# Cisco Service Control Application for Broadband

# Reference Guide

Version 3.0
OL-8410-01

**Corporate 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: 408 526-4100

Customer Order Number: OL-8410-01
Text Part Number: OL-8410-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: The equipment described in this manual generates and may radiate radio-frequency energy. If it is not installed in accordance with Cisco's installation instructions, it may cause interference with radio and television reception. This equipment has been tested and found to comply with the limits for a Class B digital device in accordance with the specifications in part 15 of the FCC rules. These specifications are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation.

Modifying the equipment without Cisco's written authorization may result in the equipment no longer complying with FCC requirements for Class A or Class B digital devices. In that event, your right to use the equipment may be limited by FCC regulations, and you may be required to correct any interference to radio or television communications at your own expense.

You can determine whether your equipment is causing interference by turning it off. If the interference stops, it was probably caused by the Cisco equipment or one of its peripheral devices. If the equipment causes interference to radio or television reception, try to correct the interference by using one or more of the following measures:

- Turn the television or radio antenna until the interference stops.
- Move the equipment to one side or the other of the television or radio.
- Move the equipment farther away from the television or radio.
- Plug the equipment into an outlet that is on a different circuit from the television or radio. (That is, make certain the equipment and the television or radio are on circuits controlled by different circuit breakers or fuses.)

Modifications to this product not authorized by Cisco Systems, Inc. could void the FCC approval and negate your authority to operate the product.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILTY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.



Printed in the USA on recycled paper containing 10% postconsumer waste.

***Cisco SCA BB*** *Reference Guide*

# CONTENTS

# Preface

This preface describes who should read the *Cisco Service Control Application for Broadband User 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.

## Document Revision History

| Cisco Service Control Release | Part Number | Publication Date |
|---|---|---|
| Release 3.0.0 | OL-8410-01 | December, 2005 |

### DESCRIPTION OF CHANGES

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

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

## Audience

This guide is intended to provide information about the data structures created and used by ***SCA BB*** for:

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

# Document Content

This guide is organized as follows:

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

# Related Publications

The following publications are available for the ***Cisco Service Control Application for Broadband***:

- *Cisco Service Control Application for Broadband User Guide*
- *Cisco SCA BB Service Configuration API Programmer's Guide*
- *Cisco Service Control Management Suite Collection Manager User Guide*
- *Cisco Service Control Management Suite Subscriber Manager User Guide*
- *Cisco Service Control Application Suite Reporter User Guide*
- *Cisco Service Control Engine Software Configuration Guide*

# Conventions

This document uses the following conventions:

| Convention | Description |
|---|---|
| **boldface** font | Commands and keywords are in **boldface**. |
| *italic* font | Arguments for which you supply values are in i*talics*. |
| [ ] | Elements in square brackets are optional. |
| {x \| y \| z} | 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 will include the quotation marks. |
| `screen` font | Terminal sessions and information the system displays are in `screen` font. |
| **`boldface screen`** font | Information you must enter is in **`boldface screen`** font. |
| *`italic screen`* font | Arguments for which you supply values are in *`italic screen`* font. |
| → | This pointer highlights an important line of text in an example. |
| ^ | The symbol ^ represents the key labeled **Control**—for example, the key combination **^D** in a screen display means hold down the **Control** key while you press the **D** key. |
| < > | Non printing 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. |

Notes use the following conventions:

**Note** Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

Cautions use the following conventions:

**Caution** Means *reader be careful*. You are capable of doing something that might result in equipment damage or loss of data.

Warnings use the following conventions:

**Warning** Means *reader be warned*. You are capable of doing something that might result in bodily injury.

# Obtaining Documentation

The following sections provide sources for obtaining documentation from Cisco Systems.

## World Wide Web

You can access the most current Cisco documentation on the World Wide Web at the following sites:

- *http://www.cisco.com*
- *http://www-china.cisco.com*
- *http://www-europe.cisco.com*

## Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or as an annual subscription.

## Ordering Documentation

Cisco documentation is available in the following ways:

- Registered Cisco Direct Customers can order Cisco Product documentation from the networking Products MarketPlace:

  *http://www.cisco.com/cgi-bin/order/order_root.pl*

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  *http://www.cisco.com/pcgi-bin/marketplace/welcome.pl*

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, in North America, by calling 800 553-NETS(6387).

## Documentation Feedback

If you are reading Cisco product documentation on the World Wide Web, you can submit technical comments electronically. Click **Feedback** in the toolbar and select **Documentation**. After you complete the form, click **Submit** to send it to Cisco.

You can email your comments to bug-doc@cisco.com.

To submit your comments by mail, use the response card behind the front cover of your document, or write to the following address:

Attn Document Resource Connection

Cisco Systems, Inc.

170 West Tasman Drive

San Jose, CA 95134-9883

We appreciate your comments.

# Obtaining Technical Assistance

Cisco provides *Cisco.com* (on page ix) as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools. For Cisco.com registered users, additional troubleshooting tools are available from the TAC website.

## Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information and resources at anytime, from anywhere in the world. This highly integrated Internet application is a powerful, easy-to-use tool for doing business with Cisco.

Cisco.com provides a broad range of features and services to help customers and partners streamline business processes and improve productivity. Through Cisco.com, you can find information about Cisco and our networking solutions, services, and programs. In addition, you can resolve technical issues with online technical support, download and test software packages, and order Cisco learning materials and merchandise. Valuable online skill assessment, training, and certification programs are also available.

Customers and partners can self-register on Cisco.com to obtain additional personalized information and services. Registered users can order products, check on the status of an order, access technical support, and view benefits specific to their relationships with Cisco.

To access Cisco.com, go to the following website:

http://www.cisco.com

## Technical Assistance Center

The Cisco TAC website is available to all customers who need technical assistance with a Cisco product or technology that is under warranty or covered by a maintenance contract.

## Contacting TAC by Using the Cisco TAC Website

If you have a priority level 3 (P3) or priority level 4 (P4) problem, contact TAC by going to the TAC website:

*http://www.cisco.com/tac*

P3 and P4 level problems are defined as follows:

- P3—Your network is degraded. Network functionality is noticeably impaired, but most business operations continue.
- P4—You need information or assistance on Cisco product capabilities, product installation, or basic product configuration.

In each of the above cases, use the Cisco TAC website to quickly find answers to your questions.

To register for *Cisco.com* (on page ix), go to the following website:

*http://tools.cisco.com/RPF/register/register.do*

If you cannot resolve your technical issue by using the TAC online resources, Cisco.com registered users can open a case online by using the TAC Case Open tool at the following website:

*http://www.cisco.com/tac/caseopen*

## Contacting TAC by Telephone

If you have a priority level 1 (P1) or priority level 2 (P2) problem, contact TAC by telephone and immediately open a case. To obtain a directory of toll-free numbers for your country, go to the following website:

*http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml*

P1 and P2 level problems are defined as follows:

- P1—Your production network is down, causing a critical impact to business operations if service is not restored quickly. No workaround is available.
- P2—Your production network is severely degraded, affecting significant aspects of your business operations. No workaround is available.

CHAPTER 1

# Default Service Configuration Reference Tables

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

This chapter contains the following sections:



## Filter Rules

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

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

**Table 1-1 Filter Rules**

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

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

| Flow Filter Name | Default State | Description |
|---|---|---|
| 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 - 1646, packets bypass the policy engine and are mapped to CoS BE |

# Protocols

Protocols are divided into four groups:

- Generic Protocols—Generic IP, Genetic TCP, and Generic UDP protocols. These protocols are used for transactions that were not mapped to a service by one of the more specific protocol types.
- Signature-Based Protocols—Protocols classified according to a Layer 7 application signature. This group includes the most common protocols, such as HTTP and FTP, as well as a large group of popular P2P protocols.
- IP Protocols—Non-TCP/UDP protocols (such as ICMP), 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.

## Generic Protocols

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

A transaction is classified as belonging to one of the generic protocols if:

- It was not classified as belonging to a signature-based protocol

  **and**

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

Table 1-2 Generic Protocols

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

## Signature-Based Protocols

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

Table 1-3 Signature-Based Protocols

| Protocol Name | ID | TCP Ports | UDP Ports |
|---|---|---|---|
| DHCP Sniff | 33 | | |
| FTP | 4 | 21 | |
| HTTP Browsing | 2 | 80, 8080 | |
| IRC | 62 | | |
| MMS | 6 | 1755 | |
| NNTP | 15 | 119 | |
| POP3 | 9 | 110 | |
| PTT Winphoria | 61 | | |
| RTP | 57 | | |
| RTSP Streaming | 5 | 554, 1554, 7070 | |
| SMTP | 8 | 25 | |
| dns | 933 | | |

| Protocol Name | ID | TCP Ports | UDP Ports |
|---|---|---|---|
| https | 358 | | |
| imap | 59 | 143 | 143 |
| tftp | 60 | 69 | 69 |
| yahoo messenger | 40 | 5000-5001 | 5000-5001 |

**Table 1-4 Signature-Based P2P Protocols**

| Protocol Name | ID | TCP Ports | UDP Ports |
|---|---|---|---|
| BitTorrent | 24 | 6881-6889, 6969 | |
| DirectConnect | 19 | 411-413 | |
| FastTrack KaZaA File Transfer | 14 | | |
| FastTrack KaZaA Networking | 13 | 1214 | |
| Filetopia | 31 | | |
| Gnutella File Transfer | 12 | | |
| Gnutella Networking | 11 | 6346-6349 | |
| Hotline | 20 | 5498, 5500-5503 | |
| Manolito | 22 | | |
| Mute | 34 | | |
| Napster | 32 | | |
| NeoNet | 37 | | |
| NodeZilla | 35 | | |
| Share | 27 | | |
| Soulseek | 29 | | |
| Warez | 39 | | |
| Waste | 36 | | |
| WinMX/OpenNap | 16 | 6257, 6699 | 6257 |
| Winny | 17 | 7742-7745, 7773 | |
| eDonkey | 18 | 4661-4665, 4672-4673, 4711, 5662, 5773, 5783 | 4661-4665, 4672-4673, 4711, 5662, 5773, 5783 |
| ITunes | 30 | | |

**Table 1-5 Signature-Based Protocols**

| Protocol Name | ID | TCP Ports | UDP Ports |
|---|---|---|---|
| Dingo Tel | 42 | | |
| H323 | 28 | 1720 | |
| MGCP | 38 | | 2427, 2727 |
| SIP | 23 | 5060-5061 | 5060-5061 |
| Skinny | 41 | | |
| Skype | 25 | 33033 | |

## IP Protocols

This section lists the IP protocols supported by ***SCA BB***.

**Table 1-6 IP Protocols**

| IP Protocol Number | Protocol Name | Protocol ID |
|---|---|---|
| 0 | HOPOPT | 756 |
| 1 | ICMP | 757 |
| 2 | IGMP | 758 |
| 3 | GGP | 759 |
| 4 | IP | 760 |
| 5 | ST | 761 |
| 6 | Generic TCP | 0 |
| 7 | CBT | 762 |
| 8 | EGP | 763 |
| 9 | IGP | 764 |
| 10 | BBN-RCC-MON | 765 |
| 11 | NVP-II | 766 |
| 12 | PUP | 767 |
| 13 | ARGUS | 768 |
| 14 | EMCON | 769 |
| 15 | XNET | 770 |
| 16 | CHAOS | 771 |
| 17 | Generic UDP | 1 |
| 18 | MUX | 772 |
| 19 | DCN-MEAS | 773 |
| 20 | HMP | 774 |

| IP Protocol Number | Protocol Name | Protocol ID |
|---|---|---|
| 21 | PRM | 775 |
| 22 | XNS-IDP | 776 |
| 23 | TRUNK-1 | 777 |
| 24 | TRUNK-2 | 778 |
| 25 | LEAF-1 | 779 |
| 26 | LEAF-2 | 780 |
| 27 | RDP | 781 |
| 28 | IRTP | 782 |
| 29 | ISO-TP4 | 783 |
| 30 | NETBLT | 784 |
| 31 | MFE-NSP | 785 |
| 32 | MERIT-INP | 786 |
| 33 | SEP | 787 |
| 34 | 3PC | 788 |
| 35 | IDPR | 789 |
| 36 | XTP | 790 |
| 37 | DDP | 791 |
| 38 | IDPR-CMTP | 792 |
| 39 | TP++ | 793 |
| 40 | IL | 794 |
| 41 | IPv6-Over-IPv4 | 795 |
| 42 | SDRP | 796 |
| 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 |

| IP Protocol Number | Protocol Name | Protocol ID |
|---|---|---|
| 56 | TLSP | 810 |
| 57 | SKIP | 811 |
| 58 | IPv6-ICMP | 812 |
| 59 | IPv6-NoNxt | 813 |
| 60 | IPv6-Opts | 814 |
| 61 | Any host internal protocol | 815 |
| 62 | CFTP | 816 |
| 63 | Any local network | 817 |
| 64 | SAT-EXPAK | 818 |
| 65 | KRYPTOLAN | 819 |
| 66 | RVD | 820 |
| 67 | IPPC | 821 |
| 68 | Any distributed file system | 822 |
| 69 | SAT-MON | 823 |
| 70 | VISA | 824 |
| 71 | IPCV | 825 |
| 72 | CPNX | 826 |
| 73 | CPHB | 827 |
| 74 | WSN | 828 |
| 75 | PVP | 829 |
| 76 | BR-SAT-MON | 830 |
| 77 | SUN-ND | 831 |
| 78 | WB-MON | 832 |
| 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 |

| IP Protocol Number | Protocol Name | Protocol ID |
|---|---|---|
| 91 | LARP | 845 |
| 92 | MTP | 846 |
| 93 | AX.25 | 847 |
| 94 | IPIP | 848 |
| 95 | MICP | 849 |
| 96 | SCC-SP | 850 |
| 97 | ETHERIP | 851 |
| 98 | ENCAP | 852 |
| 99 | Any private encryption scheme | 853 |
| 100 | GMTP | 854 |
| 101 | IFMP | 855 |
| 102 | PNNI | 856 |
| 103 | PIM | 857 |
| 104 | ARIS | 858 |
| 105 | SCPS | 859 |
| 106 | QNX | 860 |
| 107 | A/N | 861 |
| 108 | IPComp | 862 |
| 109 | SNP | 863 |
| 110 | Compaq-Peer | 864 |
| 111 | IPX-in-IP | 865 |
| 112 | VRRP | 866 |
| 113 | PGM | 867 |
| 114 | Any 0-hop protocol | 868 |
| 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 |

| IP Protocol Number | Protocol Name | Protocol ID |
|---|---|---|
| 125 | FIRE | 879 |
| 126 | CRTP | 880 |

## Port-Based Protocols

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

**Table 1-7 Port-Based Protocols**

| Protocol Name | ID | TCP Ports | UDP Ports |
|---|---|---|---|
| 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 |
| 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 |
| 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 |
| ntp | 992 | 123 | 123 |
| ansatrader | 993 | 124 | 124 |
| 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 |
| cisco-tna | 1000 | 131 | 131 |
| cisco-sys | 1001 | 132 | 132 |

| | | | |
|---|---|---|---|
| statsrv | 1002 | 133 | 133 |
| ingres-net | 1003 | 134 | 134 |
| 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 |
| pcmail-srv | 151 | 158 | 158 |
| nss-routing | 152 | 159 | 159 |
| sgmp-traps | 153 | 160 | 160 |
| snmp | 154 | 161 | 161 |
| snmptrap | 155 | 162 | 162 |
| cmip-man | 156 | 163 | 163 |
| 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 |
| ocbinder | 176 | 183 | 183 |
| 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 | 194 |
| 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 |
| 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 |
| subntbcst_tftp | 223 | 247 | 247 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| dantz | 411 | 497 | 497 |
| siam | 412 | 498 | 498 |
| iso-ill | 413 | 499 | 499 |
| isakmp | 414 | 500 | 500 |
| stmf | 415 | 501 | 501 |

| | | | |
|---|---|---|---|
| asa-appl-proto | 416 | 502 | 502 |
| intrinsa | 417 | 503 | 503 |
| citadel | 418 | 504 | 504 |
| mailbox-lm | 419 | 505 | 505 |
| ohimsrv | 420 | 506 | 506 |
| crs | 421 | 507 | 507 |
| xvttp | 422 | 508 | 508 |
| snare | 423 | 509 | 509 |
| fcp | 424 | 510 | 510 |
| passgo | 425 | 511 | 511 |
| exec | 426 | 512 | |
| biff | 427 | | 512 |
| login | 428 | 513 | |
| who | 429 | | 513 |
| shell | 430 | 514 | |
| syslog | 431 | | 514 |
| printer | 432 | 515 | 515 |
| videotex | 433 | 516 | 516 |
| talk | 434 | 517 | 517 |
| ntalk | 435 | 518 | 518 |
| utime | 436 | 519 | 519 |
| efs | 437 | 520 | |
| router | 438 | | 520 |
| ripng | 439 | 521 | 521 |
| ulp | 440 | 522 | 522 |
| ibm-db2 | 441 | 523 | 523 |
| ncp | 442 | 524 | 524 |
| timed | 443 | 525 | 525 |
| tempo | 444 | 526 | 526 |
| stx | 445 | 527 | 527 |
| custix | 446 | 528 | 528 |
| irc-serv | 447 | 529 | 529 |
| courier | 448 | 530 | 530 |
| conference | 449 | 531 | 531 |
| netnews | 450 | 532 | 532 |
| netwall | 451 | 533 | 533 |

| | | | |
|---|---|---|---|
| mm-admin | 452 | 534 | 534 |
| iiop | 453 | 535 | 535 |
| opalis-rdv | 454 | 536 | 536 |
| nmsp | 455 | 537 | 537 |
| gdomap | 456 | 538 | 538 |
| apertus-ldp | 457 | 539 | 539 |
| uucp | 458 | 540 | 540 |
| uucp-rlogin | 459 | 541 | 541 |
| commerce | 460 | 542 | 542 |
| klogin | 461 | 543 | 543 |
| kshell | 462 | 544 | 544 |
| appleqtcsrvr | 463 | 545 | 545 |
| dhcpv6-client | 464 | 546 | 546 |
| dhcpv6-server | 465 | 547 | 547 |
| idfp | 466 | 549 | 549 |
| new-rwho | 467 | 550 | 550 |
| cybercash | 468 | 551 | 551 |
| deviceshare | 469 | 552 | 552 |
| pirp | 470 | 553 | 553 |
| remotefs | 471 | 556 | 556 |
| openvms-sysipc | 472 | 557 | 557 |
| sdnskmp | 473 | 558 | 558 |
| teedtap | 474 | 559 | 559 |
| rmonitor | 475 | 560 | 560 |
| monitor | 476 | 561 | 561 |
| chshell | 477 | 562 | 562 |
| nntps | 478 | 563 | 563 |
| 9pfs | 479 | 564 | 564 |
| whoami | 480 | 565 | 565 |
| streettalk | 481 | 566 | 566 |
| banyan-rpc | 482 | 567 | 567 |
| ms-shuttle | 483 | 568 | 568 |
| ms-rome | 484 | 569 | 569 |
| meter | 485 | 570-571 | 570-571 |
| sonar | 486 | 572 | 572 |
| banyan-vip | 487 | 573 | 573 |

| | | | |
|---|---|---|---|
| ftp-agent | 488 | 574 | 574 |
| vemmi | 489 | 575 | 575 |
| ipcd | 490 | 576 | 576 |
| vnas | 491 | 577 | 577 |
| ipdd | 492 | 578 | 578 |
| decbsrv | 493 | 579 | 579 |
| sntp-heartbeat | 494 | 580 | 580 |
| bdp | 495 | 581 | 581 |
| scc-security | 496 | 582 | 582 |
| philips-vc | 497 | 583 | 583 |
| keyserver | 498 | 584 | 584 |
| imap4-ssl | 499 | 585 | 585 |
| password-chg | 500 | 586 | 586 |
| submission | 501 | 587 | 587 |
| cal | 502 | 588 | 588 |
| eyelink | 503 | 589 | 589 |
| tns-cml | 504 | 590 | 590 |
| http-alt | 505 | 591 | 591 |
| eudora-set | 506 | 592 | 592 |
| http-rpc-epmap | 507 | 593 | 593 |
| tpip | 508 | 594 | 594 |
| cab-protocol | 509 | 595 | 595 |
| smsd | 510 | 596 | 596 |
| ptcnameservice | 511 | 597 | 597 |
| sco-websrvrmg3 | 512 | 598 | 598 |
| acp | 513 | 599 | 599 |
| ipcserver | 514 | 600 | 600 |
| urm | 515 | 606 | 606 |
| nqs | 516 | 607 | 607 |
| sift-uft | 517 | 608 | 608 |
| npmp-trap | 518 | 609 | 609 |
| npmp-local | 519 | 610 | 610 |
| npmp-gui | 520 | 611 | 611 |
| hmmp-ind | 521 | 612 | 612 |
| hmmp-op | 522 | 613 | 613 |
| sshell | 523 | 614 | 614 |

| | | | |
|---|---|---|---|
| sco-inetmgr | 524 | 615 | 615 |
| sco-sysmgr | 525 | 616 | 616 |
| sco-dtmgr | 526 | 617 | 617 |
| dei-icda | 527 | 618 | 618 |
| digital-evm | 528 | 619 | 619 |
| sco-websrvrmgr | 529 | 620 | 620 |
| escp-ip | 530 | 621 | 621 |
| collaborator | 531 | 622 | 622 |
| aux_bus_shunt | 532 | 623 | 623 |
| cryptoadmin | 533 | 624 | 624 |
| dec_dlm | 534 | 625 | 625 |
| asia | 535 | 626 | 626 |
| passgo-tivoli | 536 | 627 | 627 |
| qmqp | 537 | 628 | 628 |
| 3com-amp3 | 538 | 629 | 629 |
| rda | 539 | 630 | 630 |
| ipp | 540 | 631 | 631 |
| bmpp | 541 | 632 | 632 |
| servstat | 542 | 633 | 633 |
| ginad | 543 | 634 | 634 |
| rlzdbase | 544 | 635 | 635 |
| ldaps | 545 | 636 | 636 |
| lanserver | 546 | 637 | 637 |
| mcns-sec | 547 | 638 | 638 |
| msdp | 548 | 639 | 639 |
| entrust-sps | 549 | 640 | 640 |
| repcmd | 550 | 641 | 641 |
| esro-emsdp | 551 | 642 | 642 |
| sanity | 552 | 643 | 643 |
| dwr | 553 | 644 | 644 |
| pssc | 554 | 645 | 645 |
| ldp | 555 | 646 | 646 |
| dhcp-failover | 556 | 647 | 647 |
| rrp | 557 | 648 | 648 |
| aminet | 558 | 649 | 649 |
| obex | 559 | 650 | 650 |

| | | | |
|---|---|---|---|
| ieee-mms | 560 | 651 | 651 |
| hello-port | 561 | 652 | 652 |
| repscmd | 562 | 653 | 653 |
| aodv | 563 | 654 | 654 |
| tinc | 564 | 655 | 655 |
| spmp | 565 | 656 | 656 |
| rmc | 566 | 657 | 657 |
| tenfold | 567 | 658 | 658 |
| mac-srvr-admin | 568 | 660 | 660 |
| hap | 569 | 661 | 661 |
| pftp | 570 | 662 | 662 |
| purenoise | 571 | 663 | 663 |
| secure-aux-bus | 572 | 664 | 664 |
| sun-dr | 573 | 665 | 665 |
| doom | 574 | 666 | 666 |
| disclose | 575 | 667 | 667 |
| mecomm | 576 | 668 | 668 |
| meregister | 577 | 669 | 669 |
| vacdsm-sws | 578 | 670 | 670 |
| vacdsm-app | 579 | 671 | 671 |
| vpps-qua | 580 | 672 | 672 |
| cimplex | 581 | 673 | 673 |
| acap | 582 | 674 | 674 |
| dctp | 583 | 675 | 675 |
| vpps-via | 584 | 676 | 676 |
| vpp | 585 | 677 | 677 |
| ggf-ncp | 586 | 678 | 678 |
| mrm | 587 | 679 | 679 |
| entrust-aaas | 588 | 680 | 680 |
| entrust-aams | 589 | 681 | 681 |
| xfr | 590 | 682 | 682 |
| corba-iiop | 591 | 683 | 683 |
| corba-iiop-ssl | 592 | 684 | 684 |
| mdc-portmapper | 593 | 685 | 685 |
| hcp-wismar | 594 | 686 | 686 |
| asipregistry | 595 | 687 | 687 |

| | | | |
|---|---|---|---|
| realm-rusd | 596 | 688 | 688 |
| nmap | 597 | 689 | 689 |
| vatp | 598 | 690 | 690 |
| msexch-routing | 599 | 691 | 691 |
| hyperwave-isp | 600 | 692 | 692 |
| connendp | 601 | 693 | 693 |
| ha-cluster | 602 | 694 | 694 |
| ieee-mms-ssl | 603 | 695 | 695 |
| rushd | 604 | 696 | 696 |
| uuidgen | 605 | 697 | 697 |
| olsr | 606 | 698 | 698 |
| accessnetwork | 607 | 699 | 699 |
| elcsd | 608 | 704 | 704 |
| agentx | 609 | 705 | 705 |
| silc | 610 | 706 | 706 |
| borland-dsj | 611 | 707 | 707 |
| entrust-kmsh | 612 | 709 | 709 |
| entrust-ash | 613 | 710 | 710 |
| cisco-tdp | 614 | 711 | 711 |
| netviewdm1 | 615 | 729 | 729 |
| netviewdm2 | 616 | 730 | 730 |
| netviewdm3 | 617 | 731 | 731 |
| netgw | 618 | 741 | 741 |
| netrcs | 619 | 742 | 742 |
| flexlm | 620 | 744 | 744 |
| fujitsu-dev | 621 | 747 | 747 |
| ris-cm | 622 | 748 | 748 |
| kerberos-adm | 623 | 749 | 749 |
| rfile | 624 | 750 | |
| kerberos-iv | 625 | | 750 |
| pump | 626 | 751 | 751 |
| qrh | 627 | 752 | 752 |
| rrh | 628 | 753 | 753 |
| tell | 629 | 754 | 754 |
| nlogin | 630 | 758 | 758 |
| con | 631 | 759 | 759 |

| | | | |
|---|---|---|---|
| ns | 632 | 760 | 760 |
| rxe | 633 | 761 | 761 |
| quotad | 634 | 762 | 762 |
| cycleserv | 635 | 763 | 763 |
| omserv | 636 | 764 | 764 |
| webster | 637 | 765 | 765 |
| phonebook | 638 | 767 | 767 |
| vid | 639 | 769 | 769 |
| cadlock | 640 | 770 | 770 |
| rtip | 641 | 771 | 771 |
| cycleserv2 | 642 | 772 | 772 |
| submit | 643 | 773 | |
| notify | 644 | | 773 |
| rpasswd | 645 | 774 | |
| acmaint_dbd | 646 | | 774 |
| entomb | 647 | 775 | |
| acmaint_transd | 648 | | 775 |
| wpages | 649 | 776 | 776 |
| multiling-http | 650 | 777 | 777 |
| wpgs | 651 | 780 | 780 |
| concert | 652 | 786 | 786 |
| qsc | 653 | | 787 |
| mdbs_daemon | 654 | 800 | 800 |
| device | 655 | 801 | 801 |
| itm-mcell-s | 656 | 828 | 828 |
| pkix-3-ca-ra | 657 | 829 | 829 |
| dhcp-failover2 | 658 | 847 | 847 |
| rsync | 659 | 873 | 873 |
| iclcnet-locate | 660 | 886 | 886 |
| iclcnet_svinfo | 661 | 887 | 887 |
| accessbuilder | 662 | 888 | 888 |
| omginitialrefs | 663 | 900 | 900 |
| smpnameres | 664 | 901 | 901 |
| ideafarm-chat | 665 | 902 | 902 |
| ideafarm-catch | 666 | 903 | 903 |
| xact-backup | 667 | 911 | 911 |

| | | | |
|---|---|---|---|
| ftps-data | 668 | 989 | 989 |
| ftps | 669 | 990 | 990 |
| nas | 670 | 991 | 991 |
| telnets | 671 | 992 | 992 |
| imaps | 672 | 993 | 993 |
| ircs | 673 | 994 | 994 |
| pop3s | 674 | 995 | 995 |
| vsinet | 675 | 996 | 996 |
| maitrd | 676 | 997 | 997 |
| busboy | 677 | 998 | |
| puparp | 678 | | 998 |
| garcon | 679 | 999 | |
| applix | 680 | | 999 |
| surf | 681 | 1010 | 1010 |
| rmiactivation | 682 | 1098 | 1098 |
| rmiregistry | 683 | 1099 | 1099 |
| GLT Poliane | 882 | 1201 | |
| ms-sql-s | 684 | 1433 | 1433 |
| ms-sql-m | 685 | 1434 | 1434 |
| oracle | 690 | 1521 | 1521 |
| orasrv | 691 | 1525 | 1525 |
| tlisrv | 692 | 1527 | 1527 |
| coauthor | 693 | 1529 | 1529 |
| micromuse-lm | 702 | 1534 | 1534 |
| orbixd | 703 | 1570 | 1570 |
| rdb-dbs-disp | 694 | 1571 | 1571 |
| oraclenames | 695 | 1575 | 1575 |
| shockwave | 707 | 1626 | 1626 |
| oraclenet8cman | 696 | 1630 | 1630 |
| l2tp | 742 | 1701 | 1701 |
| pptp | 739 | 1723 | 1723 |
| radius | 738 | 1812-1813 | 1812-1813 |
| net8-cman | 697 | 1830 | 1830 |
| msnp | 713 | 1836 | 1836 |
| MSN Messenger | 883 | 1863 | 1863 |
| gtp-user | 740 | 2152 | 2152 |

| | | | |
|---|---|---|---|
| kali | 718 | 2213 | 2213 |
| directplay | 716 | 2234 | 2234 |
| ms-olap | 686 | 2382-2383, 2393-2394 | 2382-2383, 2393-2394 |
| groove | 715 | 2492 | 2492 |
| citrixima | 698 | 2512 | 2512 |
| citrixadmin | 699 | 2513 | 2513 |
| worldfusion | 719 | 2595-2596 | 2595-2596 |
| citriximaclient | 701 | 2598 | 2598 |
| sitaraserver | 708 | 2629 | 2629 |
| sitaramgmt | 709 | 2630 | 2630 |
| sitaradir | 710 | 2631 | 2631 |
| wta-wsp-s | 724 | 2805 | 2805 |
| citrix-rtmp | 700 | 2897 | 2897 |
| wap-push | 725 | 2948 | 2948 |
| wap-pushsecure | 726 | 2949 | 2949 |
| xbox live | 898 | 3074 | 3074 |
| orbix-locator | 704 | 3075 | 3075 |
| orbix-config | 705 | 3076 | 3076 |
| orbix-loc-ssl | 706 | 3077 | 3077 |
| xdtp | 741 | 3088 | 3088 |
| msft-gc | 687 | 3268 | 3268 |
| msft-gc-ssl | 688 | 3269 | 3269 |
| net-assistant | 712 | 3283 | 3283 |
| mysql | 711 | 3306 | 3306 |
| directv-web | 720 | 3334 | 3334 |
| directv-soft | 721 | 3335 | 3335 |
| directv-tick | 722 | 3336 | 3336 |
| directv-catlg | 723 | 3337 | 3337 |
| ms-term-services | 689 | 3389 | 3389 |
| PeerEnabler | 881 | 3531 | 3531 |
| wap-push-http | 727 | 4035 | 4035 |
| wap-push-https | 728 | 4036 | 4036 |
| aim | 714 | 5190-5193 | |
| directplay8 | 717 | 6073 | 6073 |
| fsgs | 743 | 6112 | 6112 |
| game-spy | 755 | 6500, 28900, 29000 | 6515, 27900 |

| | | | |
|---|---|---|---|
| parsec-game | 744 | 6582 | 6582 |
| ibprotocol | 737 | 6714 | 6714 |
| UnReal_UT | 745 | 7778 | 7778 |
| 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 |
| ps2 | 899 | 10070-10080 | 10070 |
| SiN | 746 | 22450 | 22450 |
| halflife | 747 | | 27015 |
| quake-server | 754 | 27960 | 27910, 27960 |
| tribes | 748 | 28001 | 28001 |
| heretic2 | 749 | 28910 | |
| starsiege | 750 | | 29001-29009 |
| game-search | 751 | 29001 | |
| KingPin | 752 | 31510 | 31510 |
| runescape | 753 | 43594 | |

# Services

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

The following table lists the services defined in the default service configuration. Both counters used to accumulate information on transactions classified to the service have the same name.

**Table 1-8 Installed Services**

| Name | ID | Name of Parent Service | Global Counter and Subscriber Counter |
|---|---|---|---|
| Default Service | 0 | | Default Service* |
| Generic | 1 | Default Service | Default Service* |
| Generic TCP | 2 | Generic | Generic TCP |
| Generic UDP | 3 | Generic | Generic UDP |
| Generic IP | 6 | Generic | Generic IP |

| Name | ID | Name of Parent Service | Global Counter and Subscriber Counter |
|---|---|---|---|
| E-Mail | 4 | Default Service | E-Mail* |
| POP3 | 21 | E-Mail | E-Mail* |
| SMTP | 22 | E-Mail | E-Mail* |
| IMAP | 23 | E-Mail | E-Mail* |
| Browsing | 7 | Default Service | Browsing* |
| HTTP | 16 | Browsing | Browsing* |
| HTTPS | 17 | Browsing | Browsing* |
| Newsgroups | 8 | Default Service | Newsgroups |
| P2P | 9 | Default Service | P2P |
| eDonkey/eMule | 14 | P2P | eDonkey/eMule |
| Kazaa | 15 | P2P | Kazaa |
| Bittorrent | 24 | P2P | Bittorrent |
| Commercial File Sharing | 26 | P2P | Commercial File Sharing |
| Winny | 27 | P2P | Winny |
| Gnutella | 30 | P2P | Gnutella |
| WinMX | 31 | P2P | WinMX |
| VoIP | 12 | Default Service | VoIP |
| MGCP | 5 | VoIP | MGCP |
| SIP | 10 | VoIP | SIP |
| H323 | 11 | VoIP | H323 |
| Vonage | 13 | VoIP | Vonage |
| Skype | 25 | VoIP | Skype |
| Skinny | 35 | VoIP | Skinny |
| DingoTel | 36 | VoIP | DingoTel |
| Instant Messaging | 28 | Default Service | Instant Messaging |
| Gaming | 29 | Default Service | Gaming |
| FTP | 32 | Default Service | FTP |
| Net Admin | 33 | Default Service | Net Admin |
| Streaming | 34 | Default Service | Streaming* |
| Streaming over HTTP | 18 | Streaming | Streaming* |
| RTSP | 19 | Streaming | Streaming* |
| MMS | 20 | Streaming | Streaming* |

# 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 1-9 Default RDR Settings**

| RDR Family | RDR Name | State | Rate | Rate Limit | Notes |
|---|---|---|---|---|---|
| Usage | Link | ON | Every 5 minutes | | |
| | Package | ON | Every 5 minutes | | |
| | Subscriber | ON | Every 10 minutes | 200 per second | |
| Transaction | Transaction | ON | | 100 per second | All services have same relative weight |
| Transaction Usage | Transaction Usage (TUR) | OFF | | | No threshold |
| | Interim TUR | OFF | | | |
| Quota | Breach | OFF | | | |
| | Remaining | OFF | Every 5 minutes | 100 per second | |
| | Threshold | OFF | | | Generate RDR when balance goes below 10 MB |
| Log | Block | ON | | 20 per second | |
| Real-Time | Real-Time Subscriber Usage | ON | Every 1 minutes | 100 per second | Enable for each subscriber separately using CLI |
| Real-Time Signaling | Flow Signaling | OFF | | | |
| | Attack Signaling | OFF | | | |

# Rules

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

The default service configuration contains a single rule for the Default Service. Until the user creates 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 which have unlimited BW
- The rule applies no quota limitations to the flows

# System Mode

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

CHAPTER 2

# Raw Data Records: Formats and Field Contents

Raw Data Records (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). This chapter contains a list of the 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.

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

This chapter contains the following sections:

# Universal RDR Fields

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

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

**Note** For port-based protocols (for example, TCP port 666 for DOOM) and IP-protocol-based protocols (for example, IP protocol 1 for ICMP), the PROTOCOL_ID will be the TCP_GENERIC / UDP_GENERIC / IP_PROTOCOL value, according to the specific base protocol of the transaction.

- PROTOCOL_SIGNATURE—This field contains the ID of the protocol signature associated with this session.
- ZONE_ID—This field contains the ID of the zone associated with this session.
- FLAVOR_ID—For protocol signatures that have flavors, this field contains the ID of the flavor associated with this session.

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

# Transaction RDR

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

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

The following table lists the RDR fields and their descriptions.

**Table 2-1 Transaction RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3). |
| PACKAGE_ID | INT16 | See *Universal RDR Fields* (on page 2-3). |
| SERVICE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| PROTOCOL_ID | INT16 | See *Universal RDR Fields* (on page 2-3). |
| SKIPPED_SESSIONS | INT32 | The number of unreported sessions since the previous RDR. |
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| ACCESS_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| INFO_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3). |

| RDR Field Name | Type | Description |
|---|---|---|
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| MILLISEC_DURATION | INT32 | Duration, in milliseconds, of the transaction reported in this RDR. |
| TIME_FRAME | INT8 | See *Universal RDR Fields* (on page 2-3). |
| 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 downstream volume on both links of all the flows bundled in the transaction. |
| SUBSCRIBER_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 32 subscriber counters. |
| GLOBAL_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 64 global counters. |
| PACKAGE_COUNTER_ID | UINT16 | Each package is mapped to a counter. There are 64 package counters. |
| IP_PROTOCOL | UINT8 | IP protocol type. |
| PROTOCOL_SIGNATURE | INT32 | See *Universal RDR Fields* (on page 2-3). |
| ZONE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLAVOR_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLOW_CLOSE_MODE | UINT8 | The reason for the end of flow. |

# Transaction Usage RDR

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

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

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

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

The following table lists the RDR fields and their descriptions.

**Table 2-2 Transaction Usage RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3). |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| SERVICE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| PROTOCOL_ID | INT16 | See *Universal RDR Fields* (on page 2-3). |
| SKIPPED_SESSIONS | INT32 | Number of unreported sessions since the previous RDR. |
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| ACCESS_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| INFO_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3). |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| MILLISEC_DURATION | UINT32 | Duration, in milliseconds, of the transaction reported in this RDR. |
| TIME_FRAME | INT8 | See *Universal RDR Fields* (on page 2-3). |
| 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 | Each service is mapped to a counter. There are 32 subscriber counters. |
| GLOBAL_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 64 global counters. |
| PACKAGE_COUNTER_ID | UINT16 | Each package is mapped to a counter. There are 64 package counters. |
| IP_PROTOCOL | UINT8 | IP protocol type. |
| PROTOCOL_SIGNATURE | INT32 | See *Universal RDR Fields* (on page 2-3). |
| ZONE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLAVOR_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLOW_CLOSE_MODE | UINT8 | The reason for the end of flow. |

# HTTP Transaction Usage RDR

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

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

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

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

The following table lists the RDR fields and their descriptions.

**Table 2-3 HTTP Transaction Usage RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3). |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| SERVICE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| PROTOCOL_ID | INT16 | See *Universal RDR Fields* (on page 2-3). |
| SKIPPED_SESSIONS | INT32 | Number of unreported sessions since the previous RDR. |
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| ACCESS_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| INFO_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3). |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| MILLISEC_DURATION | UINT32 | Duration, in milliseconds, of the transaction reported in this RDR. |
| TIME_FRAME | INT8 | See *Universal RDR Fields* (on page 2-3). |
| 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. |

| RDR Field Name | Type | Description |
|---|---|---|
| SESSION_DOWNSTREAM_VOLUME | UINT32 | Downstream volume of the transaction, in bytes. The volume refers to the aggregated stream volume on both links of all the flows bundled in the transaction. |
| SUBSCRIBER_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 32 subscriber counters. |
| GLOBAL_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 64 global counters. |
| PACKAGE_COUNTER_ID | UINT16 | Each package is mapped to a counter. There are 64 package counters. |
| IP_PROTOCOL | UINT8 | IP protocol type. |
| PROTOCOL_SIGNATURE | INT32 | See *Universal RDR Fields* (on page 2-3). |
| ZONE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLAVOR_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLOW_CLOSE_MODE | UINT8 | The reason for the end of flow. |
| USER_AGENT | STRING | The user agent field extracted from the HTTP transaction. |
| HTTP_URL | STRING | The URL extracted from the HTTP transaction. |

# RTSP Transaction Usage RDR

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

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

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

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

The following table lists the RDR fields and their descriptions.

**Table 2-4 RTSP Transaction Usage RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3). |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| SERVICE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| PROTOCOL_ID | INT16 | See *Universal RDR Fields* (on page 2-3). |
| SKIPPED_SESSIONS | INT32 | Number of unreported sessions since the previous RDR. |
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| ACCESS_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| INFO_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3). |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| MILLISEC_DURATION | UINT32 | Duration, in milliseconds, of the transaction reported in this RDR. |
| TIME_FRAME | INT8 | See *Universal RDR Fields* (on page 2-3). |
| 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 | Each service is mapped to a counter. There are 32 subscriber counters. |
| GLOBAL_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 64 global counters. |
| PACKAGE_COUNTER_ID | UINT16 | Each package is mapped to a counter. There are 64 package counters. |
| IP_PROTOCOL | UINT8 | IP protocol type. |
| PROTOCOL_SIGNATURE | INT32 | See *Universal RDR Fields* (on page 2-3). |
| ZONE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLAVOR_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLOW_CLOSE_MODE | UINT8 | The reason for the end of flow. |
| RTSP_SESSION_ID | STRING | RTSP session ID as seen on an RTSP SETUP request. |
| RTSP_URL | STRING | RTSP URL. |

| RDR Field Name | Type | Description |
|---|---|---|
| RESPONSE_DATE | STRING | RTSP DESCRIBE date. |
| TOTAL_ENCODING_RATE | UINT32 | Sum of encoding rates of data flows. |
| NUMBER_OF_VIDEO_STREAMS | UINT8 | Number of video streams for this RTSP session. |
| NUMBER_OF_AUDIO_STREAMS | UINT8 | Number of audio streams for this RTSP session. |
| SESSION_TITLE | STRING | Title for this RTSP stream. |
| SERVER_NAME | STRING | Name of the RTSP server. |

# VoIP Transaction Usage RDR

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

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

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

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

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

The following table lists the RDR fields and their descriptions.

**Table 2-5 VoIP Transaction RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3). |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| SERVICE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| PROTOCOL_ID | INT16 | See *Universal RDR Fields* (on page 2-3). |
| SKIPPED_SESSIONS | INT32 | Number of unreported sessions since the previous RDR |
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| ACCESS_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| INFO_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3). |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| MILLISEC_DURATION | UINT32 | Duration, in milliseconds, of the transaction reported in this RDR. |
| TIME_FRAME | INT8 | See *Universal RDR Fields* (on page 2-3). |
| 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 downstream volume on both links of all the flows bundled in the transaction. |
| SUBSCRIBER_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 32 subscriber counters. |
| GLOBAL_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 64 global counters. |
| PACKAGE_COUNTER_ID | UINT16 | Each package is mapped to a counter. There are 64 package counters. |
| IP_PROTOCOL | UINT8 | IP protocol type. |
| PROTOCOL_SIGNATURE | INT32 | See *Universal RDR Fields* (on page 2-3). |
| ZONE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLAVOR_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| FLOW_CLOSE_MODE | UINT8 | The reason for the end of flow. |
| APPLICATION_ID | UINT32 | The ITU-U vendor ID of the application.<br>A value of 0xFFFFFFFF indicates that this field was not found in the traffic. |

| RDR Field Name | Type | Description |
|---|---|---|
| UPSTREAM_PACKET_LOSS | UINT16 | The average fractional upstream packet loss for the session, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.)<br>A value of 0xFFFF indicates that this field is undefined (no RTCP flows were opened). |
| DOWNSTREAM_PACKET_LOSS | UINT16 | The average fractional downstream packet loss for the session, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.)<br>A value of 0xFFFF indicates that this field is undefined (no RTCP flows were opened). |
| UPSTREAM_AVERAGE_JITTER | UINT32 | The average upstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.)<br>A value of 0xFFFFFFFF indicates that this field is undefined (no RTCP flows were opened). |
| DOWNSTREAM_AVERAGE_JITTER | UINT32 | The average downstream jitter for the session in units of 1/65 millisecond, taken from the RTCP flow. (Refer to the note following this table for an explanation of this value.)<br>A value of 0xFFFFFFFF indicates that this field is undefined (no RTCP flows were opened). |
| CALL_DESTINATION | STRING | The Q931 Alias address of the session destination.<br>A value of 'N/A' indicates that this field was not found in the traffic. |
| CALL_SOURCE | STRING | The Q931 Alias address of the session source.<br>A value of 'N/A' indicates that this field was not found in the traffic. |
| UPSTREAM_PAYLOAD_TYPE | UINT8 | The upstream RTP payload type for the session.<br>A value of 0xFF indicates that this field was not available (no RTP flows were opened). |
| DOWNSTREAM_PAYLOAD_TYPE | UINT8 | The downstream RTP payload type for the session.<br>A value of 0xFF indicates that this field is undefined (no RTP flows were opened). |
| CALL_TYPE | UINT8 | The call type (taken from H225 packet).<br>A value of 0xFF indicates that this field is undefined (no RTP flows were opened). |
| MEDIA_CHANNELS | UINT8 | The number of data flows that were opened during the session. |

**Note**

Packet Loss
This field is taken from the RTCP field 'fraction lost'. It is the average value of all RTCP packets seen during the flow life for the specified direction. The value is the numerator of a fraction whose denominator is 256. To get the packet loss value as percentage, divide this value by 2.56.
Average Jitter
This field is taken from the RTCP field ‘interval jitter’. The reported value is the average value of all RTCP packets seen during the flow life for the specified direction. This value is multiplied by the NTP time-stamp delta (middle 32 bits) and divided by the RTCP time-stamp delta to convert it to normal time units. These two time stamps are also taken from the RTCP packet. The reported value is the average jitter in units of 1/65536 second. To convert to milliseconds divide by 65.536.
See RFC 1889 for further information about the RCP/RTCP standard.

# Subscriber Usage RDR

The SUBSCRIBER_USAGE_RDR is generated periodically, at user-configured intervals, if the subscriber consumed resources associated with the service 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 the service since the previous RDR generation point, a Subscriber Usage RDR is generated
- If the subscriber did *not* consume resources associated with the service since the previous RDR generation point, *no* Subscriber Usage RDR is generated now

**Note**

Unlike other Usage RDRs, the generation logic for Subscriber Usage RDRs does NOT use the *zeroing methodology* (as described in *Periodic RDR Zero Adjustment Mechanism* (on page 2-36)).

In addition, a Subscriber Usage RDR may 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 the service since the previous Subscriber Usage RDR, a Subscriber Usage RDR is generated now. If the subscriber did not consume resources since the previous RDR, no RDR is generated for that service.

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

The following table lists the RDR fields and their descriptions.

**Table 2-6 Subscriber Usage RDR**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3). |
| PACKAGE_ID | INT16 | See *Universal RDR Fields* (on page 2-3). |
| SERVICE_USAGE_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 32 counters in the subscriber scope. |
| BREACH_STATE | UINT8 | See *Universal RDR Fields* (on page 2-3).<br>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:<br>• 0—Period time passed<br>• 1—Subscriber logout<br>• 2—Package switch<br>• 3—Wraparound<br>• 4—End of aggregation period |
| CONFIGURED_DURATION | INT32 | See *Universal RDR Fields* (on page 2-3). |
| DURATION | INT32 | This release—Not implemented (always the same as CONFIGURED_DURATION).<br>Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR. |
| END_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| UPSTREAM_VOLUME | INT32 | Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period. |
| DOWNSTREAM_VOLUME | INT32 | Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period. |
| SESSIONS | UINT16 | Aggregated number of sessions for the reported service, for the current reporting period. |
| SECONDS | UINT16 | Aggregated number of session seconds for the reported service, for the current reporting period. |

# Real-Time Subscriber Usage RDR

The REALTIME_SUBSCRIBER_USAGE_RDR is generated periodically, at user-configured intervals, if the subscriber consumed resources associated with the service during the current reporting period.

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

At fixed, user-configurable intervals (for example, every 30 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 Suite 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 the service since the previous RDR generation point, a Real-Time Subscriber Usage RDR is generated
- If the subscriber did *not* consume resources associated with the service 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* (on page 2-36)); if the subscriber consumes resources associated with the service at some later time, this will cause the *immediate* generation of either one or two zero-consumption Real-Time Subscriber Usage RDRs. (In addition to the eventual generation of the Real-Time Subscriber Usage RDR associated with this latest consumption of resources).

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

In addition, Real-Time Subscriber Usage RDRs may 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 the service since the previous Real-Time Subscriber Usage RDR, a Real-Time Subscriber Usage RDR is generated and then a zero-consumption Real-Time Subscriber Usage RDR is generated
  - If the subscriber did not consume resources since the previous RDR, no RDR is generated for that service

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

- The subscriber performed a login in a subscriber-integrated installation or was introduced from the SCE platform
  - Before the first time Real-Time Subscriber Usage RDR is generated for this service for this subscriber, a zero-consumption Real-Time Subscriber Usage RDR is generated

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

The following table lists the RDR fields and their descriptions.

**Table 2-7 Real-Time Subscriber Usage RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3). |
| PACKAGE_ID | INT16 | See *Universal RDR Fields* (on page 2-3). |
| SERVICE_USAGE_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 32 counters in the subscriber scope. |
| AGGREGATION_OBJECT_ID | INT16 | Externally assigned:<br>• 0—Offline subscriber<br>• 1—Online subscriber |
| BREACH_STATE | UINT8 | See *Universal RDR Fields* (on page 2-3).<br>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:<br>• 0—Period time passed<br>• 1—Subscriber logout<br>• 2—Package switch<br>• 3—Wraparound<br>• 4—End of aggregation period |
| CONFIGURED_DURATION | INT32 | See *Universal RDR Fields* (on page 2-3). |
| DURATION | INT32 | This release—Not implemented (always the same as CONFIGURED_DURATION).<br>Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR. |
| END_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| UPSTREAM_VOLUME | INT32 | Aggregated upstream volume on both links of all sessions, in kilobytes, for the current reporting period. |
| DOWNSTREAM_VOLUME | INT32 | Aggregated downstream volume on both links of all sessions, in kilobytes, for the current reporting period. |

| RDR Field Name | Type | Description |
|---|---|---|
| SESSIONS | UINT16 | Aggregated number of sessions for the reported service, for the current reporting period. |
| SECONDS | UINT16 | Aggregated number of session seconds for the reported service, for the current reporting period. |

# Link Usage RDR

The LINK_USAGE_RDR is generated periodically, at user-configured intervals, if the subscriber consumed resources associated with the service 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 the service have been consumed since the previous RDR generation point, a Link Usage RDR is generated
- If network resources associated with the service have *not* been 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* (on page 2-36)); if network resources associated with the service are again consumed at some later time, this will cause the *immediate* generation of either one or two zero-consumption Link Usage RDRs. (In addition to the eventual generation of the Link Usage RDR associated with this latest consumption of network resources).

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

**Note** A *separate* RDR is generated *for each link* (on a single traffic processor) within the SCE platform, where each RDR represents the total traffic processed and analyzed by that processor. To compute the total traffic in any given time frame, take the sum of the RDRs of all the processors. (A traffic processor that did not process any traffic of a specific service will not generate the corresponding RDR.)

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

The following table lists the RDR fields and their descriptions.

**Table 2-8 Link Usage RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| LINK_ID | INT8 | A numeric value associated with the reported network link. Possible values are 0 and 1 (referring to physical links 1 and 2 respectively).<br><br>For future use. |
| GENERATOR_ID | INT8 | A numeric value identifying the processor generating the RDR. Possible values are 0 through 3. |
| SERVICE_USAGE_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 64 global counters. |
| CONFIGURED_DURATION | INT32 | See *Universal RDR Fields* (on page 2-3). |
| DURATION | INT32 | This release—Not implemented (always the same as CONFIGURED_DURATION).<br><br>Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR. |
| END_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| UPSTREAM_VOLUME | INT32 | Aggregated upstream volume of all sessions, in kilobytes, for the current reporting period. |
| DOWNSTREAM_VOLUME | INT32 | Aggregated downstream volume of all sessions, in kilobytes, for the current reporting period. |
| SESSIONS | INT32 | Aggregated number of sessions for the reported service, for the current reporting period. |
| SECONDS | INT32 | Aggregated number of session seconds for the reported service, for the current reporting period. |
| CONCURRENT_SESSIONS | INT32 | Concurrent number of sessions using the reported service at this point in time. |
| ACTIVE_SUBSCRIBERS | INT32 | Concurrent number of subscribers using the reported service at this point in time. |
| TOTAL_ACTIVE_SUBSCRIBERS | INT32 | Concurrent number of subscribers in the system at this point in time. |

# Package Usage RDR

The PACKAGE_USAGE_RDR aggregates network usage information for all subscribers to the same package.

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 the service have been 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 the service 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* (on page 2-36)); if network resources associated with the service are once again consumed by a subscriber of the Package at some later time, this will cause the *immediate* generation of either one or two zero-consumption Package Usage RDRs. (In addition to the eventual generation of the Package Usage RDR associated with this latest consumption of network resources by a subscriber of the Package).

  - If there was only one interval (for example, 0805–0810) for which there was no consumption of network resources by a 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 a 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 within the SCE platform generates a separate RDR, where each RDR represents the total traffic processed and analyzed by that processor. To compute the total traffic (for a package) in any given time frame, take the sum of the RDRs of all the processors. (A traffic processor that did not process any traffic of a specific service for a specific package will not generate the corresponding RDR.)

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

The following table lists the RDR fields and their descriptions.

**Table 2-9 Package Usage RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| PACKAGE_COUNTER_ID | UINT16 | Each package is mapped to a counter. There are 64 package counters. |
| GENERATOR_ID | INT8 | A numeric value identifying the processor generating the RDR. |
| SERVICE_USAGE_COUNTER_ID | UINT16 | Each service is mapped to a counter. There are 64 global counters. |
| CONFIGURED_DURATION | INT32 | See *Universal RDR Fields* (on page 2-3). |
| DURATION | INT32 | This release—Not implemented (always the same as CONFIGURED_DURATION).<br>Future release—Indicates the number of seconds that have passed since the previous SUBSCRIBER_USAGE_RDR. |
| END_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| UPSTREAM_VOLUME | INT32 | Aggregated upstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period. |
| DOWNSTREAM_VOLUME | INT32 | Aggregated downstream volume on both links (for a single processor) of all sessions, in kilobytes, for the current reporting period. |
| SESSIONS | INT32 | Aggregated number of sessions for the reported service, for the current reporting period. |
| SECONDS | INT32 | Aggregated number of session seconds for the reported service, for the current reporting period. |
| CONCURRENT_SESSIONS | INT32 | Concurrent number of sessions using the reported service in the reported package at this point in time. |
| ACTIVE_SUBSCRIBERS | INT32 | Concurrent number of subscribers using the reported service in the reported package at this point in time. |
| TOTAL_ACTIVE_SUBSCRIBERS | INT32 | Concurrent number of subscribers in the system at this point in time. |

# 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.

Note the following regarding RDR generation:

- This RDR is generated when a session is blocked. A session can be blocked for various reasons; for example, access is blocked or concurrent session limit has been reached.
- Generation of this RDR is subject to two requirements, as follows:

- Quota—Each subscriber has a maximum quota of Blocking RDRs that can be generated for that subscriber in a specific aggregation period (day, week, month, and so forth). The quota is package-dependent; that is, its value is set according to the Package assigned to the subscriber.
- Rate—The rate is the global, maximum number of Blocking RDRs that can be generated per second by an SCE platform. The rate is a global value that sets an upper limit for the total number of RDRs to be generated for all subscribers.

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

The following table lists the RDR fields and their descriptions.

**Table 2-10 Blocking RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3). |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| SERVICE_ID | INT32 | See *Universal RDR Fields* (on page 2-3). |
| PROTOCOL_ID | INT16 | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3). |
| ACCESS_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| INFO_STRING | STRING | See *Universal RDR Fields* (on page 2-3). |
| BLOCK_REASON | UINT8 | Indicates the reason why this session was blocked. *See Block Reason (uint*8) (on page 2-32) for possible values and their interpretation. |
| BLOCK_RDR_COUNT | INT32 | Total number of blocked flows reported so far (from the beginning of the current time frame). |
| REDIRECTED | INT8 | Indicates whether the flow has been redirected (1) or not (0), after being blocked. Redirection will take place only for HTTP and RTSP flows that were mapped to a rule ordering them to be blocked and redirected. |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |

# Quota Breach RDR

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

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

This RDR is generated subject to the following conditions:

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

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

The following table lists the RDR fields and their descriptions.

**Table 2-11 Quota Breach RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3) |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| BUCKET_ID | UINT8 | 1-16, according to the number of the breached bucket |
| END_TIME | INT32 | See *Universal RDR Fields* (on page 2-3) |
| BUCKET_QUOTA | INT32 | The remaining quota in the indicated bucket:<br>• Volume bucket—Kilobytes<br>• Number of sessions bucket—Integer |
| AGGREGATION_PERIOD_TYPE | UINT8 | Defines how often the bucket is refilled<br>See *Aggregation period (uint8)* (on page 2-34) for possible values and their interpretations |

# Remaining Quota RDR

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

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

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

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

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

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

The following table lists the RDR fields and descriptions.

**Table 2-12 Remaining Quota RDR Fields**

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

# Quota Threshold Breach RDR

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

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

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

The following table lists the RDR fields and their descriptions.

**Table 2-13 Quota Threshold Breach RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3) |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| BUCKET_ID | UINT8 | 1-16, according to the number of the breached bucket |
| GLOBAL_THRESHOLD | UINT32 | The globally configured threshold in kilobytes |
| END_TIME | INT32 | See *Universal RDR Fields* (on page 2-3) |
| BUCKET_QUOTA | INT32 | The remaining quota in the indicated bucket in kilobytes |

# DHCP RDR

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

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

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

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

The following table lists the RDR fields and descriptions.

**Table 2-14 DHCP RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| CPE_MAC | STRING | A DHCP protocol field |
| CMTS_IP | UINT32 | A DHCP protocol field |
| ASSIGNED_IP | UINT32 | A DHCP protocol field |
| RELEASED_IP | UINT32 | A DHCP protocol field |
| TRANSACTION_ID | UINT32 | A DHCP protocol field |

| RDR Field Name | Type | Description |
|---|---|---|
| MESSAGE_TYPE | UINT8 | DHCP message type |
| OPTION_TYPE_0 through OPTION_TYPE_7 | UINT8 | A list of DHCP options extracted from the message |
| OPTION_VALUE_0 through OPTION_VALUE_7 | STRING | The values associated with the above DHCP options |
| END_TIME | INT32 | See *Universal RDR Fields* (on page 2-3) |

# RADIUS RDR

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

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

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

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

The following table lists the RDR fields and descriptions.

**Table 2-15 RADIUS RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3). |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3). |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3). |
| RADIUS_PACKET_CODE | UINT8 | The type of the RADIUS message intercepted. |
| RADIUS_ID | UINT8 | The RADIUS transaction ID. |
| ATTRIBUTE_VALUE_1 through ATTRIBUTE_VALUE_20 | STRING | Attributes extracted from the message. Sent as string format TLV. The last attribute field filled takes the value 0. |

# Flow Start RDR

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

**Note** This RDR is designed for services and packages where specific per-transaction RDRs are required (for example, higher Quality of Service). 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. *The generation scheme for this RDR should be configured with extra care.*

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

The following table lists the RDR fields and their descriptions.

**Table 2-16 Flow Start RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3) |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| SERVICE_ID | INT32 | See *Universal RDR Fields* (on page 2-3) |
| IP_PROTOCOL | UINT8 | IP protocol type |
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3) |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3) |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3) |
| START_TIME | UINT32 | Flow start time |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3) |
| BREACH_STATE | INT8 | See *Universal RDR Fields* (on page 2-3) |
| FLOW ID | UINT32 | Internal flow ID |
| GENERATOR_ID | INT8 | A numeric value identifying the processor generating the RDR |

# Flow End RDR

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

**Note** This RDR is designed for services and packages where specific per-transaction RDRs are required (for example, higher Quality of Service). 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. *The generation scheme for this RDR should be configured with extra care.*

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

The following table lists the RDR fields and their descriptions.

**Table 2-17 Flow End RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3) |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| SERVICE_ID | INT32 | See *Universal RDR Fields* (on page 2-3) |
| IP_PROTOCOL | UINT8 | IP protocol type |
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3) |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3) |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3) |
| START_TIME | UINT32 | Flow start time |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3) |
| BREACH_STATE | INT8 | See *Universal RDR Fields* (on page 2-3) |
| FLOW ID | UINT32 | Internal flow ID |
| GENERATOR_ID | INT8 | A numeric value identifying the processor generating the RDR |

# Ongoing Flow RDR

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

**Note** This RDR is designed for services and packages where specific per-transaction RDRs are required (for example, higher Quality of Service). 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. *The generation scheme for this RDR should be configured with extra care.*

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

The following table lists the RDR fields and their descriptions.

**Table 2-18 Ongoing Flow RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3) |
| PACKAGE_ID | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| SERVICE_ID | INT32 | See *Universal RDR Fields* (on page 2-3) |
| IP_PROTOCOL | UINT8 | IP protocol type |
| SERVER_IP | UINT32 | See *Universal RDR Fields* (on page 2-3) |
| SERVER_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| CLIENT_IP | UINT32 | See *Universal RDR Fields* (on page 2-3) |
| CLIENT_PORT | UINT16 | See *Universal RDR Fields* (on page 2-3) |
| INITIATING_SIDE | INT8 | See *Universal RDR Fields* (on page 2-3) |
| START_TIME | UINT32 | Flow start time |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3) |
| BREACH_STATE | INT8 | See *Universal RDR Fields* (on page 2-3) |
| FLOW ID | UINT32 | Internal flow ID |
| GENERATOR_ID | INT8 | A numeric value identifying the processor generating the RDR |

# 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.

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

The following table lists the RDR fields and their descriptions.

**Table 2-19 Attack Start RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| ATTACK_ID | UINT32 | Unique attack ID |
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3) |
| ATTACKING_IP | UINT32 | The IP address related to the attack (for example: in the case of DDoS, this will be the IP address under attack; in the case of a scan this will be the IP address of the source of the scan) |
| ATTACKED_IP | UINT32 | The other IP address related to the attack, if one exists; otherwise, 0xFFFFFFFF |
| ATTACKED_PORT | UINT16 | Attacked port; 0xFFFF if not present |
| ATTACKING_SIDE | INT8 | On which side of the SCE ATTACKING_IP resides:<br>• 1—Network<br>• 0—Subscriber |
| IP_PROTOCOL | UINT8 | IP protocol type |
| ATTACK_TYPE | UINT32 | To whom ATTACKING_IP belongs:<br>• 1—Attacker<br>• 0—Attacked |
| GENERATOR_ID | INT8 | A numeric value identifying the processor generating the RDR |
| ATTACK_TIME | UINT32 | Time since attack started in seconds |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3) |

# Attack End RDR

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

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

The following table lists the RDR fields and their descriptions.

**Table 2-20 Attack End RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| ATTACK_ID | UINT32 | Unique attack ID |
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3) |
| ATTACKING_IP | UINT32 | The IP address related to the attack (for example: in the case of DDoS, this will be the IP address under attack; in the case of a scan this will be the IP address of the source of the scan) |
| ATTACKED_IP | UINT32 | The other IP address related to the attack, if one exists; otherwise, 0xFFFFFFFF |
| ATTACKED_PORT | UINT16 | Attacked port; 0xFFFF if not present |
| ATTACKING_SIDE | INT8 | On which side of the SCE ATTACKING_IP resides:<br>• 1—Network<br>• 0—Subscriber |
| IP_PROTOCOL | UINT8 | IP protocol type |
| ATTACK_TYPE | UINT32 | To whom ATTACKING_IP belongs:<br>• 1—Attacker<br>• 0—Attacked |
| GENERATOR_ID | INT8 | A numeric value identifying the processor generating the RDR |
| ATTACK_TIME | UINT32 | Time since attack started in seconds |
| REPORT_TIME | INT32 | See *Universal RDR Fields* (on page 2-3) |

# Malicious Traffic Periodic RDR

The MALICIOUS_TRAFFIC_PERIODIC_RDR is generated at the detection of an attack. A MALICIOUS_TRAFFIC_PERIODIC_RDR is then generated periodically, at user-configured intervals, for the duration of the attack. The MALICIOUS_TRAFFIC_PERIODIC_RDR reports the details of the attack or malicious traffic.

Once the attack ends, a report of the resources consumed since the start of the attack is sent.

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

The following table lists the RDR fields and their descriptions.

**Table 2-21 Malicious Traffic Periodic RDR Fields**

| RDR Field Name | Type | Description |
|---|---|---|
| ATTACK_ID | INT32 | Unique attack ID. |
| SUBSCRIBER_ID | STRING | See *Universal RDR Fields* (on page 2-3). |
| ATTACK_IP | UINT32 | The IP address related to this attack. |
| OTHER_IP | UINT32 | The other IP address related to this attack, if such exists (if this is a DOS attack), or -1 otherwise. |
| PORT_NUMBER | UINT16 | The port number related to this attack, if such exists (if this is an IP scan, for example), or -1 otherwise. |
| ATTACK_TYPE | INT32 | Who ATTACK_IP belongs to:<br>• 1—Attacker<br>• 0—Attacked |
| SIDE | INT8 | The IP address side:<br>• 1—Network<br>• 0—Subscriber |
| IP_PROTOCOL | UINT8 | IP protocol type:<br>• 0—Other<br>• 1—ICMP<br>• 6—TCP<br>• 17—UDP |
| CONFIGURED_DURATION | INT32 | See *Universal RDR Fields* (on page 2-3). |
| DURATION | INT32 | Indicates the number of seconds that have passed since the previous MALICIOUS_TRAFFIC_RDR. |
| END_TIME | INT32 | See *Universal RDR Fields* (on page 2-3). |
| ATTACKS | INT8 | The number of attacks in the current reporting period. Since this report is generated per attack, the value is 0 or 1. |

| RDR Field Name | Type | Description |
|---|---|---|
| MALICIOUS_SESSIONS | INT32 | Aggregated number of sessions for the reported attack, for the current reporting period.<br>If the SCE platform blocks the attack, this field takes the value -1. |

# RDR Enumeration Fields

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

## Block Reason (uint8)

The BLOCK_REASON field can be interpreted as a bit field. The following table lists the possible values of the field separated into bits.

**Table 2-22 Block Reason Field Bit Values**

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

## String Fields

The following table lists the ACCESS_STRING and INFO_STRING field values.

**Table 2-23 String Field Values**

| Name | TR ACCESS_STRING | TR INFO_STRING | Description |
|---|---|---|---|
| PROTOCOL_TCP_GENERIC | Null | Null | |
| PROTOCOL_UDP_GENERIC | Null | Null | |
| PROTOCOL_HTTP_BROWSING | Host name | URL | |
| PROTOCOL_HTTP_STREAMING | Host name | URL | |
| PROTOCOL_FTP | Null | Null | |
| PROTOCOL_RTSP | Host name | Null | |
| PROTOCOL_MMS | Null | Null | |
| PROTOCOL_PROXY_HTTP | Host name | Null | |

| Name | TR ACCESS_STRING | TR INFO_STRING | Description |
|---|---|---|---|
| PROTOCOL_SMTP | Server IP | Sender | |
| PROTOCOL_POP3 | Server name | Login name | |
| PROTOCOL_IP_GENERIC | Null | Null | Non–TCP/UDP transaction |
| PROTOCOL_GNUTELLA_NETWORKING | Null | Null | Peer to peer |
| PROTOCOL_GNUTELLA_FILE_TRANSFER | Null | Null | Peer to peer |
| PROTOCOL_FASTTRACK_NETWORKING | Null | Null | Peer to peer |
| PROTOCOL_FASTTRACK_TRANSFER | Network name | Null | Peer to peer |
| PROTOCOL_NNTP | Null | Group name | |
| PROTOCOL_NAP_WINMX_TRANSFER | Null | Null | Peer to peer |
| PROTOCOL_WINNY | Null | Null | Peer to peer |
| PROTOCOL_EDONKEY | Null | Null | Peer to peer |
| PROTOCOL_DIRECT_CONNECT | Null | Null | Peer to peer |
| PROTOCOL_HOTLINE | Null | Null | Peer to peer |
| PROTOCOL_DYNAMIC_SIGNATURE | Null | Null | |
| PROTOCOL_MANOLITO | Null | Null | Peer to peer |
| PROTOCOL_SIP | SIP Method | SIP Domain | |
| PROTOCOL_BITTORRENT | Null | Null | Peer to peer |
| PROTOCOL_SKYPE | Null | Null | Peer to peer |
| PROTOCOL_VONAGE | SIP Method | SIP Subscriber ID | |
| PROTOCOL_SHARE | Null | Null | Peer to peer |
| PROTOCOL_H323 | Null | Is FastStart | |
| PROTOCOL_SOULSEEK | Null | Null | Peer to peer |
| PROTOCOL_ITUNES | Null | Null | Peer to peer |
| PROTOCOL_FILETOPIA | Null | Null | Peer to peer |
| PROTOCOL_NAPSTER | Null | Null | Peer to peer |
| PROTOCOL_DHCP | Null | Null | |
| PROTOCOL_MUTE | Null | Null | Peer to peer |
| PROTOCOL_NODEZILLA | Null | Null | Peer to peer |

| Name | TR ACCESS_STRING | TR INFO_STRING | Description |
|---|---|---|---|
| PROTOCOL_WASTE | Null | Null | Peer to peer |
| PROTOCOL_NEONET | Null | Null | Peer to peer |
| PROTOCOL_MGCP | Null | Null | |
| PROTOCOL_WAREZ | Null | Null | Peer to peer |

## Aggregation Period (uint8)

The following table lists the AGG_PERIOD field values.

**Table 2-24 AGG_PERIOD Field Values**

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

## Time Frames (uint16)

The following table lists the TIME_FRAME field values.

**Table 2-25 Time Frame Field Values**

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

# RDR Tag Assignment Summary

The following is a summary of RDR tag assignments.

The RDR categories can be configured using the SCE CLI. See the *Cisco Service Control Engine CLI Command Reference* for more information.

**Table 2-26 RDR Tag Assignments**

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

# 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 has been 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

# Database Tables: Formats and Field Contents

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

For additional information, such as RDR structure, RDR column and field descriptions, and how the RDRs are generated, see *Raw Data Records: Formats and Field Contents* (on page 2-1).

This chapter contains the following sections:



## Overview

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

In addition to the RDR fields that are specific to each RDR type, the tables RPT_NUR, RPT_SUR, RPT_PUR, RPT_LUR, and RPT_TR contain two universal columns: RECORD_SOURCE and TIME_STAMP. The following values are placed in these two universal columns (field numbers 0 and 1, respectively):

- 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.

- 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.

## Database Tables

This section contains information about the following database tables:

- RPT_NUR
- RPT_SUR

- RPT_PUR
- RPT_LUR
- RPT_TR
- RPT_MALUR
- RPT_TOPS_PERIOD0
- RPT_TOPS_PERIOD1
- INI_VALUES

## Table RPT_NUR

Database table RPT_NUR stores data from SUBSCRIBER_USAGE_RDRs.

**Note** This table does not exist in the default configuration.

These RDRs have the tag **4042321920**.

**Table 3-1 Columns for Table RPT_NUR**

| Field Number | Field Name | Type |
|---|---|---|
| 1 | TIME_STAMP | DateTime |
| 2 | RECORD_SOURCE | Number |
| 3 | SUBSCRIBER_ID | String |
| 4 | PACKAGE_ID | Number |
| 5 | SUBS_USG_CNT_ID | Number |
| 6 | BREACH_STATE | Number |
| 7 | REASON | Number |
| 8 | CONFIGURED_DURATION | Number |
| 9 | DURATION | Number |
| 10 | END_TIME | Number |
| 11 | UPSTREAM_VOLUME | Number |
| 12 | DOWNSTREAM_VOLUME | Number |
| 13 | SESSIONS | Number |
| 14 | SECONDS | Number |

## Table RPT_SUR

Database table RPT_SUR stores data from REALTIME_SUBSCRIBER_USAGE_RDRs. These RDRs have the tag **4042321922**.

**Table 3-2 Columns for Table RPT_SUR**

| Field Number | Field Name | Type |
|---|---|---|
| 1 | TIME_STAMP | DateTime |
| 2 | RECORD_SOURCE | Number |
| 3 | SUBSCRIBER_ID | String |
| 4 | PACKAGE_ID | Number |
| 5 | SUBS_USG_CNT_ID | Number |
| 6 | MONITORED_OBJECT_ID | Number |
| 7 | BREACH_STATE | Number |
| 8 | REASON | Number |
| 9 | CONFIGURED_DURATION | Number |
| 10 | DURATION | Number |
| 11 | END_TIME | Number |
| 12 | UPSTREAM_VOLUME | Number |
| 13 | DOWNSTREAM_VOLUME | Number |
| 14 | SESSIONS | Number |
| 15 | SECONDS | Number |

## Table RPT_PUR

Database table RPT_PUR stores data from PACKAGE_USAGE_RDRs. These RDRs have the tag **4042321924**.

**Table 3-3 Columns for Table RPT_PUR**

| Field Number | Field Name | Type |
|---|---|---|
| 1 | TIME_STAMP | DateTime |
| 2 | RECORD_SOURCE | Number |
| 3 | PKG_USG_CNT_ID | Number |
| 4 | GENERATOR_ID | Number |
| 5 | GLBL_USG_CNT_ID | Number |
| 6 | CONFIGURED_DURATION | Number |
| 7 | DURATION | Number |
| 8 | END_TIME | Number |

| Field Number | Field Name | Type |
|---|---|---|
| 9 | UPSTREAM_VOLUME | Number |
| 10 | DOWNSTREAM_VOLUME | Number |
| 11 | SESSIONS | Number |
| 12 | SECONDS | Number |
| 13 | CONCURRENT_SESSIONS | Number |
| 14 | ACTIVE_SUBSCRIBERS | Number |
| 15 | TOTAL_ACTIVE_SUBSCRIBERS | Number |

## Table RPT_LUR

Database table RPT_LUR stores data from LINK_USAGE_RDRs.

These RDRs have the tag **4042321925**.

**Table 3-4 Columns for Table RPT_LUR**

| Field Number | Field Name | Type |
|---|---|---|
| 1 | TIME_STAMP | DateTime |
| 2 | RECORD_SOURCE | Number |
| 3 | PKG_USG_CNT_ID | Number |
| 4 | GENERATOR_ID | Number |
| 5 | GLBL_USG_CNT_ID | Number |
| 6 | CONFIGURED_DURATION | Number |
| 7 | DURATION | Number |
| 8 | END_TIME | Number |
| 9 | UPSTREAM_VOLUME | Number |
| 10 | DOWNSTREAM_VOLUME | Number |
| 11 | SESSIONS | Number |
| 12 | SECONDS | Number |
| 13 | CONCURRENT_SESSIONS | Number |
| 14 | ACTIVE_SUBSCRIBERS | Number |
| 15 | TOTAL_ACTIVE_SUBSCRIBERS | Number |

## Table RPT_TR

Database table RPT_TR stores data from TRANSACTION_RDRs.

These RDRs have the tag **4042321936**.

**Table 3-5 Columns for Table RPT_TR**

| Field Number | Field Name | Type |
|---|---|---|
| 1 | TIME_STAMP | DateTime |
| 2 | RECORD_SOURCE | Number |
| 3 | SUBSCRIBER_ID | String |
| 4 | PACKAGE_ID | Number |
| 5 | SERVICE_ID | Number |
| 6 | PROTOCOL_ID | Number |
| 7 | SAMPLE_SIZE | Number |
| 8 | PEER_IP | Number |
| 9 | PEER_PORT | Number |
| 10 | ACCESS_String | String |
| 11 | INFO_String | String |
| 12 | SOURCE_IP | Number |
| 13 | SOURCE_PORT | Number |
| 14 | INITIATING_SIDE | Number |
| 15 | END_TIME | Number |
| 16 | MILISEC_DURATION | Number |
| 17 | TIME_FRAME | Number |
| 18 | UPSTREAM_VOLUME | Number |
| 19 | DOWNSTREAM_VOLUME | Number |
| 20 | SUBS_CNT_ID | Number |
| 21 | GLBL_CNT_ID | Number |
| 22 | PKG_USG_CNT_ID | Number |
| 23 | IP_PROTOCOL | Number |
| 24 | PROTOCOL_SIGNATURE | Number |
| 25 | ZONE_ID | Number |
| 26 | FLAVOR_ID | Number |
| 27 | FLOW_CLOSE_MODE | Number |

## Table RPT_MALUR

Database table RPT_MALUR stores data from MALICIOUS_TRAFFIC_PERIODIC_RDRs. These RDRs have the tag **4042322000**.

**Table 3-6 Columns for Table RPT_MALUR**

| Field Number | Field Name | Type |
|---|---|---|
| 1 | TIME_STAMP | DateTime |
| 2 | RECORD_SOURCE | Number |
| 3 | ATTACK_ID_HIGH | Number |
| 4 | SUBSCRIBER-ID | String |
| 5 | ATTACK_IP | Number |
| 6 | OTHER_IP | Number |
| 7 | PORT_NUMBER | Number |
| 8 | ATTACK_TYPE | Number |
| 9 | SIDE | Number |
| 10 | IP_PROTOCOL | Number |
| 11 | CONFIGURED_DURATION | Number |
| 12 | DURATION | Number |
| 13 | END_TIME | Number |
| 14 | ATTACKS | Number |
| 15 | 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 3-7 Columns for Table RPT_TOPS_PERIOD0**

| Field Number | Field Name | Type |
|---|---|---|
| 1 | RECORD_SOURCE | Number |
| 2 | METRIC_ID | Number |
| 3 | SUBS_USG_CNT_ID | Number |
| 4 | TIME_STAMP | DateTime |
| 5 | AGG_PERIOD | Number |
| 6 | SUBSCRIBER_ID | String |
| 7 | 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.

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

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

**Table 3-8 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 3-9 Columns for Table RPT_TOPS_PERIOD1**

| Field Number | Field Name | Type |
|---|---|---|
| 1 | RECORD_SOURCE | Number |
| 2 | METRIC_ID | Number |
| 3 | SUBS_USG_CNT_ID | Number |
| 4 | TIME_STAMP | DateTime |
| 5 | AGG_PERIOD | Number |
| 6 | SUBSCRIBER_ID | String |
| 7 | 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.

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

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

**Table 3-10 Metric_ID Values**

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

# Table INI_VALUES

Database table INI_VALUES is updated whenever a 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 mappings contained in this table are used by the SCAS Reporter.

**Table 3-11 Columns for Table INI_VALUES**

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

# 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 3-12 Columns for Table CONF_SE_TZ_OFFSET**

| Field Number | Field Name | Type |
|---|---|---|
| 1 | TIME_STAMP | DateTime |
| 2 | OFFSET_MIN | Number |

CHAPTER 4

# CSV File Formats

The ***Cisco Service Control Application for Broadband*** (***SCA BB***) provides several types of Comma-Separated Value (CSV) flat files that can be used for reviewing and configuration. For example, CSV files can be viewed with applications such as Excel.

This chapter contains the following sections:



## 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 a service configuration.

For more information about exporting and importing service configuration entities, see the *Cisco Service Control Application for Broadband User Guide*.

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

### Services

Service CSV files have a fixed format; all lines have the same structure:

```
# CSV line 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
- By default, the Default Service is the parent of all other services
- If the service is to be counted with its parent, it should have a counter index of -1

- One service can have multiple entries in the file (see the following example); in this case 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)

**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
```

## Protocols

Protocol CSV files have a fixed format; all lines have the same structure:

```
# CSV line 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.

**EXAMPLE**

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
```

## Zones

Zone CSV files have a fixed format; all lines have the same structure:

```
# CSV line format: zone name, zone index, IP range
```

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

**EXAMPLE**

The following is an example of a zone CSV file:

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

## Flavors

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

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

```
# CSV line format: 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.

**EXAMPLE**

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

```
HttpUrlFlavor,1,HTTP_URL
```

## HTTP URL

HTTP URL CSV files have a fixed format; all lines have the same structure:

```
# CSV line format: flavor name, flavor index, flavor type, host suffix,
params prefix, URI suffix, URI prefix
```

**EXAMPLE**

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

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

## HTTP User Agent

HTTP User Agent CSV files have a fixed format; all lines have the same structure:

```
# CSV line format: flavor name, flavor index, flavor type, user agent
```

## HTTP Composite

HTTP Composite CSV files have a fixed format; all lines have the same structure:

```
# CSV line 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

RTSP User Agent CSV files have a fixed format; all lines have the same structure:

```
# CSV line format: flavor name, flavor index, flavor type, user agent
```

## RTSP Host Name

RTSP Host Name CSV files have a fixed format; all lines have the same structure:

```
# CSV line format: flavor name, flavor index, flavor type, host suffix
```

## RTSP Composite

HTTP Composite CSV files have a fixed format; all lines have the same structure:

```
# CSV line 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

SIP Destination Domain CSV files have a fixed format; all lines have the same structure:

```
# CSV line format: flavor name, flavor index, flavor type, host suffix
```

### SIP Source Domain

SIP Source Domain CSV files have a fixed format; all lines have the same structure:

```
# CSV line format: flavor name, flavor index, flavor type, host suffix
```

### SIP Composite

HTTP Composite CSV files have a fixed format; all lines have the same structure:

```
# CSV line 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

SMTP Host Name CSV files have a fixed format; all lines have the same structure:

```
# CSV line format: flavor name, flavor index, flavor type, host suffix
```

# 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 regarding these CSV file formats, see the *Cisco Service Control Engine Software Configuration Guide* and the *Cisco Service Control Management Suite Subscriber Manager User Guide*.

## Import/Export File: Format of the mappings Field

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

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

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

**EXAMPLES**

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

## SCE Subscriber Files

```
# CSV line format: subscriber-id, mappings, package-id
```

**EXAMPLE**

The following is a sample CSV file for use with the SCE CLI:

```
JerryS,80.179.152.159;80.179.152.179,0
ElainB,194.90.12.2,3
```

## SCMS SM Subscriber Files

```
# CSV line format: subscriber-id, domain, mappings, package-id
```

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

**EXAMPLE**

The following is a sample CSV file for use with the SM CLI:

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

## Anonymous Group CSV Files

Anonymous Group CSV files have a fixed format. All lines have the same structure, as described below:

```
# CSV line format: anonymous-group-name, IP-range[, subscriber-template-
number]
```

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

**EXAMPLE**

The following is an example of an anonymous group CSV file:

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

# Collection Manager CSV File Formats

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

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

## 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 that is represented in the file. (See the appropriate section of the chapter *Raw Data Records: Formats and Field Contents* (on 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:

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

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

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

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

## RAG Adapter CSV Files

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

The name of the CSV file is the date and time of its creation. The default format of the file name is `yyyy-MM-dd_HH-mm-ss.csv` (for example, `2005-09-27_18-30-01.csv`). By default, the CSV repository is flat (all CSV files in one directory), and located at `~/cm/adapters/RAGAdapter/csvfiles`. Alternatively, the adapter can be configured 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 output to the CSV file may have some synthesized fields added to it, such as time stamps of the first and last RDRs that contributed to this bucket and the total number of RDRs in this bucket. Other fields may be removed altogether. Fields in the output line that are not used for aggregation will have values corresponding to the values in the first RDR that contributed to the bucket. However, the time stamp field that is prepended to the line in the CSV file will have a value corresponding to the time stamp of the last RDR in the bucket.


# SCAS BB Proprietary MIB Reference

This chapter describes the Cisco Service Control Application Suite for Broadband (SCAS BB) proprietary Management Information Base (MIB) support by the Service Control Engine (SCE) platform.

A MIB is a database of objects that can be monitored by a network management system (NMS). The SCE platform supports both the standard MIB-II and a proprietary MIB. The proprietary SCAS BB MIB enables the external management system to monitor counters and metrics specific to ***SCA BB***, which are not provided by the standard MIB.

This chapter contains the following sections:



## 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; this is recommended as a security measure) An access-list (ACL) of IP addresses. This limits access to SNMP information to a set of known locations. A different community string can be defined for each ACL.
- The destination IP address to which the SCE platform will send SNMP traps; you can enable or disable specific traps.

See the *SNMP Configuration and Management* chapter of the *Cisco Service Control Engine Software Configuration Guide* for complete documentation of SNMP configuration.

## Loading the MIB Files for Use with a MIB Browser

To access the SNMP variables on the SCE platform, load the proprietary MIB files (`pcube.mib`, `pcubeSEMib.mib`, and `PCubeEngageMib.mib`) in the SNMP browser. (The SNMPv2-SMI & SNMPv2-TC must be loaded before you load the proprietary MIB files.)

### Loading the MIB Files

The ***SCA BB*** proprietary MIB uses definitions that are defined in other MIBs, such as `pcube.mib`, and in the SNMPv2-SMI. Therefore, the order in which the MIBs are loaded is important. To avoid errors, the MIBs must be loaded in the correct order.

**Note** The ***SCA BB*** MIB file (`PCubeEngageMib.mib`) can be downloaded from the ***SCA BB*** software download page. Other MIB files (`pcube.mib` and `pcubeSEMib.mib`) can be downloaded from the SCE OS software download page.

To load the MIBs:

**Step 1** Load the SNMPv2-SMI

**Step 2** Load the SNMPv2-TC

**Step 3** Load `pcube.mib`

**Step 4** Load `pcubeSEMib.mib`

**Step 5** Load `PCubeEngageMib.mib`

# Service Control Enterprise MIB

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

- The pcubeProducts sub-tree contains the sysObjectIDs of the Service Control products

Service Control product sysObjectIDs are defined in a MIB file named *Pcube-Products-MIB*

- The pcubeModules sub-tree provides a root object identifier from which MIB modules can be defined
- The pcubeMgmt sub-tree contains the configuration copy MIB
- The pcubeWorkgroup sub-tree contains the SCE MIB, which is the main MIB for the Service Control products

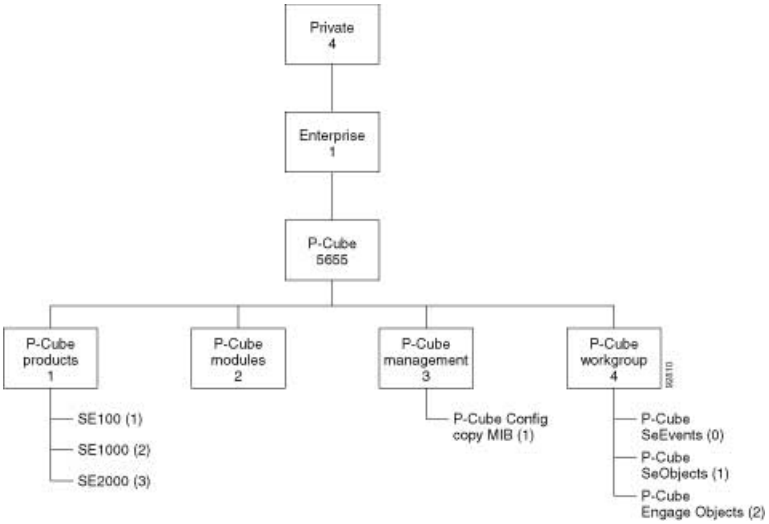
  The SCE MIB is divided into three main groups:

  - **pcubeSeEvents**
  - **pcubeSEObjs**
  - **pcubeEngageObjs**

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

The following figure illustrates the Service Control Enterprise MIB structure.

*Figure 5-1: Cisco Service Control MIB Structure*




Currently, the proprietary Cisco MIB consists of two main sub-trees:

- The pcubeMgmt sub-tree:
  - *pcubeConfigCopyMib* enables saving the running configuration of Cisco products (documented in the *Cisco Service Control Engine Software Configuration Guide*)
- The pcubeWorkgroup sub-tree:
  - *pcubeSeMib* provides a wide variety of configuration and runtime statistics (documented in the *Cisco Service Control Engine Software Configuration Guide*)

- The ***SCA BB*** MIB provides configuration and runtime status for ***SCA BB*** (see the following section)

# The SCA BB MIB

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

The ***SCA 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, the following information is presented:

| | |
|---|---|
| Description | Description of the object, including format and legal values, if applicable |
| Access | Access control associated with the object |
| **INDEX** | *{Indexes used by the table}* (where applicable) |
| **SYNTAX** | **The data type of the object** *(The general format of the object)* |

# pcubeEngageObjs (pcubeWorkgroup 2)

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

## pcubeEngageObjs Objects

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

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

# pcubeEngageObjs Structure

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

**serviceGrp**

*serviceTable* - deprecated

**linkGrp**

*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

*subscribersServiceUsageTable*

*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 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 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 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 in kilobytes of services in this service counter carried over the link

Access read-only

**SYNTAX**

**Counter32**

**Note** Although volume counters on the SCE platform hold 32-bit integers, 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 in kilobytes of services in this service counter carried over the link

Access read-only

**SYNTAX**

**Counter32**

## linkServiceUsageNumSessions (linkServiceUsageEntry 3)

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

Access read-only

**SYNTAX**

**Counter32**

## linkServiceUsageDuration (linkServiceUsageEntry 4)

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

Access read-only

**SYNTAX**

**Counter32**

## linkServiceUsageConcurrentSessions (linkServiceUsageEntry 5)

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

Access read-only

**SYNTAX**

**Counter32**

## linkServiceUsageActiveSubscribers (linkServiceUsageEntry 6)

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

Access read-only

**SYNTAX**

**Counter32**

## linkServiceUpDroppedPackets (linkServiceUsageEntry 7)

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

Access read-only

**SYNTAX**

**Counter32**

**Note** To enable the SCE application to count dropped packets and dropped bytes, the `accelerate-packet-drops` feature should be disabled 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 the *Cisco Service Control Engine Software Configuration Guide*.

## linkServiceDownDroppedPackets (linkServiceUsageEntry 8)

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

Access read-only

**SYNTAX**

**Counter32**

## linkServiceUpDroppedBytes (linkServiceUsageEntry 9)

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

Access read-only

**SYNTAX**

**Counter32**

### linkServiceDownDroppedBytes (linkServiceUsageEntry 10)

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

Access read-only

**SYNTAX**

**Counter32**

# Package Group: packageGrp (pcubeEngageObjs 3)

The Package group provides general and usage information for each global-scope package 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 counter

Access not-accessible

**SYNTAX**

```
Sequence of packageCounterEntry
```

### packageCounterEntry (packageCounterTable 1)

A Package Counter table entry containing parameters defining one package counter

Access not-accessible

**INDEX**

```
{moduleIndex, packageCounterIndex}
```

**SYNTAX**

**SEQUENCE** {

*packageCounterIndex*

*packageCounterStatus*

*packageCounterName*

*packageCounterActiveSubscribers*

}

## packageCounterIndex (packageCounterEntry 1)

The package counter index

Access not-accessible

**SYNTAX**

**INTEGER** *(1...255)*

## packageCounterStatus (packageCounterEntry 2)

The package counter status

Access read-only

**SYNTAX**

**INTEGER** *{*

*0 (disabled)*

*1 (enabled)*

*}*

## packageCounterName (packageCounterEntry 3)

The name of the package counter

Access read-only

**SYNTAX**

**DisplayString** *(SIZE 0...255)*

## packageCounterActiveSubscribers (packageCounterEntry 4)

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

Access read-only

**SYNTAX**

**Counter32**

## packageServiceUsageTable (packageGrp 2)

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

Access not-accessible

**INDEX**

```
{moduleIndex, packageCounterIndex, globalScopeServiceCounterIndex}
```

**SYNTAX**

**SEQUENCE** {

packageServiceUsageUpVolume

packageServiceUsageDownVolume

packageServiceUsageNumSessions

packageServiceUsageDuration

packageServiceUsageConcurrentSessions

packageServiceUsageActiveSubscribers

packageServiceUpDroppedPackets

packageServiceDownDroppedPackets

packageServiceUpDroppedBytes

packageServiceDownDroppedBytes

}

## packageServiceUsageUpVolume (packageServiceUsageEntry 1)

The upstream volume in kilobytes of packages in this package counter

Access read-only

**SYNTAX**

**Counter32**

**Note** Although volume counters on the SCE platform hold 32-bit integers, 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 in kilobytes of packages in this package counter

Access read-only

**SYNTAX**

**Counter32**

## packageServiceUsageNumSessions (packageServiceUsageEntry 3)

The number of sessions of packages in this package counter

Access read-only

**SYNTAX**

**Counter32**

## packageServiceUsageDuration (packageServiceUsageEntry 4)

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

Access read-only

**SYNTAX**

**Counter32**

## packageServiceUsageConcurrentSessions (packageServiceUsageEntry 5)

The number of concurrent sessions of packages in this package counter

Access read-only

**SYNTAX**

**Counter32**

## packageServiceUsageActiveSubscribers (packageServiceUsageEntry 6)

The number of active subscribers of packages in this package counter

Access read-only

**SYNTAX**

**Counter32**

## packageServiceUpDroppedPackets (packageServiceUsageEntry 7)

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

Access read-only

**SYNTAX**

**Counter32**

**Note** To enable the SCE application to count dropped packets and dropped bytes, the `accelerate-packet-drops` feature should be disabled 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 the *Cisco Service Control Engine Software Configuration Guide*.

## packageServiceDownDroppedPackets (packageServiceUsageEntry 8)

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

Access read-only

**SYNTAX**

**Counter32**

## packageServiceUpDroppedBytes (packageServiceUsageEntry 9)

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

Access read-only

**SYNTAX**

**Counter32**

## packageServiceDownDroppedBytes (packageServiceUsageEntry 10)

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

Access read-only

**SYNTAX**

**Counter32**

# Subscriber Group: subscriberGrp (pcubeEngageObjs 4)

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

**Note** To use the tables in this group, first create an entry to reference a particular subscriber in the subscribersPropertiesValueTable object of the subscriberGrp in the SE MIB (not the ***SCA BB*** MIB). Using the index of this table (spvIndex), information about the subscriber can be collected. See *Accessing Subscriber Information (the spvIndex)* (on page 5-21) for more information on how to access subscriber level information using the SNMP interface.

## subscribersTable (subscriberGrp 1)

The Subscribers Table provides information for each subscriber

Access not-accessible

**SYNTAX**

*Sequence of subscriberEntry*

## subscriberEntry (subscribersTable 1)

A Subscribers Table entry containing the package index of each subscriber

Access not-accessible

**INDEX**

*{moduleIndex, spvIndex}*

**SYNTAX**

**SEQUENCE** *{*

*subscriberPackageIndex*

*}*

## subscriberPackageIndex (subscriberEntry 1)

The package index of the subscriber's package

Access read-only

**SYNTAX**

**INTEGER** *(1...255)*

## subscribersServiceUsageTable (subscriberGrp 2)

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

Access not-accessible

**SYNTAX**

*Sequence of subscriberServiceUsageEntry*

## subscriberServiceUsageEntry (subscribersServiceUsageTable 1)

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

Access not-accessible

**INDEX**

*{moduleIndex, spvIndex, subscriberScopeServiceCounterIndex}*

**SYNTAX**

**SEQUENCE** *{*

*subscriberServiceUsageUpVolume*

*subscriberServiceUsageDownVolume*

*subscriberServiceUsageNumSessions*

*subscriberServiceUsageDuration*

*}*

## subscriberServiceUsageUpVolume (subscriberServiceUsageEntry 1)

The upstream volume in kilobytes of services in this service counter used by this customer

Access read-only

**SYNTAX**

**Counter32**

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

### subscriberServiceUsageDownVolume (subscriberServiceUsageEntry 2)

The downstream volume in kilobytes of services in this service counter used by this customer

Access read-only

**SYNTAX**

**Counter32**

### subscriberServiceUsageNumSessions (subscriberServiceUsageEntry 3)

The number of sessions of services in this service counter used by this customer

Access read-only

**SYNTAX**

**INTEGER** *(1...65535)*

### subscriberServiceUsageDuration (subscriberServiceUsageEntry 4)

Aggregated session duration in seconds of services in this service counter used by this customer

Access read-only

**SYNTAX**

**INTEGER** *(1...65535)*

## Service Counter Group: serviceCounterGrp (pcubeEngageObjs 5)

The Service Counter group provides general information for each global-scope and subscriber-scope service counter. It can be used, for example, to read the names of the services as defined in the ***SCA BB*** Service Configuration.

### globalScopeServiceCounterTable (serviceCounterGrp 1)

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

Access not-accessible

**SYNTAX**

*Sequence of globalScopeServiceCounterEntry*

## globalScopeServiceCounterEntry (globalScopeServiceCounterTable 1)

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

Access not-accessible

**INDEX**

*{moduleIndex, globalScopeServiceCounterIndex}*

**SYNTAX**

**SEQUENCE** *{*

*globalScopeServiceCounterIndex*

*globalScopeServiceCounterStatus*

*globalScopeServiceCounterName*

*}*

## globalScopeServiceCounterIndex (globalScopeServiceCounterEntry 1)

The global-scope service counter index

Access not-accessible

**SYNTAX**

**INTEGER** *(1...255)*

## globalScopeServiceCounterStatus (globalScopeServiceCounterEntry 2)

The global-scope service counter status

Access read-only

**SYNTAX**

**INTEGER** *{*

*0 (disabled)*

*1 (enabled)*

*}*

## globalScopeServiceCounterName (globalScopeServiceCounterEntry 3)

The name of the global-scope service counter

Access read-only

**SYNTAX**

**DisplayString** *(SIZE 0...255)*

## subscriberScopeServiceCounterTable (serviceCounterGrp 2)

The Subscriber-Scope Service Counter table consists of data regarding each service 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 counter

Access not-accessible

**INDEX**

*{moduleIndex, subscriberScopeServiceCounterIndex}*

**SYNTAX**

**SEQUENCE** *{*

*subscriberScopeServiceCounterIndex*

*subscriberScopeServiceCounterStatus*

*subscriberScopeServiceCounterName*

*}*

## subscriberScopeServiceCounterIndex (subscriberScopeServiceCounterEntry 1)

The subscriber-scope service counter index

Access not-accessible

**SYNTAX**

**INTEGER** *(1...255)*

### subscriberScopeServiceCounterStatus (subscriberScopeServiceCounterEntry 2)

The subscriber-scope service counter status

Access read-only

**SYNTAX**

**INTEGER** *{*

*0 (disabled)*

*1 (enabled)*

*}*

### subscriberScopeServiceCounterName (subscriberScopeServiceCounterEntry 3)

The name of the subscriber-scope service counter

Access read-only

**SYNTAX**

**DisplayString** *(SIZE 0...255)*

# Guidelines for Using the SCA BB MIB

This section provides guidelines to help access SNMP information on the SCE platform using the ***SCA BB*** MIB.

**Important 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, you should add 1 to the index of the entity. For example, the Global Counter with index 0 will be located at globalScopeServiceCounter index 1.

**Note** Although volume counters on the SCE platform hold 32-bit integers, 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, the `accelerate-packet-drops` feature should be disabled 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 the *Cisco Service Control Engine Software Configuration Guide*.

## globalScopeServiceCounterTable and subscriberScopeServiceCounterTable

The index of a service counter as defined in the ***SCA BB*** service configuration is used to reference services in the ***SCA BB*** MIB. Since MIB index values count from 1, while ***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, `LinkServiceTable.lnkServiceUpVolume` (part of the `linkGrp`) should be used. 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 SCAS BB Console (see the *Using the Service Configuration Editor: Traffic Classification* chapter in the *Cisco Service Control Application for Broadband User Guide*). For example, to reference the P2P service (which has a (default) service index of 9) in the MIB, a `serviceIndex` of 10 (= 9 + 1) must be used.

## packageCounterTable

The package index, defined in the ***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 SCAS BB Console (see the *Using the Service Configuration Editor: Traffic Control* chapter in the *Cisco Service Control Application for Broadband User Guide*). For example, to reference the Default Package (which has a package index of 0) in the MIB, a `packageIndex` of 1 (= 0 + 1) must be used.

## Accessing Subscriber Information (the spvIndex)

In order to collect subscriber level information using the SNMP interface, you must first create an entry in the `subscriberPropertiesValuesTable` part of the `subscriberGrp` in `pcubeSEMib` (not `PCubeEngageMib`). Once an entry in this table is created and associated with a subscriber name, its index (`spvIndex`) can then be referred to in `PCubeEngageMib` to collect usage statistics for this subscriber.

An entry is created in the `subscriberPropertiesValuesTable` table by setting the entry `spvRowStatus` object with `CreateAndGo(4)` then setting the name of the subscriber in the `spvSubName` property and the `spvIndex` variable to be used as an index to the subscriber.

For example, to poll the downstream volume of subscriber “sub123” for the P2P service using `PCubeEngageMib`, do the following:

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

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

**Step 3** Set the object indices:

- For `moduleIndex` use 1
- Set `spvIndex` to the desired value [in this example we will use 1]

**Step 4** Set `spvRowStatus` to 4 (using `CreateAndGo`).

**Step 5** Set `spvSubName` to “sub123”.

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

# Glossary of Terms

## A

### Active subscriber

An online subscriber who is actually generating IP traffic.

### Anonymous subscriber mode

A mode in which entities defined as IP addresses or VLANs are treated as subscribers. The correlation to actual subscriber IDs is not performed by the system, but can be performed externally by the collection system. Anonymous subscriber mode does not require an SCMS-SM.

## C

### CLI

One of the management interfaces to the SCE platform. It is accessed through a Telnet session or directly via the console port on the front panel of the SCE platform.

### CM

A software application running on a Solaris or Linux platform that is responsible for receiving RDRs from the SCE platform and processing them.

### Command-Line Interface

*See* CLI.

## D

### Domains

A group of SCE platforms that share a group of subscribers. The subscriber traffic can pass through any SCE platform in the domain.

A subscriber and an SCE can belong to only one domain.

### Downstream traffic

Traffic entering the SCE platform from the network side (that is, toward the subscribers).

## E

### External Quota Management

Provisioning of per-service quotas for individual subscribers by an external system, such as a pre-paid server or a policy-controller.

In External Quota Provisioning, usage counters are not automatically reset at the end of an aggregation period, nor is a specific quota limit provided uniformly to all subscribers as part of the package Parameters. Rather the quotas are provisioned individually via the external Quota Management system.

# F

## Flow

All packets traveling in both directions on a single application layer connection (such as a TCP or UDP connection). A flow is identified by the tuple information: <Source IP, Destination IP, Source Port, Destination Port, IP Protocol>. (Note that if the IP protocol is neither TCP nor UDP, the port number is defined as '0'.)

In this guide, the term 'flow' represents bidirectional flows (packets from both the client and server of each connection). When referencing a unidirectional flow, this is explicitly mentioned.

## Flow bundle

A group of one or more flows comprising the set of application-layer connections (such as a TCP or UDP connection) used in a single, logical application session. The semantics of flow-bundles are application dependant, and relate to the way each application spawns and negotiates additional flows as part of a single session. A few common examples are:

- An SIP (VoIP) flow bundle comprises the signaling flow as well as all the RTP/UDP flows containing the actual media data (voice)
- An RTSP (Streaming) flow bundle comprises the signaling flow as well as the RTP/UDP flows containing the audio and video transmissions
- AN FTP (file transfer) flow bundle comprises the control flow (used to login to an FTP server) and the actual file-transfer flows

In each of these cases, the SCE platform tracks the application communication to identify new connections created and bundle them into a single context. This is important for classification and accounting purposes, as otherwise these spawned flows would be unclassifiable.

# O

## Online subscriber

A subscriber that is currently online. At any particular time, a number of online subscribers will be idle.

# P

## Package

A collection of business policy rules, defining access levels to various services, charging parameters, and traffic control actions to be taken upon predefined events. Subscribers are assigned packages (plans) that determine how their network transactions are controlled and charged.

# Q

## Quota

A (subscriber's) limit for a specific metric, such as bandwidth or volume.

## Quota buckets

When the external quota management mode is selected, subscriber usage of a service is consumed from a predefined subscriber quota bucket. Each subscriber has four subscriber quota buckets. When a quota bucket is depleted, services that try to consume from that bucket are *breached*.

A quota manager that is external to the Service Control system replenishes quota buckets.

# R

## Raw Data Record

*See* RDR.

### RDR

A data record produced by the SCE platform that reports on events in the traffic. RDRs produced by the SCE platform are sent to the Cisco Service Control Management Suite Collection Manager and then stored in the Collection Manager database or forwarded to third-party systems. The RDR typically contains a quota (*see* Quota) request or reports service usage.

### RDR Formatter

An internal component of the SCE platform that gathers the Raw Data Records (RDRs), formats them, and sends them to an external Cisco Service Control Management Suite Collection Manager.

### Real-time subscriber usage monitoring

Subscribers are monitored in detail; usage information is frequently reported by the SCE platform to facilitate detailed reports.

## S

### SCAS BB Console

The user interface used for controlling the Cisco Service Control Application for Broadband; used to create, modify, and apply service configurations.

### SCE platform

The SCE platform is a purpose-built service component and active enforcing system designed for enhancing service providers and backbone carrier networks. By identifying, classifying, and manipulating complex traffic flows at wire-speed, the SCE platform transforms simple transport networks into differentiated service delivery infrastructures for a wide variety of value-added IP applications, such as video streaming, VoIP, tiered services, and bilateral application-level SLAs.

The SCE platform seamlessly interfaces with existing network elements—including routers, switches, aggregators, subscriber management devices, and operational support systems—using industry standard interfaces and communications protocols.

The need to guarantee that packets passing through the network are processed at the rate they arrive makes it necessary to provide a custom-made hardware solution.

The SCE platform comes in three models—SCE 1000, SCE 2000 4xGBE, and SCE 2000 4/8xFE. There may be one or more SCE platforms in the provider network. Within the SCE platform, network transactions are analyzed and mapped to services that enforce the provider's policies.

In addition, the SCE platform implements the business logic of the system solution and performs transaction analysis in real-time. When so instructed, the SCE platform creates a Raw Data Record (RDR) to be sent for storage to the system's data repository, the Cisco Service Control Management Suite Collection Manager; or carries out other operations such as bandwidth or volume control.

### Service

A value-added offering given by the service provider to its subscribers on top of its access network.

For each such commercial service the providers offer to their subscribers, a corresponding service is defined in the Cisco Service Control solution for classifying and identifying network transaction associated with the service, reporting on its usage, and controlling its traffic according to the business policy.

### Service Configuration

The definition of services within the Cisco Service Control solution, the mapping of network transactions to their corresponding services, and the behavior of the SCE platform on them. The service configuration includes the definition of services, packages, Bandwidth Controllers, filter rules, and so on.

### Service Control

The basic Cisco concept for enabling service providers to differentiate subscribers, detect real-time events, create premium services, actively control applications, and leverage their existing infrastructure.

### Service Control Engine platform

*See* SCE platform.

### Service Control Management Suite Collection Manager

*See* CM.

### Service Control Management Suite Subscriber Manager

*See* SM.

### Service rule

A service is assigned to a package by defining a service rule for the package.

### Session (also called Transaction)

An instance of communication between network hosts. A precise definition of a session is application protocol (Layer 7) dependent.

### Signature

A set of parameters that uniquely identify a protocol.

### SM

A middleware software component used in cases where dynamic binding of subscriber information and service configurations is required. The SM manages subscriber information and provisions it in real time to multiple SCE platforms. It can store subscriber service configurations information internally, and act as a state-full bridge between the AAA system (for example, RADIUS and DHCP) and the SCE platforms.

### Subscriber

Service Provider client. There are two types of subscribers:

- Introduced Subscriber—A specific customer with an externally generated name. May be mapped to more than one IP address.
- Anonymous subscriber group—A subscriber with an internally generated name, generated automatically by the SCE platform according to an anonymous subscriber group specification. Always mapped to a single IP address. The actual identity of the subscriber is unknown to the system.

### Subscriber-initiated transactions

Transactions that are initiated by a host of a subscriber.

### Subscriberless mode

A mode of the Cisco Service Control solution that requires no integration, so that the SCMS-SM is not required. This mode is not influenced by the number of subscribers or inbound IP addresses; the total number of subscribers using the monitored link is unlimited from the perspective of the SCE platform. It is the choice for sites where control and level analysis functions are required only at a global platform resolution.

## T

### Traffic Discovery Reports

Statistics reports on network activity based on transaction usage records.

### Transaction (also called Session)

An event in traffic that is recognized by a service control application. A transaction is distinguished according to its L3, L4, or L7 characteristics. Different protocols may have different transaction types.

## U

### Upstream traffic

Traffic entering the SCE platform from the subscriber side.

# Index

## H



## I



## L



## M



## O



## P

## Q



## R



## S

## T



## U



## V



## W



## Z