507	507
xvttp	422	508	508
snare	423	509	509
fcp	424	510	510
passgo	425	511	511
exec	426	512	
biff	427		512
login	428	513	
who	429		513
shell	430	514	
syslog	431		514
printer	432	515	515
videotex	433	516	516
talk	434	517	517
ntalk	435	518	518
utime	436	519	519
efs	437	520	
router	438		520
ripng	439	521	521
ulp	440	522	522
ibm-db2	441	523	523
ncp	442	524	524
timed	443	525	525
tempo	444	526	526

Table 1-8 Port-Based Protocols (Ports 401 to 700) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
stx	445	527	527
custix	446	528	528
irc-serv	447	529	529
courier	448	530	530
conference	449	531	531
netnews	450	532	532
netwall	451	533	533
mm-admin	452	534	534
iiop	453	535	535
opalis-rdv	454	536	536
nmsp	455	537	537
gdomap	456	538	538
apertus-ldp	457	539	539
uucp	458	540	540
uucp-rlogin	459	541	541
commerce	460	542	542
klogin	461	543	543
kshell	462	544	544
appleqtcsrvr	463	545	545
dhcpv6-client	464	546	546
dhcpv6-server	465	547	547
idfp	466	549	549
new-rwho	467	550	550
cybercash	468	551	551
deviceshare	469	552	552
pirp	470	553	553
remotefs	471	556	556
openvms-sysipc	472	557	557
sdnskmp	473	558	558
teedtap	474	559	559
rmonitor	475	560	560
monitor	476	561	561
chshell	477	562	562
nntps	478	563	563
9pfs	479	564	564
whoami	480	565	565

Table 1-8 Port-Based Protocols (Ports 401 to 700) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
streettalk	481	566	566
banyan-rpc	482	567	567
ms-shuttle	483	568	568
ms-rome	484	569	569
meter	485	570-571	570-571
sonar	486	572	572
banyan-vip	487	573	573
ftp-agent	488	574	574
vemmi	489	575	575
ipcd	490	576	576
vnas	491	577	577
ipdd	492	578	578
decbsrv	493	579	579
sntp-heartbeat	494	580	580
bdp	495	581	581
scc-security	496	582	582
philips-vc	497	583	583
keyserver	498	584	584
imap4-ssl	499	585	585
password-chg	500	586	586
submission	501	587	587
cal	502	588	588
eyelink	503	589	589
tns-cml	504	590	590
http-alt	505	591	591
eudora-set	506	592	592
http-rpc-epmap	507	593	593
tpip	508	594	594
cab-protocol	509	595	595
smsd	510	596	596
ptcnameservice	511	597	597
sco-websrvrmg3	512	598	598
acp	513	599	599
ipcserver	514	600	600
urm	515	606	606
nqs	516	607	607

Table 1-8 Port-Based Protocols (Ports 401 to 700) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
sift-uft	517	608	608
npmp-trap	518	609	609
npmp-local	519	610	610
npmp-gui	520	611	611
hmmp-ind	521	612	612
hmmp-op	522	613	613
sshell	523	614	614
sco-inetmgr	524	615	615
sco-sysmgr	525	616	616
sco-dtmgr	526	617	617
dei-icda	527	618	618
digital-evm	528	619	619
sco-websrvrmgr	529	620	620
escp-ip	530	621	621
collaborator	531	622	622
aux_bus_shunt	532	623	623
cryptoadmin	533	624	624
dec_dlm	534	625	625
asia	535	626	626
passgo-tivoli	536	627	627
qmqp	537	628	628
3com-amp3	538	629	629
rda	539	630	630
ipp	540	631	631
bmpp	541	632	632
servstat	542	633	633
ginad	543	634	634
rlzdbase	544	635	635
ldaps	545	636	636
lanserver	546	637	637
mcns-sec	547	638	638
msdp	548	639	639
entrust-sps	549	640	640
repcmd	550	641	641
esro-emsdp	551	642	642
sanity	552	643	643

Table 1-8 Port-Based Protocols (Ports 401 to 700) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
dwr	553	644	644
pssc	554	645	645
ldp	555	646	646
dhcp-failover	556	647	647
rrp	557	648	648
aminet	558	649	649
obex	559	650	650
ieee-mms	560	651	651
hello-port	561	652	652
repscmd	562	653	653
aodv	563	654	654
tinc	564	655	655
spmp	565	656	656
rmc	566	657	657
tenfold	567	658	658
mac-srvr-admin	568	660	660
hap	569	661	661
pftp	570	662	662
purenoise	571	663	663
secure-aux-bus	572	664	664
sun-dr	573	665	665
doom	574	666	666
disclose	575	667	667
mecomm	576	668	668
meregister	577	669	669
vacdsm-sws	578	670	670
vacdsm-app	579	671	671
vpps-qua	580	672	672
cimplex	581	673	673
acap	582	674	674
dctp	583	675	675
vpps-via	584	676	676
vpp	585	677	677
ggf-ncp	586	678	678
mrm	587	679	679
entrust-aaas	588	680	680

Table 1-8 Port-Based Protocols (Ports 401 to 700) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
entrust-aams	589	681	681
xfr	590	682	682
corba-iiop	591	683	683
corba-iiop-ssl	592	684	684
mdc-portmapper	593	685	685
hcp-wismar	594	686	686
asipregistry	595	687	687
realm-rusd	596	688	688
nmap	597	689	689
vatp	598	690	690
msexch-routing	599	691	691
hyperwave-isp	600	692	692
connendp	601	693	693
ha-cluster	602	694	694
ieee-mms-ssl	603	695	695
rushd	604	696	696
uuidgen	605	697	697
olsr	606	698	698
accessnetwork	607	699	699

Table 1-9 Port-Based Protocols (Ports 701+)

Protocol Name	ID	TCP Ports	UDP Ports
elcsd	608	704	704
agentx	609	705	705
silc	610	706	706
borland-dsj	611	707	707
entrust-kmsh	612	709	709
entrust-ash	613	710	710
cisco-tdp	614	711	711
netviewdm1	615	729	729
netviewdm2	616	730	730
netviewdm3	617	731	731
netgw	618	741	741
netrcs	619	742	742
flexlm	620	744	744
fujitsu-dev	621	747	747

Table 1-9 Port-Based Protocols (Ports 701+) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
ris-cm	622	748	748
kerberos-adm	623	749	749
rfile	624	750	
kerberos-iv	625		750
pump	626	751	751
qrh	627	752	752
rrh	628	753	753
tell	629	754	754
nlogin	630	758	758
con	631	759	759
ns	632	760	760
rxe	633	761	761
quotad	634	762	762
cycleserv	635	763	763
omserv	636	764	764
webster	637	765	765
phonebook	638	767	767
vid	639	769	769
cadlock	640	770	770
rtip	641	771	771
cycleserv2	642	772	772
submit	643	773	
notify	644		773
rpasswd	645	774	
acmaint_dbd	646		774
entomb	647	775	
acmaint_transd	648		775
wpages	649	776	776
multiling-http	650	777	777
wpgs	651	780	780
concert	652	786	786
qsc	653		787
mdbs_daemon	654	800	800
device	655	801	801
itm-mcell-s	656	828	828
pkix-3-ca-ra	657	829	829

Table 1-9 Port-Based Protocols (Ports 701+) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
dhcp-failover2	658	847	847
rsync	659	873	873
iclcnet-locate	660	886	886
iclcnet_svinfo	661	887	887
accessbuilder	662	888	888
omginitialrefs	663	900	900
smpnameres	664	901	901
ideafarm-chat	665	902	902
ideafarm-catch	666	903	903
xact-backup	667	911	911
ftps-data	668	989	989
ftps	669	990	990
nas	670	991	991
telnets	671	992	992
imaps	672	993	993
ircs	673	994	994
pop3s	674	995	995
vsinet	675	996	996
maitrd	676	997	997
busboy	677	998	
puparp	678		998
garcon	679	999	
applix	680		999
surf	681	1010	1010
Need For Speed 3	1018	1030	1030
rmiactivation	682	1098	1098
rmiregistry	683	1099	1099
Westwood Online	1028	1140, 1234	1140, 1234
GLT Poliane	882	1201	
ms-sql-s	684	1433	1433
ms-sql-m	685	1434	1434
oracle	690	1521	1521
orasrv	691	1525	1525
tlisrv	692	1527	1527
coauthor	693	1529	1529
micromuse-lm	702	1534	1534

Table 1-9 Port-Based Protocols (Ports 701+) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
orbixd	703	1570	1570
rdb-dbs-disp	694	1571	1571
oraclenames	695	1575	1575
shockwave	707	1626	1626
oraclenet8cman	696	1630	1630
l2tp	742	1701	1701
pptp	739	1723	1723
net8-cman	697	1830	1830
msnp	713	1836	1836
MSN Messenger	883	1863	1863
gtp-user	740	2152	2152
kali	718	2213	2213
directplay	716	2234	2234
Rainbox six	1026	2346	2346
ms-olap	686	2382-2383, 2393-2394	2382-2383, 2393-2394
groove	715	2492	2492
citrixima	698	2512	2512
citrixadmin	699	2513	2513
worldfusion	719	2595-2596	2595-2596
citriximaclient	701	2598	2598
Black And White	1006	2611-2612	
sitaraserver	708	2629	2629
sitaramgmt	709	2630	2630
sitaradir	710	2631	2631
wta-wsp-s	724	2805	2805
citrix-rtmp	700	2897	2897
wap-push	725	2948	2948
wap-pushsecure	726	2949	2949
xbox live	898	3074	3074
orbix-locator	704	3075	3075
orbix-config	705	3076	3076
orbix-loc-ssl	706	3077	3077
xdtp	741	3088	3088
Delta Force	1025	3100, 3999	3100, 3999, 3568, 3569
msft-gc	687	3268	3268
msft-gc-ssl	688	3269	3269

Table 1-9 Port-Based Protocols (Ports 701+) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
net-assistant	712	3283	3283
mysql	711	3306	3306
directv-web	720	3334	3334
directv-soft	721	3335	3335
directv-tick	722	3336	3336
directv-catlg	723	3337	3337
ms-term-services	689	3389	3389
Myth	1016	3453	3453
Warcraft	1023	3724	3724
Kohan Immortal Sovereigns	1014	3855, 17437	3855, 17437
F16	1011		3862, 3863
F22 Simulator (lightning 3)	1012		3874-3875, 4533, 4534
wap-push-http	727	4035	4035
wap-push-https	728	4036	4036
Ultima	1022	5002-5010, 7775-7777, 8888, 9999, 7875	
aim	714	5190-5193	
Google Talk	1030	5222	
Outlaws	1020	5310	5310
directplay8	717	6073	6073
Konspire2b	1031	6085	6085
fsgs	743	6112	6112
Diablo	1009	6113-6119	6113-6119
game-spy	755	6500, 28900, 29000	6515, 27900
parsec-game	744	6582	6582
ibprotocol	737	6714	6714
Anarchy	1004	7013, 7500-7501	7013, 7500-7501
UnReal_UT	745	7778	7777-7783
Znes	1024		7845
Asherons Call	1005	9000-9013	9000-9013
wap-wsp	729	9200	9200
wap-wsp-wtp	730	9201	9201
wap-wsp-s	731	9202	9202
wap-wsp-wtp-s	732	9203	9203
wap-vcard	733	9204	9204

Table 1-9 Port-Based Protocols (Ports 701+) (continued)

Protocol Name	ID	TCP Ports	UDP Ports
wap-vcal	734	9205	9205
wap-vcard-s	735	9206	9206
wap-vcal-s	736	9207	9207
Need For Speed	1017	9442	9442
ps2	899	10070-10080	10070
Yahoo Games	1029	11999	
Motorhead	1015	16000, 16010-16030	16000, 16010-16030
Swat3	1021	16639	16638
SiN	746	22450	22450
Elite Force	1010		26000, 27500
Dark Reign	1008	26214	26214
Hexen	1013		26900
halflife	747		27015
Counter strike	1007	27020-27039	1200, 27000-27014
quake-server	754	27960	27910, 27960
tribes	748	28001	28001
heretic2	749	28910	
Soldier of fortune	1027		28911-28915
starsiege	750		29001-29009
game-search	751	29001	
KingPin	752	31510	31510
runescape	753	43594	
Operation Flash Point	1019	47624	

Protocols Identified on Unidirectional Flows

When asymmetric routing classification mode is enabled, the protocols listed in the following table can be detected on unidirectional flows.

- When a unidirectional flow (inbound or outbound) passes through the SCE platform it is matched against this set of protocol signatures.
- When a bidirectional flow passes through the SCE platform the protocol library tries to match it to one of its standard (bidirectional) protocol signatures.

Table 1-10 Unidirectionally-Detected Protocols

Protocol Name	Protocol ID
AntsP2P	113
BaiBao	43
Behavioral Upload/Download	127

Table 1-10 Unidirectionally-Detected Protocols

Protocol Name	Protocol ID
BitTorrent	24
CUWorld	117
Dijjer	120
DingoTel	42
DirectConnect	19
EmuleEncrypted	105
Entropy	125
Exosee	121
FastTrack KaZaA File Transfer	14
Filetopia	31
Flash	1033
Flash MySpace	1035
Flash Yahoo	1036
Flash YouTube	1034
Furthur	123
Generic TCP	0
Gnutella File Transfer	12
Gnutella Networking	11
GoogleEarth	118
HTTP Browsing	2
HTTP Tunnel	55
Hopster	115
Hotline	20
ICQ	119
Jabber	116
Kontiki	124
Manolito	22
Mobile MMS	46
Mute	34
Napster	32
NeoNet	37
NodeZilla	35
POCO	51
PPLive	44
PPStream	49
PeerEnabler	122

Table 1-10 Unidirectionally-Detected Protocols

Protocol Name	Protocol ID
QQ-Live	1032
Skype	25
Sling	112
TVAnts	109
Thunder	50
UC	48
Warez/FileCroc	39
WinMX/OpenNap	16
Yahoo Messenger	40
Yahoo Messenger VoIP	45
eDonkey	18
guruguru	66
iTunes	30
soribada	69
v-share	71

Services

Services are the building blocks of service configurations. Classification of a transaction to a service determines the accounting and control that applies to the transaction. Services are organized in a hierarchal structure used for both accounting and control.

The following table lists the services defined in the default service configuration. Both service usage counters, which are used to accumulate information about transactions classified to the service, have the same name.

Table 1-11 Installed Services

Name	ID	Name of Parent Service	Global Usage Counter and Subscriber Usage Counter
Default Service	0		Default Service*
Generic	1	Default Service	Default Service*
Generic TCP	2	Generic	Generic TCP
Generic UDP	3	Generic	Generic UDP
Generic IP	6	Generic	Generic IP
Behavioral Upload/Download	39	Generic	Behavioral Upload/Download
E-Mail	4	Default Service	E-Mail*
POP3	21	E-Mail	E-Mail*
SMTP	22	E-Mail	E-Mail*

Table 1-11 Installed Services (continued)

Name	ID	Name of Parent Service	Global Usage Counter and Subscriber Usage Counter
IMAP	23	E-Mail	E-Mail*
Browsing	7	Default Service	Browsing*
HTTP	16	Browsing	Browsing*
HTTPS	17	Browsing	Browsing*
Newsgroups	8	Default Service	Newsgroups
P2P	9	Default Service	P2P
eDonkey/eMule	14	P2P	eDonkey/eMule
Kazaa	15	P2P	Kazaa
BitTorrent	24	P2P	BitTorrent
Commercial File Sharing	26	P2P	Commercial File Sharing
Winny	27	P2P	Winny
Gnutella	30	P2P	Gnutella
WinMX	31	P2P	WinMX
VoIP	12	Default Service	VoIP
MGCP	5	VoIP	MGCP
SIP	10	VoIP	SIP
H323	11	VoIP	H323
Vonage	13	VoIP	Vonage
Skype	25	VoIP	Skype
Skinny	35	VoIP	Skinny
DingoTel	36	VoIP	DingoTel
Yahoo Messenger VoIP	37	VoIP	Yahoo Messenger VoIP
ICQ VoIP	40	VoIP	ICQ VoIP
Instant Messaging	28	Default Service	Instant Messaging
Gaming	29	Default Service	Gaming
FTP	32	Default Service	FTP
Net Admin	33	Default Service	Net Admin
Streaming	34	Default Service	Streaming*
Streaming over HTTP	18	Streaming	Streaming*
RTSP	19	Streaming	Streaming*
MMS	20	Streaming	Streaming*
Tunneling	38	Default Service	Tunneling

Note An asterisk is appended to a service usage counter name whenever the counter applies to more than one service.

RDR Settings

SCE platforms generate and transmit Raw Data Records (RDRs) that contain a wide variety of information and statistics, depending on the configuration of the system.

Table 1-12 Default RDR Settings

RDR Family	RDR Name	State	Rate	Rate Limit	Notes
Usage	Link	ON	Every 5 minutes		
	Package	ON	Every 5 minutes		
	Subscriber	ON	Every 10 minutes	200 per second	
	Virtual Links	OFF	Every 10 minutes		Default is ON for service configurations created in Virtual Links mode.
Transaction	Transaction	ON		100 per second	All services have the same relative weight.
Transaction Usage	Transaction Usage (TUR)	OFF			No threshold.
	Interim TUR	OFF			
	Media Flow	ON			
Quota	Breach	OFF			
	Remaining	OFF	Every 5 minutes	100 per second	
	Threshold	OFF			Generate RDR when balance goes below 10 MB.
	Restore Quota	OFF			Generated upon subscriber introduction.
Log	Block	ON		20 per second	

Table 1-12 Default RDR Settings (continued)

RDR Family	RDR Name	State	Rate	Rate Limit	Notes
Real-Time Subscriber	Real-Time Subscriber Usage	ON	Every 1 minutes	100 per second	Enable for each subscriber separately, using CLI.
Real-Time Signaling	Flow Signaling	OFF			
	Attack Signaling	OFF			
Malicious Traffic	Malicious Traffic	ON			

Rules

Rules are set of configurable instructions telling the application how to handle flows classified to a service.

The default service configuration contains a single rule for the default service. Until you create other rules, the default service rule applies to all traffic processed by the SCE platform.

The default service rule places no restrictions on traffic:

- Flows are routed through the default BWCs, which have unlimited BW.
- No quota limitations are applied to the flows and external quota management mode is selected.

System Mode

The default System Operational Mode is Report Only, which means that the system is used for reporting but does not control traffic.

The default System Topological Mode is Duplex, which means that all inbound and outbound traffic goes through the SCE platform.

Note When asymmetric routing classification mode is enabled, there are some changes to the default service configuration:

- There are no predefined flavors.
- No service elements include a specified flavor.
- Periodic quota management mode is selected.

CHAPTER 2

Raw Data Records: Formats and Field Contents

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

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

- Raw Data Records Overview
- Universal RDR Fields
- Transaction RDR
- Transaction Usage RDR
- HTTP Transaction Usage RDR
- RTSP Transaction Usage RDR
- VoIP Transaction Usage RDR
- Subscriber Usage RDR
- Real-Time Subscriber Usage RDR
- Link Usage RDR
- Package Usage RDR
- Virtual Links Usage RDR
- Blocking RDR
- Quota Breach RDR
- Remaining Quota RDR
- Quota Threshold Breach RDR
- Quota State Restore RDRs
- DHCP RDR
- RADIUS RDR
- Flow Start RDR
- Flow End RDR
- Ongoing Flow RDR
- Media Flow RDR
- Attack Start RDR

- Attack End RDR
- Malicious Traffic Periodic RDR
- Information About RDR Enumeration Fields

Raw Data Records Overview

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

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

Universal RDR Fields

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

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

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

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

Note Note All volumes in RDRs are reported in L3 bytes.

Transaction RDR

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

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

The following table lists the RDR fields and their descriptions.

Table 2-1 Transaction RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	INT16	See Universal RDR Fields.
SERVICE_ID	INT32	See Universal RDR Fields.
PROTOCOL_ID	INT16	See Universal RDR Fields.

Table 2-1 Transaction RDR Fields (continued)

RDR Field Name	Type	Description
SKIPPED_SESSIONS	INT32	The number of unreported sessions since the previous RDR.
SERVER_IP	UINT32	See Universal RDR Fields.
SERVER_PORT	UINT16	See Universal RDR Fields.
ACCESS_STRING	STRING	See Universal RDR Fields.
INFO_STRING	STRING	See Universal RDR Fields.
CLIENT_IP	UINT32	See Universal RDR Fields.
CLIENT_PORT	UINT16	See Universal RDR Fields.
INITIATING_SIDE	INT8	See Universal RDR Fields.
REPORT_TIME	INT32	See Universal RDR Fields.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	See Universal RDR Fields.
SESSION_UPSTREAM_VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	See Universal RDR Fields.
ZONE_ID	INT32	See Universal RDR Fields.
FLAVOR_ID	INT32	See Universal RDR Fields.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow.

Transaction Usage RDR

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-2 Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	UINT16	See Universal RDR Fields.
SERVICE_ID	INT32	See Universal RDR Fields.
PROTOCOL_ID	INT16	See Universal RDR Fields.
SKIPPED_SESSIONS	INT32	Number of unreported sessions since the previous RDR.
SERVER_IP	UINT32	See Universal RDR Fields.
SERVER_PORT	UINT16	See Universal RDR Fields.
ACCESS_STRING	STRING	See Universal RDR Fields.
INFO_STRING	STRING	See Universal RDR Fields.
CLIENT_IP	UINT32	See Universal RDR Fields.
CLIENT_PORT	UINT16	See Universal RDR Fields.
INITIATING_SIDE	INT8	See Universal RDR Fields.
REPORT_TIME	INT32	See Universal RDR Fields.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	See Universal RDR Fields.
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.

Table 2-2 Transaction Usage RDR Fields (continued)

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

HTTP Transaction Usage RDR

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-3 HTTP Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	UINT16	See Universal RDR Fields.
SERVICE_ID	INT32	See Universal RDR Fields.

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

RDR Field Name	Type	Description
PROTOCOL_ID	INT16	See Universal RDR Fields.
SKIPPED_SESSIONS	INT32	Number of unreported sessions since the previous RDR.
SERVER_IP	UINT32	See Universal RDR Fields.
SERVER_PORT	UINT16	See Universal RDR Fields.
ACCESS_STRING	STRING	See Universal RDR Fields.
INFO_STRING	STRING	See Universal RDR Fields.
CLIENT_IP	UINT32	See Universal RDR Fields.
CLIENT_PORT	UINT16	See Universal RDR Fields.
INITIATING_SIDE	INT8	See Universal RDR Fields.
REPORT_TIME	INT32	See Universal RDR Fields.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	See Universal RDR Fields.
SESSION_UPSTREAM_VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated stream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	See Universal RDR Fields.
ZONE_ID	INT32	See Universal RDR Fields.
FLAVOR_ID	INT32	See Universal RDR Fields.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow.

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

RDR Field Name	Type	Description
USER_AGENT	STRING	The user agent field extracted from the HTTP transaction.
HTTP_URL	STRING	The URL extracted from the HTTP transaction.

RTSP Transaction Usage RDR

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-4 RTSP Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	UINT16	See Universal RDR Fields.
SERVICE_ID	INT32	See Universal RDR Fields.
PROTOCOL_ID	INT16	See Universal RDR Fields.
SKIPPED_SESSIONS	INT32	Number of unreported sessions since the previous RDR.
SERVER_IP	UINT32	See Universal RDR Fields.
SERVER_PORT	UINT16	See Universal RDR Fields.
ACCESS_STRING	STRING	See Universal RDR Fields.
INFO_STRING	STRING	See Universal RDR Fields.
CLIENT_IP	UINT32	See Universal RDR Fields.
CLIENT_PORT	UINT16	See Universal RDR Fields.
INITIATING_SIDE	INT8	See Universal RDR Fields.
REPORT_TIME	INT32	See Universal RDR Fields.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	See Universal RDR Fields.

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

RDR Field Name	Type	Description
SESSION_UPSTREAM_VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated stream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	See Universal RDR Fields.
ZONE_ID	INT32	See Universal RDR Fields.
FLAVOR_ID	INT32	See Universal RDR Fields.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow.
RTSP_SESSION_ID	STRING	RTSP session ID as seen on an RTSP SETUP request.
RTSP_URL	STRING	RTSP URL.
RESPONSE_DATE	STRING	RTSP DESCRIBE date.
TOTAL_ENCODING_RATE	UINT32	Sum of encoding rates of data flows.
NUMBER_OF_VIDEO_STREAMS	UINT8	Number of video streams for this RTSP session.
NUMBER_OF_AUDIO_STREAMS	UINT8	Number of audio streams for this RTSP session.
SESSION_TITLE	STRING	Title for this RTSP stream.
SERVER_NAME	STRING	Name of the RTSP server.

VoIP Transaction Usage RDR

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

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-5 VoIP Transaction RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	UINT16	See Universal RDR Fields.
SERVICE_ID	INT32	See Universal RDR Fields.
PROTOCOL_ID	INT16	See Universal RDR Fields.
SKIPPED_SESSIONS	INT32	Number of unreported sessions since the previous RDR
SERVER_IP	UINT32	See Universal RDR Fields.
SERVER_PORT	UINT16	See Universal RDR Fields.
ACCESS_STRING	STRING	See Universal RDR Fields.
INFO_STRING	STRING	See Universal RDR Fields.
CLIENT_IP	UINT32	See Universal RDR Fields.
CLIENT_PORT	UINT16	See Universal RDR Fields.
INITIATING_SIDE	INT8	See Universal RDR Fields.
REPORT_TIME	INT32	See Universal RDR Fields.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	See Universal RDR Fields.
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.

Table 2-5 VoIP Transaction RDR Fields (continued)

RDR Field Name	Type	Description
SESSION_DOWNSTREAM_VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	See Universal RDR Fields.
ZONE_ID	INT32	See Universal RDR Fields.
FLAVOR_ID	INT32	See Universal RDR Fields.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow.
APPLICATION_ID	UINT32	The ITU-U vendor ID of the application. A value of 0xFFFFFFFF indicates that this field was not found in the traffic.
UPSTREAM_PACKET_LOSS	UINT16	The average fractional upstream packet loss for the session, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.) A value of 0xFFFF indicates that this field is undefined (no RTCP flows were opened).
DOWNSTREAM_PACKET_LOSS	UINT16	The average fractional downstream packet loss for the session, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.) A value of 0xFFFF indicates that this field is undefined (no RTCP flows were opened).

Table 2-5 VoIP Transaction RDR Fields (continued)

RDR Field Name	Type	Description
UPSTREAM_AVERAGE_JITTER	UINT32	The average upstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.) A value of 0xFFFFFFFF indicates that this field is undefined (no RTCP flows were opened).
DOWNSTREAM_AVERAGE_JITTER	UINT32	The average downstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.) A value of 0xFFFFFFFF indicates that this field is undefined (no RTCP flows were opened).
CALL_DESTINATION	STRING	The Q931 Alias address of the session destination. A value of N/A indicates that this field was not found in the traffic.
CALL_SOURCE	STRING	The Q931 Alias address of the session source. A value of N/A indicates that this field was not found in the traffic.
UPSTREAM_PAYLOAD_TYPE	UINT8	The upstream RTP payload type for the session. A value of 0xFF indicates that this field was not available (no RTP flows were opened).
DOWNSTREAM_PAYLOAD_TYPE	UINT8	The downstream RTP payload type for the session. A value of 0xFF indicates that this field is undefined (no RTP flows were opened).

Table 2-5 VoIP Transaction RDR Fields (continued)

RDR Field Name	Type	Description
CALL_TYPE	UINT8	The call type (taken from H225 packet). A value of 0xFF indicates that this field is undefined (no RTP flows were opened).
MEDIA_CHANNELS	UINT8	The number of data flows that were opened during the session.

Note Packet Loss

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

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. See RFC 1889 for more information about the RCP/RTCP standard.

Subscriber Usage RDR

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

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

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

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-6 Subscriber Usage RDR

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	INT16	See Universal RDR Fields.
SERVICE_USAGE_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 counters in the subscriber scope.
BREACH_STATE	UINT8	See Universal RDR Fields. Holds the breach state of a service. However, this RDR reports usage counters, which cannot be breached, so the value is always zero.
REASON	UINT8	Reason for RDR generation: • 0—Period time passed • 1—Subscriber logout • 2—Package switch • 3—Wraparound • 4—End of aggregation period
CONFIGURED_DURATION	INT32	See Universal RDR Fields.
DURATION	INT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	INT32	See Universal RDR Fields.
UPSTREAM_VOLUME	INT32	Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period.

Table 2-6 Subscriber Usage RDR (continued)

RDR Field Name	Type	Description
DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT16	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT16	Aggregated number of session seconds for the reported service, for the current reporting period.

Real-Time Subscriber Usage RDR

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

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

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

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

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

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

- If there was only one interval (for example, 0805–0810) for which there was no subscriber consumption of resources, only one zero-consumption Real-Time Subscriber Usage RDR is generated.

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

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

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-7 Real-Time Subscriber Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	INT16	See Universal RDR Fields.
SERVICE_USAGE_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 counters in the subscriber scope.
AGGREGATION_OBJECT_ID	INT16	Externally assigned: • 0—Offline subscriber • 1—Online subscriber
BREACH_STATE	UINT8	See Universal RDR Fields. Holds the breach state of a service. However, this RDR reports usage counters, which cannot be breached, so the value is always zero.

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

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

Link Usage RDR

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

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

- If network resources associated with a service usage counter were consumed since the previous RDR generation point, a Link Usage RDR is generated.

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

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-8 Link Usage RDR Fields

RDR Field Name	Type	Description
LINK_ID	INT8	A numeric value associated with the reported network link. Possible values are 0 and 1 (referring to physical links 1 and 2 respectively). For future use.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR. Possible values are 0 to 3.
SERVICE_USAGE_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
CONFIGURED_DURATION	INT32	See Universal RDR Fields.
DURATION	INT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	INT32	See Universal RDR Fields.
UPSTREAM_VOLUME	INT32	Aggregated upstream volume of all sessions, in kilobytes, for the current reporting period.

Table 2-8 Link Usage RDR Fields (continued)

RDR Field Name	Type	Description
DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume of all sessions, in kilobytes, for the current reporting period.
SESSIONS	INT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	INT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	INT32	Concurrent number of sessions using the reported service at this point in time.
ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers using the reported service at this point in time.
TOTAL_ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers in the system at this point in time.

Package Usage RDR

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

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

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

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

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-9 Package Usage RDR Fields

RDR Field Name	Type	Description
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.
SERVICE_USAGE_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 64 global usage counters.
CONFIGURED_DURATION	INT32	See Universal RDR Fields.
DURATION	INT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	INT32	See Universal RDR Fields.
UPSTREAM_VOLUME	INT32	Aggregated upstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period.
SESSIONS	INT32	Aggregated number of sessions for the reported service, for the current reporting period.

Table 2-9 Package Usage RDR Fields (continued)

RDR Field Name	Type	Description
SECONDS	INT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	INT32	Concurrent number of sessions using the reported service in the reported package at this point in time.
ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers using the reported service in the reported package at this point in time.
TOTAL_ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers in the system at this point in time.

Virtual Links Usage RDR

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

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

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

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-10 Virtual Links Usage RDR Fields

RDR Field Name	Type	Description
VLINK_ID	INT16	The virtual link ID
VLINK_DIRECTION	INT8	The virtual link direction: • 0—Upstream • 1—Downstream
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.
SERVICE_USAGE_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 1024 global usage counters.
CONFIGURED_DURATION	INT32	See Universal RDR Fields.
DURATION	INT32	Not implemented (always the same as CONFIGURED_DURATION).
END_TIME	INT32	See Universal RDR Fields.
UPSTREAM_VOLUME	INT32	Aggregated upstream volume on the virtual link (for a single processor) of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume on the virtual link (for a single processor) of all sessions, in kilobytes, for the current reporting period.
SESSIONS	INT32	Reserved for future use.
SECONDS	INT32	Reserved for future use.
CONCURRENT_SESSIONS	INT32	Reserved for future use.
ACTIVE_SUBSCRIBERS	INT32	Reserved for future use.
TOTAL_ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers in the system at this point in time.

Blocking RDR

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-11 Blocking RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	UINT16	See Universal RDR Fields.
SERVICE_ID	INT32	See Universal RDR Fields.
PROTOCOL_ID	INT16	See Universal RDR Fields.
CLIENT_IP	UINT32	See Universal RDR Fields.
CLIENT_PORT	UINT16	See Universal RDR Fields.
SERVER_IP	UINT32	See Universal RDR Fields.
SERVER_PORT	UINT16	See Universal RDR Fields.
INITIATING_SIDE	INT8	See Universal RDR Fields.
ACCESS_STRING	STRING	See Universal RDR Fields.
INFO_STRING	STRING	See Universal RDR Fields.
BLOCK_REASON	UINT8	Indicates the reason why this session was blocked. See Block Reason (uint8) for possible values and their interpretation.
BLOCK_RDR_COUNT	INT32	Total number of blocked flows reported so far (from the beginning of the current time frame).

Table 2-11 Blocking RDR Fields (continued)

RDR Field Name	Type	Description
REDIRECTED	INT8	Indicates whether the flow has been redirected after being blocked. • 0—Not redirected • 1—Redirected Redirection is performed only for HTTP and RTSP flows that were mapped to a rule ordering them to be blocked and redirected.
REPORT_TIME	INT32	See Universal RDR Fields.

Quota Breach RDR

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

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

This RDR is generated subject to the following conditions:

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

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

The following table lists the RDR fields and their descriptions.

Table 2-12 Quota Breach RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	UINT16	See Universal RDR Fields.
BUCKET_ID	UINT8	1 to 16, according to the number of the breached bucket.
END_TIME	INT32	See Universal RDR Fields.

Table 2-12 Quota Breach RDR Fields (continued)

RDR Field Name	Type	Description
BUCKET_QUOTA	INT32	The remaining quota in the indicated bucket: • Volume bucket—Kilobytes • Number of sessions bucket—Integer
AGGREGATION_PERIOD_TYPE	UINT8	Defines how often the bucket is refilled. See Aggregation Period (uint8) for possible values and their interpretations.

Remaining Quota RDR

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

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

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

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

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

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

The following table lists the RDR fields and descriptions.

Table 2-13 Remaining Quota RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	UINT16	See Universal RDR Fields.
RDR_REASON	UINT8	• 0—Period time passed • 1—Logout • 2—Package switch • 3—Wraparound • 4—End of aggregation period
END_TIME	INT32	See Universal RDR Fields.

Table 2-13 Remaining Quota RDR Fields (continued)

RDR Field Name	Type	Description
REMAINING_QUOTA_1 through REMAINING_QUOTA_16	INT32	The remaining quota in the bucket that was breached, in kilobytes. There are sixteen Remaining Quota fields, one for each bucket.
TOTAL_VOLUME_USAGE	UINT32	Total Volume Usage for all services that are not quota provisioned, in kilobytes, for the current reporting period.

Quota Threshold Breach RDR

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-14 Quota Threshold Breach RDR Fields

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

Quota State Restore RDRs

The QUOTA_STATE_RESTORE_RDR is generated each time a subscriber is introduced.

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

The following table lists the RDR fields and their descriptions.

Table 2-15 Quota State Restore RDR Fields

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

DHCP RDR

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

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

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

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

The following table lists the RDR fields and descriptions.

Table 2-16 DHCP RDR Fields

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

RADIUS RDR

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

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

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

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

The following table lists the RDR fields and descriptions.

Table 2-17 RADIUS RDR Fields

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

Flow Start RDR

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-18 Flow Start RDR Fields

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

Flow End RDR

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-19 Flow End RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	UINT16	See Universal RDR Fields.
SERVICE_ID	INT32	See Universal RDR Fields.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	See Universal RDR Fields.
SERVER_PORT	UINT16	See Universal RDR Fields.

Table 2-19 Flow End RDR Fields (continued)

RDR Field Name	Type	Description
CLIENT_IP	UINT32	See Universal RDR Fields.
CLIENT_PORT	UINT16	See Universal RDR Fields.
INITIATING_SIDE	INT8	See Universal RDR Fields.
START_TIME	UINT32	Flow start time.
REPORT_TIME	INT32	See Universal RDR Fields.
BREACH_STATE	INT8	See Universal RDR Fields.
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.

Ongoing Flow RDR

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-20 Ongoing Flow RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
PACKAGE_ID	UINT16	See Universal RDR Fields.
SERVICE_ID	INT32	See Universal RDR Fields.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	See Universal RDR Fields.
SERVER_PORT	UINT16	See Universal RDR Fields.
CLIENT_IP	UINT32	See Universal RDR Fields.
CLIENT_PORT	UINT16	See Universal RDR Fields.
INITIATING_SIDE	INT8	See Universal RDR Fields.
START_TIME	UINT32	Flow start time.
REPORT_TIME	INT32	See Universal RDR Fields.
BREACH_STATE	INT8	See Universal RDR Fields.

Table 2-20 Ongoing Flow RDR Fields (continued)

RDR Field Name	Type	Description
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.

Media Flow RDR

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

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

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

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

The following table lists the RDR fields and their descriptions.

Table 2-21 Media Flow RDR Fields

Field name	Type	Description
SUBSCRIBER_ID	String	See Universal RDR Fields.
PACKAGE_ID	INT16	See Universal RDR Fields.
SERVICE_ID	INT32	See Universal RDR Fields.
PROTOCOL_ID	INT16	See Universal RDR Fields.
DESTINATION_IP	UINT32	SIP: Destination IP address of RTP flow. Skype: Destination IP address of Skype flow.
DESTINATION_PORT	UINT16	SIP: Destination port of RTP flow. Skype: Destination port of Skype flow.
SOURCE_IP	UINT32	SIP: Source IP address of RTP flow. Skype: Source IP address of Skype flow.
SOURCE_PORT	UINT16	SIP: Source port of RTP flow. Skype: Source port of Skype flow.

Table 2-21 Media Flow RDR Fields (continued)

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

Table 2-21 Media Flow RDR Fields (continued)

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

Note Packet Loss

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

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. See RFC 1889 for more information about the RCP/RTCP standard.

Attack Start RDR

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

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

The following table lists the RDR fields and their descriptions.

Table 2-22 Attack Start RDR Fields

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

Attack End RDR

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

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

The following table lists the RDR fields and their descriptions.

Table 2-23 Attack End RDR Fields

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

Malicious Traffic Periodic RDR

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

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

The following table lists the RDR fields and their descriptions.

Table 2-24 Malicious Traffic Periodic RDR Fields

RDR Field Name	Type	Description
ATTACK_ID	INT32	Unique attack ID.
SUBSCRIBER_ID	STRING	See Universal RDR Fields.
ATTACK_IP	UINT32	The IP address related to this attack.
OTHER_IP	UINT32	The other IP address related to this attack, if such exists (if this is a DOS attack), or -1 otherwise.
PORT_NUMBER	UINT16	The port number related to this attack, if such exists (if this is an IP scan, for example), or -1 otherwise.
ATTACK_TYPE	INT32	Who ATTACK_IP belongs to: • 0—Attacked • 1—Attacker
SIDE	INT8	The IP address side: • 0—Subscriber • 1—Network
IP_PROTOCOL	UINT8	IP protocol type: • 0—Other • 1—ICMP • 6—TCP • 17—UDP
CONFIGURED_DURATION	INT32	See Universal RDR Fields.
DURATION	INT32	Indicates the number of seconds that have passed since the previous MALICIOUS_TRAFFIC_RDR.
END_TIME	INT32	See Universal RDR Fields.
ATTACKS	INT8	The number of attacks in the current reporting period. Since this report is generated per attack, the value is 0 or 1.
MALICIOUS_SESSIONS	UINT32	Aggregated number of sessions for the reported attack, for the current reporting period. If the SCE platform blocks the attack, this field takes the value -1.

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

- Scan—OTHER_IP=-1 and ATTACK_TYPE=1 (the RDR contains the source (attacker) IP address)
- DDOS attack—OTHER_IP=-1 and ATTACK_TYPE=0 (the RDR contains the destination (attacked) IP address)
- DOS attack—OTHER_IP contains an IP address (the RDR contains two IP addresses)

Information About RDR Enumeration Fields

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

- Block Reason (uint8)
- String Fields
- Aggregation Period (uint8)
- Time Frames (uint16)
- RDR Tag Assignment Summary
- Periodic RDR Zero Adjustment Mechanism

Block Reason (uint8)

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

Table 2-25 Block Reason Field Bit Values

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

String Fields

The following table lists the ACCESS_STRING and INFO_STRING field values.

Table 2-26 String Field Values

Name	TR ACCESS_STRING	TR INFO_STRING	Description
PROTOCOL_TCP_GENERIC	Null	Null	
PROTOCOL_UDP_GENERIC	Null	Null	
PROTOCOL_HTTP_BROWSING	Host name	URL	
PROTOCOL_HTTP_STREAMING	Host name	URL	
PROTOCOL_FTP	Null	Null	
PROTOCOL_RTSP	Host name	Null	
PROTOCOL_MMS	Null	Null	
PROTOCOL_PROXY_HTTP	Host name	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_FASTTRACK_TRANSFER	Network name	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	
PROTOCOL_BITTORRENT	Null	Null	Peer to peer
PROTOCOL_SKYPE	Null	Null	Peer to peer
PROTOCOL_VONAGE	SIP Method	SIP Subscriber ID	

Table 2-26 String Field Values (continued)

Name	TR ACCESS_STRING	TR INFO_STRING	Description
PROTOCOL_SHARE	Null	Null	Peer to peer
PROTOCOL_H323	Null	Is FastStart	
PROTOCOL_SOULSEEK	Null	Null	Peer to peer
PROTOCOL_ITUNES	Null	Null	Peer to peer
PROTOCOL_FILETOPIA	Null	Null	Peer to peer
PROTOCOL_NAPSTER	Null	Null	Peer to peer
PROTOCOL_DHCP	Null	Null	
PROTOCOL_MUTE	Null	Null	Peer to peer
PROTOCOL_NODEZILLA	Null	Null	Peer to peer
PROTOCOL_WASTE	Null	Null	Peer to peer
PROTOCOL_NEONET	Null	Null	Peer to peer
PROTOCOL_MGCP	Null	Null	
PROTOCOL_WAREZ	Null	Null	Peer to peer

Aggregation Period (uint8)

The following table lists the AGG_PERIOD field values.

Table 2-27 AGG_PERIOD Field Values

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

Time Frames (uint16)

The following table lists the TIME_FRAME field values.

Table 2-28 Time Frame Field Values

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

RDR Tag Assignment Summary

The following is a summary of RDR tag assignments.

You can configure the RDR categories using the SCE CLI. See the *Cisco Service Control Engine (SCE) CLI Command Reference* for more information.

Table 2-29 RDR Tag Assignments

RDR Name	Default Category (explained below)	Tag Value (decimal)	Tag Value (hexa)
SUBSCRIBER USAGE RDR (NUR)	CM-DB (1)	4,042,321,920	F0 F0 F0 00
REALTIME SUBSCRIBER USAGE RDR (SUR)	CM-DB (1)	4,042,321,922	F0 F0 F0 02
PACKAGE USAGE RDR	CM-DB (1)	4,042,321,924	F0 F0 F0 04
LINK USAGE RDR	CM-DB (1)	4,042,321,925	F0 F0 F0 05
VIRTUAL LINK RDR	CM-DB (1)	4,042,321,926	F0 F0 F0 06
TRANSACTION RDR	CM-DB (1)	4,042,321,936	F0 F0 F0 10
TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,000	F0 F0 F4 38
HTTP TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,004	F0 F0 F4 3C
RTSP TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,008	F0 F0 F4 40
VOIP TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,050	F0 F0 F4 6A
BLOCKING RDR	CM-CSV (1)	4,042,321,984	F0 F0 F0 40
QUOTA BREACH RDR	QP (4)	4,042,321,954	F0 F0 F0 22
REMAINING QUOTA RDR	QP (4)	4,042,321,968	F0 F0 F0 30
QUOTA THRESHOLD RDR	QP (4)	4,042,321,969	F0 F0 F0 31

Table 2-29 RDR Tag Assignments

RDR Name	Default Category (explained below)	Tag Value (decimal)	Tag Value (hexa)
QUOTA STATE RESTORE RDR	QP (4)	4,042,321,970	F0 F0 F0 32
RADIUS RDR	SM (3)	4,042,321,987	F0 F0 F0 43
DHCP RDR	SM (3)	4,042,321,986	F0 F0 F0 42
FLOW START RDR	RT (2)	4,042,321,942	F0 F0 F0 16
FLOW END RDR	RT (2)	4,042,321,944	F0 F0 F0 18
MEDIA FLOW RDR	CM-DB (1)	4,042,323,052	F0 F0 F4 6C
FLOW ONGOING RDR	RT (2)	4,042,321,943	F0 F0 F0 17
ATTACK_START RDR	RT (2)	4,042,321,945	F0 F0 F0 19
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

Table 2-30 RDR Tag Default Categories

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

Periodic RDR Zero Adjustment Mechanism

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

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

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

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

Example 1:

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

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

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

Example 2:

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

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

CHAPTER 3

NetFlow Records: Formats and Field Contents

This chapter describes the fields that may be contained in a NetFlow record.

NetFlow records can be generated for the data contained in the following RDRs:

- Subscriber Usage RDR (SUR)
- Package Usage RDR (PUR)
- Link Usage RDR (LUR)
- Virtual Link Usage RDR (VUR)
- Malicious Usage RDR (MALUR)
- NetFlow
- NetFlow Field Types

NetFlow

- The Cisco Service Control Application for Broadband (SCA BB) supports NetFlow v5 and v9.
- For more information about NetFlow, refer to
  - *Cisco IOS NetFlow Version 9 Flow-Record Format, EDCS-307741*
  - RFC 3954

NetFlow Field Types

Table 3-1 NetFlow Fields

Field Type	Value	Length (Bytes)	Description
sceSubscriberId	300	64	The subscriber identification string, introduced through the subscriber management interfaces. For an unknown subscriber this field may contain an empty string. The string is padded with zeros.
scePackageId	301	2	The ID of the service configuration package/profile assigned to the subscriber.
sceServiceId	302	4	The service classification of the reported session.
sceProtocolId	303	2	The unique ID of the protocol associated with the reported session. The PROTOCOL_ID will be the Generic IP / Generic TCP / Generic UDP protocol ID value, according to the specific transport protocol of the transaction, unless a more specific protocol definition (such as a signature-based or a port-based protocol) that matches the reported session is assigned to a service.
sceSkipppedSessions	304	4	The number of unreported sessions since the previous reporting record of this kind.

Table 3-1 NetFlow Fields (continued)

Field Type	Value	Length (Bytes)	Description
sceInitiatingSide	305	1	The initiating side of the transaction: • 0—Subscriber side • 1—Network side
sceReportTime	306	4	Ending time stamp of this reporting record. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
sceTransactionDuration Millisec	307	4	Duration, in milliseconds, of the transaction reported in this reporting record.
sceTimeFrame	308	1	Which of the four possible time frames was used for the period during which the reporting record was generated. The field takes a value in the range 0 to 3.
sceSessionUpstreamVolume	309	4	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.
sceSessionDownstream Volume	310	4	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.
sceIpProtocolType	311	1	The IP protocol type.
sceProtocolSignature	312	4	The ID of the protocol signature associated with this session
sceZoneId	313	4	The ID of the zone associated with this session

Table 3-1 NetFlow Fields (continued)

Field Type	Value	Length (Bytes)	Description
sceFlavorId	314	4	For protocol signatures that have flavors, this field contains the ID of the flavor associated with this session.
sceFlowCloseMode	315	1	The reason for the end of the flow.
	316-319		Reserved.
sceAccessString	320	128, 256, 512, 1024	A Layer 7 property, extracted from the transaction.
sceInfoString	324	128, 256, 512, 1024	A Layer 7 property, extracted from the transaction.
	328-350		Reserved.
sceServiceUsageSubscriberCounterId	351	2	Each service is mapped to a counter. There are 32 counters in the subscriber scope.
sceBreachState	352	1	Indicates whether the subscriber's quota was breached: • 0—The quota was not breached • 1—The quota was breached
sceReason	353	1	The reason that the reporting record was generated: • 0—Periodic record • 1—Subscriber logout • 2—Package switch • 3—Wraparound • 4—End of aggregation period
sceConfiguredDuration	354	4	Configured period, in seconds, between successive reporting records.

Table 3-1 NetFlow Fields (continued)

Field Type	Value	Length (Bytes)	Description
sceDuration	355	4	The number of seconds that have passed since the previous reporting record of this type.
sceEndTime	356	4	Ending time stamp of this reporting record. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970
sceUpstreamVolume	357	4	Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period.
sceDownstreamVolume	358	4	Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period.
sceSessions	359	2	Aggregated number of sessions for the reported service, for the current reporting period.
sceSeconds	360	2	Aggregated number of session seconds for the reported service, for the current reporting period.
scePackageCounterId	361	2	Each package is mapped to a counter. There are 64 package usage counters.
sceGeneratorId	362	1	A numeric value identifying the processor generating the reporting record.
sceServiceGlobalCounterId	363	2	Each service is mapped to a counter. There 64 global usage counters

Table 3-1 NetFlow Fields (continued)

Field Type	Value	Length (Bytes)	Description
sceConcurrentSessions	364	4	Concurrent number of sessions using the reported service when this reporting record was generated.
sceActiveSubscribers	365	4	Concurrent number of subscribers using the reported service when this reporting record was generated.
sceTotalActiveSubscribers	366	4	Concurrent number of subscribers in the system when this reporting record was generated.
sceLinkId	367	1	A numeric value associated with the reported network link: • 0—Physical link 1 • 1—Physical link 2
sceVirtualLinkId	368	1	A numeric value associated with the reported virtual network link.
	369-399		Reserved
sceAttackId	400	4	Unique attack ID.
sceAttackIp	401	4	The IP address related to this attack.
sceAttackOtherIp	402	4	The other IP address related to this attack if it exists, -1 otherwise.
sceAttackPortNumber	403	2	The port number related to this attack if one exists (if this is an IP scan, for example), -1 otherwise.
sceAttackType	404	4	Who sceAttackIp belongs to: • 0—Attacked • 1—Attacker
sceAttackSide	405	1	The IP address side: • 0—Subscriber • 1—Network

Table 3-1 NetFlow Fields (continued)

Field Type	Value	Length (Bytes)	Description
sceAttackIpProtocol	406	1	The IP protocol type: • 0—Other • 1—ICMP • 6—TCP • 17—UDP
sceAttacks	407	1	The number of attacks in the current reporting period. Since attack reports are generated per attack, the value is 0 or 1.
sceAttackMaliciousSessions	408	4	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.
	409-499		Reserved

CHAPTER 4

Database Tables: Formats and Field Contents

Each Raw Data Record (RDR) is sent to the Cisco Service Control Management Suite (SCMS) Collection Manager (CM). On the CM, adapters convert the RDRs and store them in database tables. There is a separate table for each RDR type. This chapter presents these tables and their columns (field names and types).

For additional information, such as RDR structure, RDR column and field descriptions, and how the RDRs are generated, see Raw Data Records Overview .

- Database Tables Overview
- Table RPT_NUR
- Table RPT_SUR
- Table RPT_PUR
- Table RPT_LUR
- Table RPT_TR
- Table RPT_MEDIA
- Table RPT_MALUR
- Table RPT_TOPS_PERIOD0
- Table RPT_TOPS_PERIOD1
- Table INI_VALUES
- Table VLINK_INI
- Table CONF_SE_TZ_OFFSET

Database Tables Overview

Each RDR is routed to the appropriate adapter—the JDBC Adapter or the Topper/Aggregator (TA) Adapter—converted, and written into a database table row. There is a separate table for each RDR type, with a column designated for each RDR field.

In addition to the RDR fields that are specific to each RDR type, the tables RPT_NUR, RPT_SUR, RPT_PUR, RPT_LUR, and RPT_TR contain two universal columns: TIME_STAMP and RECORD_SOURCE. The following values are placed in these two universal columns (field numbers 1 and 2, respectively):

- TIME_STAMP—The RDR time stamp assigned by the SCMS-CM. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.

- RECORD_SOURCE—Contains the IP address of the Service Control Engine (SCE) platform that generated the RDR.

  The IP address is in 32-bit binary format (displayed as a 4-byte integer).

Table RPT_NUR

Database table RPT_NUR stores data from SUBSCRIBER_USAGE_RDRs.

Note This table is not part of the default configuration.

These RDRs have the tag **4042321920** .

Table 4-1 Columns for Table RPT_NUR

Field Name	Type
TIME_STAMP	Date_Time
RECORD_SOURCE	Number
SUBSCRIBER_ID	String
PACKAGE_ID	Number
SUBS_USG_CNT_ID	Number
BREACH_STATE	Number
REASON	Number
CONFIGURED_DURATION	Number
DURATION	Number
END_TIME	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
SESSIONS	Number
SECONDS	Number

Table RPT_SUR

Database table RPT_SUR stores data from REALTIME_SUBSCRIBER_USAGE_RDRs.

These RDRs have the tag **4042321922** .

Table 4-2 Columns for Table RPT_SUR

Field Name	Type
TIME_STAMP	Date_Time
RECORD_SOURCE	Number
SUBSCRIBER_ID	String
PACKAGE_ID	Number

Table 4-2 Columns for Table RPT_SUR (continued)

Field Name	Type
SUBS_USG_CNT_ID	Number
MONITORED_OBJECT_ID	Number
BREACH_STATE	Number
REASON	Number
CONFIGURED_DURATION	Number
DURATION	Number
END_TIME	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
SESSIONS	Number
SECONDS	Number

Table RPT_PUR

Database table RPT_PUR stores data from PACKAGE_USAGE_RDRs.

These RDRs have the tag **4042321924** .

Table 4-3 Columns for Table RPT_PUR

Field Name	Type
TIME_STAMP	Date_Time
RECORD_SOURCE	Number
PKG_USG_CNT_ID	Number
GENERATOR_ID	Number
GLBL_USG_CNT_ID	Number
CONFIGURED_DURATION	Number
DURATION	Number
END_TIME	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
SESSIONS	Number
SECONDS	Number
CONCURRENT_SESSIONS	Number
ACTIVE_SUBSCRIBERS	Number
TOTAL_ACTIVE_SUBSCRIBERS	Number

Table RPT_LUR

Database table RPT_LUR stores data from LINK_USAGE_RDRs.

These RDRs have the tag **4042321925** .

Table 4-4 Columns for Table RPT_LUR

Field Name	Type
TIME_STAMP	Date_Time
RECORD_SOURCE	Number
LINK_ID	Number
GENERATOR_ID	Number
GLBL_USG_CNT_ID	Number
CONFIGURED_DURATION	Number
DURATION	Number
END_TIME	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
SESSIONS	Number
SECONDS	Number
CONCURRENT_SESSIONS	Number
ACTIVE_SUBSCRIBERS	Number
TOTAL_ACTIVE_SUBSCRIBERS	Number

Table RPT_TR

Database table RPT_TR stores data from TRANSACTION_RDRs.

These RDRs have the tag **4042321936** .

Table 4-5 Columns for Table RPT_NUR

Field Name	Type
TIME_STAMP	Date_Time
RECORD_SOURCE	Number
SUBSCRIBER_ID	String
PACKAGE_ID	Number
SERVICE_ID	Number
PROTOCOL_ID	Number
SAMPLE_SIZE	Number
PEER_IP	Number
PEER_PORT	Number
ACCESS_String	String

Table 4-5 Columns for Table RPT_NUR (continued)

Field Name	Type
INFO_String	String
SOURCE_IP	Number
SOURCE_PORT	Number
INITIATING_SIDE	Number
END_TIME	Number
MILISEC_DURATION	Number
TIME_FRAME	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
SUBS_CNT_ID	Number
GLBL_CNT_ID	Number
PKG_USG_CNT_ID	Number
IP_PROTOCOL	Number
PROTOCOL_SIGNATURE	Number
ZONE_ID	Number
FLAVOR_ID	Number
FLOW_CLOSE_MODE	Number

Table RPT_MEDIA

Database table RPT_MEDIA stores data from MEDIA_FLOW_RDRs.

These RDRs have the tag **4042323052** .

Table 4-6 Columns for Table RPT_MEDIA

Field Name	Type
TIME_STAMP	DateTime
RECORD_SOURCE	Number
SUBSCRIBER_ID	String
PACKAGE_ID	Number
SERVICE_ID	Number
PROTOCOL_ID	Number
PEER_IP	Number
PEER_PORT	Number
SOURCE_IP	Number
SOURCE_PORT	Number
INITIATING_SIDE	Number

Table 4-6 Columns for Table RPT_MEDIA (continued)

Field Name	Type
ZONE_ID	Number
FLAVOR_ID	Number
SIP_DOMAIN	String
SIP_USER_AGENT	String
START_TIME	Number
END_TIME	Number
SEC_DURATION	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
IP_PROTOCOL	Number
FLOW_TYPE	Number
SESSION_ID	Number
UPSTREAM_AVERAGE_JITTER	Number
DOWNSTREAM_AVERAGE_JITTER	Number
UPSTREAM_PACKET_LOSS	Number
DOWNSTREAM_PACKET_LOSS	Number
UPSTREAM_PAYLOAD_TYPE	Number
DOWNSTREAM_PAYLOAD_TYPE	Number

Table RPT_MALUR

Database table RPT_MALUR stores data from MALICIOUS_TRAFFIC_PERIODIC_RDRs.

These RDRs have the tag **4042322000** .

Table 4-7 Columns for Table RPT_MALUR

Field Name	Type
TIME_STAMP	DateTime
RECORD_SOURCE	Number
ATTACK_ID	Number
SUBSCRIBER_ID	String
ATTACK_IP	Number
OTHER_IP	Number
PORT_NUMBER	Number
ATTACK_TYPE	Number
SIDE	Number
IP_PROTOCOL	Number

Table 4-7 Columns for Table RPT_MALUR (continued)

Field Name	Type
CONFIGURED_DURATION	Number
DURATION	Number
END_TIME	Number
ATTACKS	Number
MALICIOUS_SESSIONS	Number

Table RPT_TOPS_PERIOD0

The Topper/Aggregator (TA) Adapter generates database table RPT_TOPS_PERIOD0 for its shorter aggregation interval (by default, one hour).

Table 4-8 Columns for Table RPT_TOPS_PERIOD0

Field Name	Type
RECORD_SOURCE	Number
METRIC_ID	Number
SUBS_USG_CNT_ID	Number
TIME_STAMP	DateTime
AGG_PERIOD	Number
SUBSCRIBER_ID	String
CONSUMPTION	Number

For each Top Report, the TA Adapter sorts the subscriber/consumption pairs from the highest consumption to lowest. At the end of each report is a statistic giving the sum of all subscribers for this metric.

If the report is empty, typically when no traffic was reported for the designated service/metric pair during the aggregation period, the DB will still be updated, but the only row in the report will be the final row showing a total consumption of zero. The DB is updated to avoid the perception in the Cisco Service Control Application (SCA) Reporter that the report is not there because of a malfunction.

The possible values for the field METRIC_ID are presented in the following table.

Table 4-9 Metric_ID Values

Metric_ID	Metric
0	Up Volume
1	Down Volume
2	Combined Volume
3	Sessions
4	Seconds

Table RPT_TOPS_PERIOD1

The Topper/Aggregator (TA) Adapter generates database table RPT_TOPS_PERIOD1 for its longer aggregation interval (by default, 24 hour).

Table 4-10 Columns for Table RPT_TOPS_PERIOD1

Field Name	Type
RECORD_SOURCE	Number
METRIC_ID	Number
SUBS_USG_CNT_ID	Number
TIME_STAMP	DateTime
AGG_PERIOD	Number
SUBSCRIBER_ID	String
CONSUMPTION	Number

For each Top Report, the TA Adapter sorts the subscriber/consumption pairs from the highest consumption to lowest. At the end of each report is a statistic giving the sum of all subscribers for this metric.

If the report is empty, typically when no traffic was reported for the designated service/metric pair during the aggregation period, the DB will still be updated, but the only row in the report will be the final row showing a total consumption of zero. The DB is updated to avoid the perception in the SCA Reporter that the report is not there because of a malfunction.

The possible values for the field METRIC_ID are presented in the following table.

Table 4-11 Metric_ID Values

Metric_ID	Metric
0	Up Volume
1	Down Volume
2	Combined Volume
3	Sessions
4	Seconds

Table INI_VALUES

Database table INI_VALUES is updated whenever the service configuration is applied to the SCE platform. This table contains, for each SCE IP address, mappings between numeric identifiers and textual representation for services, packages, and other service configuration components. The mapping is represented as a standard properties file in string form, where each mapping file is stored in one row. The SCA Reporter uses the mappings contained in this table.

Table 4-12 Columns for Table INI_VALUES

Field Name	Type	Description
TIME_STAMP	DateTime	
SE_IP	String	Identification of the SCE platform where these values were applied.

Table 4-12 Columns for Table INI_VALUES (continued)

Field Name	Type	Description
VALUE_TYPE	Number	Key/Value family type. The possible values are: 1—Service ID / service name 2—Package ID / package name 3—TCP port number / port name 4—Time frame ID / time frame name 5—SCE address 32-bit / dotted notation 6—IP protocol number / IP protocol name 7—Signature protocol ID / protocol name 8—P2P signature protocol ID / protocol name 11—Global service usage counter ID / counter name 12—Subscriber service usage counter ID / counter name 13—Package usage counter ID / counter name 15—UDP port number / port name 1002—VoIP signature protocol ID / protocol name 2001—P2P subscriber service usage counter ID / counter 2002—VoIP subscriber service usage counter ID / counter 3001—P2P global service usage counter ID / counter 3002—VoIP global service usage counter ID / counter
VALUE_KEY	String	Key name. For example: Gold, Silver, or Adult Browsing.
VALUE	Number	Numeric

Table VLINK_INI

Database table VLINK_INI is updated when the CM utility `update_vlinks.sh` is run. This table contains the name and id of each virtual link defined in the SCE platform. The SCA Reporter uses the mappings contained in this table for the Virtual Links reports.

Table 4-13 Columns for Table VLINK_INI

Field Name	Type	Description
TIME_STAMP	DateTime	
SCE_IP	String	Identification of the SCE platform where these values were applied
VLINK_ID	INT16	Virtual link ID
VLINK_DIRECTION	INT8	Virtual link direction
VLINK_NAME	String	Virtual link name

Table CONF_SE_TZ_OFFSET

Database table CONF_SE_TZ_OFFSET contains the time-zone offset in minutes for each SCE platform's clock as configured by the `select-sce-tz.sh` script.

Table 4-14 Columns for Table CONF_SE_TZ_OFFSET

Field Name	Type
TIME_STAMP	DateTime
OFFSET_MIN	Number

CHAPTER 5

CSV File Formats

The Cisco Service Control Application for Broadband (SCA BB) provides several types of Comma-Separated Value (CSV) flat files that you can review and configure using third-party applications such as Excel.

- Information About Service Configuration Entities CSV File Formats
- Information About Subscriber CSV File Formats
- Information About Collection Manager CSV File Formats

Information About Service Configuration Entities CSV File Formats

This section describes the file formats of the CSV files created when exporting service configuration entities into CSV files. The same format must be used for importing such entities into service configurations.

For more information about exporting and importing service configuration entities, see “Managing Service Configurations” in the “Using the Service Configuration Editor” chapter of the *Cisco Service Control Application for Broadband User Guide* .

Note There is no need to repeat the same values in subsequent rows of the CSV file. If a field is left empty in a row, the value of that field from the previous row is used.

Service CSV Files

Lines in Service CSV files have the following fixed format:

```
service name,service numeric ID,[description],sample rate,parent name,global counter
index,subscriber counter index,[flavor],initiating side,protocol,[zone]
```

- The only service that does not have a parent service is the default service.
- The default service is the parent of all other services.
- If the service will be counted with its parent, it must have a counter index of -1.
- One service can have multiple entries in the file (see the following example). There is no need to state the service properties for each of its items.
- Some fields can take a null value (see the last line of the following example).

The following is an example of a service CSV file:

```
P2P,9,,10,Default Service,9,9,,EitherSide,DirectConnect,zone1
P2P,9,,10,Default Service,9,9,flavor1,EitherSide,Manolito, zone1
,,,,,,,,EitherSide,Hotline, zone1
,,,,,,, flavor2,EitherSide,Share, zone1
Generic,1,,10,Default Service,-1,-1,No items,null,null,null
```

Protocol CSV Files

Lines in Protocol CSV files have the following fixed format:

```
protocol name,protocol index,[IP protocol],[port range],signature
```

One protocol can have multiple entries in the file (see the following example). Port range has the format: **MinPort-MaxPort**. For example, **1024-5000** means port 1024 to port 5000.

The following is an example of a protocol CSV file:

```
HTTP Browsing,2,TCP,80-80,Generic
HTTP Browsing,2,TCP,8080-8080,Generic
HTTP Browsing,2,,,HTTP
```

Zone CSV Files

Lines in Zone CSV files have the following fixed format:

```
zone name,zone index,IP range
```

where IP range is an IP address in dotted notation, followed by a mask.

The following is an example of a zone CSV file:

```
zone1,1,10.1.1.0/24
,,10.1.2.0/24
```

Information About Flavor CSV Files

The format of flavor CSV files depends on the flavor type.

Each line of every flavor CSV files begins with the same three fields:

```
flavor name,flavor index,flavor type[,flavor specific field[s]]
```

The formats of the CSV files of different flavors are described in the following sections.

The following is an example of a line from a flavor CSV file:

```
HttpUrlFlavor,1,HTTP_URL
```

HTTP URL

Lines in HTTP URL CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,host suffix,params prefix,
URI suffix,URI prefix
```

The following is an example of an HTTP URL CSV file:

```
NEWS,0,HTTP_URL,*.reuters.com,,,/news/*
,,,*.msnbc.msn.com,,,
,,,*.wired.com,,,/news/technology/*
,,,*.cbsnews.com,,,/sections/world/*
,,,*.cnn.com,,,/WORLD/*
```

HTTP User Agent

Lines in HTTP User Agent CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,user agent
```

RTSP User Agent

Lines in RTSP User Agent CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,user agent
```

RTSP Host Name

Lines in RTSP Host Name CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,host suffix
```

RTSP Composite

Lines in HTTP Composite CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,RTSP_Host_Name,RTSP_User_Agent_name
```

where **RTSP_Host_Name** and **RTSP_User_Agent_name** are the names of existing flavors of types RTSP Host Name and RTSP User Agent respectively

SIP Destination Domain

Lines in SIP Destination Domain CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,host suffix
```

SIP Source Domain

Lines in SIP Source Domain CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,host suffix
```

SIP Composite

Lines in HTTP Composite CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,SIP_Destination_Domain_name,
SIP_Source_Domain_name
```

where **SIP_Destination_Domain_name** and **SIP_Source_Domain_name** are the names of existing flavors of types SIP Destination Domain and SIP Source Domain respectively

SMTP Host Name

Lines in SMTP Host Name CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,host suffix
```

Information About Subscriber CSV File Formats

This section describes the file formats of various subscriber CSV files used by the Cisco Service Control Management Suite (SCMS) Subscriber Manager(SM). For more information about these CSV file formats, see "Subscriber Files" in the "Managing Subscribers" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide* and see the *Cisco Service Control Management Suite Subscriber Manager User Guide* .

- Import/Export File: Format of the mappings Field
- SCE Subscriber CSV Files
- SCMS SM Subscriber CSV Files
- SCE Anonymous Group CSV Files
- SCE Subscriber Template CSV File

Import/Export File: Format of the mappings Field

Some of the CSV files include a mappings field. This field can include one or more of the following values delimited by colons (":") or semicolons (";"):

- A single IP address in dotted notation (xx.xx.xx.xx).
- An IP address range in dotted notation (xx.xx.xx.xx/mask).
- A single VLAN (xx) as an integer in decimal notation in the range of 0 to 2044.
- A VLAN range (xx-yy) where both values are integers in decimal notation in the range of 0 to 2044.

Note Note Specifying VLAN and IP Mappings together in the same line is not allowed.

- Multiple IP mappings—10.1.1.0/24;10.1.2.238
- Multiple VLAN mappings—450:896-907

SCE Subscriber CSV Files

Lines in SCE Subscriber CSV files have the following fixed format:

```
subscriber-id,mappings,package-id,upstream Virtual Link id,downstream Virtual Link id
```

The following is an example CSV file for use with the SCE CLI:

```
JerryS,80.179.152.159;80.179.152.179,0,1,3
ElainB,194.90.12.2,3,55,87
```

SCMS SM Subscriber CSV Files

Lines in SCMS SM Subscriber CSV files have the following fixed format:

```
subscriber-id,domain,mappings,package-id,upstream Virtual Link id,downstream Virtual Link id
```

If no domain is specified, the default domain (subscribers) is assigned.

The following is an example CSV file for use with the SM CLI:

```
JerryS,subscribers,80.179.152.159,0,0,0
ElainB,,194.90.12.2,3,12,1
```

SCE Anonymous Group CSV Files

Lines in SCE Anonymous Group CSV files have the following fixed format:

```
anonymous-group-name,IP-range[,subscriber-template-number]
```

If no subscriber-template-number is specified, then the anonymous subscribers of that group will use the default template (equivalent to using a subscriber-template-number value of zero).

The mapping between subscriber-template-number and package-id is defined in the SCE Subscriber Template CSV file, which is described in the following section.

The following is an example of an anonymous group CSV file

```
group1,10.1.0.0/16;10.5.0.0/16,2
group2,176.23.34.0/24,3
group3,10.7.0.0/16
```

SCE Subscriber Template CSV File

Lines in Subscriber Template CSV files have the following fixed format, as described below:

```
subscriber-template-number,package-id
```

SCA BB includes a default one-to-one mapping between package-id and subscriber-template-number for values from 0 to 63.

Subscriber-template-numbers can take values between 0 and 199. You can map more than one subscriber-template-number to the same package-id.

For more information about this file, see the *Cisco Service Control Engine (SCE) Software Configuration Guide* .

Information About Collection Manager CSV File Formats

This section describes the file formats of the CSV files created by adapters of the Cisco Service Control Management Suite (SCMS) Collection Manager (CM). For more information about the CM and its adapters, see the *Cisco Service Control Management Suite Collection Manager User Guide* .

Each RDR is routed to the appropriate adapter—the Comma-Separated Value (CSV) Adapter, the Topper/Aggregator (TA Adapter), or the Real-Time Aggregating (RAG) Adapter—converted, and written to a CSV file.

- CSV Adapter CSV Files
- TA Adapter CSV Files
- RAG Adapter CSV Files

CSV Adapter CSV Files

By default, the CSV Adapter writes files to subdirectories of **~/cm/adapters/CSVAdapter/csvfiles** , where each subdirectory name is the RDR tag of the RDR that generated the CSV file.

Each CSV file created by the CSV Adapter has a structure matching the RDR represented in the file.

TA Adapter CSV Files

The TA Adapter receives Subscriber Usage RDRs, aggregates the data they contain, and outputs statistics to CSV files. By default, these files are created once every 24 hours, at midnight.

The name of the CSV file is the date and time of its creation. The default format of the file name is **yyyy-MM-dd_HH-mm-ss.csv** (for example, **2005-09-27_18-30-01.csv**). By default, the location of the CSV files is **~/cm/adapters/TAAdapter/csvfiles**.

By default, the fields in each row of the CSV file are as follows:

```
TIMESTAMP,TAG,subsID,svcALLup,svcALLdown,svcALLsessions,svcALLseconds,
svc0up,svc0down,svc0sessions,svc0seconds,svc1up,svc1down,svc1sessions,
svc1seconds,...,svcNup,svcNdown,svcNsessions,svcNseconds
```

where **subsID** is the Subscriber ID and **svcXY** is the aggregated volume of metric Y for service X. (The N in **svcN** is the highest service number, which is the configured number of services minus 1.)

The combined volume is not stored in the CSV file, since it is easily obtained by adding the upstream and downstream volumes.

You can configure the adapter to insert a comment at the beginning of every CSV file. This comment contains a time stamp showing when the file was created, and an explanation of its format. By default, this feature is disabled. To turn this option on, edit the file **csvadapter.conf** and change the value of **includeRecordSource**.

RAG Adapter CSV Files

The RAG Adapter processes RDRs of one or more types and aggregates the data from predesignated field positions into buckets . When a RAG Adapter bucket is flushed, its content is written as a single line into a CSV file, one file per RDR, in the adapters' CSV repository.

The name of the CSV file is the date and time of its creation. The default format of the file name is **yyyy-MM-dd_HH-mm-ss.csv** (for example, **2005-09-27_18-30-01.csv**). By default, the CSV repository is flat (all CSV files in one directory), and located at **~/cm/adapters/RAGAdapter/csvfiles**. Alternatively, you can configure the adapter to use a subdirectory structure; the CSV files are written to subdirectories of **~/cm/adapters/RAGAdapter/csvfiles** , where each subdirectory name is the RDR tag of the RDR type that was written to this CSV file.

Each line written to the CSV file may have some synthesized fields added to it, such as time stamps of the first and last RDRs that contributed to this bucket and the total number of RDRs in this bucket. Other fields may be removed altogether. Fields in the output line that are not used for aggregation will have values corresponding to the values in the first RDR that contributed to the bucket. However, the time stamp field that is prepended to the line in the CSV file will have a value corresponding to the time stamp of the last RDR in the bucket.

CHAPTER 6

SCA BB Proprietary MIB Reference

This chapter describes the proprietary CISCO-SCAS-BB Management Information Base (MIB) supported by the Service Control Engine (SCE) platform.

A MIB is a database of objects that can be monitored by a network management system (NMS). The SCE platform supports both the standard MIB-II and the proprietary Cisco Service Control Enterprise MIB. The CISCO-SCAS-BB MIB is the part of the Service Control Enterprise MIB that enables the external management system to monitor counters and metrics specific to the Cisco Service Control Application for Broadband (SCA BB).

Information About SNMP Configuration and Management

This section explains how to configure the SNMP interface, and how to load the MIB files.

Configuring the SNMP Interface on the SCE platform

Before using the SNMP interface:

- Enable SNMP access on the SCE platform (by default, SNMP access is disabled).
- Set the values of SNMP parameters:
  - The community string to be used for client authentication.
  - (Optional, recommended as a security measure) An access-list (ACL) of IP addresses. This limits access to SNMP information to a set of known locations. You can define a different community string for each ACL.
  - The destination IP address to which the SCE platform will send SNMP traps.

Note You can enable or disable specific traps.

Related Info

For more information about SNMP configuration, see “SNMP Configuration and Management” in the “Configuring the Management Interface and Security” chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.

Required MIB Files

Service Control Enterprise MIB

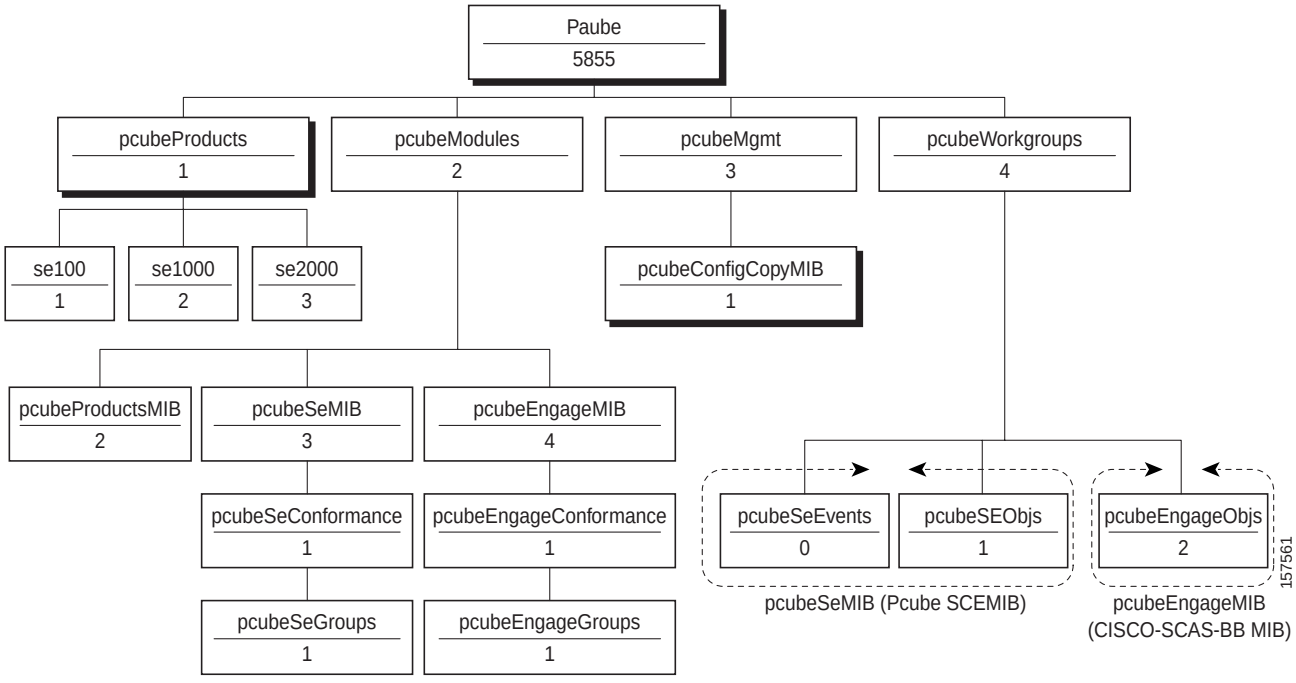
The Service Control Enterprise MIB includes four main groups: Products, Modules, Management, and Workgroup. The Service Control enterprise tree structure is defined in a MIB file named **pcube.mib**.

- The pcubeProducts subtree contains the sysObjectIDs of the Service Control products.

  Service Control product sysObjectIDs are defined in a MIB file named **Pcube-Products-MIB**.
- The pcubeModules subtree provides a root object identifier from which MIB modules are defined.
- The pcubeMgmt subtree contains the configuration copy MIB:
  - **pcubeConfigCopyMib** enables saving the running configuration of Cisco products. This MIB is documented in the "Proprietary MIB Reference" appendix of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.
- The pcubeWorkgroup subtree contains:
  - **pcubeSeEvents** and **pcubeSEObjs**—**pcubeSeMib**, the SCE MIB, is the main MIB for the Service Control products and provides a wide variety of configuration and runtime statistics. This MIB is also documented in the "Proprietary MIB Reference" appendix of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.
  - **pcubeEngageObjs**—The CISCO-SCAS-BB MIB provides configuration and runtime status for SCA BB, and is described in the following section.

The following figure illustrates the Service Control Enterprise MIB structure.

Figure 6-1



Note The following object identifier represents the Service Control Enterprise MIB: **1.3.6.1.4.1.5655** or **iso.org.dod.internet.private.enterprise.pcube**.

Information About The CISCO-SCAS-BB MIB

The CISCO-SCAS-BB MIB provides access to service usage counters through the SNMP interface. Using this MIB, a network administrator can collect usage information per service at link, package, or subscriber granularity.

The CISCO-SCAS-BB MIB is defined in the file *PCubeEngageMib.mib.*

The MIB is documented in the remainder of this module.

Using this Reference

This reference is divided into sections according to the MIB object groups. For each object, information is presented in the following format:

<Description of the object>

Access	access control associated with the object
Units	unit of measurement used for the object

INDEX

{Indexes used by the table}

Syntax

```
OBJECT DATA TYPE { The general format of the object }
```

pcubeEngageObjs (pcubeWorkgroup 2)

The pcubeEngageObjs objects provide current information about packages, service, and subscribers.

pcubeEngageObjs Objects

This is a list of the pcubeEngageObjs objects. Each object consists of a number of subordinate object types, as summarized in the following section.

serviceGrp	{pcubeEngageObjs 1}
linkGrp	{pcubeEngageObjs 2}
packageGrp	{pcubeEngageObjs 3}
subscriberGrp	{pcubeEngageObjs 4}
serviceCounterGrp	{pcubeEngageObjs 5}

pcubeEngageObjs Structure

This is a summary of the structure of pcubeEngageObjs. Note the table structure for objects that may have multiple entries.

serviceGrp
serviceTable—deprecated

linkGrp *linkServiceUsageTablelinkServiceUsageEntry*linkServiceUsageUpVolume
linkServiceUsageDownVolume
linkServiceUsageNumSessions
linkServiceUsageDuration
linkServiceUsageConcurrentSessions
linkServiceUsageActiveSubscribers
linkServiceUpDroppedPackets
linkServiceDownDroppedPackets
linkServiceUpDroppedBytes
linkServiceDownDroppedBytes
packageGrp *packageCounterTablepackageCounterEntry*packageCounterIndex
packageCounterStatus
packageCounterName
packageCounterActiveSubscribers*packageServiceUsageTablepackageServiceUsageEntry*
packageServiceUsageUpVolume
packageServiceUsageDownVolume
packageServiceUsageNumSessions
packageServiceUsageDuration
packageServiceUsageConcurrentSessions
packageServiceUsageActiveSubscribers
packageServiceUpDroppedPackets
packageServiceDownDroppedPackets
packageServiceUpDroppedBytes
packageServiceDownDroppedBytes
subscriberGrp
*subscribersTablesubscriberEntry*subscriberPackageIndex*subscriberServiceUsageTablesubscriberServiceUsageEntry*subscriberServiceUsageUpVolume
subscriberServiceUsageDownVolume
subscriberServiceUsageNumSessions
subscriberServiceUsageDuration
serviceCounterGrp
*globalScopeServiceCounterTableglobalScopeServiceCounterEntry*globalScopeServiceCounterIndex
globalScopeServiceCounterStatus
globalScopeServiceCounterName*subscriberScopeServiceCounterTablesubscriberScopeServiceCounterEntry*subscriberScopeServiceCounterIndex
subscriberScopeServiceCounterStatus
subscriberScopeServiceCounterName

Service Group: serviceGrp (pcubeEngageObjs 1)

The Service group is deprecated. Use the Service Counter group.

serviceTable (serviceGrp 1)

Deprecated—Use the tables in the Service Counter group.

Access	not-accessible

Syntax

Counter32

Link Group: linkGrp (pcubeEngageObjs 2)

The Link Service group provides usage information per link for each global-scope service usage counter (for example, traffic statistics of a service for all subscribers using a particular link).

linkServiceUsageTable (linkGrp 1)

The Link Service Usage table provides usage information per link for each global-scope service usage counter.

Access	not-accessible

Syntax

SEQUENCE OF linkServiceUsageEntry

linkServiceUsageEntry (linkServiceUsageTable 1)

A Link Service Usage table entry containing parameters defining resource usage of one link for services included in one global-scope service usage counter.

Access	not-accessible

Index

`{linkModuleIndex, linkIndex, globalScopeServiceCounterIndex}`

Syntax

SEQUENCE{*linkServiceUsageUpVolumelinkServiceUsageDownVolumelinkServiceUsageNumSessionslinkServiceUsageDurationlinkServiceUsageConcurrentSessionslinkServiceUsageActiveSubscriberslinkServiceUpDroppedPacketslinkServiceDownDroppedPacketslinkServiceUpDroppedByteslinkServiceDownDroppedBytes*}

linkServiceUsageUpVolume (linkServiceUsageEntry 1)

The upstream volume of services in this service usage counter carried over the link.

Access	read-only
Units	kilobytes

Syntax

Counter32

Note Although volume counters on the SCE platform hold 32-bit integers, CISCO-SCAS-BB MIB volume counters wraparound (turn back to zero) when the maximum 29-bit integer value (0x1FFFFFFF) is reached.

linkServiceUsageDownVolume (linkServiceUsageEntry 2)

The downstream volume of services in this service usage counter carried over the link.

Access	read-only
Units	kilobytes

Syntax

`Counter32`

linkServiceUsageNumSessions (linkServiceUsageEntry 3)

The number of sessions of services in this service usage counter carried over the link.

Access	read-only
Units	sessions

Syntax

`Counter32`

linkServiceUsageDuration (linkServiceUsageEntry 4)

The aggregated session duration of services in this service usage counter carried over the link.

Access	read-only
Units	seconds

Syntax

`Counter32`

linkServiceUsageConcurrentSessions (linkServiceUsageEntry 5)

The number of concurrent sessions of services in this service usage counter carried over the link.

Access	read-only
Units	sessions

Syntax

`Counter32`

linkServiceUsageActiveSubscribers (linkServiceUsageEntry 6)

The number of active subscribers of services in this service usage counter carried over the link.

Access	read-only
Unit	subscribers

Syntax

Counter32

linkServiceUpDroppedPackets (linkServiceUsageEntry 7)

The number of dropped upstream packets of services in this service usage counter carried over the link.

Access	read-only
Units	packets

Syntax

Counter32

Note To enable the SCE application to count dropped packets and dropped bytes, disable the `accelerate-packet-drops` feature on the SCE platform; if `accelerate-packet-drops` is enabled, the MIB dropped packets and dropped bytes counters constantly show the value 0xFFFFFFFF. For more information about the `accelerate-packet-drops` feature, see "Counting Dropped Packets" in the "Configuring the Line Interface" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.

linkServiceDownDroppedPackets (linkServiceUsageEntry 8)

The number of dropped downstream packets of services in this service usage counter carried over the link.

Access	read-only
Units	packets

Syntax

Counter32

Note To enable the SCE application to count dropped packets and dropped bytes, disable the `accelerate-packet-drops` feature on the SCE platform; if `accelerate-packet-drops` is enabled, the MIB dropped packets and dropped bytes counters constantly show the value 0xFFFFFFFF. For more information about the `accelerate-packet-drops` feature, see "Counting Dropped Packets" in the "Configuring the Line Interface" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.

linkServiceUpDroppedBytes (linkServiceUsageEntry 9)

The number of dropped upstream bytes of services in this service usage counter carried over the link.

Access	read-only
Units	bytes

Syntax

Counter32

Note To enable the SCE application to count dropped packets and dropped bytes, disable the `accelerate-packet-drops` feature on the SCE platform; if `accelerate-packet-drops`is enabled, the MIB dropped packets and dropped bytes counters constantly show the value 0xFFFFFFFF. For more information about the `accelerate-packet-drops` feature, see "Counting Dropped Packets" in the "Configuring the Line Interface" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.

linkServiceDownDroppedBytes (linkServiceUsageEntry 10)

The link service-counter number of dropped downstream bytes of services in this service usage counter carried over the link.

Access	read-only
Units	bytes

Syntax

Counter32

Note To enable the SCE application to count dropped packets and dropped bytes, disable the `accelerate-packet-drops` feature on the SCE platform; if `accelerate-packet-drops` is enabled, the MIB dropped packets and dropped bytes counters constantly show the value 0xFFFFFFFF. For more information about the `accelerate-packet-drops`feature, see "Counting Dropped Packets" in the "Configuring the Line Interface" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.

Package Group: packageGrp (pcubeEngageObjs 3)

The Package group provides general and usage information for each global-scope package usage counter (for example, traffic statistics of a service for all subscribers assigned to a particular package or group of packages).

The Package group provides general and usage information for each global-scope package usage counter (for example, traffic statistics of a service for all subscribers assigned to a particular package or group of packages).

Syntax

```
SEQUENCE OF packageCounterEntry
```

packageCounterEntry (packageCounterTable 1)

A Package Counter table entry containing parameters defining one package usage counter.

Access	not-accessible

Index

```
{pmoduleIndex, packageCounterIndex}
```

Syntax

SEQUENCE{*packageCounterIndexpackageCounterStatuspackageCounterNamepackageCounterActiveSubscribers*}

packageCounterIndex (packageCounterEntry 1)

The package usage counter index.

Access	not-accessible

Syntax

Integer32 *(1...1023)*

packageCounterStatus (packageCounterEntry 2)

The package usage counter status.

Access	read-only

Syntax
INTEGER{*0 (disabled)1 (enabled)*}

packageCounterName (packageCounterEntry 3)

The name of the package usage counter.

Access	read-only

Syntax

SnmpAdminString

packageCounterActiveSubscribers (packageCounterEntry 4)

The total number of active subscribers of packages included in the package usage counter.

Access	read-only

Syntax

Counter32

packageServiceUsageTable (packageGrp 2)

The Package Service Usage table provides usage information for each global-scope package usage counter.

Access	not-accessible

Syntax

SEQUENCE OF packageServiceUsageEntry

packageServiceUsageEntry (packageServiceUsageTable 1)

A Package Service Usage table entry containing parameters defining resource usage of packages included in one global-scope package usage counter.

Access	not-accessible

Index

{pmoduleIndex, packageCounterIndex, globalScopeServiceCounterIndex}

Syntax

SEQUENCE{*packageServiceUsageUpVolumepackageServiceUsageDownVolumepackageServiceUsageNumSessionspackageServiceUsageDurationpackageServiceUsageConcurrentSessionspackageServiceUsageActiveSubscriberspackageServiceUpDroppedPacketspackageServiceDownDroppedPacketspackageServiceUpDroppedBytespackageServiceDownDroppedBytes*}

packageServiceUsageUpVolume (packageServiceUsageEntry 1)

The upstream volume of packages in this package usage counter.

Access	read-only
Units	kilobytes

Syntax

Counter32

Note Although volume counters on the SCE platform hold 32-bit integers, CISCO-SCAS-BB MIB volume counters wraparound (turn back to zero) when the maximum 29-bit integer value (0x1FFFFFFF) is reached.

packageServiceUsageDownVolume (packageServiceUsageEntry 2)

The downstream volume of packages in this package usage counter.

Access	read-only
Units	kilobytes

Syntax

Counter32

packageServiceUsageNumSessions (packageServiceUsageEntry 3)

The number of sessions of packages in this package usage counter.

Access	read-only
Units	sessions

Syntax

Counter32

packageServiceUsageDuration (packageServiceUsageEntry 4)

The aggregated session duration seconds of packages in this package usage counter.

Access	read-only
Units	seconds

Syntax

Counter32

packageServiceUsageConcurrentSessions (packageServiceUsageEntry 5)

The number of concurrent sessions of packages in this package usage counter.

Access	read-only
Units	sessions

Syntax

Counter32

packageServiceUsageActiveSubscribers (packageServiceUsageEntry 6)

The number of active subscribers of packages in this package usage counter.

Access	read-only
Units	subscribers

Syntax

Counter32

packageServiceUpDroppedPackets (packageServiceUsageEntry 7)

The number of dropped upstream packets of packages in this package usage counter.

Access	read-only
Units	packets

Syntax

Counter32

Note To enable the SCE application to count dropped packets and dropped bytes, disable the `accelerate-packet-drops` feature on the SCE platform; if `accelerate-packet-drops` is enabled, the MIB dropped packets and dropped bytes counters constantly show the value 0xFFFFFFFF. For more information about the `accelerate-packet-drops`feature, see "Counting Dropped Packets" in the "Configuring the Line Interface" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.

packageServiceDownDroppedPackets (packageServiceUsageEntry 8)

The number of dropped downstream packets of packages in this package usage counter.

Access	read-only
Units	packets

Syntax

Counter32

Note To enable the SCE application to count dropped packets and dropped bytes, disable the `accelerate-packet-drops` feature on the SCE platform; if `accelerate-packet-drops` is enabled, the MIB dropped packets and dropped bytes counters constantly show the value 0xFFFFFFFF. For more information about the `accelerate-packet-drops`feature, see "Counting Dropped Packets" in the "Configuring the Line Interface" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.

packageServiceUpDroppedBytes (packageServiceUsageEntry 9)

The number of dropped upstream bytes of packages in this package usage counter.

Access	read-only
Units	bytes

Syntax

Counter32

Note To enable the SCE application to count dropped packets and dropped bytes, disable the `accelerate-packet-drops` feature on the SCE platform; if `accelerate-packet-drops` is enabled, the MIB dropped packets and dropped bytes counters constantly show the value 0xFFFFFFFF. For more information about the `accelerate-packet-drops`feature, see "Counting Dropped Packets" in the "Configuring the Line Interface" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.

packageServiceDownDroppedBytes (packageServiceUsageEntry 10)

The number of dropped downstream bytes of packages in this package usage counter.

Access	read-only
Units	bytes

Syntax

Counter32

Note To enable the SCE application to count dropped packets and dropped bytes, disable the `accelerate-packet-drops` feature on the SCE platform; if `accelerate-packet-drops` is enabled, the MIB dropped packets and dropped bytes counters constantly show the value 0xFFFFFFFF. For more information about the `accelerate-packet-drops`feature, see "Counting Dropped Packets" in the "Configuring the Line Interface" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide*.

Subscriber Group: subscriberGrp (pcubeEngageObjs 4)

The Subscriber group provides general information for each subscriber and usage information per service usage counter for each subscriber (for example, traffic statistics of a service for a particular subscriber defined in the system).

Note To use the tables in this group, first create an entry to reference a particular subscriber in the subscribersPropertiesValueTable object of the subscriberGrp in the SCE MIB (not the CISCO-SCAS-BB MIB). Using the index of this table (spvIndex), information about the subscriber can be collected. See *Accessing Subscriber Information (the spvIndex)*for more information about how to access subscriber-level information using the SNMP interface.

subscribersTable (subscriberGrp 1)

The Subscribers Table provides information for each subscriber.

Access	not-accessible

Syntax

SEQUENCE OF subscribersEntry

subscribersEntry (subscribersTable 1)

A Subscribers Table entry containing the package index of each subscriber.

Access	not-accessible

Index

{pmoduleIndex, spvIndex}

Syntax

SEQUENCE{*subscriberPackageIndex*}

subscriberPackageIndex (subscribersEntry 1)

The package index of the subscriber's package.

Access	read-only

Syntax

Integer32 *(1...255)*

subscriberServiceUsageTable (subscriberGrp 2)

The Subscriber Service Usage table provides usage information per service usage counter for each subscriber.

Access	not-accessible

Syntax

Sequence of subscriberServiceUsageEntry

subscriberServiceUsageEntry (subscriberServiceUsageTable 1)

A Subscriber Service Usage table entry containing parameters defining resource usage by one subscriber of services included in one service usage counter.

Access	not-accessible

Index

{pmoduleIndex, spvIndex, subscriberScopeServiceCounterIndex}

Syntax

SEQUENCE{*subscriberServiceUsageUpVolumesubscriberServiceUsageDownVolumesubscriberServiceUsageNumSessionssubscriberServiceUsageDuration*}

subscriberServiceUsageUpVolume (subscriberServiceUsageEntry 1)

The upstream volume of services in this service usage counter used by this subscriber.

Access	read-only
Unit	kilobytes

Syntax

Counter32

Note Although volume counters on the SCE platform hold 32-bit integers, CISCO-SCAS-BB MIB volume counters wraparound (turn back to zero) when the maximum 29-bit integer value (0x1FFFFFFF) is reached.

subscriberServiceUsageDownVolume (subscriberServiceUsageEntry 2)

The downstream volume of services in this service usage counter used by this subscriber.

Access	read-only
Unit	kilobytes

Syntax

Counter32

Note Although volume counters on the SCE platform hold 32-bit integers, CISCO-SCAS-BB MIB volume counters wraparound (turn back to zero) when the maximum 29-bit integer value (0x1FFFFFFF) is reached.

subscriberServiceUsageNumSessions (subscriberServiceUsageEntry 3)

The number of sessions of services in this service usage counter used by this subscriber.

Access	read-only
Unit	sessions

Syntax

Integer32 *(1...65535)*

subscriberServiceUsageDuration (subscriberServiceUsageEntry 4)

Aggregated session duration of services in this service usage counter used by this subscriber.

Access	read-only
Units	seconds

Syntax

Integer32 *(1...65535)*

Service Counter Group: serviceCounterGrp (pcubeEngageObjs 5)

The Service Counter group provides general information for each global-scope and subscriber-scope service usage counter. You can use it, for example, to read the names of the services as defined in a SCA BB service configuration.

globalScopeServiceCounterTable (serviceCounterGrp 1)

The Global-Scope Service Counter table consists of data about each service usage counter used by the link and by packages.

Access	not-accessible

Syntax

SEQUENCE OF globalScopeServiceCounterEntry

globalScopeServiceCounterEntry (globalScopeServiceCounterTable 1)

A Global-Scope Service Counter table entry containing parameters defining one global-scope service usage counter.

Access	not-accessible

Index

{pmoduleIndex, globalScopeServiceCounterIndex}

Syntax

SEQUENCE{*globalScopeServiceCounterIndexglobalScopeServiceCounterStatusglobalScopeServiceCounterName*}

globalScopeServiceCounterIndex (globalScopeServiceCounterEntry 1)

The global-scope service usage counter index.

Access	not-accessible

Syntax

Integer32 *(1...255)*

globalScopeServiceCounterStatus (globalScopeServiceCounterEntry 2)

The global-scope service usage counter status.

Access	read-only

Syntax

INTEGER{*0 (disabled)1 (enabled)*}

globalScopeServiceCounterName (globalScopeServiceCounterEntry 3)

The name of the global-scope service usage counter.

Access	read-only

Syntax

SnmpAdminString

subscriberScopeServiceCounterTable (serviceCounterGrp 2)

The Subscriber-Scope Service Counter table consists of data about each service usage counter used by subscribers.

Access	not-accessible

Syntax

SEQUENCE OF subscriberScopeServiceCounterEntry

subscriberScopeServiceCounterEntry (subscriberScopeServiceCounterTable 1)

A Subscriber-Scope Service Counter table entry containing parameters defining one subscriber-scope service usage counter.

Access	not-accessible

Index

{pmoduleIndex, subscriberScopeServiceCounterIndex}

Syntax

SEQUENCE{*subscriberScopeServiceCounterIndexsubscriberScopeServiceCounterStatussubscriberScopeServiceCounterName*}

subscriberScopeServiceCounterIndex (subscriberScopeServiceCounterEntry 1)

The subscriber-scope service usage counter index.

Access	not-accessible

Syntax

Integer32 *(1...255)*

subscriberScopeServiceCounterStatus (subscriberScopeServiceCounterEntry 2)

The subscriber-scope service usage counter status.

Access	read-only

Syntax

INTEGER{*0 (disabled)1 (enabled)*}

subscriberScopeServiceCounterName (subscriberScopeServiceCounterEntry 3)

The name of the subscriber-scope service usage counter.

Access	read-only

Syntax

SnmpAdminString

Accessing Subscriber InformationGuidelines for Using the CISCO-SCAS-BB MIB

This section provides guidelines to help access SNMP information about the SCE platform using the CISCO-SCAS-BB MIB.

Note Indices in SNMP start from 1; SCA BB indices start from 0. When accessing a counter in the SCA BB SNMP MIB by its index, add 1 to the index of the entity. For example, the global usage counter with index 0 will be located at globalScopeServiceCounter index 1.

Note Although volume counters on the SCE platform hold 32-bit integers, CISCO-SCAS-BB MIB volume counters wraparound (turn back to zero) when the maximum 29-bit integer value (0x1FFFFFFF) is reached.

Note To enable the SCE application to count dropped packets and dropped bytes, disable the **accelerate-packet-drops** feature on the SCE platform; if **accelerate-packet-drops** is enabled, the MIB dropped packets and dropped bytes counters constantly show the value 0xFFFFFFFF.

Note For more information about the accelerate-packet-drops feature, see "Counting Dropped Packets" in the "Configuring the Line Interface" chapter of the *Cisco Service Control Engine (SCE) Software Configuration Guide* .

globalScopeServiceCounterTable and subscriberScopeServiceCounterTable

The index of a service usage counter as defined in a SCA BB service configuration is used to reference services in the CISCO-SCAS-BB MIB. Since MIB index values count from 1, but SCA BB indices count from 0, the index used in the MIB must always be one greater than the index of the service it is referencing.

For example, to get the number of upstream bytes used by a service on a link, use **LinkServiceTable.lnkServiceUpVolume** (part of the linkGrp). The value assigned to **serviceIndex** for this table must be one greater than service index defined for this service in the service configuration.

To identify or change the index of a service, go to the Advanced tab of the Service Settings dialog box in the SCA BB Console (see the "Using the Service Configuration Editor: Traffic Classification" chapter of the *Cisco Service Control Application for Broadband User Guide*). For example, to reference the P2P service (which has a (default) service index of 9) in the MIB, a **serviceIndex** of 10 (= 9 + 1) must be used.

packageCounterTable

The package index, defined in a SCA BB service configuration, is used to reference entries in **packageTable** and **packageServiceTable** (part of the **packageGrp**). As with serviceIndex the value assigned to **packageIndex** must be one greater than the package index in the service configuration.

To identify or change the index of a package, go to the Advanced tab of the Package Settings dialog box in the SCA BB Console (see the "Using the Service Configuration Editor: Traffic Control" chapter of the *Cisco Service Control Application for Broadband User Guide*). For example, to reference the default package (which has a package index of 0) in the MIB, a **packageIndex** of 1 (= 0 + 1) must be used.

Accessing Subscriber Information (the spvIndex)

In order to collect subscriber-level information using the SNMP interface, you must first create an entry in the subscriberPropertiesValuesTable part of the subscriberGrp in pcubeSEMib (not PCubeEngageMib). After an entry in this table is created and associated with a subscriber name, its index (spvIndex) can be referred to in PCubeEngageMib to collect usage statistics for this subscriber. An entry is created in the subscriberPropertiesValuesTable table by setting the entry spvRowStatus object with CreateAndGo(4) then setting the name of the subscriber in the spvSubName property and the spvIndex variable to be used as an index to the subscriber. For example, to poll the downstream volume of subscriber “sub123” for the P2P service using PCubeEngageMib, perform the following steps.

Step 1 Obtain the index of the P2P service from the SCA BB Console. (This is a one-time operation that you should perform only if services are changed in the policy.) [In this example, assume that the P2P service index has its default value of 9.]

Step 2 Create an entry in `SEMib:subscriberGrp:subscriberPropertiesValuesTable`.

Step 3 Set the object indices:

- For pmoduleIndex use 1.
- Set spvIndex to the desired value. [In this example we will use 1.]

Step 4 Set spvRowStatus to 4 (using CreateAndGo).

Step 5 Set spvSubName to “sub123”.

Step 6 Read the subscriberServiceDownVolume property out of EngageMib:subscriberGrp:subscriberServiceTable where spvIndex is set to 1 and serviceIndex is set to 10.