Cisco Service Control Application for Broadband Reference Guide

Release 3.6.x
November 23, 2012

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

Text Part Number: OL-21066-04

Cisco Service Control Application for Broadband Reference Guide

CONTENTS

About this Guide

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

Introduction

This chapter describes who should read *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 guide provides information about the data structures created and used by Cisco Service Control Application for Broadband (SCA BB). It is intended for:

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

Document Revision History

Table 1 records changes to this document.

Table 1 Document Revision History

Revision	Cisco Service Control Release Number and Date	Change Summary
OL-21066-03	Supports all 3.6.x releases November 23, 2012	Updated Chapter 2, "Raw Data Records: Formats and Field Contents."
OL-21066-03	Supports all 3.6.x releases November 19, 2012	Updated Chapter 2, "Raw Data Records: Formats and Field Contents."
OL-21066-03	Supports all 3.6.x releases August 29, 2012	Chapter 2, "Raw Data Records: Formats and Field Contents".
OL-21066-03	Supports all 3.6.x releases February 04, 2011	Chapter 2, "Raw Data Records: Formats and Field Contents".
OL-21066-03	Supports all 3.6.x releases February 04, 2011	Updated "Information About Protocols".

Table 1 Document Revision History (continued)

Revision	Cisco Service Control Release Number and Date	Change Summary
OL-21066-02	Supports all 3.6.x releases November 8, 2010	Changes made to include release 3.6.5 updates. The following RDR was added: • Zone Usage RDR, page 2-39 The following RAG adapter tables were added: • Table RPT_TOP_APN, page 4-21 • Table RPT_TOP_DEVICE_TYPE, page 4-22 • Table RPT_TOP_NETWORK_TYPE, page 4-23 • Table RPT_TOP_SGSN, page 4-24 • Table RPT_TOP_USER_LOCATION, page 4-25 • Table RPT_DVLINK, page 4-26 • Table RPT_UVLINK, page 4-27 • Table RPT_TOP_HTTP_DOMAINS, page 4-28 • Table RPT_TOP_HTTP_HOSTS, page 4-29 • Table RPT_TOP_VIDEO_DOMAINS, page 4-30 • Table RPT_TOP_VIDEO_HOSTS, page 4-31 • Table RPT_ZUR, page 4-32 • Table RPT_SPAM, page 4-33 • Table RPT_FUR, page 4-34 • Table IMEI_DEVICETYPE, page 4-35 The following TA adapter tables were added: • Table RPT_TOPS_PEAK_PERIOD, page 4-16 • Table RPT_TOPS_PEAK_CUMULATIVE, page 4-17 CSV Files Format Chapter updated with details on Standard Format and Easy Format for Flavors and Zones.

Table 1 Document Revision History (continued)

Revision	Cisco Service Control Release Number and Date	Change Summary
OL-21066-01	3.6.x March 28, 2010	First version of this document (new for the Release 3.6.x train).
		The following changes were made from the last release of the 3.5.x train: The following Quota RDR formats were updated: • Quota Breach RDR, page 2-47 • Quota Status RDR, page 2-50 • Quota Threshold Breach RDR, page 2-54 • Session Creation RDRs, page 2-57 The following new RDR format was added: • Video Transaction Usage RDR, page 2-23 The following new section was added: • ADDITIONAL_INFO Field, page 2-4

Organization

Table 2 lists the document organization of this guide.

Table 2 Document Organization

Section	Title	Description
1	Default Service Configuration Reference Tables	Describes the default service configuration provided with SCA BB
2	Raw Data Records: Formats and Field Contents	Lists the various RDRs produced by the 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
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
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)
5	CSV File Formats	Describes the location and structure of CSV files pertaining to service configuration, subscriber management, and data collection management
6	SCA BB Proprietary MIB Reference	Describes that part of the Cisco SCE proprietary MIB that provides configuration and runtime status for SCA BB

Related Documentation

Use *Cisco Service Control Application for Broadband Reference Guide* in conjunction with the following Cisco documentation:

- *Cisco Service Control Application for Broadband User Guide*
- *Cisco SCA BB Service Configuration API Programmer Guide*
- *Cisco Service Control Management Suite Collection Manager User Guide*
- *Cisco Service Control Management Suite Subscriber Manager User Guide*
- *Cisco Service Control Application Reporter User Guide*
- SCE platform installation and configuration guides:
  - *Cisco SCE 1000 2xGBE Installation and Configuration Guide*
  - *Cisco SCE 2000 Installation and Configuration Guide*
  - *Cisco SCE8000 10GBE Software Configuration Guide*
  - *Cisco SCE8000 GBE Software Configuration Guide*
- *Cisco SCE 2000 and SCE 1000 CLI Command Reference*
- *Cisco SCE 2000 and SCE 1000 Software Configuration Guide*
- *Cisco SCE8000 CLI Command Reference*
- *Cisco SCE8000 10GBE Installation and Configuration Guide*
- *Cisco SCE8000 GBE Installation and Configuration Guide*

Conventions

This document uses the following conventions:

Table 3 Document Conventions

Convention	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation mark.
`courier` font	Terminal sessions and information the system displays appear in `courier` font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

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

Tip Means *the following information will help you solve a problem.* The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

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

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

Warning **Means *reader be warned*. In this situation, you might perform an action that could result in bodily injury.**

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, 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

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

CHAPTER 1

Default Service Configuration Reference Tables

Revised: October 21, 2011, OL-21066-04

Introduction

This chapter describes the default service configuration provided with SCA BB. The default service configuration serves as a starting point for creating a service configuration tailored to customer needs.

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.

Table 1-1 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.
net-bios (to subscriber)	Active	Applies to UDP packets, network-side port is equal to 137, packets bypass the policy engine and are mapped to CoS BE.
eDonkey UDP (to network)	Inactive	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)	Inactive	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)	Inactive	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)	Inactive	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)	Inactive	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)	Inactive	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)	Inactive	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)	Inactive	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.
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—Protocols that are used for transactions not mapped to a service by one of the more specific protocol types.
- Signature-Based Protocols—Protocols that are 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 that are 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, page 1-4
- Signature-Based Protocols, page 1-5
- IP Protocols, page 1-10
- Port-Based Protocols, page 1-14
- Protocols Identified on Unidirectional Flows, page 1-35

Generic Protocols

Three generic protocols (IP, TCP, and UDP) serve as default containers for classifying transactions of the relevant type (IP, TCP, or UDP) that are 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 list the generic protocols.

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.

Note Table 1-3 only lists signature-based protocols that are not P2P, VoIP, or SIP protocols (these protocols are listed in the following tables). However, the Signature-Based Protocols Filter in the Console lists all signature-based protocols.

Table 1-3 Signature-Based Protocols

Protocol Name	ID	TCP Ports	UDP Ports
Audio over HTTP	1041	—	—
Baidu Movie	1043	—	—
Behavioral Upload/Download See note following table (page 6)	127	—	—
Binary over HTTP	1042	—	—
Call Of Duty	1127	—	—
CUWorld	117	—	—
Club Box	1038	—	—
DHCP	33	—	—
DHT	106	—	—
DNS	47	—	—
DingoTel	42	—	—
FIX	1113	—	—
FTP	4	21	—
Flash	2033	—	—
Flash MySpace	2035	—	—
Flash Yahoo	2036	—	—
Flash YouTube	2034	—	—
Fring	1052	—	—
Generic Non-Established TCP See note in the following table (page 7)	126	—	—
Google Talk	1030	—	—
GoogleEarth	118	—	—
HTTP Browsing	2	80,8080	—
HTTP Tunnel	55	—	—
Hopster	115	—	—
ICQ	119	—	—

Table 1-3 Signature-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
IRC	62	—	—
Jabber	116	—	—
MMS	6	1755	—
MS Exchange Desktop	1111	—	—
MS Push Mail	1048	—	—
Mobile MMS	46	—	—
MyJabber	1056	—	—
NNTP	15	119	—
NTP	54	—	—
Napster	32	—	—
POP3	9	110	—
QQ	52	—	—
RTSP Streaming	5	554, 1554, 7070	—
RayV	1112	—	—
SMTP	8	25	—
SSDP	53	—	—
STUN	114	—	—
Second Life	1060	—	—
SkeedReceiver	1109	—	—
Sling	112	—	—
UC	48	—	—
Utagoe UGLive2	1108	—	—
Video over HTTP	1040	—	—
Windows Update	1107	—	—
WebEx	1110		
Yahoo Messenger	40	5000–5001	5000–5010
iTunes	30	—	—
imap	59	143	143
radius	738	—	—
tftp	60	69	69

Note 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.

Note 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
Angle Media	1063	—	—
AntsP2P	113	—	—
BBBroadcast	1058	—	—
BBC iPlayer	1057	—	—
BaiBao	43	—	—
Behavioral P2P	2044	—	—
BitTorrent	24	6881–6889	—
Dijjer	120	—	—
DirectConnect	19	411–413	—
EmuleEncrypted	105	—	—
Entropy	125	—	—
Exosee	121	—	—
FastTrack KaZaA File Transfer	14	—	—
FastTrack KaZaA Networking	13	1214	—
Feidian	1037	—	—
Filetopia	31	—	—
Freenet	107	—	—
Furthur	123	—	—
Gnutella File Transfer	12	—	—
Gnutella Networking	11	6346–6349	—
Hotline	20	—	—
Joost	1046	—	—
Kontiki	124	—	—
KuGoo	1050	—	—
Manolito	22	—	—
Mute	34	—	—
NeoNet	37	—	—
NodeZilla	35	—	—
POCO	51	—	—
PPLive	44	—	—
PPStream	49	—	—
PacketiX	1059	—	—

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

Protocol Name	ID	TCP Ports	UDP Ports
Pando	1049	—	—
PeerEnabler	122	—	—
QQ-Live	2032	—	—
Rodi	111	—	—
Share	27	—	—
SopCast	1064	—	—
Soulseek	29	—	—
TVAnts	109	—	—
Thunder	50	—	—
Warez/FileCroc	39	—	—
Waste	36	—	—
WebThunder	1055	—	—
WinMX/OpenNap	16	6257, 6699	6257
Winny	17	7742–7745, 7773	—
Zattoo	1047	—	—
eDonkey	18	4661–4665, 4672–4673, 4711, 5662, 5773, 5783	4661–4665, 4672–4673, 4711, 5662, 5773, 5783
guruguru	66	—	—
kuro	67	—	—
soribada	69	—	—
v-share	71	—	—

Table 1-5 Signature-Based VoIP Protocols

Protocol Name	ID	TCP Ports	UDP Ports
Behavioral VoIP	1062	—	—
Fring VoIP	1053	—	—
H323	28	1720	—
ICQ VoIP	110	—	—
MGCP	38		2427, 2727
MSN Messenger VoIP	1054	—	—
PTT Winphoria	61	—	—
Primus	108	—	—
RTP	57	—	—
SIP	23	5060-5061	5060-5061
Skinny	41	—	—
Skype	25	—	—
Vivox	1061	—	—
Yahoo Messenger VoIP	45	—	—
Yahoo VoIP over SIP	2039	—	—

Note The protocols ICQ VoIP, Primus, SIP, and Yahoo VoIP over SIP are also signature-based SIP protocols.

IP Protocols

Table 1-6 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

Table 1-7 lists the TCP/UDP port-based protocols defined in the SCA BB default service configuration.

Table 1-7 Port-Based Protocols

Protocol Name	ID	TCP Ports	UDP Ports
FTP	4	21	—
Gnutella Networking	11	6346-6349	—
FastTrack KaZaA Networking	13	1214	
Bittorrent	24	6881-6889	—
NTP	54	123	123
epmap	128	135	135
profile	129	136	136
netbios-ns	130	137	137
netbios-dgm	131	138	138
netbios-ssn	132	139	139
emfis-data	133	140	140
emfis-cntl	134	141	141
bl-idm	135	142	142
uma	137	144	144
uaac	138	145	145
iso-tp0	139	146	146
iso-ip	140	147	147
jargon	141	148	148
aed-512	142	149	149
sql-net	143	150	150
hems	144	151	151
bftp	145	152	152
sgmp	146	153	153
netsc-prod	147	154	154
netsc-dev	148	155	155
sqlsrv	149	156	156
knet-cmp	150	157	157
nss-routing	152	159	159
sgmp-traps	153	160	160
snmp	154	161	161
snmptrap	155	162	162
cmip-man	156	163	163

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
cmip-agent	157	164	164
xns-courier	158	165	165
s-net	159	166	166
namp	160	167	167
rsvd	161	168	168
send	162	169	169
print-srv	163	170	170
multiplex	164	171	171
cl/1	165	172	172
xyplex-mux	166	173	173
mailq	167	174	174
vmnet	168	175	175
genrad-mux	169	176	176
xdmcp	170	177	177
nextstep	171	178	178
bgp	172	179	179
ris	173	180	180
unify	174	181	181
audit	175	182	182
ocserver	177	184	184
remote-kis	178	185	185
kis	179	186	186
aci	180	187	187
mumps	181	188	188
qft	182	189	189
gacp	183	190	190
prospero	184	191	191
osu-nms	185	192	192
srmp	186	193	193
IRC	187	194, 6665-6669	194, 6665-6669
dn6-nlm-aud	188	195	195
dn6-smm-red	189	196	196
dls	190	197	197
dls-mon	191	198	198
smux	192	199	199
src	193	200	200

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
at-rtmp	194	201	201
at-nbp	195	202	202
at-3	196	203	203
at-echo	197	204	204
at-5	198	205	205
at-zis	199	206	206
at-7	200	207	207
at-8	201	208	208
qmtp	202	209	209
z39.50	203	210	210
914c/g	204	211	211
anet	205	212	212
ipx	206	213	213
vmpwscs	207	214	214
softpc	208	215	215
CAIlic	209	216	216
dbase	210	217	217
mpp	211	218	218
uarps	212	219	219
imap3	213	220	220
fln-spx	214	221	221
rsh-spx	215	222	222
cdc	216	223	223
masqdialer	217	224	224
direct	218	242	242
sur-meas	219	243	243
inbusiness	220	244	244
link	221	245	245
dsp3270	222	246	246
bhfhs	224	248	248
set	225	257	257
yak-chat	226	258	258
esro-gen	227	259	259
openport	228	260	260
nsiiops	229	261	261
arcisdms	230	262	262

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
hdap	231	263	263
bgmp	232	264	264
x-bone-ctl	233	265	265
sst	234	266	266
td-service	235	267	267
td-replica	236	268	268
http-mgmt	237	280	280
personal-link	238	281	281
cableport-ax	239	282	282
rescap	240	283	283
corerjd	241	284	284
fxp-1	242	286	286
k-block	243	287	287
novastorbakcup	244	308	308
entrusttime	245	309	309
bhmds	246	310	310
asip-webadmin	247	311	311
vslmp	248	312	312
magenta-logic	249	313	313
opalis-robot	250	314	314
dpsi	251	315	315
decauth	252	316	316
zannet	253	317	317
pkix-timestamp	254	318	318
ptp-event	255	319	319
ptp-general	256	320	320
pip	257	321	321
rtsps	258	322	322
texar	259	333	333
pdap	260	344	344
pawserv	261	345	345
zserv	262	346	346
fatserv	263	347	347
csi-sgwp	264	348	348
mftp	265	349	349
matip-type-a	266	350	350

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
matip-type-b	267	351	351
dtag-ste-sb	268	352	352
ndsauth	269	353	353
bh611	270	354	354
datex-asn	271	355	355
cloanto-net-1	272	356	356
bhevent	273	357	357
shrinkwrap	274	358	358
nsrmp	275	359	359
scoi2odialog	276	360	360
semantix	277	361	361
srssend	278	362	362
rsvp_tunnel	279	363	363
aurora-cmgr	280	364	364
dtk	281	365	365
odmr	282	366	366
mortgageware	283	367	367
qbikgdp	284	368	368
rpc2portmap	285	369	369
codaauth2	286	370	370
clearcase	287	371	371
ulistproc	288	372	372
legent-1	289	373	373
legent-2	290	374	374
hassle	291	375	375
nip	292	376	376
tnETOS	293	377	377
dsETOS	294	378	378
is99c	295	379	379
is99s	296	380	380
hp-collector	297	381	381
hp-managed-node	298	382	382
hp-alarm-mgr	299	383	383
arns	300	384	384
ibm-app	301	385	385
asa	302	386	386

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
aurp	303	387	387
unidata-ldm	304	388	388
ldap	305	—	389
uis	306	390	390
synotics-relay	307	391	391
synotics-broker	308	392	392
meta5	309	393	393
embl-ndt	310	394	394
netware-ip	311	396	396
mptn	312	397	397
kryptolan	313	398	398
iso-tsap-c2	314	399	399
work-sol	315	400	400
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
ariel3	337	422	422
opc-job-start	338	423	423

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
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
ampr-rcmd	373	459	459
skronk	374	460	460

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
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
intecourier	409	495	495
pim-rp-disc	410	496	496

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
dantz	411	497	497
siam	412	498	498
iso-ill	413	499	499
isakmp	414	500, 4500	500, 4500
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
stx	445	527	527
custix	446	528	528

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
irc-serv	447	529	529
courier	448	530	530
conference	449	531	531
netnews	450	432	432
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
streettalk	481	566	566
banyan-rpc	482	567	567

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
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
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
sift-uft	517	608	608
npmp-trap	518	609	609
npmp-local	519	610	610

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
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
dwr	553	644	644
pssc	554	645	645
ldp	555	646	646

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
dhcp-failover	556	647	647
rrp	557	648	648
aminet	558	649	659
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
entrust-aams	589	681	681
xfr	590	682	682
corba-iiop	591	683	683

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
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
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
netrcs619	619	742	742
flexlm	620	744	744
fujitsu-dev	621	747	747
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

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
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
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

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
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
rmiactivation	682	1098	1098
rmiregistry	683	1099	1099
ms-sql-s	684	1433	1433
oracle	690	1521	1521
orasrv	691	1525	1525
tlisrv	692	1527	1527
coauthor	693	1529	1529
rdb-dbs-disp	694	1571	1571
oraclenames	695	1575	1575
oraclenet8cman	696	1630	1630
net8-cman	697	1830	1830
ms-olap	686	2382–2383, 2393–2394	2382–2383, 2393–2394
msft-gc	687	3268	3268
msft-gc-ssl	688	3269	3269
citrixima	698	2512	2512
citrixadmin	699	2513	2513
citrix-rtmp	700	2897	2897
citriximaclient	701	2598	2598

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
micromuse-lm	702	1534	1534
orbixd	703	1570	1570
orbix-locator	704	3075	3075
orbix-config	705	3076	3076
orbix-loc-ssl	706	3077	3077
shockwave	707	1626	1626
sitaraserver	708	2629	2629
sitaramgmt	709	2630	2630
sitaradir	710	2631	2631
mysql	711	3306	3306
msnp	713	1836	1826
aim	714	5190–5193	—
groove	715	2492	2492
directplay	716	2234	2234
directplay8	717	6073	6073
kali	718	2213	2213
worldfusion	719	2595–2596	2595–2596
directv-web	720	3334	3334
directv-soft	721	3335	3335
directv-tick	722	3336	3336
directv-catlg	723	3337	3337
wta-wsp-s	724	2805	2805
wap-push	725	2948	2948
wap-pushsecure	726	2949	2949
wap-push-http	727	4035	4035
wap-push-https	728	4036	4036
game-spy	755	6500, 28900	6515, 27900
ibprotocol	737	6714	6714
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
wap-vcal	734	9205	9205
wap-vcard-s	735	9206	9206
wap-vcal-s	736	9207	9207

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
pptp	739	1723	1723
gtp-user	740	2152	2152
xdtp	741	3088	3088
l2tp	742	1701	1701
fsgs	743	6112	6112
parsec-game	744	6582	6582
UnReal_UT	745	—	7777-7783
SiN	746	22450	22450
halflife	747	—	27015
tribes	748	28001	28001
Heretic II	749	28910	—
starsiege	750	—	29001–29009
game-search	751	29001	—
KingPin	752	31510	31510
runescape	753	43594	—
GLT Poliane	882	1201	—
MSN Messenger	883	1863	1863
xbox live	898	3074	3074
ps2	899	10070–10080	10070
compressnet	900	2–3	2–3
rje	901	5	5
echo	902	7	7
discard	903	9	9
systat	904	11	11
daytime	905	13	13
qotd	906	17	17
msp	907	18	18
chargen	908	19	19
ftp-data	909	20	20
ssh	910	22	22
telnet	911	23	23
nsw-fe	912	27	27
msg-icp	913	29	29
msg-auth	916	31	31
dsp	917	33	33
time	918	37	37

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
rap	919	38	38
rlp	920	39	39
graphics	921	41	41
name	922	42	42
nicname	923	43	43
mpm-flags	924	44	44
mpm	925	45	45
mpm-snd	926	46	46
ni-ftp	927	47	47
auditd	928	48	48
tacacs	929	49	49
re-mail-ck	930	50	50
la-maint	931	51	51
xns-time	932	52	52
xns-ch	934	54	54
isi-gl	935	55	55
xns-auth	936	56	56
xns-mail	937	58	58
ni-mail	938	61	61
acas	939	62	62
whois	940	63	63
covia	941	64	64
tacacs-ds	942	65	65
sql*net	943	66	66
bootps	944	67	67
bootpc	945	68	68
gopher	947	70	70
netrjs-1	948	71	71
netrjs-2	949	72	72
netrjs-3	950	73	73
netrjs-4	951	74	74
deos	952	76	76
finger	953	79	79
hosts2-ns	954	81	81
xfer	955	82	82
mit-ml-dev	956	83, 85	83, 85

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
ctf	957	84	84
mfcobol	958	86	86
kerberos	959	88	88
su-mit-tg	960	89	89
dnsix	961	90	90
mit-dov	962	91	91
npp	963	92	92
dcp	964	93	93
objcall	965	94	94
supdup	966	95	95
dixie	967	96	96
swift-rvf	968	97	97
tacnews	969	98	98
metagram	970	99	99
newacct	971	100	
hostname	972	101	101
iso-tsap	973	102	102
gppitnp	974	103	103
acr-nema	975	104	104
csnet-ns	976	105	105
3com-tsmux	977	106	106
rtelnet	978	107	107
snagas	979	108	108
pop2	980	109	109
sunrpc	981	111	111
mcidas	982	112	112
auth	983	113	113
audionews	984	114	114
sftp	985	115	115
ansanotify	986	116	116
uucp-path	987	117	117
sqlserv	988	118	118
cfdptkt	989	120	120
erpc	990	121	121
smakynet	991	122	122
ansatrader	993	124	124

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
locus-map	994	125	125
nxedit	995	126	126
locus-con	996	127	127
gss-xlicen	997	128	128
pwdgen	998	129	129
cisco-fna	999	130	130
LapLink	1105	1547	
cisco-tna	2000	131	131
cisco-sys	2001	132	132
statsrv	2002	133	133
ingres-net	2003	134	134
Anarchy	2004	7013, 7500–7501	7013, 7500–7501
Asherons Call	2005	9000–9013	9000–9013
Black And White	2006	2611–2612	—
Counter strike	2007	27020–27039	1200, 27000–27018
Dark Reign	2008	26214	26214
Diablo	2009	6113–6119, 4000	6113–6119
Elite Force	2010	—	26000, 27500
F16	2011	—	3862, 3863
F22 Simulator (lightning 3)	2012	—	3874–3875, 4533, 4534
Hexen	2013	—	26900
Kohan Immortal Sovereigns	2014	3855, 17437	3855, 17437
Motorhead	2015	16000, 16010–16030	16000, 16010–16030
Myth	2016	3453	3453
Need For Speed	2017	9442	9442
Need For Speed 3	2018	1030	1030
Operation Flash Point	2019	47624	—
Outlaws	2020	5310	5310
Swat3	2021	16639	16638
Ultima	2022	5002-5010, 7775-7777, 8888, 9999, 7875	—
Warcraft	2023	3724	3724
Znes	2024	—	7845
Delta Force	2025	3100, 3999	3100, 3999, 3568, 3569
Rainbox six	2026	2346	2346

Table 1-7 Port-Based Protocols (continued)

Protocol Name	ID	TCP Ports	UDP Ports
Soldier of fortune	2027	—	28911–28915
Westwood Online	2028	1140, 1234	1140, 1234
Yahoo Games	2029	11999	—
Konspire2b	2031	6085	6085

Protocols Identified on Unidirectional Flows

When unidirectional classification is enabled, the protocols listed in Table 1-8 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 1-8 Unidirectionally-Detected Protocols

Protocol Name	Protocol ID
AntsP2P	113
Audio over HTTP	1041
BBC iPlayer	1057
BaiBao	43
Baidu Movie	1043
Behavioral Upload/Download	127
Binary over HTTP	1042
BitTorrent	24
CUWorld	117
Club Box	1038
Dijjer	120
DingoTel	42
DirectConnect	19
EmuleEncrypted	105
Entropy	125
Exosee	121
FastTrack KaZaA File Transfer	14
Feidian	1037
Filetopia	31
Flash	2033
Flash MySpace	2035

Table 1-8 Unidirectionally-Detected Protocols (continued)

Protocol Name	Protocol ID
Flash Yahoo	2036
Flash YouTube	2034
Fring	1052
Furthur	123
Generic TCP	0
Gnutella File Transfer	12
Gnutella Networking	11
Google Talk	1030
GoogleEarth	118
HTTP Browsing	2
HTTP Tunnel	55
Hopster	115
Hotline	20
ICQ	119
Jabber	116
Joost	1046
Kontiki	124
Location Free	1045
MMS	6
MS Push Mail	1048
MSN Messenger	883
Manolito	22
Mobile MMS	46
Mute	34
Napster	32
NeoNet	37
NodeZilla	35
POCO	51
POP3	9
PPLive	44
PPStream	49
Pando	1049
PeerEnabler	122
QQ-Live	2032
SMTP	8
Skype	25

Table 1-8 Unidirectionally-Detected Protocols (continued)

Protocol Name	Protocol ID
Sling	112
TVAnts	109
Thunder	50
Tor	1065
UC	48
Video over HTTP	1040
Warez/FileCroc	39
WebThunder	1055
WinMX/OpenNap	16
Winny	17
Yahoo Messenger	40
Yahoo Messenger VoIP	45
Zattoo	1047
eDonkey	18
guruguru	66
iTunes	30
imap	59
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.

Table 1-9 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 1-9 Installed Services

Name	ID	Name of Parent Service	Global Usage Counter and Subscriber Usage Counter
Default Service	0		Default Service*
Browsing	7	Default Service	Global: Default Service*, Subscriber: Browsing*
HTTP	16	Browsing	Global: HTTP, Subscriber: Browsing*
HTTPS	17	Browsing	Global: HTTPS, Subscriber: Browsing*
Location Based Services	48	Browsing	Global: Location Based Services, Subscriber: Browsing*
E-Mail	4	Default Service	E-Mail*
IMAP	23	E-Mail	Global: IMAP, Subscriber: E-Mail*
MS Push Mail	47	E-Mail	Global: MS Push Mail, Subscriber: E-Mail*
POP3	21	E-Mail	Global: POP3, Subscriber: E-Mail*
SMTP	22	E-Mail	Global: SMTP, Subscriber: E-Mail*
Web-Based E-Mail	71	E-Mail	Global: Web-Based E-Mail, Subscriber: E-Mail*
File Sharing	49	Default Service	Default Service*
Download over HTTP	44	File Sharing	Download over HTTP
FTP	32	File Sharing	FTP
IM File Transfer	51	File Sharing	Global: Default Service*, Subscriber: IM File Transfer*
Google Talk File Transfer	54	IM File Transfer	Global: Google Talk File Transfer, Subscriber: IM File Transfer*
ICQ File Transfer	55	IM File Transfer	Global: ICQ File Transfer, Subscriber: IM File Transfer*
QQ File Transfer	52	IM File Transfer	Global: QQ File Transfer, Subscriber: IM File Transfer*
Skype File Transfer	98	IM File Transfer	Global: Skype File Transfer, Subscriber: IM File Transfer*

Table 1-9 Installed Services (continued)

Name	ID	Name of Parent Service	Global Usage Counter and Subscriber Usage Counter
Windows Live Messenger File Transfer	57	IM File Transfer	Global: Windows Live Messenger File Transfer, Subscriber: IM File Transfer*
Yahoo Messenger File Transfer	53	Global: Yahoo Messenger File	Global: Yahoo Messenger File Transfer, Subscriber: IM File Transfer*
Other IM File Transfer	56	IM File Transfer	Global: Other IM File Transfer, Subscriber: IM File Transfer*
One-Click Hosting	50	File Sharing	One-Click Hosting
P2P	9	File Sharing	Default Service*
Ares/Warez	58	P2P	Arez/Warez
Bittorrent	24	P2P	Global: Default Service*, Subscriber: Bittorrent*
Encrypted Bittorrent	62	Bittorrent	Global: Encrypted Bittorrent, Subscriber: Bittorrent*
Non-Encrypted Bittorrent	63	Bittorrent	Global: Non-Encrypted Bittorrent, Subscriber: Bittorrent*
Gnutella	30	P2P	Gnutella
Winny	27	P2P	Winny
eDonkey/eMule	14	P2P	Global: Default Service*, Subscriber: eDonkey/eMule*
Encrypted eMule	60	eDonkey/eMule	Global: Encrypted eMule, Subscriber: eDonkey/eMule*
Non-Encrypted eMule	61	eDonkey/eMule	Global: Non-Encrypted eMule, Subscriber: eDonkey/eMule*
Behavioral P2P	43	P2P	Behavioral P2P
Other P2P	59	P2P	Other P2P
Behavioral Upload/Download	39	File Sharing	Behavioral Upload/Download
Gaming	29	Default Service	Global: Default Service*, Subscriber: Gaming*
Nintendo Wii	90	Gaming	Global: Nintendo Wii, Subscriber: Gaming*
PC Gaming	87	Gaming	Global: PC Gaming, Subscriber: Gaming*
Playstation	89	Gaming	Global: Playstation, Subscriber: Gaming*
Xbox	88	Gaming	Global: Xbox, Subscriber: Gaming*
Instant Messaging	28	Default Service	Global: Default Service*, Subscriber: Instant Messaging*

Table 1-9 Installed Services (continued)

Name	ID	Name of Parent Service	Global Usage Counter and Subscriber Usage Counter
Google Talk	83	Instant Messaging	Global: Google Talk, Subscriber: Instant Messaging*
ICQ	85	Instant Messaging	Global: ICQ, Subscriber: Instant Messaging*
Windows Live Messenger	82	Instant Messaging	Global: Windows Live Messenger, Subscriber: Instant Messaging*
Yahoo Messenger	84	Instant Messaging	Global: Yahoo Messenger, Subscriber: Instant Messaging*
Other Instant Messaging	86	Instant Messaging	Global: Other Instant Messaging, Subscriber: Instant Messaging*
Internet Privacy	94	Default Service	Global: Default Service*, Subscriber: Internet Privacy*
Anonimity Networks	95	Internet Privacy	Global: Anonimity Networks, Subscriber: Internet Privacy*
Tunneling	38	Internet Privacy	Global: Tunneling, Subscriber: Internet Privacy*
VPN	41	Internet Privacy	Global: Default Service*, Subscriber: Internet Privacy*
IPSec VPN	42	VPN	Global: IPSec VPN, Subscriber: Internet Privacy*
Internet Video	70	Default Service	Default Service*
Audio and Video over HTTP	76	Internet Video	Audio and Video over HTTP
Commercial Media Distribution	26	Internet Video	Commercial Media Distribution
Flash	45	Internet Video	Global: Default Service*, Subscriber: Flash*
Flash MySpace	73	Flash	Global: Flash MySpace, Subscriber: Flash*
Flash Yahoo	75	Flash	Global: Flash Yahoo, Subscriber: Flash*
Flash YouTube	74	Flash	Global: Flash YouTube, Subscriber: Flash*
Other Flash	72	Flash	Global: Other Flash, Subscriber: Flash*
P2P TV	77	Internet Video	Global: Default Service*, Subscriber: P2P TV*
Joost	81	P2P TV	Global: Joost, Subscriber: P2P TV*
PPLive	79	P2P TV	Global: PPLive, Subscriber: P2P TV*
PPStream	80	P2P TV	Global: PPStream, Subscriber: P2P TV*
Other P2P TV	78	P2P TV	Global: Other P2P TV, Subscriber: P2P TV*

Table 1-9 Installed Services (continued)

Name	ID	Name of Parent Service	Global Usage Counter and Subscriber Usage Counter
Streaming	34	Internet Video	Global: Default Service*, Subscriber: Streaming*
MMS	20	Streaming	Global: MMS, Subscriber: Streaming*
RTMP	99	Streaming	Global: RTMP, Subscriber: Streaming*
RTSP	19	Streaming	Global: RTSP, Subscriber: Streaming*
Net Admin	33	Default Service	Global: Default Service*, Subscriber: Net Admin*
Naming Services	91	Net Admin	Global: Naming Services, Subscriber: Net Admin*
Terminals	92	Net Admin	Global: Terminals, Subscriber: Net Admin*
Other Net Admin	93	Net Admin	Global: Other Net Admin, Subscriber: Net Admin*
Newsgroups	8	Default Service	Newsgroups
Voice and Video Calls	12	Default Service	Global: Default Service*, Subscriber: Voice and Video Calls*
Google Talk VoIP	68	Voice and Video Calls	Global: Google Talk VoIP, Subscriber: Voice and Video Calls*
H323	11	Voice and Video Calls	Global: H323, Subscriber: Voice and Video Calls*
ICQ VoIP	40	Voice and Video Calls	Global: ICQ VoIP, Subscriber: Voice and Video Calls*
MGCP	5	Voice and Video Calls	Global: MGCP, Subscriber: Voice and Video Calls*
QQ VoIP	69	Voice and Video Calls	Global: QQ VoIP, Subscriber: Voice and Video Calls*
SIP	10	Voice and Video Calls	Global: SIP, Subscriber: Voice and Video Calls*
Skype	25	Voice and Video Calls	Global: Default Service*, Subscriber: Voice and Video Calls*
Skype VoIP	97	Skype	Global: Skype VoIP, Subscriber: Voice and Video Calls*
SkypeIn	65	Skype	Global: SkypeIn, Subscriber: Voice and Video Calls*
SkypeOut	66	Skype	Global: SkypeOut, Subscriber: Voice and Video Calls*
Other Skype	67	Skype	Global: Other Skype, Subscriber: Voice and Video Calls*
Vonage	13	Voice and Video Calls	Global: Vonage, Subscriber: Voice and Video Calls*

Table 1-9 Installed Services (continued)

Name	ID	Name of Parent Service	Global Usage Counter and Subscriber Usage Counter
Windows Live Messenger VoIP and Video	15	Voice and Video Calls	Global: Default Service*, Subscriber: Voice and Video Calls*
Windows Live Messenger Video	18	Windows Live Messenger VoIP and Video	Global: Windows Live Messenger Video, Subscriber: Voice and Video Calls*
Windows Live Messenger VoIP	46	Windows Live Messenger VoIP and Video	Global: Windows Live Messenger VoIP, Subscriber: Voice and Video Calls*
Yahoo Messenger VoIP and Video	31	Voice and Video Calls	Global: Default Service*, Subscriber: Voice and Video Calls*
Yahoo Messenger Video	35	Yahoo Messenger VoIP and Video	Global: Yahoo Messenger Video, Subscriber: Voice and Video Calls*
Yahoo Messenger VoIP	37	Yahoo Messenger VoIP and Video	Global: Yahoo Messenger VoIP, Subscriber: Voice and Video Calls*
Behavioral VoIP	64	Voice and Video Calls	Global: Behavioral VoIP, Subscriber: Voice and Video Calls*
Other VoIP	36	Voice and Video Calls	Global: Other VoIP, Subscriber: Voice and Video Calls*
Other	1	Default Service	Default Service*
Other IP	6	Other	Other IP
Other TCP	2	Other	Other TCP
Other UDP	3	Other	Other UDP
Other Well-Known Ports	96	Other	Other Well-Known Ports

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 1-10 lists the RDR settings defined in the default service configuration.

Table 1-10 Default RDR Settings

RDR Family	RDR Name	State	Rate	Rate Limit	Notes
Usage	Generic	ON	Every 5 minutes	—	—
	Link	ON	Every 5 minutes	—	—
	Package	ON	Every 5 minutes	—	—
	Subscriber	ON	Every 10 minutes	—	—
	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.
	HTTP Transaction Usage	OFF	—	—	—
	Anonymized HTTP Transaction Usage	OFF	—	—	—
	RTSP Transaction Usage	OFF	—	—	—
	Video Transaction Usage	OFF	—	—	—
	VoIP Transaction Usage	OFF	—	—	—

Table 1-10 Default RDR Settings (continued)

RDR Family	RDR Name	State	Rate	Rate Limit	Notes
Quota	Quota Breach	OFF	—	—	Generate RDR when bucket is breached.
	Quota Status	OFF	user configured	user configured	—
	Quota Threshold Breach	OFF	—	—	Generate RDR each time bucket exceeds threshold.
	Session Creation	OFF	—	—	Generated upon subscriber introduction or package switch.
Log	Block	ON	—	20 per second	—
Real-Time Subscriber	Real-Time Subscriber Usage	ON	Every 1 minutes	100 per second	Enable for each subscriber separately, using CLI.
Attack	Attack Start	OFF	—	—	—
	Attack Stop		—	—	—
Malicious Traffic	Malicious Traffic Periodic	ON	Every 60 seconds	—	Only generated during attack.
Spam	Spam	OFF	—	—	—
DHCP	DHCP	OFF	—	—	—
RADIUS	RADIUS	OFF	—	—	—

Rules

Rules are a 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 Bandwidth Controllers (BWCs), which have unlimited Bandwidth (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 unidirectional classifications 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

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

Introduction

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

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

Raw Data Records Overview

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

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

Universal RDR Fields

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

- SUBSCRIBER_ID—Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
- PACKAGE_ID—ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
- ACCESS_STRING—Layer 7 property, extracted from the transaction. For possible values see String Fields, page 2-78.
- BREACH_STATE—Indicates whether the subscriber's quota was breached.
  - 0—Not breached
  - 1—Breached

- CLIENT_IP—IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
- CLIENT_PORT—Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
- CONFIGURED_DURATION—Configured period, in seconds for periodic RDRs, between successive RDRs.
- END_TIME—Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
- FLAVOR_ID—ID of the flavor associated with this session. This is for protocol signatures that have flavors.
- INFO_STRING—Layer 7 property extracted from the transaction. For possible values see String Fields, page 2-78.
- INITIATING_SIDE—Side of the SCE platform on which the initiator of the transaction resides.
  - 0—The subscriber side
  - 1—The network side
- PROTOCOL_ID—Unique ID of the protocol associated with the reported session.

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

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

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

Related Topics

- String Fields, page 2-78

ADDITIONAL_INFO Field

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

Table 2-1 ADDITIONAL_INFO Field Definition

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

Transaction RDR

This section contains descriptions of transaction RDRs

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

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

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

  See the Sizing Tool for the appropriate sample rate; a sample rate which is too high may cause CM sizing problems. A sample rate which is too low reduces the accuracy of TR-based reports.
- RDR tag— 0xf0f0f010 / 4042321936

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

Table 2-2 Transaction RDR Fields

RDR Field Name	Type	Description	Example Value
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.	john
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.	0 [Default Package]
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.	16 [HTTP]
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.	2 [HTTP]
SKIPPED_SESSIONS	UINT32	Number of unreported sessions since the previous RDR *plus one*. The default value is 1. A value of 2 means that *one* RDR was unreported.	10
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.	198.133.219.25
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.	80
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.	www.cisco.com
INFO_STRING	STRING	Layer 7 property extracted from the transaction.	/en/US/partner/

Table 2-2 Transaction RDR Fields (continued)

RDR Field Name	Type	Description	Example Value
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.	192.118.76.130
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.	3221
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side	0 [subscriber-initiated]
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.	—
MILLISEC_ DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.	310
TIME_FRAME	INT8	Time frame during which the RDR was generated. The field's value is in the range 0 to 3, indicating which of the four time frames was used. The system supports time-dependent policies, by using different rules for different time frames.	0
SESSION_ UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.	32
SESSION_ DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.	117
SUBSCRIBER_ COUNTER_ID	UINT16	Counter to which each service is mapped.There are 32 subscriber usage counters.	1
GLOBAL_COUNTER_ ID	UINT16	Counter to which each service is mapped.There are 64 global usage counters.	9

Table 2-2 Transaction RDR Fields (continued)

RDR Field Name	Type	Description	Example Value
PACKAGE_ COUNTER_ID	UINT16	Counter to which each package is mapped.There are 1024 package usage counters.	0
IP_PROTOCOL	UINT8	IP protocol type.	6 [TCP]
PROTOCOL_ SIGNATURE	INT32	ID of the protocol signature associated with this session.	0x3010000 [HTTP]
ZONE_ID	INT32	ID of the zone associated with this session.	0
FLAVOR_ID	INT32	ID of the flavor associated with this session.	0
FLOW_CLOSE_ MODE	UINT8	Reason for the end of flow. • 0 [TCP_NORMAL_CLOSE] • 2 [The flow was closed by the aging mechanism.]	0

Related Topics

• Universal RDR Fields, page 2-2

Transaction Usage RDR

- RDR Purpose—Log network transactions for transaction-based billing or offline data mining.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single transaction; its connection attributes, extracted L7 attributes, duration and volume.
- RDR Generation Logic—Generated at the end of a session, for all transactions on packages and services that are configured to generate such an RDR.

  This RDR is not generated for sessions that were blocked by a rule.
- RDR tag—0xf0f0f438 / 4042323000

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

The Transaction Usage RDR is designed for services and packages where specific, per-transaction RDRs are required (such as, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate. See the Sizing Tool.

Note Configure the generation scheme for this RDR with extra care.

Table 2-3 lists the Transaction Usage RDR fields and their descriptions.

Table 2-3 Transaction Usage RDR Fields

RDR Field Name	Type	Description	Example Value
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.	john
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.	0 [Default Package
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.	16 [HTTP]
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.	2 [HTTP]
SKIPPED_SESSIONS	UINT32	Reason for RDR generation: • 0 (INTERIM)—Interim Transaction Usage RDR • 1 (SESSION_END)—Normal Transaction Usage RDR for a flow that had no interim Transaction Usage RDRs • 2 (LAST_TUR)—The last Transaction Usage RDR for a flow that had interim Transaction Usage RDRs	1 [SESSION_END]
SERVER_IP	UINT32	Contains the destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.	198.133.219.25
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.	80

Table 2-3 Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description	Example Value
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.	www.cisco.com
INFO_STRING	STRING	Layer 7 property extracted from the transaction.	/en/US/partner/
CLIENT_IP	UNIT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.	192.118.76.130
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.	3221
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side	0 [subscriber-initiated]
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.	—
MILLISEC_ DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.	310
TIME_FRAME	INT8	Time frame during which the RDR was generated. The field's value can be in the range 0 to 3, indicating which of the four time frames was used. The system supports time-dependent policies, by using different rules for different time frames.	0
SESSION_ UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.	32
SESSION_ DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.	117

Table 2-3 Transaction Usage RDR Fields (continued)

RDR Field Name	Type	Description	Example Value
SUBSCRIBER_ COUNTER_ID	UINT16	Counter to which each service is mapped.There are 32 subscriber usage counters.	1
GLOBAL_COUNTER_ ID	UINT16	Counter to which each service is mapped.There are 64 global usage counters.	9
PACKAGE_ COUNTER_ID	UINT16	Counter to which each package is mapped.There are 1024 package usage counters.	0
IP_PROTOCOL	UINT8	IP protocol type.	6 [TCP]
PROTOCOL_ SIGNATURE	INT32	ID of the protocol signature associated with this session.	0x3010000 [HTTP]
ZONE_ID	INT32	ID of the zone associated with this session.	0
FLAVOR_ID	INT32	ID of the flavor associated with this session.	0
FLOW_CLOSE_ MODE	UINT8	Reason for the end of flow. • 0 [TCP_NORMAL_CLOSE] • 2 [The flow was closed by the aging mechanism.]	0

Related Topics

- Universal RDR Fields, page 2-2

HTTP Transaction Usage RDR

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

- RDR Purpose—Log HTTP network transactions for transaction-based billing or offline data mining.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single HTTP transaction; its connection attributes, extracted L7 attributes, duration, and volume.
- RDR Generation Logic—Generated at the end of an HTTP session, for all transactions on packages and services that are configured to generate a Transaction Usage RDR.

  This RDR is not generated for sessions that were blocked by a rule.
- RDR tag—0xf0f0f43C / 4042323004

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

This RDR is designed for services and packages where specific, per-transaction RDRs are required (such as, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate.

Note Configure the generation scheme for this RDR with extra care.

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

Table 2-4 HTTP Transaction Usage RDR Fields

RDR Field Name	Type	Description	Example Value
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.	john
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.	0 [Default Package]
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.	16 [HTTP]
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.	2 [HTTP]
SKIPPED_SESSIONS	UINT32	Number of unreported sessions since the previous RDR. Since an HTTP Transaction Usage RDR is generated only at the end of a flow, this field always has the value 1.	1 [SESSION_END]
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.	198.133.219.25
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.	80
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.	www.cisco.com

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

RDR Field Name	Type	Description	Example Value
INFO_STRING	STRING	Layer 7 property extracted from the transaction.	/en/US/partner/
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.	192.118.76.130
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.	3221
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side	0 [subscriber-initiated]
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.	—
MILLISEC_ DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.	310
TIME_FRAME	INT8	Time frame during which the RDR was generated. The field's value can be in the range 0 to 3, indicating which of the four time frames was used. The system supports time-dependent policies, by using different rules for different time frames.	0
SESSION_ UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.	32
SESSION_ DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.	117
SUBSCRIBER_ COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.	1

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

RDR Field Name	Type	Description	Example Value
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.	9
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.	0
IP_PROTOCOL	UINT8	IP protocol type.	6 [TCP]
PROTOCOL_SIGNATURE	INT32	ID of the protocol signature associated with this session.	0x3010000 [HTTP]
ZONE_ID	INT32	ID of the zone associated with this session.	0
FLAVOR_ID	INT32	ID of the flavor associated with this session.	0
FLOW_CLOSE_MODE	UINT8	Reason for the end of flow. • 0 [TCP_NORMAL_CLOSE] • 2 [The flow was closed by the aging mechanism.]	0
USER_AGENT	STRING	User agent field extracted from the HTTP transaction.	Moselle
HTTP_URL	STRING	URL extracted from the HTTP transaction.	/en/US/partner/
HTTP_REFERER	STRING	REFERER extracted from the HTTP transaction.	http://addition.cnn.com
HTTP_COOKIE	STRING	COOKIE extracted from the HTTP transaction.	SelectedAddition=Addition;CNNid=3459286729-09

Related Topics

- Universal RDR Fields, page 2-2

Anonymized HTTP Transaction Usage RDR

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

- RDR Purpose—Log HTTP network transactions for transaction-based billing or offline data mining without personal subscriber data.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single HTTP transaction; its connection attributes, extracted L7 attributes, duration, and volume.
- RDR Generation Logic—Generated at the end of an HTTP session, for all transactions on packages and services that are configured to generate a Transaction Usage RDR.

  This RDR is not generated for sessions that were blocked by a rule.
- RDR tag—0xf0f0f53C / 4042323260

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

This RDR is designed for services and packages where specific, per-transaction RDRs are required (such as, transaction level billing). It is easy to configure this RDR, in error, so that it is generated for every transaction, which may result in an excessive RDR rate.

Note Configure the generation scheme for this RDR with extra care.

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

Table 2-5 Anonymized HTTP Transaction Usage RDR Fields

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

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

RDR Field Name	Type	Description
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the networking session.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0–The subscriber side • 1–The network side
REPORT_TIME	UINT32	Ending time stamp of this RDR.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	Time frame during which the RDR was generated. (0 – 3)
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated stream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	UINT32	ID of the protocol signature associated with this session.
ZONE_ID	UINT32	ID of the zone associated with this session.
FLAVOR_ID	UINT32	ID of the flavor associated with this session.
FLOW_CLOSE_MODE	UINT8	Reason for the end of flow.
HASHED_SUBSCRIBER_IP	STRING	Subscriber IP, may be hashed if configured.
USER_AGENT	STRING	User agent field extracted from the HTTP transaction.
HTTP_REFERER	STRING	REFERER extracted from the HTTP transaction.
HTTP_COOKIE	STRING	COOKIE extracted from the HTTP transaction.

RTSP Transaction Usage RDR

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

- RDR Purpose—Log RTSP network transactions for transaction-based billing or offline data mining.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single RTSP transaction; its connection attributes, extracted L7 attributes, duration, and volume.
- RDR Generation Logic—Generated at the end of a session, for all RTSP transactions on packages and services that are configured to generate a Transaction Usage RDR.

  This RDR is not generated for sessions that were blocked by a rule.
- RDR tag—0xf0f0f440 / 4042323008

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

This RDR is designed for services and packages where specific, per-transaction RDRs are required (such as, transaction level billing). It is easy to configure this RDR in error, so that it is generated for every transaction, which may result in an excessive RDR rate.

Note Configure the generation scheme for this RDR with extra care.

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

Table 2-6 RTSP Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
SKIPPED_SESSIONS	UINT32	Number of unreported sessions since the previous RDR. Since an RTSP Transaction Usage RDR is generated only at the end of a flow, this field always has the value 1.

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

RDR Field Name	Type	Description
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	System supports time-dependent policies, by using different rules for different time frames. This field indicates the time frame during which the RDR was generated. The field's value can be in the range 0 to 3, indicating which of the four time frames was used.
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.

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

RDR Field Name	Type	Description
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.
IP_PROTOCOL	UNIT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	ID of the protocol signature associated with this session.
ZONE_ID	INT32	ID of the zone associated with this session.
FLAVOR_ID	INT32	ID of the protocol signature with flavor associated with this session.
FLOW_CLOSE_MODE	UINT8	The reason for the end of flow. • 0 [TCP_NORMAL_CLOSE] • 2 [The flow was closed by the aging mechanism.]
RTSP_SESSION_ID	STRING	RTSP session ID as seen on an RTSP SETUP request.
RTSP_URL	STRING	RTSP URL.
RESPONSE_DATE	STRING	RTSP DESCRIBE date.
TOTAL_ENCODING_RATE	UINT32	Sum of encoding rates of data flows.
NUMBER_OF_VIDEO_ STREAMS	UINT8	Number of video streams for this RTSP session.
NUMBER_OF_AUDIO_ STREAMS	UINT8	Number of audio streams for this RTSP session.
SESSION_TITLE	STRING	Title for this RTSP stream.
SERVER_NAME	STRING	Name of the RTSP server.

Related Topics

- Universal RDR Fields, page 2-2

VoIP Transaction Usage RDR

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

- RDR Purpose—Log VOIP network transactions for transaction-based billing or offline data mining.
- RDR Default destination—Sent to the CM, and stored in CSV files.
- RDR Content—Describes a single RTSP transaction; its connection attributes, extracted Layer 7 attributes, duration, and volume.
- RDR Generation Logic—Generated at the end of a session, for all transactions on packages and services that are configured to generate such an RDR.

  This RDR is not generated for sessions that were blocked by a rule.
- RDR tag—0xf0f0f46a / 4042323050

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

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

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

Note Configure the generation scheme for this RDR with extra care.

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

Table 2-7 VoIP Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.

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

RDR Field Name	Type	Description
SKIPPED_SESSIONS	UINT32	Number of unreported sessions since the previous RDR. Since a VoIP Transaction Usage RDR is generated only at the end of a flow, this field always has the value 1.
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	Time frame during which the RDR was generated. The field's value can be in the range 0 to 3, indicating which of the four time frames was used. The system supports time-dependent policies, by using different rules for different time frames.
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.

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

RDR Field Name	Type	Description
GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped. There are 1024 package usage counters.
IP_PROTOCOL	UINT8	IP protocol type.
PROTOCOL_SIGNATURE	INT32	ID of the protocol signature associated with this session.
ZONE_ID	INT32	ID of the zone associated with this session.
FLAVOR_ID	INT32	ID of the protocol signatures with flavor associated with this session.
FLOW_CLOSE_MODE	UINT8	The ITU-U vendor ID of the application. A value of 0xFFFFFFFF indicates that this field was not found in the traffic.
APPLICATION_ID	UINT32	ITU-U vendor ID of the application. A value of 0xFFFFFFFF indicates that this field was not found in the traffic.
UPSTREAM_PACKET_LOSS (see Note, page 22)	UINT16	Average fractional upstream packet loss for the session, taken from the RTCP flow. (See the note following this table for an explanation of this value.) A value of 0xFFFF indicates that this field is undefined (no RTCP flows were opened).
DOWNSTREAM_PACKET_ LOSS (see Note, page 22)	UINT16	Average fractional downstream packet loss for the session, taken from the RTCP flow. (See the note following this table for an explanation of this value.) A value of 0xFFFF indicates that this field is undefined (no RTCP flows were opened).
UPSTREAM_AVERAGE_ JITTER (see Note, page 22)	UINT32	Average upstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (See the note following this table for an explanation of this value.) A value of 0xFFFFFFFF indicates that this field is undefined (no RTCP flows were opened).
DOWNSTREAM_AVERAGE_ JITTER (see Note, page 22)	UINT32	Average downstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (See the note following this table for an explanation of this value.) A value of 0xFFFFFFFF indicates that this field is undefined (no RTCP flows were opened).
CALL_DESTINATION	STRING	Q931 Alias address of the session destination. A value of N/A indicates that this field was not found in the traffic.
CALL_SOURCE	STRING	Q931 Alias address of the session source. A value of N/A indicates that this field was not found in the traffic.

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

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

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

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

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

Related Topics

- Universal RDR Fields, page 2-2

Video Transaction Usage RDR

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

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

  This RDR is not generated for sessions that were blocked by a rule.
- RDR tag-0xf0f0f480 / 4042323072

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

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

Note Configure the generation scheme for this RDR with extra care.

Table 2-8 lists the VIDEO_TRANSACTION_USAGE_RDR fields and their descriptions.

Table 2-8 Video Transaction Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.

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

RDR Field Name	Type	Description
SKIPPED_SESSIONS	UINT32	Number of unreported sessions since the previous RDR. Since an RTSP Transaction Usage RDR is generated only at the end of a flow, this field always has the value 1.
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.
TIME_FRAME	INT8	Time frame during which the RDR was generated. The field's value can be in the range 0 to 3, indicating which of the four time frames was used. The system supports time-dependent policies, by using different rules for different time frames.
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.
SESSION_DOWNSTREAM_ VOLUME	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated downstream volume on both links of all the flows bundled in the transaction.
SUBSCRIBER_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.

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

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

Generic Usage RDR

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

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

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

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

Table 2-9 Generic Usage RDR

Key/Data	RDR Field Name	Type	Description
Key	GUR_TYPE	INT32	Type of the GUR—defines the usage of the rest of the fields
Key	LINK_ID	INT8	Numeric value associated with the reported network link. Possible values are 0 and 1 (referring to physical links 1 and 2 respectively). For future use.
Key	GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR. Possible values are 0 to 3.
Key	GLOBAL_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters
Key	SUBSCRIBER_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 subscriber usage counters.
Key	PACKAGE_COUNTER_ID	UINT16	Counter to which each package is mapped.There are 1024 package usage counters.
Key	SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
Key	PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
Key	SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
Key	PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
Key	SIGNATURE_ID	INT32	ID of the protocol signature associated with this session.

Table 2-9 Generic Usage RDR

Key/Data	RDR Field Name	Type	Description
Key	DESTINATION_IP	UINT32	• SIP: Destination IP address of RTP flow. • Skype: Destination IP address of Skype flow.
Key	DESTINATION_PORT	UINT16	• SIP: Destination port of RTP flow. • Skype: Destination port of Skype flow.
Key	SOURCE_IP	UINT32	• SIP: Source IP address of RTP flow. • Skype: Source IP address of Skype flow.
Key	SOURCE_PORT	UINT16	• SIP: Source port of RTP flow. • Skype: Source port of Skype flow.
Key	INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side For Skype, this is the initiating side of the flow (not necessarily the initiating side of the voice call).
Key	ZONE_ID	INT32	ID of the zone associated with this session.
Key	FLAVOR_ID	INT32	ID of protocol signatures with flavor associated with this session.
Key	SESSION_ID	UINT32	• SIP: The flow-context ID of the control flow. • Skype: The flow-context ID of the flow.
Key	START_TIME	UINT32	Flow start time.
Key	END_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
Key	ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
Key	INFO_STRING	STRING	Layer 7 property extracted from the transaction.
Key	For future use	INT32	—
Key	For future use	INT32	—
Key	For future use	STRING	—
Key	For future use	STRING	—
Data	UPSTREAM_VOLUME	INT32	Aggregated upstream volume of all sessions, in kilobytes, for the current reporting period.
Data	DOWNSTREAM_VOLUME	INT32	Aggregated downstream volume of all sessions, in kilobytes, for the current reporting period.
Data	TOTAL_VOLUME	INT32	Aggregated total volume of all sessions, in kilobytes, for the current reporting period.
Data	SESSIONS	INT32	Aggregated number of sessions for the reported service for the current reporting period.
Data	SECONDS	INT32	Aggregated number of session seconds for the reported service for the current reporting period.

Table 2-9 Generic Usage RDR

Key/Data	RDR Field Name	Type	Description
Data	CONCURRENT_SESSIONS	INT32	Concurrent number of sessions using the reported service at this point in time.
Data	ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers using the reported service at this point in time.
Data	TOTAL_ACTIVE_SUBSCRIBERS	INT32	Concurrent number of subscribers in the system at this point in time.
Data	CONFIGURED_DURATION	INT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
Data	DURATION	INT32	• This release—Not implemented (always the same as CONFIGURED_DURATION). • Future releases—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
Data	For future use	INT32	—
Data	For future use	INT32	—
Data	For future use	INT32	—
Data	For future use	INT32	—

Using the Generic Usage RDR to Report IPv6 Usage

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

Both reports use the GUR type '1'.

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

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

Table 2-10 Generic Usage RDR Fields for IPv6 Usage

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

Subscriber Usage RDR

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

- RDR Purpose—Compare subscribers for the Top Subscribers report, and create daily subscriber usage summary records.
- RDR Default destination—Sent to the CM, and processed by the Topper Adapter, which stores the processing results in the database and in CSV files. The Reporter tool uses the database records for creating the Top Subscribers reports.
- RDR Content—Summary of the activity of a single subscriber on a defined service for the last user-configured number of minutes, including aggregated number of flows, total volume, and duration.
- RDR Generation Logic—Generated periodically, at user-configured intervals, for each subscriber. A separate RDR is generated for each service usage counter. The RDR is generated only if the subscriber consumed resources associated with the service usage counter during the current reporting period.

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

  – If the subscriber consumed resources associated with a service usage counter since the previous RDR generation point, a Subscriber Usage RDR is generated.

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

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

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

- The subscriber performed a logout in a subscriber-integrated installation or was un-introduced from the SCE platform:

  – If the subscriber consumed resources associated with a service usage counter since the previous Subscriber Usage RDR, a Subscriber Usage RDR is generated.

  – If the subscriber did not consume resources since the previous RDR, no RDR is generated for that service usage counter.

- RDR tag—0xf0f0f000 / 4042321920

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

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

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

Table 2-11 Subscriber Usage RDR

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	The subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	The ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_USAGE_ COUNTER_ID_	UINT16	Counter to which each service is mapped. There are 32 counters in the subscriber scope.
BREACH_STATE	UINT8	Indicates whether the subscriber's quota was breached. • 0—Not breached • 1—Breached Holds the breach state of a service. However, this RDR reports usage counters, which cannot be breached, so the value is always zero.
REASON	UINT8	Reason for RDR generation: • 0—Period time passed • 1—Subscriber logout • 3—Wraparound • 5—Subscriber VLink change
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	Indicates the number of seconds that have passed since the previous Subscriber Usage RDR.
END_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT32	Aggregated number of session seconds for the reported service, for the current reporting period.

Table 2-11 Subscriber Usage RDR (continued)

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

Related Topics

- Periodic RDR Zero Adjustment Mechanism, page 2-83

Real-Time Subscriber Usage RDR

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

- RDR Purpose—Create detailed subscriber-level reports of network usage per service.
- RDR Default destination—Sent to the CM, stored in the database, and used by the Reporter tool for subscriber usage reports such as the Subscriber Bandwidth per Service report.
- RDR Content—Summary of the activity of a single subscriber on a specific service for the last user-configured number of minutes, including aggregated number of flows, total volume, and duration.
- RDR Generation Logic—Generated periodically, at user-configured intervals, for each subscriber that has real-time monitoring enabled. A separate RDR is generated for each service usage counter. The RDR is generated only if the subscriber consumed resources associated with the service usage counter during the current reporting period.

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

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

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

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

  However, the generation logic for Subscriber Usage RDRs uses the zeroing methodology (as described in Periodic RDR Zero Adjustment Mechanism, page 2-83). If the subscriber consumes resources associated with the service usage counter at some later time, this causes the immediate generation of either one or two zero-consumption Real-Time Subscriber Usage RDRs. (In addition to the eventual generation of the Real-Time Subscriber Usage RDR associated with this latest consumption of resources).

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

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

- The subscriber performed a logout in a subscriber-integrated installation or was un-introduced from the SCE platform:
  - If the subscriber consumed resources associated with a service usage counter since the previous Real-Time Subscriber Usage RDR, a Real-Time Subscriber Usage RDR is generated and then a zero-consumption Real-Time Subscriber Usage RDR is generated.
  - If the subscriber consumed resources associated with a service usage counter since the previous Real-Time Subscriber Usage RDR, a Real-Time Subscriber Usage RDR is generated and then a zero-consumption Real-Time Subscriber Usage RDR is generated.

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

- The subscriber performed a login in a subscriber-integrated installation or was introduced from the SCE platform:
  - Before the first Real-Time Subscriber Usage RDRs reporting actual consumption are generated, a zero-consumption Real-Time Subscriber Usage RDR is generated.
- RDR tag—0xf0f0f002 / 4042321922

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

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

Table 2-12 Real-Time Subscriber Usage RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	The subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	The ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_package. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_USAGE_ COUNTER_ID	UINT16	Counter to which each service is mapped. There are 32 counters in the subscriber scope.
AGGREGATION_OBJECT_ ID	INT16	Externally assigned: • 0—Offline subscriber • 1—Online subscriber

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

RDR Field Name	Type	Description
BREACH_STATE	UINT8	Indicates whether the subscriber's quota was breached. • 0—Not breached • 1—Breached Holds the breach state of a service. However, this RDR reports usage counters, which cannot be breached, so the value is always zero.
REASON	UINT8	Reason for RDR generation: • 0—Period time passed • 1—Subscriber logout • 3—Wraparound • 5—Subscriber VLink change
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	Indicates the number of seconds that have passed since the previous Real-Time Subscriber Usage RDR. **Note** This field is not valid for zeroing RDR, "1" with appear.
END_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period.
SESSIONS	INT16	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	INT16	Aggregated number of session seconds for the reported service, for the current reporting period.

Related Topics

- Periodic RDR Zero Adjustment Mechanism, page 2-83

Link Usage RDR

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

- RDR Purpose—Create link-level reports of network usage per service.
- RDR Default destination—Sent to the CM, stored in the database, and used by the reporter for global usage reports such as the Global Bandwidth per Service report, and subscriber demographics reports, such as the Active Subscribers per Service report.
- RDR Content—Summary of the activity on one of the SCE links for a specific service for the last user-configured minutes, including aggregated number of flows, total volume, duration, and active subscribers.
- RDR Generation Logic—Generated periodically, at user-configured intervals, for each link. A separate RDR is generated for each service usage counter. The RDR is generated only if resources associated with the service usage counter were consumed during the current reporting period.

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

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

    However, the generation logic for Link Usage RDRs uses the zeroing methodology (as described in Periodic RDR Zero Adjustment Mechanism, page 2-83). If network resources associated with the service are again consumed at some later time, this causes the immediate generation of either one or two zero-consumption Link Usage RDRs. (In addition to the eventual generation of the Link Usage RDR associated with this latest consumption of network resources).
  - If there was only one interval (for example, 0830–0900) for which there was no consumption of network resources, only one zero-consumption Link Usage RDR is generated.
  - If there were multiple consecutive intervals (for example, 0830–0900, 0900–0930, 0930–1000, 1000–1030) for which there was no consumption of network resources, two zero-consumption Link Usage RDR are generated: one for the first such time interval (0830–0900) and one for the last (1000–1030).

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

- RDR tag—0xf0f0f005 / 4042321925

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

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

Table 2-13 Link Usage RDR Fields

RDR Field Name	Type	Description
LINK_ID	INT8	A numeric value associated with the reported network link. Possible values are 0 and 1 (referring to physical links 1 and 2 respectively). For future use.
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR. Possible values are 0 to 3.
SERVICE_USAGE_ COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	UINT32	Concurrent number of sessions using the reported service at this point in time. A value 0 is reported for all links except link 0. Although the values are not reported in the respective links, a cumulative value is reported in link 0.
ACTIVE_SUBSCRIBERS	UINT32	Concurrent number of subscribers using the reported service at this point in time. A value 0 is reported for all links except link 0. Although the values are not reported in the respective links, a cumulative value is reported in link 0.
TOTAL_ACTIVE_ SUBSCRIBERS	UINT32	Subscribers having active bidirectional flows in the system. A value 0 is reported for all links except link 0. Although the values are not reported in the respective links, a cumulative value is reported in link 0.

Related Topics

- Periodic RDR Zero Adjustment Mechanism, page 2-83

Zone Usage RDR

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

- RDR Purpose—Create zone-level reports of network usage per service.
- RDR Default destination—Sent to the CM, stored in the database, and used by the reporter for global usage reports such as the Global Bandwidth per Service report, and subscriber demographics reports such as the Active Subscribers per Service report.
- RDR Content—Summary of the activity on one of the SCE zones for a specific service for the last user-configured minutes, including aggregated number of flows, total volume, duration, and active subscribers.
- RDR Generation Logic—Generated periodically, at user-configured intervals, for each zone. A separate RDR is generated for each service usage counter. The RDR is generated only if resources associated with the service usage counter were consumed during the current reporting period.

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

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

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

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

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

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

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

- RDR tag—4042321928

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

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

Table 2-14 Zone Usage RDR Fields

RDR Field Name	Type	Description
ZONE_COUNTER_ID	UINT16	ID of the zone associated with this session.
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR. Possible values are in the range from 0 to 3.
SERVICE_USAGE_ COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	INT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	UINT32	Concurrent number of sessions using the reported service at this point in time. Currently not supported in Release 3.6.5, so counter always returns 0 value.
ACTIVE_SUBSCRIBERS	UINT32	Concurrent number of subscribers using the reported service at this point in time. Currently not supported in Release 3.6.5, so counter always returns 0 value.
TOTAL_ACTIVE_ SUBSCRIBERS	UINT32	Concurrent number of subscribers in the system at this point in time. Currently not supported in Release 3.6.5, so counter always returns 0 value.

Related Topics

- Periodic RDR Zero Adjustment Mechanism, page 2-83

Package Usage RDR

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

- RDR Purpose—Create reports about network usage per service for a group of subscribers.
- RDR Default destination—Sent to the CM, stored in the database, and used by the Reporter tool for package usage reports such as the Package Bandwidth per Service report.
- RDR Content—Summary of the activity of a specific group of subscribers (belonging to the same package) for a specific service for the last user-configured number of minutes, including aggregated number of flows, total volume, and duration.
- RDR Generation Logic—Generated periodically, at user-configured intervals, for each package usage counter. A separate RDR is generated for each service usage counter. The RDR is generated only if resources associated with the service usage counter were consumed during the current reporting period. The RDR contains aggregated network usage information for all subscribers to the package or group of packages represented by the package usage counter.

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

  – If network resources associated with a service usage counter were consumed by a subscriber of the Package since the previous RDR generation point, a Package Usage RDR is generated.

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

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

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

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

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

- RDR tag—0xf0f0f004 / 4042321924

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

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

Table 2-15 Package Usage RDR Fields

RDR Field Name	Type	Description
PACKAGE_COUNTER_ID	UINT16	Counter to which each service is mapped. There are 1024 package usage counters.
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR.
SERVICE_USAGE_ COUNTER_ID	UINT16	Counter to which each service is mapped. There are 64 global usage counters.
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	This release—Not implemented (always the same as CONFIGURED_DURATION). Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR.
END_TIME	INT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Aggregated number of sessions for the reported service, for the current reporting period.
SECONDS	UINT32	Aggregated number of session seconds for the reported service, for the current reporting period.
CONCURRENT_SESSIONS	UINT32	Concurrent number of sessions using the reported service in the reported package at this point in time.
ACTIVE_SUBSCRIBERS	UINT32	Concurrent number of subscribers using the reported service in the reported package at this point in time.
TOTAL_ACTIVE_ SUBSCRIBERS	UINT32	Concurrent number of subscribers in the system at this point in time.

Related Topics

- Periodic RDR Zero Adjustment Mechanism, page 2-83

Virtual Links Usage RDR

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

- RDR Purpose—Create reports relating to network usage per service for a specific virtual link.
- RDR Default destination—Sent to the CM, stored in the database, and used by the reporter for virtual link reports such as the Virtual Link Bandwidth per Service report.
- RDR Content—Summary of the activity on one of the virtual links for a specific service for the last user-configured number of minutes, including aggregated number of flows, total volume, and duration.
- RDR Generation Logic—Generated periodically, at user-configured intervals, for each service usage counter. A separate RDR is generated for each virtual link. The RDR is generated only if resources associated with the virtual link were consumed during the current reporting period. The RDR contains aggregated network usage information for all subscribers to the same virtual link.

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

  – If network resources associated with the service usage counter were consumed by any subscriber of the virtual link since the previous RDR generation point, a Virtual Links Usage RDR is generated.

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

    However, the generation logic for Virtual Links Usage RDRs uses the zeroing methodology (as described in Periodic RDR Zero Adjustment Mechanism, page 2-83). If network resources associated with the service usage counter are again consumed by subscribers of the virtual link at some later time, this causes the immediate generation of either one or two zero-consumption Virtual Links Usage RDRs. (In addition to the eventual generation of the Virtual Links Usage RDR associated with this latest consumption of network resources by subscribers of the virtual link.)

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

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

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

- RDR tag—0xf0f0f006 / 4042321926

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

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

Table 2-16 Virtual Links Usage RDR Fields

RDR Field Name	Type	Description
VLINK_ID	INT16	Virtual link ID
VLINK_DIRECTION	INT8	Virtual link direction: • 0—Upstream • 1—Downstream
GENERATOR_ID	INT8	A numeric value identifying the processor generating the RDR.
SERVICE_USAGE_ COUNTER_ID	UINT16	Counter to which each service is mapped. There are 1024 global usage counters.
CONFIGURED_DURATION	UINT32	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	UINT32	Not implemented (always the same as CONFIGURED_DURATION).
END_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
UPSTREAM_VOLUME	UINT32	Aggregated upstream volume on the virtual link (for a single processor) of all sessions, in kilobytes, for the current reporting period.
DOWNSTREAM_VOLUME	UINT32	Aggregated downstream volume on the virtual link (for a single processor) of all sessions, in kilobytes, for the current reporting period.
SESSIONS	UINT32	Reserved for future use.
SECONDS	UINT32	Reserved for future use.
CONCURRENT_SESSIONS	UINT32	Reserved for future use.
ACTIVE_SUBSCRIBERS	UINT32	Reserved for future use.
TOTAL_ACTIVE_ SUBSCRIBERS	UINT32	Concurrent number of subscribers in the system at this point in time.

Related Topics

- Periodic RDR Zero Adjustment Mechanism, page 2-83

Blocking RDR

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

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

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

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

Table 2-17 Blocking RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.

Table 2-17 Blocking RDR Fields (continued)

RDR Field Name	Type	Description
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.
BLOCK_REASON	UINT8	Indicates the reason why this session was blocked.
BLOCK_RDR_COUNT	INT32	Total number of blocked flows reported so far (from the beginning of the current aggregation period).
REDIRECTED	INT8	Indicates whether the flow has been redirected after being blocked. • 0—Not redirected • 1—Redirected
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.

Related Topics

- Block Reason (uint8), page 2-77

Quota Breach RDR

The QUOTA_BREACH_RDR is generated each time a bucket is breached.

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

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

Table 2-18 lists the Quota Breach RDR fields and their descriptions.

Table 2-18 Quota Breach RDR Fields

RDR Field Name	Type	Description
QUOTA_MODEL_TYPE	UINT8	Quota model type: • 1 - Gy Quota Model • 2 - QM Quota Model • 3 - Internal Quota Model
RDR_REASON	UINT8	Reason the RDR was sent. Not in use, RESERVED - 0xfe
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 40 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
ADDITIONAL_INFO	UINT32	See ADDITIONAL_INFO Field, page 2-4 for details.
END_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
BUCKET_ID	UINT16	Bucket ID to report.

Table 2-18 Quota Breach RDR Fields (continued)

RDR Field Name	Type	Description
BUCKET_TYPE	UINT16	Bucket type: • 1—Volume_UP Only the upstream volume is reported in the RDR. UNIT_AMOUNT_IN is 0 and UNIT_AMOUNT_OUT indicates the upstream volume. • 2—Volume_DOWN Only the downstream volume is reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT is 0. • 3—Total Volume The sum of downstream and upstream volumes, that is, the total volume consumed, and the remaining volume, that is, bucket size – total volume is reported in the RDR. UNIT_AMOUNT_IN indicates the total volume consumed and UNIT_AMOUNT_OUT indicates the remaining volume. • 4—VolumeUpDown Both upstream and downstream volumes are reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT indicates the upstream volume. • 5—Events (sessions) UNIT_AMOUNT_IN indicates the number of sessions that has used the bucket. UNIT_AMOUNT_OUT indicates the remaining number of sessions for the bucket. • 6—Time UNIT_AMOUNT_IN indicates how long a bucket has been used. This unit is represented in seconds. The UNIT_AMOUNT_OUT field is 0. **Note** For the following bucket types, only the UNIT_AMOUNT_IN field is valid: – Time – Events (sessions) – Total Volume
UNIT_AMOUNT_IN	UINT32	Consumed downstream volume in volume units/ Seconds/ Sessions.

Table 2-18 Quota Breach RDR Fields (continued)

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

Quota Status RDR

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

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

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

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

The following events trigger the sending of this RDR:

- Periodically, at user-configured intervals. The intervals are defined globally.

  Applies to all quota models, including internal and QM external quota models.
- Package switch event: Indicates consumed quota before the package switch.

  Applies to all quota models.
- Subscriber logout event.

  Applies to all quota models.
- Quota Validity Time/Quota Holding Time expiration.

  Applies to Gy quota model only.

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

Table 2-19 lists the Quota Status RDR fields and descriptions.

Table 2-19 Quota Status RDR Fields

RDR Field Name	Type	Description
QUOTA_MODEL_TYPE	UINT8	Quota model type: • 1 - Gy Quota Model • 2 - QM Quota Model • 3 - Internal Quota Model
RDR_REASON	UINT8	Reason the RDR was sent: • 0 - Period time passed • 1 - Logout • 2 - Package Switch • 5 - Request (RAR)
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 40 characters. For unknown subscribers this field may contain an empty string.

Table 2-19 Quota Status RDR Fields

RDR Field Name	Type	Description
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
ADDITIONAL_INFO	UINT32	See ADDITIONAL_INFO Field, page 2-4 for details.
END_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
BUCKET_ID	UINT16	Bucket ID to report.

Table 2-19 Quota Status RDR Fields

RDR Field Name	Type	Description
BUCKET_TYPE	UINT16	Bucket type: • 1—Volume_UP Only the upstream volume is reported in the RDR. UNIT_AMOUNT_IN is 0 and UNIT_AMOUNT_OUT indicates the upstream volume. • 2—Volume_DOWN Only the downstream volume is reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT is 0. • 3—Total Volume The sum of downstream and upstream volumes, that is, the total volume consumed, and the remaining volume, that is, bucket size – total volume is reported in the RDR. UNIT_AMOUNT_IN indicates the total volume consumed and UNIT_AMOUNT_OUT indicates the remaining volume. • 4—VolumeUpDown Both upstream and downstream volumes are reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT indicates the upstream volume. • 5—Events (sessions) UNIT_AMOUNT_IN indicates the number of sessions that has used the bucket. UNIT_AMOUNT_OUT indicates the remaining number of sessions for the bucket. • 6—Time UNIT_AMOUNT_IN indicates how long a bucket has been used. This unit is represented in seconds. The UNIT_AMOUNT_OUT field is 0. **Note** For the following bucket types, only the UNIT_AMOUNT_IN field is valid: – Time – Events (sessions) – Total Volume

Table 2-19 Quota Status RDR Fields

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

Note The following fields report information per bucket:

- BUCKET_ID
- BUCKET_TYPE
- UNIT_AMOUNT_IN
- UNIT_AMOUNT_OUT

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

Quota Threshold Breach RDR

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

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

The RDR tag of the QUOTA_THRESHOLD_BREACH_RDR is 0xf0f0f073 / 4,042,322,035.

Table 2-20 lists the Quota Threshold Breach RDR fields and their descriptions.

Table 2-20 Quota Threshold Breach RDR Fields

RDR Field Name	Type	Description
QUOTA_MODEL_TYPE	UINT8	Quota model type: • 1 - Gy Quota Model • 2 - QM Quota Model • 3 - Internal Quota Model
RDR_REASON	UINT8	Reason the RDR was sent. Not in use, RESERVED - 0xfe
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 40 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_package. The value maximum_number_of_packages is reserved for unknown subscribers.
ADDITIONAL_INFO	UINT32	See ADDITIONAL_INFO Field, page 2-4 for details.
END_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
BUCKET_ID	UINT16	Bucket ID to report.

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

RDR Field Name	Type	Description
BUCKET_TYPE	UINT16	Bucket type: • 1—Volume_UP Only the upstream volume is reported in the RDR. UNIT_AMOUNT_IN is 0 and UNIT_AMOUNT_OUT indicates the upstream volume. • 2—Volume_DOWN Only the downstream volume is reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT is 0. • 3—Total Volume The sum of downstream and upstream volumes, that is, the total volume consumed, and the remaining volume, that is, bucket size – total volume is reported in the RDR. UNIT_AMOUNT_IN indicates the total volume consumed and UNIT_AMOUNT_OUT indicates the remaining volume. • 4—VolumeUpDown Both upstream and downstream volumes are reported in the RDR. UNIT_AMOUNT_IN indicates the downstream volume and UNIT_AMOUNT_OUT indicates the upstream volume. • 5—Events (sessions) UNIT_AMOUNT_IN indicates the number of sessions that has used the bucket. UNIT_AMOUNT_OUT indicates the remaining number of sessions for the bucket. • 6—Time UNIT_AMOUNT_IN indicates how long a bucket has been used. This unit is represented in seconds. The UNIT_AMOUNT_OUT field is 0. **Note** For the following bucket types, only the UNIT_AMOUNT_IN field is valid: – Time – Events (sessions) – Total Volume
UNIT_AMOUNT_IN	UINT32	Consumed downstream volume in volume units/ Seconds/ Sessions.

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

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

Session Creation RDRs

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

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

The following events trigger the sending of this RDR:

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

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

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

Table 2-21 Session Creation RDR Fields

RDR Field Name	Type	Description
QUOTA_MODEL_TYPE	UINT8	Quota model type: • 1 - Gy Quota Model • 2 - QM Quota Model • 3 - Internal Quota Model
RDR_REASON	UINT8	Reason the RDR was sent: • 2 - Package Switch • 3 - Login
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 40 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
ADDITIONAL_INFO	UINT32	See ADDITIONAL_INFO Field, page 2-4 for details.
END_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
BUCKET_ID	UINT16	Use for Gy quota model only. If **request quota upon login** is specified for the bucket, this field contains bucket id. This indicates to the server that quota should be provided to specified bucket id. '0xFFFF' – reserved.
BUCKET_TYPE	UINT16	Not used. 0

Table 2-21 Session Creation RDR Fields (continued)

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

Note The following fields report information per bucket:

- BUCKET_ID
- BUCKET_TYPE
- UNIT_AMOUNT_IN
- UNIT_AMOUNT_OUT

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

DHCP RDR

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

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

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

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

Table 2-22 lists the DHCP RDR fields and descriptions.

Table 2-22 DHCP RDR Fields

RDR Field Name	Type	Description
CPE_MAC	STRING	DHCP protocol field.
CMTS_IP	UINT32	DHCP protocol field.
ASSIGNED_IP	UINT32	DHCP protocol field.
RELEASED_IP	UINT32	DHCP protocol field.
TRANSACTION_ID	UINT32	DHCP protocol field.
MESSAGE_TYPE	UINT8	DHCP message type.
OPTION_TYPE_0 through OPTION_TYPE_7	UINT8	List of DHCP options extracted from the message.
OPTION_TYPE_0 through OPTION_TYPE_7	STRING	Values associated with the above DHCP options.
END_TIME	INT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.

RADIUS RDR

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

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

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

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

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

Table 2-23 RADIUS RDR Fields

RDR Field Name	Type	Description
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side
RADIUS_PACKET_CODE	UINT8	Type of the RADIUS message intercepted.
RADIUS_ID	UINT8	RADIUS transaction ID.
ATTRIBUTE_VALUE_1 through ATTRIBUTE_VALUE_20	STRING	Attributes extracted from the message. Sent as string format TLV. The last attribute field filled takes the value 0.

Flow Start RDR

The FLOW_START_RDR is generated when a flow starts, as follows:

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

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

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

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

Table 2-24 Flow Start RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	Contains the destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.

Table 2-24 Flow Start RDR Fields

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side
START_TIME	UINT32	Flow start time.
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
BREACH_STATE	INT8	Indicates whether the subscriber's quota was breached. • 0—Not breached • 1—Breached
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR.

Flow End RDR

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

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

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

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

Table 2-25 Flow End RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_package. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.

Table 2-25 Flow End RDR Fields (continued)

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side
START_TIME	UINT32	Flow start time.
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
BREACH_STATE	INT8	Indicates whether the subscriber's quota was breached. • 0—Not breached • 1—Breached
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR.

Ongoing Flow RDR

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

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

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

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

Table 2-26 Ongoing Flow RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
IP_PROTOCOL	UINT8	IP protocol type.
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.

Table 2-26 Ongoing Flow RDR Fields (continued)

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side
START_TIME	UINT32	Flow start time.
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
BREACH_STATE	INT8	Indicates whether the subscriber's quota was breached. • 0—Not breached • 1—Breached
FLOW ID	UINT32	Internal flow ID.
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR.

Media Flow RDR

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

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

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

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

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

Table 2-27 Media Flow RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers, this field may contain an empty string.
PACKAGE_ID	INT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer between 0 and maximum_number_of_packages. The maximum_number_of_packages value is reserved for unknown subscribers.
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR, this field indicates which service was accessed, and in the Breaching RDR, this field indicates which service was breached.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
DESTINATION_IP	UINT32	• SIP—Destination IP address of RTP flow. • Skype—Destination IP address of Skype flow. • MGCP—Destination IP address of RTP flow.
DESTINATION_PORT	UINT16	• SIP—Destination port of RTP flow. • Skype—Destination port of Skype flow. • MGCP—Destination port of RTP flow.
SOURCE_IP	UINT32	• SIP—Source IP address of RTP flow. • Skype—Source IP address of Skype flow. • MGCP—Source IP address of RTP flow.
SOURCE_PORT	UINT16	• SIP—Source port of RTP flow. • Skype—Source port of Skype flow. • MGCP—Source port of RTP flow.

Table 2-27 Media Flow RDR Fields (continued)

RDR Field Name	Type	Description
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides: • 0—Subscriber side • 1—Network side For Skype, this is the initiating side of the flow (not necessarily the initiating side of the voice call).
ZONE_ID	INT32	ID of the zone associated with this session.
FLAVOR_ID	INT32	ID of the protocol signatures with flavor associated with this session.
DOMAIN	STRING	• SIP—Domain name extracted from SIP header. • MGCP—Domain name extracted from MGCP header.
USER_AGENT	STRING	• SIP—User-Agent field extracted from SIP header. • MGCP—Not applicable.
START_TIME	UINT32	Flow start time.
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.
DURATION_SECONDS	INT32	• SIP—The active duration of the RTP flow, not including aging time. • Skype—The time between the start-of-call and end-of-call detection events. • MGCP—The active duration of the RTP flow, not including the aging time.
UPSTREAM_VOLUME	UINT32	• SIP—The upstream volume of the RTP flow, in bytes. • Skype—The upstream volume between the start-of-call and end-of-call detection events. • MGCP—The upstream volume of the RTP flow, in bytes.
DOWNSTREAM_VOLUME	UINT32	• SIP—The downstream volume of the RTP flow, in bytes. • Skype—The downstream volume between the start-of-call and end-of-call detection events. • MGCP—The downstream volume of the RTP flow, in bytes.
IP_PROTOCOL	UINT8	IP protocol type: • 6—TCP • 17—UDP

Table 2-27 Media Flow RDR Fields (continued)

RDR Field Name	Type	Description
FLOW_TYPE	INT8	• 0—All Skype flows • 1—Audio (SIP/MGCP) • 2—Video (SIP/MGCP)
SESSION_ID	UINT32	• SIP—The flow-context ID of the control flow. • Skype—The flow-context ID of the flow. • MGCP—The flow-context ID of the control flow.
UPSTREAM_JITTER	UINT32	• SIP—The average upstream jitter for the session, taken from the RTCP flow. N/A (0xFFFFFFFF) if RTCP flow is missing. • Skype—N/A (0xFFFFFFFF). • MGCP—The average upstream jitter for the session, taken from the RTCP flow. N/A (0xFFFFFFFF) if RTCP flow is missing.
DOWNSTREAM_JITTER	UINT32	• SIP—The average downstream jitter for the session, taken from the RTCP flow: N/A (0xFFFFFFFF) if RTCP flow is missing. • Skype—N/A (0xFFFFFFFF). • MGCP—The average downstream jitter for the session, taken from the RTCP flow. N/A (0xFFFFFFFF) if RTCP flow is missing.
UPSTREAM_PACKET_LOSS	UINT16	• SIP—The average fractional upstream packet loss for the session, taken from the RTCP flow. N/A (0xFFFF) if RTCP flow is missing. • Skype—N/A (0xFFFF). • MGCP—The average fractional upstream packet loss for the session, taken from the RTCP flow. N/A (0xFFFF) if RTCP flow is missing.
DOWNSTREAM_PACKET_ LOSS	UINT16	• SIPvThe average fractional downstream packet loss for the session, taken from the RTCP flow. N/A (0xFFFF) if RTCP flow is missing. • Skype—N/A (0xFFFF). • MGCP—The average fractional downstream packet loss for the session, taken from the RTCP flow. N/A (0xFFFF) if RTCP flow is missing.

Table 2-27 Media Flow RDR Fields (continued)

RDR Field Name	Type	Description
UPSTREAM_PAYLOAD_ TYPE	UINT8	• SIP—The upstream RTP payload type for the session. • Skype—N/A (0xFF). • MGCP—The upstream RTP payload type for the session.
DOWNSTREAM_PAYLOAD_ TYPE	UINT8	• SIP—The downstream RTP payload type for the session. • Skype—N/A (0xFF). • MGCP—The downstream RTP payload type for the session.

Note **Packet Loss Note**

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

Average Jitter

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

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

Attack Start RDR

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

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

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

Table 2-28 Attack Start RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
ATTACK_ID	UINT32	Unique attack ID.
ATTACKING_IP	UINT32	IP address related to the attack (for example: in a DDoS, this is the IP address under attack; in a scan this is the IP address of the source of the scan).
ATTACKED_IP	UINT32	Other IP address related to the attack, if one exists; otherwise, 0xFFFFFFFF.
ATTACKED_PORT	UINT16	Attacked port: 0xFFFF if not present.
ATTACKING_SIDE	INT8	Side of the SCE ATTACKING_IP on which it resides: • 0—Subscriber • 1—Network
IP_PROTOCOL	UINT8	IP protocol type.
ATTACK_TYPE	UINT32	ATTACKING_IP to whom it belongs: • 0—Attacked • 1—Attacker
GENERATOR_ID	INT8	Numeric value identifying the processor generating the RDR.
ATTACK_TIME	UINT32	Time since attack started in seconds.
REPORT_TIME	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.

Attack End RDR

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

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

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

Table 2-29 Attack End RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
ATTACK_ID	UINT32	Unique attack ID.
ATTACKING_IP	UINT32	IP address related to the attack (for example: in a DDoS, this will be the IP address under attack; in a scan this is the IP address of the source of the scan).
ATTACKED_IP	UINT32	Other IP address related to the attack, if one exists; otherwise, 0xFFFFFFFF.
ATTACKED_PORT	UINT16	Attacked port: 0xFFFF if not present.
ATTACKING_SIDE	INT8	Side of the SCE ATTACKING_IP on which it resides: • 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	UINT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.

Malicious Traffic Periodic RDR

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

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

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

Table 2-30 Malicious Traffic Periodic RDR Fields

RDR Field Name	Type	Description
ATTACK_ID	INT32	Unique attack ID.
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
ATTACK_IP	UINT32	IP address related to this attack.
OTHER_IP	UINT32	IP address other than the one displayed in ATTACK_IP. For example, in a DDoS, this is the IP address under attack; in a scan, this is the IP address of the source of the scan. If there is no attack, 0xFFFFFFFF is displayed.
PORT_NUMBER	UINT16	Displays the attacked port. If there is no attack, 0xFFFF is displayed.
ATTACK_TYPE	INT32	ATTACK_IP to whom it belongs: • 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	Configured period for periodic RDRs, in seconds, between successive RDRs.
DURATION	INT32	Indicates the number of seconds that have passed since the previous MALICIOUS_TRAFFIC_RDR.
END_TIME	INT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.

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

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

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

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

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

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

Spam RDR

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

The RDR tag of the SPAM_RDR is 4042322048.

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

Table 2-31 Spam RDR Fields

RDR Field Name	Type	Description
SUBSCRIBER_ID	STRING	Subscriber identification string, introduced through the subscriber management interfaces. It may contain up to 64 characters. For unknown subscribers this field may contain an empty string.
PACKAGE_ID	UINT16	ID of the Package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum_number_of_packages. The value maximum_number_of_packages is reserved for unknown subscribers.
SERVICE_ID	INT32	Service classification of the reported session. For example, in the Transaction RDR this field indicates which service was accessed, and in the Breaching RDR this field indicates which service was breached.
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.
CLIENT_IP	UINT32	IP address of the client side of the reported session. (The client side is defined as the initiator of the networking session.) The IP address is in a 32-bit binary format.
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field has the value zero.
SERVER_IP	UINT32	Destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
SERVER_PORT	UINT16	Destination port number of the TCP/UDP-based networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides. • 0—Subscriber side • 1—Network side
ACCESS_STRING	STRING	Layer 7 property, extracted from the transaction.
INFO_STRING	STRING	Layer 7 property extracted from the transaction.

Table 2-31 Spam RDR Fields (continued)

RDR Field Name	Type	Description
SPAM_FOUND	UINT8	Indicates whether spam was found (1) or stopped (0).
THRESHOLD_LEVEL	UINT16	Threshold level. Reserved for future use. Currently 0.
SESSION_COUNTER	UINT32	Number of sessions found.
TIME_INTERVAL	UINT32	Time that elapsed since the beginning of the period.
DEFINED_SESSION_ COUNTER	UINT32	Indicates the defined number of sessions.
DEFINED_TIME_INTERVAL	UINT32	Indicates the defined time interval.
REPORT_TIME	INT32	Ending time stamp of this RDR. The field is in UNIX time_t format, which is the number of seconds since midnight of 1 January 1970.

Related Topics

- Universal RDR Fields, page 2-2

Information About RDR Enumeration Fields

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

Block Reason (uint8)

The BLOCK_REASON field is a bit field. Table 2-32 lists the meanings of the bits of this field.

Table 2-32 Block Reason Field Bit Values

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

String Fields

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

Table 2-33 String Field Values

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

Table 2-33 String Field Values (continued)

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

Aggregation Period (uint8)

Table 2-34 lists the AGG_PERIOD field values.

Table 2-34 AGG_PERIOD Field Values

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

Flow Close Mode (uint8)

Table 2-35 lists the FLOW_CLOSE_MODE field values.

Table 2-35 Flow Close Mode Field Values

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

Time Frames (unint16)

Table 2-36 lists the TIME_FRAME field values.

Table 2-36 Time Frame Field Values

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

RDR Tag Assignment Summary

Table 2-37 summarizes RDR tag assignments.

Table 2-37 RDR Tag Assignments

RDR Name	Default Category (explained in the following table)	Tag Value (decimal)	Tag Value (hex)
SUBSCRIBER USAGE RDR (NUR)	CM-DB (1)	4,042,321,920	F0 F0 F0 00
REALTIME SUBSCRIBER USAGE RDR (SUR)	CM-DB (1)	4,042,321,922	F0 F0 F0 02
PACKAGE USAGE RDR	CM-DB (1)	4,042,321,924	F0 F0 F0 04
LINK USAGE RDR	CM-DB (1)	4,042,321,925	F0 F0 F0 05
ZONE USAGE RDR	CM-DB (1)	4,042,321,928	F0 F0 F0 08
VIRTUAL LINK RDR	CM-DB (1)	4,042,321,926	F0 F0 F0 06
TRANSACTION RDR	CM-DB (1)	4,042,321,936	F0 F0 F0 10
TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,000	F0 F0 F4 38
HTTP TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,004	F0 F0 F4 3C
RTSP TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,008	F0 F0 F4 40
VOIP TRANSACTION USAGE RDR	CM-CSV (1)	4,042,323,050	F0 F0 F4 6A
VIDEO TRANSACTION USAGE RDR	CM-CSV (1)	0xf0f0f480	4042323072
BLOCKING RDR	CM-CSV (1)	4,042,321,984	F0 F0 F0 40
QUOTA BREACH RDR	QP (4)	4,042,322,034	F0 F0 F0 72
QUOTA STATUS RDR	QP (4)	4,042,322,033	F0 F0 F0 71
QUOTA THRESHOLD RDR	QP (4)	4,042,322,035	F0 F0 F0 73
SESSION CREATION RDR	QP (4)	4,042,322,032	F0 F0 F0 70
RADIUS RDR	SM (3)	4,042,321,987	F0 F0 F0 43
DHCP RDR	SM (3)	4,042,321,986	F0 F0 F0 42
FLOW START RDR	RT (2)	4,042,321,942	F0 F0 F0 16
FLOW END RDR	RT (2)	4,042,321,944	F0 F0 F0 18
MEDIA FLOW RDR	CM-DB (1)	4,042,323,052	F0 F0 F4 6C

Table 2-37 RDR Tag Assignments (continued)

RDR Name	Default Category (explained in the following table)	Tag Value (decimal)	Tag Value (hex)
FLOW ONGOING RDR	RT (2)	4,042,321,943	F0 F0 F0 17
ATTACK_START RDR	RT (2)	4,042,321,945	F0 F0 F0 19
ATTACK_END RDR	RT (2)	4,042,321,946	F0 F0 F0 1A
MALICIOUS TRAFFIC RDR	DC-DB (1)	4,042,322,000	F0 F0 F0 50

RDR categories are the mechanism by which different types of RDRs can be sent to different collectors. You can configure the RDR categories using the SCE CLI. For more information, see the following relevant document:

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

Table 2-38 summarizes the RDR tag default categories.

Table 2-38 RDR Tag Default Categories

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

Periodic RDR Zero Adjustment Mechanism

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

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

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

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

Example 1

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

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

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

Example 2

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

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

CHAPTER 3

NetFlow Records: Formats and Field Contents

Revised: October 21, 2011, OL-21066-04

Introduction

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:

- Using the Generic Usage RDR to Report IPv6 Usage, page 2-29 (NUR)
- Package Usage RDR, page 2-41 (PUR)
- Link Usage RDR, page 2-36 (LUR)
- NetFlow, page 3-1
- NetFlow Field Types, page 3-2

NetFlow

- The Cisco Service Control Application for Broadband (SCA BB) supports NetFlow v9.
- For more information about NetFlow, see: RFC 3954.

NetFlow Field Types

Table 3-1 lists the possible fields in a NetFlow record and their descriptions.

Table 3-1 NetFlow Fields

Field Type	Value	Length (Bytes)	Description
scTag	32769	4	—
scTrafficProcessorId	32770	1	—
scSourceIpSample	32771	1	—
scDestinationIpSampl	32772	1	—
scFlowContextId	32773	4	—
scSubscriberId	32774	64	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.
scPackageId	32775	4	ID of the service configuration package/profile assigned to the subscriber.
scServiceId	32776	4	Service classification of the reported session.
scProtocolId	32777	2	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.
scSkipppedSessions	32778	4	Number of unreported sessions since the previous reporting record of this kind.
scInitiatingSide	32779	1	Initiating side of the transaction: • 0—Subscriber side • 1—Network side
scReportTime	32780	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.
scTransaction DurationMillisec	32781	4	Duration, in milliseconds, of the transaction reported in this reporting record.

Table 3-1 NetFlow Fields (continued)

Field Type	Value	Length (Bytes)	Description
scTimeFrame	32782	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.
scSessionUpstream Volume	32783	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.
scSessionDownstream Volume	32784	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.
scProtocolSignature	32785	4	ID of the protocol signature associated with this session.
scZoneId	32786	4	ID of the zone associated with this session.
scFlavorId	32787	4	ID of the protocol signatures with flavor associated with this session.
scFlowCloseMode	32788	1	Reason for the end of the flow.
scAccessString	32789	128, 256, 512, 1024	Layer 7 property, extracted from the transaction.
scInfoString	32790	128, 256, 512, 1024	Layer 7 property, extracted from the transaction.
scClientPort	32791	2	—
scServerPort	32792	2	—
scSubscriberCounterId	32793	2	—
scServiceUsageCounter Id	32794	2	—
scBreachState	32795	1	Indicates whether the subscriber's quota was breached: • 0—The quota was not breached • 1—The quota was breached
scReason	32796	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

Table 3-1 NetFlow Fields (continued)

Field Type	Value	Length (Bytes)	Description
scConfiguredDuration	32797	4	Configured period, in seconds, between successive reporting records.
scDuration	32798	4	Number of seconds that have passed since the previous reporting record of this type.
scEndTime	32799	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.
scUpstreamVolume	32800	4	Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period.
scDownstreamVolume	32801	4	Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period.
scSessions	32802	4	Aggregated number of sessions for the reported service, for the current reporting period.
scSeconds	32803	4	Aggregated number of session seconds for the reported service, for the current reporting period.
scPackageCounterId	32804	2	Counter to which each service is mapped. There are 64 package usage counters.
scGeneratorId	32805	1	Numeric value identifying the processor generating the reporting record.
scServiceGlobal CounterId	32806	2	Counter to which each service is mapped. There are 64 global usage counters.
scConcurrentSessions	32807	4	Concurrent number of sessions using the reported service when this reporting record was generated.
scActiveSubscribers	32808	4	Concurrent number of subscribers using the reported service when this reporting record was generated.
scTotalActive Subscribers	32809	4	Concurrent number of subscribers in the system when this reporting record was generated.
scLinkId	32810	1	Numeric value associated with the reported network link: • 0—Physical link 1 • 1—Physical link 2
	32811-32818	Reserved.	—
scAttackId	32819	4	Unique attack ID.
scAttackIp	32820	4	IP address related to this attack.

Table 3-1 NetFlow Fields (continued)

Field Type	Value	Length (Bytes)	Description
scAttackOtherIp	32821	4	Other IP address related to this attack if it exists, -1 otherwise.
scAttackPortNumber	32822	2	Port number related to this attack if one exists (if this is an IP scan, for example), -1 otherwise.
scAttackType	32823	4	Whom the AttackIp belongs to: • 0—Attacked • 1—Attacker
scAttackSide	32824	1	IP address side: • 0—Subscriber • 1—Network
scAttackIpProtoco	32825	1	IP protocol type: • 0—Other • 1—ICMP • 6—TCP • 17—UDP
scAttacks	32826	1	Number of attacks in the current reporting period. Since attack reports are generated per attack, the value is 0 or 1.
scAttackMalicious Sessions	32827	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.

CHAPTER 4

Database Tables: Formats and Field Contents

Revised: October 21, 2011, OL-21066-04

Introduction

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: Formats and Field Contents, page 2-1.

Database Tables Overview

Each RDR is routed to the appropriate adapter—the JDBC, Topper/Aggregator (TA), or Real-Time Aggregating (RAG) 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 RPT_NUR, RPT_SUR, RPT_PUR, RPT_LUR, and RPT_TR tables 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 list the Columns for Table RPT_NUR.

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 list the Columns for Table RPT_SUR.

Table 4-2 Columns for Table RPT_SUR

Field Name	Type
TIME_STAMP	Date_Time
RECORD_SOURCE	Number
SUBSCRIBER_ID	String
PACKAGE_ID	Number
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 list the Columns for Table RPT_PUR.

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 list the Columns for Table RPT_LUR.

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_GUR

Database table RPT_GUR stores data from GENERIC USAGE_RDRs.

These RDRs have the tag 4042322064.

Table 4-5 list the Columns for Table RPT_GUR.

Table 4-5 Columns for Table RPT_GUR

Field Name	Type
TIME_STAMP	Date_Time
RECORD_SOURCE	Number
GUR_TYPE	Number
LINK_ID	Number
GENERATOR_ID	Number
GLBL_USG_CNT_ID	Number
SUBS_USG_CNT_ID	Number
PKG_USG_CNT_ID	Number
SERVICE_ID	Number
SUBSCRIBER_ID	String
PACKAGE_ID	Number
PROTOCOL_ID	Number
SIGNATURE_ID	Number
PEER_IP	Number
PEER_PORT	Number
SOURCE_IP	Number
SOURCE_PORT	Number
INITIATING_SIDE	Number
ZONE_ID	Number
FLAVOR_ID	Number
SESSION_ID	Number
START_TIME	Number
END_TIME	Number
ACCESS_STRING	String
INFO_STRING	String
INT_KEY0	Number
INT_KEY1	Number
INT_KEY2	Number
INT_KEY3	Number
STR_KEY0	String
STR_KEY1	String

Table 4-5 Columns for Table RPT_GUR (continued)

Field Name	Type
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
TOTAL_VOLUME	Number
SESSIONS	Number
SECONDS	Number
CONCURRENT_SESSIONS	Number
ACTIVE_SUBSCRIBERS	Number
TOTAL_ACTIVE_SUBSCRIBERS	Number
CONFIGURED_DURATION	Number
DURATION	Number
DATA0	Number
DATA1	Number
DATA2	Number
DATA3	Number

Table RPT_TR

Database table RPT_TR stores data from TRANSACTION_RDRs.

These RDRs have the tag 4042321936.

Table 4-6 list the Columns for Table RPT_TR.

Table 4-6 Columns for Table RPT_TR

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
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
GLBL_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 4-7 list the Columns for Table RPT_MEDIA.

Table 4-7 Columns for Table RPT_MEDIA

Field Name	Type
TIME_STAMP	Date_Time
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
ZONE_ID	Number
FLAVOR_ID	Number
SIP_DOMAIN	String
SIP_USER_AGENT	String
MGCP_DOMAIN	String
MGCP_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 4-8 list the Columns for Table RPT_MALUR.

Table 4-8 Columns for Table RPT_MALUR

Field Name	Type
TIME_STAMP	Date_Time
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
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 4-9 list the Columns for Table RPT_TOPS_PERIOD0.

Table 4-9 Columns for Table RPT_TOPS_PERIOD0

Field Name	Type
RECORD_SOURCE	Number
METRIC_ID	Number
SUBS_USG_CNT_ID	Number
TIME_STAMP	Date_Time
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 is reported for the designated service/metric pair during the aggregation period, the DB is updated, but the only the final row in the report is updated to show a total consumption of zero. The DB is updated to avoid the perception in Cisco Service Control Application (SCA) Reporter that the report is not generated due to a malfunction.

Table 4-10 list the possible values for the field METRIC_ID.

Table 4-10 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 4-11 list the Columns for Table RPT_TOPS_PERIOD1.

Table 4-11 Columns for Table RPT_TOPS_PERIOD1

Field Name	Type
RECORD_SOURCE	Number
METRIC_ID	Number
SUBS_USG_CNT_ID	Number
TIME_STAMP	Date_Time
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 is still updated, but the only row in the report is 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.

Table 4-12 lists the possible values for the field METRIC_ID.

Table 4-12 Metric_ID Values

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

Table RPT_TOPS_PERIOD0_CUMULATIVE

The Topper/Aggregator (TA) Adapter generates database table RPT_TOPS_PERIOD0_CUMULATIVE for its shorter aggregation interval (by default, one hour).

Table 4-13 list the Columns for Table RPT_TOPS_PERIOD0_CUMULATIVE.

Table 4-13 Columns for Table RPT_TOPS_PERIOD0_CUMULATIVE

Field Name	Type
RECORD_SOURCE	Number
METRIC_ID	Number
SUBS_USG_CNT_ID	Number
TIME_STAMP	Date_Time
AGG_PERIOD	Number
SUBSCRIBERS	Number
CONSUMPTION	Number
TOTAL_SUBSCRIBERS	Number
TOTAL_CONSUMPTION	Number
LAST_SUBS_CONSUMPTION	Number

Table 4-14 list the possible values for the field METRIC_ID.

Table 4-14 Metric_ID Values

Metric_ID	Metric
0	Up Volume
1	Down Volume

Table RPT_TOPS_PERIOD1_CUMULATIVE

The Topper/Aggregator (TA) Adapter generates database table RPT_TOPS_PERIOD1_CUMULATIVE for its longer aggregation interval (by default, one day).

Table 4-15 list the Columns for Table RPT_TOPS_PERIOD1_CUMULATIVE.

Table 4-15 Columns for Table RPT_TOPS_PERIOD1_CUMULATIVE

Field Name	Type
RECORD_SOURCE	Number
METRIC_ID	Number
SUBS_USG_CNT_ID	Number
TIME_STAMP	Date_Time
AGG_PERIOD	Number
SUBSCRIBERS	Number
CONSUMPTION	Number
TOTAL_SUBSCRIBERS	Number
TOTAL_CONSUMPTION	Number
LAST_SUBS_CONSUMPTION	Number

Table 4-16 lists the possible values for the field METRIC_ID.

Table 4-16 Metric_ID Values

Metric_ID	Metric
0	Up Volume
1	Down Volume

Table RPT_TOPS_PEAK_PERIOD

The Topper/Aggregator (TA) Adapter generates database table RPT_TOPS_PEAK_PERIOD for the configured period in the peak_hours section in taadapter.conf.

Table 4-17 lists the columns for the RPT_TOPS_PEAK_PERIOD table.

Table 4-17 Columns for Table RPT_TOPS_PEAK_PERIOD

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

Table 4-18 lists the possible values for the METRIC_ID field.

Table 4-18 Metric_ID Values

Metric_ID	Metric
0	Up Volume
1	Down Volume

Table RPT_TOPS_PEAK_CUMULATIVE

The Topper/Aggregator (TA) Adapter generates database table RPT_TOPS_PEAK_CUMULATIVE for the configured period in the peak_hours section in taadapter.conf.

Table 4-19 lists the columns for the RPT_TOPS_PEAK_CUMULATIVE table.

Table 4-19 Columns for Table RPT_TOPS_PEAK_CUMULATIVE

Field Name	Type
RECORD_SOURCE	Number
METRIC_ID	Number
SUBS_USG_CNT_ID	Number
TIME_STAMP	Date_Time
AGG_PERIOD	Number
SUBSCRIBERS	Number
CONSUMPTION	Number
TOTAL_SUBSCRIBERS	Number
TOTAL_CONSUMPTION	Number
LAST_SUBS_CONSUMPTION	Number

Table 4-20 lists the possible values for the METRIC_ID field.

Table 4-20 METRIC_ID Values

Metric_ID	Metric
0	Up Volume
1	Down Volume

Table RPT_VLUR

Database table RPT_VLUR stores data from VIRTUAL_LINKS_USAGE_RDRs.

These RDRs have the tag 4042321926.

Table 4-21 lists the columns for the RPT_VLUR table.

Table 4-21 Columns for Table RPT_VLUR

Name	Type
TIME_STAMP	Date_Time
RECORD_SOURCE	Number
VLINK_ID	Number
VLINK_DIRECTION	Number
GENERATOR_ID	Number
SRVC_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 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 4-22 lists the columns for the INI_VALUES table.

Table 4-22 Columns for Table INI_VALUES

Field Name	Type	Description
TIME_STAMP	Date_Time	
SE_IP	String	Identification of the SCE platform where these values were applied.
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 global service usage counter ID / counter 3001—P2P global service usage counter ID / counter 3002—VoIP subscriber service usage counter ID / counter
VALUE_KEY	String	Key name. For example: Gold, Silver, or Adult Browsing.
VALUE	Number	Numeric reference.

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 4-23 lists the columns for table VLINK_INI.

Table 4-23 Columns for Table VLINK_INI

Field Name	Type	Description
TIME_STAMP	Date_Time	
SE_IP	String	Identification of the SCE platform where these values were applied
VLINK_ID	UINT16	Virtual link ID
VLINK_DIRECTION	INT8	Virtual link direction
VLINK_NAME	String	Virtual link name
CHANNEL_ID	UINT16	Channel ID
CHANNEL_NAME	String	Name of the channel
CMTS_NAME	String	Name of the CMTS

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 4-24 lists the columns for table CONF_SE_TZ_OFFSET.

Table 4-24 Columns for Table CONF_SE_TZ_OFFSET

Field Name	Type
TIME_STAMP	Date_Time
OFFSET_MIN	Number

Table RPT_TOP_APN

The Real-Time Aggregating (RAG) Adapter generates database table RPT_TOP_APN for the configured aggregation interval (1 hour by default) as configured in vsa_SURs.xml.

Table 4-25 lists the columns for the RPT_TOP_APN table.

Table 4-25 Columns for Table RPT_TOP_APN

Field Name	Type
TIME_STAMP	Date_Time
AGG_PERIOD	Number
APN	String
SERVICE_USAGE_COUNTER_ID	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
RANK_VOLUME	Number

At the end of the each aggregation period, the Collection Manager inserts the aggregated records into the table. Rank is a sequential numerical value that indicates the top entries. Rank of VSA attributes are based on the usage (downstream, upstream) of particular services.

From the RPT_TOP_APN table, you can generate reports such as usage per APN and application usage per APN.

Table RPT_TOP_DEVICE_TYPE

The Real-Time Aggregating (RAG) Adapter generates database table RPT_TOP_DEVICE_TYPE for the configured aggregation interval (1 hour by default) as configured in vsa_SURs.xml.

Table 4-26 lists the columns for the RPT_TOP_DEVICE_TYPE table.

Table 4-26 Columns for Table RPT_TOP_DEVICE_TYPE

Field Name	Type
TIME_STAMP	Date_Time
AGG_PERIOD	Number
IMEI_TAC	String
SERVICE_USAGE_COUNTER_ID	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
UNIQ_SUBS	Number
RANK_VOLUME	Number
RANK_UNIQ_SUBS	Number

At the end of the each aggregation period, the Collection Manager inserts the aggregated records into the table. Rank is a sequential numerical value that indicates the top entries. RANK_VOLUME is derived based on the usage (downstream, upstream) of particular services. RANK_UNIQ_SUBS is derived based on the total unique subscribers on a particular service.

From the RPT_TOP_DEVICE_TYPE table, we can generate reports such as device type distribution (IMEI), usage per device, and application usage per device.

Table RPT_TOP_NETWORK_TYPE

The Real-Time Aggregating (RAG) Adapter generates database table RPT_TOP_NETWORK_TYPE for the configured aggregation interval (1 hour by default) as configured in vsa_SURs.xml.

Table 4-27 lists the columns for the RPT_TOP_NETWORK_TYPE table.

Table 4-27 Columns for Table RPT_TOP_NETWORK_TYPE

Field Name	Type
TIME_STAMP	Date_Time
AGG_PERIOD	Number
NETWORK_TYPE	String
SERVICE_USAGE_COUNTER_ID	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
RANK_VOLUME	Number

At the end of the each aggregation period, the Collection Manager inserts the aggregated records into the table. Rank is a sequential numerical value that indicates the top entries. RANK_VOLUME is derived based on the usage (downstream, upstream) of particular services.

From the RPT_TOP_NETWORK_TYPE table, we can generate reports such as usage per network type and application usage per network type.

Table RPT_TOP_SGSN

The Real-Time Aggregating (RAG) Adapter generates database table RPT_TOP_SGSN for the configured aggregation interval (1 hour by default) as configured in vsa_SURs.xml.

Table 4-28 lists the columns for the RPT_TOP_SGSN table.

Table 4-28 Columns for Table RPT_TOP_SGSN

Field Name	Type
TIME_STAMP	Date_Time
AGG_PERIOD	Number
SGSN	String
SERVICE_USAGE_COUNTER_ID	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
RANK_VOLUME	Number

At the end of the each aggregation period, the Collection Manager inserts the aggregated records into the table. Rank is a sequential numerical value that indicates the top entries. RANK_VOLUME is derived based on the usage (downstream, upstream) of particular services.

From the RPT_TOP_SGSN table, you can generate a usage per SGSN report.

Table RPT_TOP_USER_LOCATION

The Real-Time Aggregating (RAG) Adapter generates database table RPT_TOP_USER_LOCATION for the configured aggregation interval (1 hour by default) as configured in vsa_SURs.xml.

Table 4-29 lists the columns for the RPT_TOP_USER_LOCATION table.

Table 4-29 Columns for Table RPT_TOP_USER_LOCATION

Field Name	Type
TIME_STAMP	Date_Time
AGG_PERIOD	Number
USER_LOCATION	String
SERVICE_USAGE_COUNTER_ID	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
UNIQ_SUBS	Number
RANK_VOLUME	Number

At the end of the each aggregation period, the Collection Manager inserts the aggregated records into the table. Rank is a sequential numerical value that indicates the top entries. RANK_VOLUME is derived based on the usage (downstream, upstream) of particular services.

From the RPT_TOP_USER_LOCATION table, you can generate reports such as number of subscribers per location and usage per location.

Table RPT_DVLINK

The Real-Time Aggregating (RAG) Adapter generates database table RPT_DVLINK. It aggregates the subscriber usage RDR data. Aggregation is based on per package and per VLINK (DOWN VLINK). You can generate a report.

Table 4-30 lists the columns for the RPT_DVLINK table.

Table 4-30 Columns for Table RPT_DVLINK

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
UP_VLINK_ID	Number
DOWN_VLINK_ID	Number

Table RPT_UVLINK

The Real-Time Aggregating (RAG) Adapter generates database table RPT_UVLINK. It aggregates the subscriber usage RDR data. Aggregation is based on per package and per VLINK (UP VLINK). You can generate a report.

Table 4-31 lists the columns for the RPT_UVLINK table.

Table 4-31 Columns for Table RPT_UVLINK

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
UP_VLINK_ID	Number
DOWN_VLINK_ID	Number

Table RPT_TOP_HTTP_DOMAINS

The Real-Time Aggregating (RAG) Adapter generates database table RPT_TOP_HTTP_DOMAINS for the configured aggregation interval (1 hour by default) as configured in http_TURs.xml. It aggregates the HTTP transaction usage RDR data. Aggregation is based on domain, service, and package. You can generate reports.

Table 4-32 lists the columns for the RPT_TOP_HTTP_DOMAINS table.

Table 4-32 Columns for Table RPT_TOP_HTTP_DOMAINS

Field Name	Type
TIME_STAMP	Date_Time
AGG_PERIOD	Number
DOMAIN	String
SERVICE_ID	Number
PACKAGE_ID	Number
SESSIONS	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
DURATION	Number
UNIQ_SUBS	Number
RANK_VOLUME	Number
RANK_SESSIONS	Number
RANK_UNIQ_SUBS	Number

At the end of the each aggregation period, the Collection Manager inserts the aggregated records into the table. Rank is a sequential numerical value that indicates the top entries. RANK_VOLUME is derived based on the usage (downstream, upstream), package, and service. RANK_SESSIONS is derived based on the total sessions, package, and service. RANK_UNIQ_SUBS is derived based on the total number of unique subscribers, package, and service.

Table RPT_TOP_HTTP_HOSTS

The Real-Time Aggregating (RAG) Adapter generates database table RPT_TOP_HTTP_HOSTS for the configured aggregation interval (1 hour by default) as configured in http_TURs.xml. It aggregates the HTTP transaction usage RDR data. Aggregation is based on domain, service, and package. You can generate reports.

Table 4-33 lists the columns for the RPT_TOP_HTTP_HOSTS table.

Table 4-33 Columns for Table RPT_TOP_HTTP_HOSTS

Field Name	Type
TIME_STAMP	Date_Time
AGG_PERIOD	Number
HOST	String
SERVICE_ID	Number
PACKAGE_ID	Number
SESSIONS	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
DURATION	Number
UNIQ_SUBS	Number
RANK_VOLUME	Number
RANK_SESSIONS	Number
RANK_UNIQ_SUBS	Number
DOMAIN	String

At the end of the each aggregation period, the Collection Manager inserts the aggregated records into the table. Rank is a sequential numerical value that indicates the top entries. RANK_VOLUME is derived based on the usage (downstream, upstream), package, and service. RANK_SESSIONS is derived based on the total sessions, package, and service. RANK_UNIQ_SUBS is derived based on the total number of unique subscribers, package, and service.

Table RPT_TOP_VIDEO_DOMAINS

The Real-Time Aggregating (RAG) Adapter generates database table RPT_TOP_VIDEO_DOMAINS for the configured aggregation interval (1 hour by default) as configured in video_TURs.xml. It aggregates the video transaction usage RDR data. Aggregation is based on domain, service, and package. You can generate reports.

Table 4-34 lists the columns for the RPT_TOP_VIDEO_DOMAINS table.

Table 4-34 Columns for Table RPT_TOP_VIDEO_DOMAINS

Field Name	Type
TIME_STAMP	Date_Time
AGG_PERIOD	Number
DOMAIN	String
SERVICE_ID	Number
PACKAGE_ID	Number
SESSIONS	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
DURATION	Number
UNIQ_SUBS	Number
RANK_VOLUME	Number
RANK_SESSIONS	Number
RANK_UNIQ_SUBS	Number

At the end of the each aggregation period, the Collection Manager inserts the aggregated records into the table. Rank is a sequential numerical value that indicates the top entries. RANK_VOLUME is derived based on the usage (downstream, upstream), package, and service. RANK_SESSIONS is derived based on the total sessions, package, and service. RANK_UNIQ_SUBS is derived based on the total number of unique subscribers, package, and service.

Table RPT_TOP_VIDEO_HOSTS

The Real-Time Aggregating (RAG) Adapter generates database table RPT_TOP_VIDEO_HOSTS for the configured aggregation interval (1 hour by default) as configured in video_TURs.xml. It aggregates the video transaction usage RDR data. Aggregation is based on domain, service, and package. You can generate reports.

Table 4-35 lists the columns for the RPT_TOP_VIDEO_HOSTS table.

Table 4-35 Columns for Table RPT_TOP_VIDEO_HOSTS

Field Name	Type
TIME_STAMP	Date_Time
AGG_PERIOD	Number
HOST	String
SERVICE_ID	Number
PACKAGE_ID	Number
SESSIONS	Number
UPSTREAM_VOLUME	Number
DOWNSTREAM_VOLUME	Number
DURATION	Number
UNIQ_SUBS	Number
RANK_VOLUME	Number
RANK_SESSIONS	Number
RANK_UNIQ_SUBS	Number
DOMAIN	String

At the end of the each aggregation period, the Collection Manager inserts the aggregated records into the table. Rank is a sequential numerical value that indicates the top entries. RANK_VOLUME is derived based on the usage (downstream, upstream), package, and service. RANK_SESSIONS is derived based on the total sessions, package, and service. RANK_UNIQ_SUBS is derived based on the total number of unique subscribers, package, and service.

Table RPT_ZUR

The RPT_ZUR database table stores data from ZONE_USAGE_RDRs.

These RDRs have the tag 4042321928.

Table 4-36 lists the columns for the RPT_ZUR table.

Table 4-36 Columns for Table RPT_ZUR

Field Name	Type
ZONE_COUNTER_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_SPAM

The RPT_SPAM database table stores data from SPAM_RDRs.

These RDRs have the tag 4042322048.

Table 4-37 lists the columns for the RPT_SPAM table.

Table 4-37 Columns for Table RPT_SPAM

Field Name	Type
SUBSCRIBER_ID	String
PACKAGE_ID	Number
SERVICE_ID	Number
PROTOCOL_ID	Number
CLIENT_IP	Number
CLIENT_PORT	Number
SERVER_IP	Number
SERVER_PORT	Number
INITIATING_SIDE	Number
ACCESS_STRING	String
INFO_STRING	String
SPAM_FOUND	Number
THRESHOLD_LEVEL	Number
SESSION_COUNTER	Number
TIME_INTERVAL	Number
DEFINED_SESSION_COUNTER	Number
DEFINED_TIME_INTERVAL	Number
REPORT_TIME	Number

Table RPT_FUR

These RDRs have the tag 4042321927.

Table 4-38 lists the columns for the RPT_FUR table.

Table 4-38 Columns for Table RPT_FUR

Field Name	Type
SUBSCRIBER_ID	String
PACKAGE_ID	Number
SERVICE_ID	Number
PROTOCOL_ID	Number
REASON	Number
SERVER_IP	Number
SERVER_PORT	Number
ACCESS_STRING	String
INFO_STRING	String
CLIENT_IP	Number
CLIENT_PORT	Number
INITIATING_SIDE	Number
END_TIME	Number
MILLISEC_DURATION	Number
TIME_FRAME	Number
SESSION_UPSTREAM_VOLUME	Number
SESSION_DOWNSTREAM_VOLUME	Number
SUBSCRIBER_COUNTER_ID	Number
GLOBAL_COUNTER_ID	Number
PACKAGE_COUNTER_ID	Number
IP_PROTOCOL	Number
PROTOCOL_SIGNATURE	Number
ZONE_ID	Number
FLAVOR_ID	Number
FLOW_CLOSE_MODE	Number
FLOW_ID	Number
SESSION_ID	Number

Table IMEI_DEVICETYPE

The IMEI_DEVICETYPE database table contains mappings for the Device types with IMEI_TAC. The Reporter uses the IMEI_TAC column for the Device Type Reports.

Table 4-39 lists the columns for the IMEI_DEVICETYPE table.

Table 4-39 Columns for Table IMEI_DEVICETYPE

Field Name	Type
TIME_STAMP	Timestamp
IMEI_TAC	String
DEVICE_TYPE	String

CHAPTER 5

CSV File Formats

Revised: October 21, 2011, OL-21066-04

Introduction

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

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 the "Managing Service Configurations" section in the "Using the Service Configuration Editor" chapter of *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 is 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

Two formats—Standard and Easy—are available for Zone CSV files.

Standard Format

Lines in Zone CSV files in Standard Format 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 in Standard Format:

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

Easy Format

Lines in Zone CSV files in Easy Format have only Zone items.

The following is an example of a Zone CSV file in Easy Format:

```
1.0.0.0/32
1.0.0.1/32
1.0.0.2/32
```

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
```


For information on flavors, see the “Managing Flavors” section in the “Using the Service Configuration Editor: Traffic Classification” chapter of *Cisco Service Control Application for Broadband User Guide*.

HTTP URL CSV Files

Two formats—Standard and Easy—are available for HTTP URL CSV files.

Standard Format

Lines in HTTP URL CSV files in Standard format 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 in Standard Format:

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

Easy Format

Lines in HTTP URL CSV files in Easy format have a single URL format.

The following is an example of an HTTP URL CSV file in Easy format:

```
http://*.rapidshare.com/cgi-bin/upload*
http://*.rapidshare.com/files*
```

HTTP Referer CSV Files

Two formats—Standard and Easy—are available for HTTP Referer CSV files.

Standard Format

Lines in HTTP Referer CSV files in Standard Format 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 Referer CSV file in Standard Format:

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

Easy Format

Lines in HTTP Referer CSV files in Easy format have a single URL format.

The following is an example of an HTTP Referer CSV file in Easy format:

```
http://*.rapidshare.com/cgi-bin/upload*
http://*.rapidshare.com/files*
```

HTTP User Agent CSV Files

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

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

HTTP Composite CSV Files

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

```
flavor name,flavor index,flavor type,HTTP_URL_name,HTTP_User_Agent_name
```

where HTTP_URL_name and HTTP_User_Agent_name are the names of existing flavors of types HTTP URL and HTTP User Agent respectively.

RTSP User Agent CSV Files

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

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

RTSP Host Name CSV Files

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

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

RTSP Composite CSV Files

Lines in RTSP 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 CSV Files

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

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

SIP Source Domain CSV Files

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

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

SIP Composite CSV Files

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 CSV Files

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

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

ToS CSV Files

Lines in ToS CSV files have the following fixed format:

```
flavor name,flavor index,flavor type,ToS value
```

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 the following relevant document:

- "Subscriber Files" section in the "Managing Subscribers" chapter of *Cisco SCE8000 10GBE Software Configuration Guide*.
- "Subscriber Files" section in the "Managing Subscribers" chapter of *Cisco SCE8000 GBE Software Configuration Guide*.

See also *Cisco Service Control Management Suite Subscriber Manager User Guide*.

- Import/Export File: Format of the mappings Field, page 5-6
- SCE Subscriber CSV Files, page 5-7
- SCMS SM Subscriber CSV Files, page 5-7
- SCE Anonymous Group CSV Files, page 5-7
- SCE Subscriber Template CSV File, page 5-7

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 (";"):

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

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

The following are examples of the mappings field:

- 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 uses 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:

```
group1,176.23.34.0/24,3
group2,10.7.0.0/16
```

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 199.

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 either *Cisco SCE8000 10GBE Software Configuration Guide* or *Cisco SCE8000 GBE 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 *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, page 5-8
- TA Adapter CSV Files, page 5-8
- RAG Adapter CSV Files, page 5-9

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. It is possible to include the SE IP (for example, record source) which generated the RDR in the csv line. To turn this option on, edit the file *csvadapter.conf* and set the value of *includeRecordSource* property under [csvadapter] section to *true*.

Related Topics

- Raw Data Records: Formats and Field Contents, page 2-1

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:

```
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 *taadapter.conf* and set the value of the *write_headers* property under [csv] section to *true*.

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 CSV repository of the adapter.

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 has the 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 has a value corresponding to the time stamp of the last RDR in the bucket.

CHAPTER 6

SCA BB Proprietary MIB Reference

Revised: October 21, 2011, OL-21066-04

Introduction

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

An 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 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 sends SNMP traps.

Note You can enable or disable specific traps.

Related Information

For more information about SNMP configuration, see the following relevant document:

- "Configuring and Managing the SNMP Interface" section in "Configuring the Management Interface and Security" chapter of *Cisco SCE8000 10GBE Software Configuration Guide*.
- "Configuring and Managing the SNMP Interface" section in the "Configuring the Management Interface and Security" chapter of *Cisco SCE8000 GBE Software Configuration Guide*.

Required MIB Files

To access the SNMP variables on the SCE platform, you must load the SNMP browser with a standard MIB file (*SNMPv2.mib*) and proprietary Cisco MIB files (*pcube.mib*, *pcubeSEMib.mib*, and *PCubeEngageMib.mib*).

Note You can download the CISCO-SCAS-BB MIB file (*PCubeEngageMib.mib*) and other MIB files (*pcube.mib* and *pcubeSEMib.mib*) from ftp://ftp.cisco.com/pub/mibs/.

The Order to Load the MIB Files

The SCA BB proprietary MIB uses definitions that are defined in other MIBs, such as *SNMPv2.mib* and *pcube.mib*.

This means that the order in which the MIBs are loaded is important; to avoid errors, the MIBs must be loaded in the correct order.

Load the MIBs in the following order:

1. *SNMPv2.mib*
2. *pcube.mib*
3. *pcubeSEMib.mib*
4. *PCubeEngageMib.mib*

Information About the Service Control Enterprise MIB

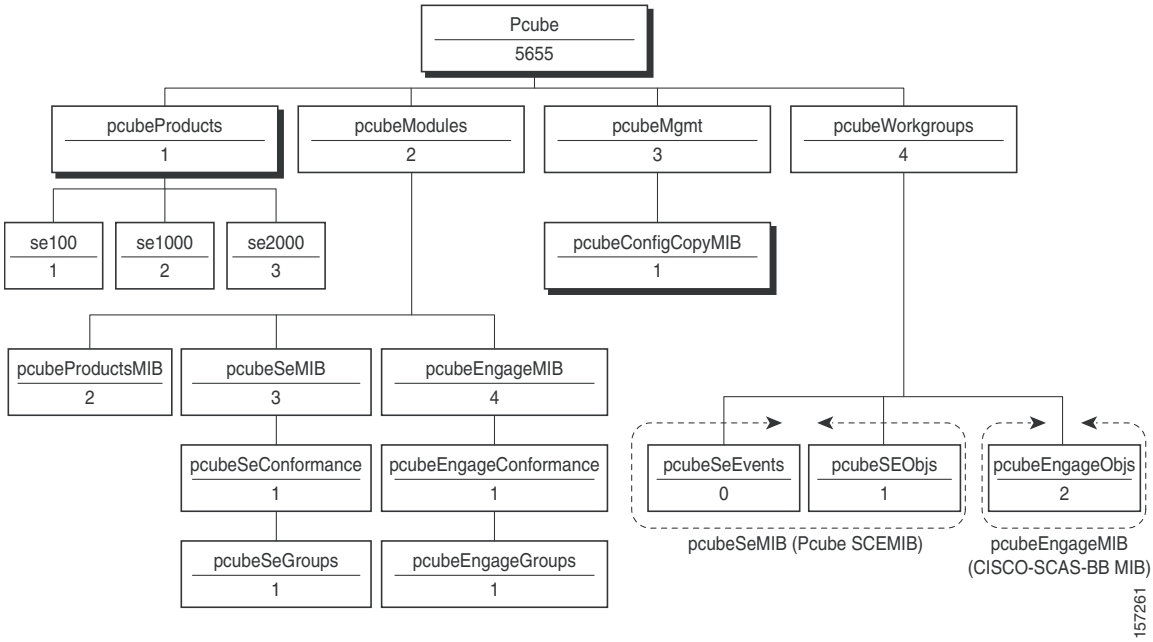
The Service Control Enterprise MIB includes four main groups: Products, Modules, Management, and Workgroup. The Service Control enterprise tree structure is defined in a MIB file named *pcube.mib*.

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

  Service Control product sysObjectIDs are defined in a MIB file named *Pcube-Products-MIB*.
- pcubeModules subtree provides a root object identifier from which MIB modules are defined.
- pcubeMgmt subtree contains the configuration copy MIB:
  - *pcubeConfigCopyMib* enables saving the running configuration of Cisco products. This MIB is documented in the "Cisco Service Control MIBs" appendix of *Cisco SCE8000 10GBE Software Configuration Guide* or in the "Cisco Service Control MIBs" appendix of *Cisco SCE8000 GBE Software Configuration Guide*.
- 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 "Cisco Service Control MIBs" appendix of *Cisco SCE8000 10GBE Software Configuration Guide* or in the "Cisco Service Control MIBs" appendix of *Cisco SCE8000 GBE Software Configuration Guide*.
  - *pcubeEngageObjs*—CISCO-SCAS-BB MIB provides configuration and runtime status for SCA BB, and is described in the following section.

Figure 6-1 illustrates the Service Control Enterprise MIB structure.

Figure 6-1 Service Control Enterprise MIB Structure



Note The following object identifier represents the Service Control Enterprise MIB: *1.3.6.1.4.1.5655* or *so.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 chapter.

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

Table 6-1 lists the pcubeEngageObjs objects. Each object consists of a number of subordinate object types, which are summarized in the following section.

Table 6-1 pcubeEngageObjs Objects

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
           linkServiceUsageTable
               linkServiceUsageEntry
                   linkServiceUsageUpVolume
                   linkServiceUsageDownVolume
                   linkServiceUsageNumSessions
                   linkServiceUsageDuration
                   linkServiceUsageConcurrentSessions
                   linkServiceUsageActiveSubscribers
                   linkServiceUpDroppedPackets
                   linkServiceDownDroppedPackets
                   linkServiceUpDroppedBytes
                   linkServiceDownDroppedBytes
           packageGrp
           packageCounterTable
               packageCounterEntry
                   packageCounterIndex
                   packageCounterStatus
                   packageCounterName
                   packageCounterActiveSubscribers
                   packageServiceUsageTable
                   packageServiceUsageEntry
                   packageServiceUsageUpVolume
                   packageServiceUsageDownVolume
                   packageServiceUsageNumSessions
                   packageServiceUsageDuration
                   packageServiceUsageConcurrentSessions
                   packageServiceUsageActiveSubscribers
                   packageServiceUpDroppedPackets
                   packageServiceDownDroppedPackets
                   packageServiceUpDroppedBytes
                   packageServiceDownDroppedBytes
           subscriberGrp
           subscribersTable
               subscriberEntry
                   subscriberPackageIndex
                   subscriberServiceUsageTable
```

```
            subscriberServiceUsageEntry
            subscriberServiceUsageUpVolume
            subscriberServiceUsageDownVolume
            subscriberServiceUsageNumSessions
            subscriberServiceUsageDuration
    serviceCounterGrp
        globalScopeServiceCounterTable
            globalScopeServiceCounterEntry
            globalScopeServiceCounterIndex
            globalScopeServiceCounterStatus
            globalScopeServiceCounterName
    subscriberScopeServiceCounterTable
        subscriberScopeServiceCounterEntry
            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{
linkServiceUsageUpVolume
linkServiceUsageDownVolume
linkServiceUsageNumSessions
linkServiceUsageDuration
linkServiceUsageConcurrentSessions
linkServiceUsageActiveSubscribers
linkServiceUpDroppedPackets
linkServiceDownDroppedPackets
linkServiceUpDroppedBytes
linkServiceDownDroppedBytes
}
```

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
```

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.

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
Units	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 either the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 10GBE Software Configuration Guide* or the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 GBE 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 either the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 10GBE Software Configuration Guide* or the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 GBE 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 either the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 10GBE Software Configuration Guide* or the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 GBE 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 either the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 10GBE Software Configuration Guide* or the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 GBE 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).

packageCounterTable (packageGrp 1)

The Package Counter table provides information for each package usage counter.

Access	non-accessible

Syntax

```
SEQUENCE OF packageCounterEntry
```

packageCounterEntry (packageCounterTable 1)

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

Access	non-accessible

Index

```
{pmoduleIndex, packageCounterIndex}
```

Syntax

```
SEQUENCE {
packageCounterIndex
packageCounterStatus
packageCounterName
packageCounterActiveSubscribers
}
```

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	non-accessible

Index

```
{pmoduleIndex, packageCounterIndex, globalScopeServiceCounterIndex}
```

Syntax

```
SEQUENCE {
packageServiceUsageUpVolume
packageServiceUsageDownVolume
packageServiceUsageNumSessions
packageServiceUsageDuration
packageServiceUsageConcurrentSessions
packageServiceUsageActiveSubscribers
packageServiceUpDroppedPackets
packageServiceDownDroppedPackets
packageServiceUpDroppedBytes
packageServiceDownDroppedBytes
}
```

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
```

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.

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 either the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 10GBE Software Configuration Guide* or the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 GBE Software Configuration Guide*.

packageServiceDownDroppedPackets (packageServiceUsageEntry 8)

The number of dropped downstream packets of packages in this package usagecounter.

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 either the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 10GBE Software Configuration Guide* or the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 GBE 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 either the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 10GBE Software Configuration Guide* or the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 GBE 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 either the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 10GBE Software Configuration Guide* or the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 GBE 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 **For the SCE8000:** To use the tables in this group, first create an entry to reference a particular subscriber in the cServiceControlSubscribersTable object of the CISCO-SERVICE-CONTROL-SUBSCRIBERS MIB (not the CISCO-SCAS-BB MIB). Using the index of this table (cServiceControlSubscribersIndex), information about the subscriber can be collected.

Note **For the SCE2000:** 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.

Related Topics

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 {
subscriberServiceUsageUpVolume
subscriberServiceUsageDownVolume
subscriberServiceUsageNumSessions
subscriberServiceUsageDuration
```

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 {
subscriberServiceUsageUpVolume
subscriberServiceUsageDownVolume
subscriberServiceUsageNumSessions
subscriberServiceUsageDuration
```

subscriberServiceUsageDownVolume (subscriberServiceUsageEntry 2)

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

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.

subscriberServiceUsageNumSessions (subscriberServiceUsageEntry 3)

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

Access	read-only
Units	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	non-accesible

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 {
globalScopeServiceCounterIndex
globalScopeServiceCounterStatus
globalScopeServiceCounterName
}
```

globalScopeServiceCounterIndex (globalScopeServiceCounterEntry 1)

The global-scope service usage counter index.

Access	not-accesible

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 {
subscriberScopeServiceCounterIndex
subscriberScopeServiceCounterStatus
subscriberScopeServiceCounterName
}
```

subscriberScopeServiceCounterIndex (subscriberScopeServiceCounterEntry 1)

The subscriber-scope service usage counter index.

Access	not-accesible

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
```

Guidelines 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.

- globalScopeServiceCounterTable and subscriberScopeServiceCounterTable, page 6-25
- packageCounterTable, page 6-25
- Accessing Subscriber Information (the spvIndex), page 6-25

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 is 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. For more information about the accelerate-packet-drops feature, see either the “Counting Dropped Packets”

section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 10GBE Software Configuration Guide* or the "Counting Dropped Packets" section in the "Configuring the Line Interface" chapter of *Cisco SCE8000 GBE 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 *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 Classification" chapter of *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 proper subscriber MIB table and associate this entry with a subscriber name. Its index can be then be referred to in order to collect usage statistics for this subscriber.

The exact MIB objects vary, depending on the particular SCE platform, as described in the following sections.

Accessing Subscriber Information in the Cisco SCE 2000

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 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 In order to create a subscriber entry, you must specify the index of the module and the desired *spvIndex* .

- Set pmoduleIndex = 1.
- In SEMib:subscriberGrp:subscriberPropertiesValuesTable, choose an index for spvIndex (for this example arbitrarily choose index 7).

Step 3 Create an entry in SEMib:subscriberGrp:subscriberPropertiesValuesTable, at the index that you have chosen:

- Set *spvRowStatus* to 4 (using CreateAndGo).
- Set *spvSubName* to “sub123”.

Step 4 Read the subscriberServiceDownVolume property out of EngageMib:subscriberGrp:subscriberServiceTable Step 6 where *spvIndex* is set to 7 and *serviceIndex* is set to 10. (In general, you may walk the cServiceControlSubscribersTable in order to find out the various subscriber indexes, but in this case we have chosen it to be 7 so we can directly use the same index for accessing the data of this subscriber).

Accessing Subscriber Information in the Cisco SCE8000

Create an entry in the cServiceControlSubscribersTable of the CISCO-SERVICE-CONTROL-SUBSCRIBERS MIB. After an entry in this table is created and associated with a subscriber name, its index (cServiceControlSubscribersIndex) can be referred to in PCubeEngageMib (as spvIndex) to collect usage statistics for this subscriber.

An entry is created in the cServiceControlSubscribersTable table (at an index chosen by the user) by setting the entry cServiceControlSubscribersRowStatus object with CreateAndGo(4) then setting the name of the subscriber in the cServiceControlSubscribersName property.

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 In order to create a subscriber entry, you must specify the indexes of the module and the desired cServiceControlSubscribersIndex.

- Set entPhyIndex according to the index of the Service Control Module (SCM) entry in the Entity MIB.
- Choose an index for cServiceControlSubscribersIndex (for this example arbitrarily choose index 7).

Step 3 Create an entry in ciscoServiceControlSubscribersMIB:cServiceControlSubscribersTable, at the index that you have chosen:

- Set cServiceControlSubscribersRowStatus to 4 (using CreateAndGo).
- Set cServiceControlSubscribersName to "sub123".

Step 4 Read the subscriberServiceUsageDownVolume property out of EngageMib:subscriberGrp:subscriberServiceUsageTable where entPhyIndex is set as instructed, spvIndex is set to 7 and serviceIndex is set to 10. (In general, you may walk the cServiceControlSubscribersTable in order to find out the various subscriber indexes, but in this case we have chosen it to be 7 so we can directly use the same index for accessing the data of this subscriber).