JUNIPER NETWORKS

SRC Software

Application Library Guide

Release

4.0.x

Published: 2010-05-18

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

SRC Software Application Library Guide
Release 4.0.x

Writing: Linda Creed, Justine Kangas, Betty Lew, Helen Shaw
Editing: Fran Mues
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
May 2010—Revision 1

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

END USER LICENSE AGREEMENT

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html .

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

About the Documentation xix

Part 1 Installing Applications

Chapter 1 Installing the SRC Applications 3

Part 2 Providing Network Security and Threat Mitigation

Chapter 2 Providing Threat Mitigation Services with the Threat Mitigation Application 13

Chapter 3 Managing Threats with the SRC TMP 31

Part 3 Managing Network Resources

Chapter 4 Providing Application-Level Session Tracking and QoS Control 47

Part 4 Controlling Volume Usage with the SRC VTA

Chapter 5 Overview of Controlling Volume Usage with the SRC VTA 65

Chapter 6 Installing and Initially Configuring the SRC VTA 81

Chapter 7 Configuring the SRC VTA with VTA Configuration Manager 103

Chapter 8 Managing Subscriber Accounts with VTA Portals 179

Chapter 9 Example of a Bucket VTA 193

Part 5 Index

Index 207

Table of Contents

About the Documentation xix

SRC Documentation and Release Notes xix
Audience xix
Documentation Conventions xix
Documentation Feedback xxi
Requesting Technical Support xxi
Self-Help Online Tools and Resources xxii
Opening a Case with JTAC xxii

Part 1 Installing Applications

Chapter 1 **Installing the SRC Applications 3**

SRC Application Library Software 3
Before You Install the SRC Applications 4
Solaris Packages and Installation Folders for the SRC Application Library 5
Installing SRC Application Packages 5
Uninstalling SRC Packages 6
Installing Sample SRC Data for Applications in the Application Library Guide 6
Installing SRC Web Applications 7
Installing Web Applications Inside JBoss 7
Removing SRC Web Applications 8
Reviewing SRC Port Settings for SRC Applications 8

Part 2 Providing Network Security and Threat Mitigation

Chapter 2 **Providing Threat Mitigation Services with the Threat Mitigation Application 13**

Overview of the Threat Mitigation Application 13
Before You Install the Threat Mitigation Application 14
Sample Implementation 15
Installing and Initially Configuring the Threat Mitigation Application 16
Configuring Threat Mitigation 17
Configuring Database Properties for the Threat Mitigation Application 18
Configuring a Database to Store Attack and Response Data 18
Configuring Attack Types in the Database 19
ATTACK_TYPE Attributes Used by the Threat Mitigation Application 20
Configuring Actions in the Database 21
ACTION Attributes Used by the Threat Mitigation Application 21
Configuring Candidate Actions in the Database 23
ATTACK_TYPE_CANDIDATE_ACTION Attributes Used by the Threat Mitigation Application 23

Configuring Logging 24
Deploying the Threat Mitigation Application 25
Applying Services to Manage Threats 26
Examples: Classifying Subscribers and Interfaces for the Threat Mitigation Application 28
Example: Subscriber Classification Scripts 28
Example: Interface Classification Scripts for JUNOS Routing Platforms 29
Example: Interface Classification Scripts for JUNOSe Routers 29

Chapter 3 Managing Threats with the SRC TMP 31

Overview of the SRC TMP 31
About the Record Servlet 31
Overview of Configuring and Deploying the SRC TMP 33
Using the NIC Resolver for the SRC TMP 33
Configuring the NIC for Provider Edge Interfaces 33
Configuring the NIC for Forwarding Interfaces 33
Configuring the NIC for Subscriber Interfaces 34
Accessing the SRC TMP 34
Managing Attacks with the SRC TMP 36
Managing Attacks Requiring Action 36
Managing Attacks Pending Service Activation 37
Managing Attacks Pending Service Deactivation 38
Managing Attacks with Activated Services 40
Fields in the thm.py File 41

Part 3 Managing Network Resources

Chapter 4 Providing Application-Level Session Tracking and QoS Control 47

Overview of Application-Level Session Tracking and QoS Control 47
Benefits of Application-Level Session Tracking and QoS Control 47
Integration of the SRC Software and the Ellacoya DPI Platform 48
Ellacoya Networks DPI Platform 48
Juniper Networks Platforms 49
IPSCS Service Offers and Service Bundles 50
Mapping Service Offers and Service Bundles to SRC Concepts 50
Synchronization Between the SRC Software and the Ellacoya System 50
Collecting Accounting Data 51
Subscriber Login and Logout in a DPI Environment 52
Service Activation and Deactivation in a DPI Environment 52
Loading the Sample Data for the DPI 53
Configuring the SRC Software for DPI Integration 54
SRC Script Services for DPI 54
Adding a Service Scope 54
Creating a DPI Script Service 55
Configuring the Script Service 56

Configuring a Virtual Router Object for DPI . . . 58
Configuring Subscriptions to DPI Services . . . 59
Configuring the Ellacoya DPI Platform for Integration . . . 59
Provisioning the IPSCS . . . 59
Service Bundles . . . 60
Service Offers . . . 60
Traffic-Accounting Profiles . . . 60
Configuring the SLE . . . 60
Synchronizing System Clocks . . . 61

Part 4 Controlling Volume Usage with the SRC VTA

Chapter 5 Overview of Controlling Volume Usage with the SRC VTA . . . 65

Overview of the SRC VTA . . . 65
Types of VTAs . . . 65
Terminology . . . 65
VTA Service and Subscriber Accounts . . . 66
VTA Sessions . . . 67
Managing Subscriber Accounts with Portals . . . 67
Volume-Based Services . . . 67
SRC VTA Architecture and Connections to SRC Components . . . 68
How the SRC VTA Works . . . 68
Events . . . 69
Event Attributes . . . 69
Event Handlers . . . 70
Actions . . . 71
Processors . . . 71
Database Engine Processor . . . 72
Mail Processor . . . 72
SAE Proxy Processor . . . 72
Script Runner Processor . . . 72
SRC VTA Operation . . . 73
Managing VTA Accounts and Sessions . . . 74
Identifying Subscribers, SAEs, and Sessions . . . 74
Managing VTA Accounts and Sessions . . . 75
Using SRC VTA Keys to Manage Accounts and Sessions . . . 75
Managing Subscriber Sessions and Service Sessions . . . 76
Using SRC VTA Keys to Manage Subscriber and Server Sessions . . . 77
Example: Limiting Subscriber Access Based on Account Balances . . . 78

Chapter 6 Installing and Initially Configuring the SRC VTA . . . 81

Before You Install the SRC VTA . . . 81
Installing the SRC VTA and Running the Configuration Script . . . 82
Using JavaScript Programs in VTA Configurations . . . 83
Additional SRC VTA Configuration Script Tasks . . . 83
Configuring a Database to Store Account and Session Data . . . 84
Configuring the J2EE Application Server . . . 85
Creating Deployment Descriptors . . . 86
Example: Configuring the Oracle Database as the VTA Database . . . 87
Troubleshooting Database Deadlocks . . . 88

Configuring VTA Services and Policies 88
Configuring Subscribers and Subscriptions to VTA Services 89
Accessing the J2EE Application Server's Client Libraries 90
Specifying How the SRC VTA Loads Configurations from the Directory 90
Properties in ejb-jar.xml file 91
Configuring the SAE to Send Tracking Events to the SRC VTA 92
Specifying Tracking Plug-Ins for Enterprise Subscribers on JUNOS Routing Platforms 94
Configuring the Event Queue 95
Setting the Size of the Event Queue 95
Calculating the Size of the Nonpersistent Event Queue 96
Specifying the Type of Event Queue 96
Event Queue Property 97
Configuring the Event Queue Size 97
Using NICs with the SRC VTA 98
Locating the SAE That Manages a Subscriber for the SRC VTA 98
Configuring a NIC 99
Configuring NIC Proxies for the VTA 99
Renaming a VTA 100
Modifying the VTA Renaming Rules 100

Chapter 7 **Configuring the SRC VTA with VTA Configuration Manager 103**

Installing VTA Configuration Manager 104
Running VTA Configuration Manager 104
Loading and Importing VTA Configurations 105
Loading a Configuration from a Directory 106
Connecting to the Directory Fields 108
Importing a VTA Configuration from a Local File 109
Accessing the VTA Configuration 110
Configuring the SRC VTA to Manage Database Accounts 111
Configuring Scripts That Update Accounts 114
Account Update Script Fields 114
Configuring the SRC VTA to Manage Subscriber Accounts 115
Database Engine Processor Buckets 116
Configuring a Usage Metric for Service Accounts 117
Defining a Formula for Determining Network Resource Usage That the SRC VTA Evaluates 119
Sample Formulas for Usage Metrics for the SRC VTA 120
Configuring an Interim Accounting Interval for Service Accounts 121
Database Engine Processor Fields 122
Adjusting the Interim Accounting Interval for a Service 123
Service Variables 124
Current Service Variables 124
Other Service Variables 125
Account Balance Variable 126
Sample Formulas for Interim Accounting Interval 126
Configuring Actions for the Database Engine Processor 126
Action Fields for the Database Engine Processor 129

Setting Up the SRC VTA to Send E-Mail Notifications 130
Mail Processor Field 132
Configuring the SRC VTA to Send E-Mail Notifications 132
E-Mail Notification Fields 134
Configuring the SAE Proxy Processor 136
Configuring Actions for the SAE Proxy Processor 137
StartBehavingService Fields 139
Configuring the SRC VTA to Run Scripts 143
Configuring JavaScript Programs 143
JavaScript Fields 146
Configuring External Scripts 147
External Script Fields 149
Configuring VTA Actions to Run Scripts 150
RunJavaScript Fields 152
Configuring Events 154
Available Events Field 156
Configuring Event Handlers 156
Event Handler Fields 160
Configuring Identifiers for Subscribers and Sessions 161
Subscriber ID and Lookup Fields 162
Using One VTA Account for Multiple Subscriber Sessions 163
Logging Event Messages for the SRC VTA 165
Logging Events Messages to a Text File 166
File Logging Fields 168
Logging Events Messages to a System Logging Server 171
System Logging Fields 172
Validating VTA Configurations 174
Committing a VTA Configuration to a Directory 175
Exporting a VTA Configuration to a Local File 176

Chapter 8 Managing Subscriber Accounts with VTA Portals 179

Overview of Managing Subscriber Accounts with VTA Portals 179
Automatic Login of Subscribers 180
Configuring Web Applications for the SRC VTA 180
Properties for VTA Portals 181
Managing Subscriber Accounts with the Administrator Portal 184
Accessing the Administrator Portal 184
Viewing Subscriber Accounts 185
Replenishing Periodic Accounts 186
Deleting Information from the VTA's Database 186
Testing the VTA Configuration 187
Allowing Subscribers to Manage Their Accounts with the Subscriber Portal 187
Accessing the Subscriber Portal 188
Viewing Information About the Account 189
Purchasing a Periodic Account 189
Suspending a Periodic Account 190
Purchasing Extra Bandwidth 191

Chapter 9 **Example of a Bucket VTA 193**

Example of a Bucket VTA 193
Overview of Bucket VTA Example 193
Events for Bucket VTA 193
Event Handlers for Bucket VTA 194
GetBucket Event Handler 195
RefillBucketWithBehavingRate Event Handler 196
UpdateBehavingUsage Event Handler 196
ToMisbehaving Event Handler 197
Database Engine Processor for Bucket VTA 197
Account Update Scripts 197
Subscriber Account 198
Service Accounts 198
SAE Proxy Processor for Bucket VTA 199
Actions for Bucket VTA 201
GetBucketBalance Action 201
CalcUsage Action 202
UpdateBucketForBehaving Action 202
RefillBucketWithBehavingRate Action 203
StartMisbehavingService Action 203
StopBehavingService Action 203

Part 5 Index

Index 207

List of Figures

Part 3 Managing Network Resources

Chapter 4 **Providing Application-Level Session Tracking and QoS Control 47**

Figure 1: DPI Integration Overview 49

Part 4 Controlling Volume Usage with the SRC VTA

Chapter 5 **Overview of Controlling Volume Usage with the SRC VTA 65**

Figure 2: SRC VTA Architecture and Position in the SRC Network 68
Figure 3: VTA Event Handler Model 71
Figure 4: Operation of the SRC VTA 73

List of Tables

About the Documentation xix

Table 1: Notice Icons xx
Table 2: Text Conventions xx

Part 1 Installing Applications

Chapter 1 **Installing the SRC Applications 3**

Table 3: Application Library Applications 3
Table 4: Solaris Packages and Installation Folders for Application Library 5
Table 5: Default Port Settings for SRC Applications 8

Part 3 Managing Network Resources

Chapter 4 **Providing Application-Level Session Tracking and QoS Control 47**

Table 6: Synchronization Between the SRC Software and the Ellacoya System 50
Table 7: Parameter Definitions for DPI Services 56

Part 4 Controlling Volume Usage with the SRC VTA

Chapter 5 **Overview of Controlling Volume Usage with the SRC VTA 65**

Table 8: SRC VTA Terms 66
Table 9: Event Attributes 69
Table 10: Keys That the SRC VTA Constructs to Manage Accounts and Sessions 76
Table 11: Keys That the SRC VTA Constructs to Manage Subscriber andService Sessions 77

Chapter 6 **Installing and Initially Configuring the SRC VTA 81**

Table 12: Names for Data Sources and JMS Queues 85
Table 13: Settings for Filter Strings 93

Chapter 7 **Configuring the SRC VTA with VTA Configuration Manager 103**

Table 14: Examples of Formulas That Calculate Use of Network Resources 120
Table 15: Examples of Interim Accounting Interval 126
Table 16: Named Severity Levels 168
Table 17: Examples of Filters for Event Messages 170
Table 18: Named Severity Levels 172
Table 19: Examples of Filters for Event Messages 174

About the Documentation

- SRC Documentation and Release Notes on page xix
- Audience on page xix
- Documentation Conventions on page xix
- Documentation Feedback on page xxi
- Requesting Technical Support on page xxi

SRC Documentation and Release Notes

For a list of related SRC documentation, see http://www.juniper.net/techpubs/.

If the information in the latest *SRC Release Notes* differs from the information in the SRC guides, follow the *SRC Release Notes*.

Audience

This documentation is intended for experienced system and network specialists working with routers running JUNOS® and JUNOSe Software in an Internet access environment. We assume that readers know how to use the routers, directories, and RADIUS servers that they will deploy in their SRC networks. If you are using the SRC software in a cable network environment, we assume that you are familiar with the PacketCable Multimedia Specification (PCMM) as defined by Cable Television Laboratories, Inc. (CableLabs) and with the Data-over-Cable Service Interface Specifications (DOCSIS) 1.1 protocol. We also assume that you are familiar with operating a multiple service operator (MSO) multimedia-managed IP network.

Documentation Conventions

Table 1 on page xx defines the notice icons used in this guide. Table 2 on page xx defines text conventions used throughout this documentation.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2: Text Conventions

Convention	Description	Examples
Bold text like this	• Represents keywords, scripts, and tools in text. • Represents a GUI element that the user selects, clicks, checks, or clears.	• Specify the keyword **exp-msg**. • Run the **install.sh** script. • Use the **pkgadd** tool. • To cancel the configuration, click **Cancel**.
Bold text like this	Represents text that the user must type.	**user@host# set cache-entry-age *cache-entry-age***

`Fixed-width text like this`	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	`nic-locators {`
		`  login {`
		`  resolution {`
		`  resolver-name /realms/`
		`  login/A1;`
		`  key-type LoginName;`
		`  value-type SaeId;`
		`  }`

Regular sans serif typeface	• Represents configuration statements. • Indicates SRC CLI commands and options in text. • Represents examples in procedures. • Represents URLs.	• **system ldap server{ stand-alone;** • Use the **request sae modify device failover** command with the **force** option • **user@host# . . .** • **http://www.juniper.net/techpubs/software/management/src/api-index.html**
Italic sans serif typeface	Represents variables in SRC CLI commands.	**user@host# set local-address *local-address***
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.

Table 2: Text Conventions *(continued)*

Key names linked with a plus sign (+)	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
Italic typeface	• Emphasizes words. • Identifies book names. • Identifies distinguished names. • Identifies files, directories, and paths in text but not in command examples.	• There are two levels of access: *user* and *privileged*. • *SRC PE Getting Started Guide* • *o=Users, o=UMC* • The */etc/default.properties* file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	Plugin.radiusAcct-1.class=\ net.juniper.smgt.sae.plugin\ RadiusTrackingPluginEvent
Words separated by the | symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	diagnostic | line

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf .
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/ .
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/ .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html .

PART 1

Installing Applications

- Installing the SRC Applications on page 3

CHAPTER 1

Installing the SRC Applications

- SRC Application Library Software on page 3
- Before You Install the SRC Applications on page 4
- Solaris Packages and Installation Folders for the SRC Application Library on page 5
- Installing SRC Application Packages on page 5
- Uninstalling SRC Packages on page 6
- Installing Sample SRC Data for Applications in the Application Library Guide on page 6
- Installing SRC Web Applications on page 7
- Installing Web Applications Inside JBoss on page 7
- Removing SRC Web Applications on page 8
- Reviewing SRC Port Settings for SRC Applications on page 8

SRC Application Library Software

You can access the software for the SRC application library, and the product *Release Notes* on the Juniper networks Web site at: https://www.juniper.net/support/csc/swdist-erx/src.html. The application library files are located in the SRC_APLIB.tar.gz file. Table 3 on page 3 lists the applications provided in the application library.

Table 3: Application Library Applications

Application	Type of Application	File or Directory in Archive File
Deep Packet Inspection Script Service	Solaris package	UMCdpi-ss
Dynamic Service Activator	Web application	dsa.war
Threat Mitigation Application	Solaris package	UMCthma
VTA	Solaris package	/vta/UMCvtacnf
Bucket VTA	Web application	/vta/bucketvta.ear

Table 3: Application Library Applications *(continued)*

Application	Type of Application	File or Directory in Archive File
Quota VTA	Web application	/vta/quotavta.ear
VTA Configuration	Web application	/vta/config.war

The SRC software also provides sample applications in the SDK+AppSupport+Demos+Samples.tar.gz on the Juniper networks Web site at: https://www.juniper.net/support/csc/swdist-erx/src.html .

For information about the sample applications, see the *SRC PE Sample Applications Guide* on the Juniper Networks Web site at https://www.juniper.net/support/csc/swdist-erx/src.html .

The SRC Admission Control Plug-In (SRC ACP) component is installed on C Series Controllers as part of the SRC core software. The documentation for SRC ACP is in the *SRC PE Network Guide*.

Related Topics

- Before You Install the SRC Applications on page 4
- Installing Sample SRC Data for Applications in the Application Library Guide on page 6
- Solaris Packages and Installation Folders for the SRC Application Library on page 5
- Installing SRC Application Packages on page 5

Before You Install the SRC Applications

Before you install applications, install necessary Solaris patches on the installation host, make sure that you understand if you want to root or nonroot users to have access to install and configure the application, and establish users and groups for software administration.

Before you install the UMCthma or UMCvtacnf package, install JBoss on the Solaris host. You can obtain the JBoss software in the SDK+AppSupport+Demos+Samples.tar.gz file available on the Juniper networks Web site at: https://www.juniper.net/support/csc/swdist-erx/src.html.

The SRC applications are distributed as Solaris packages or Web application archive (WAR) files.

Related Topics

- SRC Application Library Software on page 3
- Installing SRC Application Packages on page 5
- Solaris Packages and Installation Folders for the SRC Application Library on page 5

Solaris Packages and Installation Folders for the SRC Application Library

Table 4 on page 5 lists the components for each application, their Solaris package names, and the directories where each component is installed by default. In Table 4 on page 5, the directories listed are all subordinate to */opt/UMC*.

Table 4: Solaris Packages and Installation Folders for Application Library

Application	Components Supplied with SRC	Package	Installation Directory
Threat Mitigation Application	• Threat Mitigation Application • Python Runtime Environment	• UMCthma • SMCpython	• conf/thma • python
VTAs	• VTAs • Python Runtime Environment (includes Python additional libraries)	• UMCvtacnf • SMCpython and UMCpyadd	• conf/vta • python

Related Topics

- SRC Application Library Software on page 3
- Before You Install the SRC Applications on page 4
- Installing SRC Application Packages on page 5
- Installing Sample SRC Data for Applications in the Application Library Guide on page 6
- Uninstalling SRC Packages on page 6

Installing SRC Application Packages

To install an application library package:

1. On the UNIX host where you will install the application library software, log in as root.
2. Copy the Solaris package for an application to a directory such as /tmp.
3. Launch the **pkgadd** tool.

   pkgadd -d /tmp

   The tool displays the available Solaris packages.
4. Enter the desired package(s).

   You can enter the name or number for a single package, multiple packages separated by spaces, or the keyword **all** to select all the packages.

   The tool displays the license agreement.

5. Press Enter to move through the agreement, and then enter **y** to accept the license agreement when prompted by the tool.
6. Follow the prompt directions to accept the installation directory for the package, to permit the use of superuser scripts required for the package, and so on.

NOTE: You can use the UNIX **swmtool** command to install the application packages, but this method requires that you install each application separately. If you use **admintool** directly, you can install multiple applications at the same time.

Related Topics

- SRC Application Library Software on page 3
- Before You Install the SRC Applications on page 4
- Installing SRC Web Applications on page 7
- Uninstalling SRC Packages on page 6
- Solaris Packages and Installation Folders for the SRC Application Library on page 5
- Reviewing SRC Port Settings for SRC Applications on page 8

Uninstalling SRC Packages

Use the **pkgrm** command to uninstall application library components. For example, to remove the SRC VTA package, issue the following command, and respond as prompted by the process:

```
pkgrm UMCvtacnf
```

Related Topics

- SRC Application Library Software on page 3
- Installing SRC Application Packages on page 5
- Solaris Packages and Installation Folders for the SRC Application Library on page 5

Installing Sample SRC Data for Applications in the Application Library Guide

You can install sample data from the SRC CLI for the following applications:

- Dynamic Service Activator application
- Traffic-Mirroring Application

Related Topics

- SRC Application Library Software on page 3
- Before You Install the SRC Applications on page 4
- Installing SRC Application Packages on page 5
- Loading Sample Data in to a Juniper Networks Database (SRC CLI)
- Overview of Dynamic Service Activator

Installing SRC Web Applications

We supply one WAR file for each Web application in the application library. Web applications must be deployed in a Web application server.

The exact way you install Web applications depends on the Web application server you are using and the particular Web application.

The following procedure provides general steps for installing a Web application:

1. Install the Web application server on the host.
2. If the Web application requires configuration of a properties file, complete the following procedure:
   a. Copy the WAR file from the archive file to a temporary folder on the host.
   b. Unpack the WAR file.

      For information about unpacking and packing WAR files, see http://java.sun.com/j2se/1.4/docs/guide/jar/.

   c. Edit the properties file for the Web application.
   d. Repack the WAR file.
3. Deploy the WAR file by using the procedure appropriate for your Web application server.

   For information about deploying WAR files, see the documentation for your Web application software.

Related Topics

- SRC Application Library Software on page 3
- Before You Install the SRC Applications on page 4
- Installing Web Applications Inside JBoss on page 7
- Removing SRC Web Applications on page 8
- Reviewing SRC Port Settings for SRC Applications on page 8

Installing Web Applications Inside JBoss

We provide the JBoss Web application server with the sample and demonstration applications in the SDK+AppSupport+Demos+Samples.tar.gz file. available from the Juniper Web site at: https://www.juniper.net/support/csc/swdist-erx/src.html.

JBoss is an open-source Java application server that provides full support for J2EE application programming interfaces (APIs). To deploy a Web application inside JBoss:

1. Install the *UMCjboss* package.
2. During the installation, choose a JBoss configuration when prompted; typically choose the default configuration.
3. Customize the properties file for the Web application.
4. Deploy the WAR file by copying it into the JBoss *default/deploy* directory.

   cp <filename>.war /opt/UMC/jboss/server/default/deploy

JBoss automatically starts the Web application when a new WAR file is copied into the deploy directory.

Related Topics

- SRC Application Library Software on page 3
- Before You Install the SRC Applications on page 4
- Installing SRC Web Applications on page 7
- Removing SRC Web Applications on page 8
- Reviewing SRC Port Settings for SRC Applications on page 8

Removing SRC Web Applications

The way you remove a Web application depends on the Web application server that you are using. Refer to the documentation on removing Web applications for your server.

To undeploy a Web application from JBoss:

- Remove the WAR file from the JBoss *default/deploy* directory.

Related Topics

- SRC Application Library Software on page 3
- Installing SRC Web Applications on page 7
- Installing Web Applications Inside JBoss on page 7

Reviewing SRC Port Settings for SRC Applications

If you use firewall software within your internal network, ensure that firewall settings allow traffic to and from the following ports for the applications that you implement in your environment. Table 5 on page 8 lists the default port settings for SRC applications.

Table 5: Default Port Settings for SRC Applications

Component	Type of Communication	Default Port Setting
Threat Mitigation Application	Oracle (if used)	TCP 1521
	PostgreSQL (if used)	TCP 5432
	MySQL (if used)	TCP 3306

Table 5: Default Port Settings for SRC Applications *(continued)*

Component	Type of Communication	Default Port Setting
SRC Volume-Tracking Application (SRC VTA)	MySQL (if used)	TCP 3306

Related Topics

- SRC Application Library Software on page 3
- Before You Install the SRC Applications on page 4
- Installing SRC Application Packages on page 5
- Installing Sample SRC Data for Applications in the Application Library Guide on page 6
- Installing SRC Web Applications on page 7

PART 2

Providing Network Security and Threat Mitigation

- Providing Threat Mitigation Services with the Threat Mitigation Application on page 13
- Managing Threats with the SRC TMP on page 31

CHAPTER 2

Providing Threat Mitigation Services with the Threat Mitigation Application

- Overview of the Threat Mitigation Application on page 13
- Before You Install the Threat Mitigation Application on page 14
- Sample Implementation on page 15
- Installing and Initially Configuring the Threat Mitigation Application on page 16
- Configuring Threat Mitigation on page 17
- Configuring Database Properties for the Threat Mitigation Application on page 18
- Configuring Logging on page 24
- Deploying the Threat Mitigation Application on page 25
- Applying Services to Manage Threats on page 26
- Examples: Classifying Subscribers and Interfaces for the Threat Mitigation Application on page 28

Overview of the Threat Mitigation Application

The Threat Mitigation Application helps administrators detect and respond to attacks on the network. The Threat Mitigation Application can be customized based on customer-supplied data to control the description and recommended actions for each type of attack. If the user chooses to take an action, the Threat Mitigation Application activates a service for the source address of the event. The Threat Mitigation Application includes the ability to log all user operations to provide an audit trail of actions.

You can use the Threat Mitigation Application to respond to threats on the network by:

- Executing a script for Juniper Networks NetScreen-Security Manager that posts information about the attack to the SRC Threat Mitigation Portal (SRC TMP)
- Managing attacks with the SRC TMP that provides information about the nature of the attack and possible actions
- Applying policies to the interfaces to manage problem traffic, such as applying policies that reduce the amount of available bandwidth or that block the threat

The Threat Mitigation Application deals with threats in an SRC-managed environment by providing a solution that involves using:

- Juniper Networks Intrusion Detection and Prevention (IDP) sensors to detect the threats.

  IDP sensors are IDP hardware appliances that run the IDP sensor software. The sensors monitor network traffic to detect suspicious or anomalous traffic and respond as configured. IDP monitors network traffic to detect potentially detrimental traffic and responds to problem incidents to prevent damage to the network.

- Juniper Networks NetScreen-Security Manager to manage the IDP sensors and to signal the SRC TMP when a threat is detected.

  NetScreen-Security Manager is software that enables you to integrate and centralize management of your Juniper Networks security environment. NetScreen-Security Manager delivers integrated, policy-based security and network management for all Juniper Networks security devices. NetScreen-Security Manager is used for its elaborate authorization and auditing functionality, which provides more detailed reporting and analysis.

- The SRC TMP to display detailed information about the threat and the recommended actions to the administrator.

  The SRC TMP is the user interface for the Threat Mitigation Application that enables administrators to manage threats and act on them. The administrator can react to the threat by activating a service in the SAE. The service activation can result in pushing policies, for the originating IP address, to the upstream router running JUNOS Software in the core network or to the edge router running JUNOSe Software, depending on the configuration.

Related Topics

- Before You Install the Threat Mitigation Application on page 14
- Installing and Initially Configuring the Threat Mitigation Application on page 16
- Configuring Threat Mitigation on page 17
- Configuring Logging on page 24
- Examples: Classifying Subscribers and Interfaces for the Threat Mitigation Application on page 28

Before You Install the Threat Mitigation Application

Installing the Threat Mitigation Application into an SRC-managed environment requires:

- The UMCthma package installed with your SRC application library software.
- SRC-managed routers running JUNOSe or JUNOS Software in the network.
- Working knowledge of the NetScreen-Security Manager software and familiarity with NetScreen-Security Manager documentation. See http://www.juniper.net/techpubs/software/management/security-manager/.
- Working knowledge of the IDP software and familiarity with IDP documentation. See http://www.juniper.net/techpubs/software/management/idp/.

Before you use the Threat Mitigation Application, you typically:

- Install the transactional database. The Threat Mitigation Application provides a sample schema that includes these tables:
  - ATTACK—Attacks
  - ATTACK_TYPE—Attack types
  - ACTION—Configured actions that the application can execute
  - ATTACK_TYPE_CANDIDATE_ACTION—Candidate actions that can be taken in response to attack types

    The administrator maintains the data in the ATTACK_TYPE, ACTION, and ATTACK_TYPE_CANDIDATE_ACTION tables to ensure that the data defines the relationship between attack types and candidate actions. In cases where attacks do not belong to any defined attack types, the administrator should create a default attack type and the candidate actions for the default attack type.
- Install the IDP sensors. The sensors monitor network traffic to detect suspicious or anomalous traffic and respond as configured.
- Install NetScreen-Security Manager to monitor the IDP sensors. The administrator creates the attack types that are reported to the Threat Mitigation Application.

Related Topics

- Overview of the Threat Mitigation Application on page 13
- Sample Implementation on page 15
- Applying Services to Manage Threats on page 26
- Configuring Logging on page 24
- Deploying the Threat Mitigation Application on page 25

Sample Implementation

The SRC application library provides a robust sample implementation for mitigating threats using the Threat Mitigation Application in an SRC-managed network.

The sample implementation includes:

- Policies, services, router definitions, and SAE configurations in the sample data. Sample entries for the Threat Mitigation Application have the prefix THMA.

  For information about installing sample data, see “Installing and Initially Configuring the Threat Mitigation Application” on page 16.
- Data in the schema to detail attacks, actions that can be executed by the application, and actions that can be used to respond to attacks.

You can use the sample data and application to create a demonstration implementation. The router definitions, identified as THMA<routername> in the sample data, can be configured to act as simulated routers for a demonstration environment. For information about setting up a simulated router, see Configuring Simulated Router Drivers (SRC CLI).

You can also customize the sample data to mitigate threats in your network, or you can use the samples as a guide to create your own implementation.

The sample data uses the following terminology for the type of interface on which the service would be activated:

- Provider edge interface—Subscriber-facing interface on the router running JUNOS Software
- Forwarding interface—Forwarding interface on the router running JUNOS Software
- Subscriber interface—Subscriber interface on the router running JUNOSe Software

Related Topics

- Overview of the Threat Mitigation Application on page 13
- Configuring Threat Mitigation on page 17
- Configuring Logging on page 24
- Deploying the Threat Mitigation Application on page 25

Installing and Initially Configuring the Threat Mitigation Application

Because the Threat Mitigation Application relies on other components in the SRC network, you must complete several tasks before you install the Threat Mitigation Application software. After you install the software, you must also complete several configuration tasks before the application can function correctly.

For information about the location of the Threat Mitigation Application software, see "Installing SRC Application Packages" on page 5.

Before you install and configure the Threat Mitigation Application, you must:

1. Deploy a working SRC network.

   To support the Threat Mitigation Application, you must install SAEs to manage the routers or other devices through which subscribers connect to the network. You must also install and configure the directory in which you will store the SRC data.
2. Install a J2EE application server on the host that supports the Threat Mitigation Application.
3. On the host that supports the Threat Mitigation Application, install a transactional database to store the data for the Threat Mitigation Application.

To install the Threat Mitigation Application:

1. Install the Solaris package for the Threat Mitigation Application on a host that supports JBoss.

   See "Installing SRC Application Packages" on page 5.
2. Run the configuration script to configure the Threat Mitigation Application.

3. Run the load script to complete the configuration tasks for deploying the Threat Mitigation Application.

   See “Deploying the Threat Mitigation Application” on page 25.

4. On each host, restart JBoss.

   You must restart JBoss when you have configured the SRC TMP for the first time or you have changed the database type.

Related Topics

- Overview of the Threat Mitigation Application on page 13
- Before You Install the Threat Mitigation Application on page 14
- Configuring Threat Mitigation on page 17
- Configuring Logging on page 24

Configuring Threat Mitigation

To support threat mitigation with the Threat Mitigation Application in an SRC network, configure services that can be activated to act on threats detected by IDP sensors that are managed by NetScreen-Security Manager. We recommend that you activate the services as close as possible to the interfaces where the problem traffic entered the network.

To use the Threat Mitigation Application, perform the following tasks:

- Access the local configuration
- Deploy the application. See “Deploying the Threat Mitigation Application” on page 25
- Apply services. See “Applying Services to Manage Threats” on page 26.

Also see “Examples: Classifying Subscribers and Interfaces for the Threat Mitigation Application” on page 28.

Some sections provide references to entries in the sample data that demonstrate an implementation.

After performing these tasks, configure the script used by NetScreen-Security Manager to implement the messaging that records attacks and identifies actions that the SRC software should take in response to those attacks.

Related Topics

- Overview of the Threat Mitigation Application on page 13
- Before You Install the Threat Mitigation Application on page 14
- Installing and Initially Configuring the Threat Mitigation Application on page 16
- Configuring a Database to Store Attack and Response Data on page 18
- Configuring Logging on page 24

Configuring Database Properties for the Threat Mitigation Application

- Configuring a Database to Store Attack and Response Data on page 18
- Configuring Attack Types in the Database on page 19
- ATTACK_TYPE Attributes Used by the Threat Mitigation Application on page 20
- Configuring Actions in the Database on page 21
- ACTION Attributes Used by the Threat Mitigation Application on page 21
- Configuring Candidate Actions in the Database on page 23
- ATTACK_TYPE_CANDIDATE_ACTION Attributes Used by the Threat Mitigation Application on page 23

Configuring a Database to Store Attack and Response Data

The Threat Mitigation Application requires a transactional database to store attack types and candidate responses. For information about databases that we have tested for use with the Threat Mitigation Application, see the *SRC Application Library Release Notes*.

The Threat Mitigation Application provides sample data for a schema that includes these tables:

- ATTACK_TYPE—Contains information about the attacks that NetScreen-Security Manager is expected to send to the Threat Mitigation Application. The administrator maintains this data. See “Configuring Attack Types in the Database” on page 19.
- ACTION—Contains information about the SRC services that are activated to respond to attacks. The administrator maintains this data. See “Configuring Actions in the Database” on page 21.
- ATTACK_TYPE_CANDIDATE_ACTION—Contains information about the actions that can be taken in response to specific attack types. The administrator maintains this data. See “Configuring Candidate Actions in the Database” on page 23.
- ATTACK—Contains information about the attacks that are managed by the Threat Mitigation Application. The SRC TMP displays this information on various pages, including the Attack Details page. For information about how the SRC TMP displays the attributes, see“Overview of the Threat Mitigation Application” on page 13.

To use the Threat Mitigation Application, the administrator must create data in the ATTACK_TYPE, ACTION, and ATTACK_TYPE_CANDIDATE_ACTION tables to define the relationship between attack types and candidate actions. The information in the ATTACK table is managed by the Threat Mitigation Application and must not be modified by an administrator. The attributes specified in the tables are referenced in the XML schema for NetScreen-Security Manager attack events.

To configure the database:

1. Create a database, tables, and user for the database by using the following database schema file:

   /opt/UMC/conf/thma/etc/<database>/thma.sql

where <database> is the selected database when you run the load script. This file is created when you install the Solaris package for the Threat Mitigation Application.

2. Load the sample data for the database using the following file:

   /opt/UMC/conf/thma/etc/<database>/data.sql

   where <database> is the selected database when you run the load script. This file is created when you install the Solaris package for the Threat Mitigation Application.

Related Topics

- Installing and Initially Configuring the Threat Mitigation Application on page 16
- Configuring Logging on page 24
- ACTION Attributes Used by the Threat Mitigation Application on page 21
- ATTACK_TYPE Attributes Used by the Threat Mitigation Application on page 20
- ATTACK_TYPE_CANDIDATE_ACTION Attributes Used by the Threat Mitigation Application on page 23

Configuring Attack Types in the Database

The ATTACK_TYPE table contains data about all the attacks that NetScreen-Security Manager is expected to send to the Threat Mitigation Application. Attacks are considered to be the same attack type if their category, subcategory, and definingAttributes values are the same.

NOTE: The ATTACK_TYPE table must contain a special attack type with category and subcategory values of DEFAULT to respond to attacks that do not match a configured attack type.

The entry in the */opt/UMC/conf/thma/etc/<database>/data.sql* file contains the attributes in the format:

```
INSERT INTO ATTACK_TYPE
VALUES ('<category>', '<subcategory>', '<definingAttributes>', '<description>');
```

For example:

```
INSERT INTO ATTACK_TYPE
VALUES ('DEFAULT', 'DEFAULT', 'srcAddr', 'There is no specific information for this type
  of attack.');
```

Related Topics

- Overview of the Threat Mitigation Application on page 13
- Configuring a Database to Store Attack and Response Data on page 18
- ATTACK_TYPE Attributes Used by the Threat Mitigation Application on page 20
- ATTACK_TYPE_CANDIDATE_ACTION Attributes Used by the Threat Mitigation Application on page 23

ATTACK_TYPE Attributes Used by the Threat Mitigation Application

The SRC TMP displays the configured attributes for the attack types that are used by the Threat Mitigation Application.

category

- Category of the attack; displayed in the Attack Details page.
- Value—Text string
- Examples
  - DEFAULT
  - predefined

subcategory

- Subcategory of the attack; displayed in the Attack Type column.
- Value—Text string
- Examples
  - DEFAULT
  - FTP:USER:ROOT
  - ICMP:EXPLOIT:FLOOD

definingAttributes

- Attributes used to identify an attack. Defining attributes determine whether an attack is a new record or an update to an existing attack record. The srcAddr attribute is always considered a defining attribute for the attack, even if it is not specified as a defining attribute.
- Value—List of defining attributes separated by semicolons
  - srcAddr—Source address; displayed in the Source column
  - srcPort—Source port; displayed in the Attack Details page
  - dstAddr—Destination address; displayed in the Destination column
  - dstPort—Destination port; displayed in the Attack Details page
  - protocol—Protocol; displayed in the Attack Details page
  - user—User; displayed in the Attack Details page
  - app—Application; displayed in the Attack Details page
  - uri—Uniform resource identifier; displayed in the Attack Details page
- Examples

- srcAddr
- srcAddr;dstAddr;dstPort
- srcAddr;dstAddr

description

- Description of the attack; displayed in the Attack Details page.
- Value—Text string
- Examples
  - There is no specific information for this type of attack.
  - This attack indicates an ICMP session that contains more than 250 ICMP packets per second. This may indicate that an attacker is trying to degrade network performance, causing poor service for legitimate users.

Configuring Actions in the Database

The ACTION table contains data about services to activate in response to an attack. The administrator must add one ACTION table entry for each SRC service that is used as an action in the Threat Mitigation Application.

The entry in the */opt/UMC/conf/thma/etc/<database>/data.sql* file contains the attributes in the format:

```
INSERT INTO ACTION
VALUES ('<serviceName>', '<name>', '<description>');
```

For example:

```
INSERT INTO ACTION
VALUES ('BlockAttacker', 'Block Attacker', 'This action blocks all traffic to and from the
  attacker.');
```

Related Topics

- Overview of the Threat Mitigation Application on page 13
- For information about configuring actions, see the *SRC PE Services and Policies Guide*
- Configuring a Database to Store Attack and Response Data on page 18
- ACTION Attributes Used by the Threat Mitigation Application on page 21
- Configuring Candidate Actions in the Database on page 23

ACTION Attributes Used by the Threat Mitigation Application

The SRC TMP displays the configured attributes for the actions that are used by the Threat Mitigation Application.

serviceName

- Service activated in response to an attack.
- Value—Text string

The following values are passed to the service as parameter substitutions:

- category—Name of the category
  - subcategory—Name of the subcategory
  - severity—Severity level as a number in the range 0–5
    - 0—not set
    - 1—info
    - 2—warning
    - 3—minor
    - 4—major
    - 5—critical
  - srcAddr—IP address; enclose in single quotes if not in IPv4 format
  - srcPort—Port number
  - dstAddr—IP address; enclose in single quotes if not in IPv4 format
  - dstPort—Port number
  - protocol—Protocol number
  - user—Username
  - app—Name of the application
  - uri—Uniform resource identifier

The category, subcategory, user, app, and uri parameters are encoded as valid parameter names (not text strings) so that these parameter values can be provided to the policies.

For example, you could define a policy that takes the app parameter as the value for a policer rate with a default value of 64000. Then, you could define global parameters named after different applications, such as http=32000. When the attack includes an HTTP application, the Threat Mitigation Application would pass app=http, and 32000 would be the value in the policer definition.

- Example—BlockAttacker

name

- Name of action; displayed in the Action drop-down list.
- Value—Text string
- Example—Block Attacker

description

- Description of the action; displayed in the Action Help page.
- Value—Text string
- Example—This action blocks all traffic to and from the attacker.

Configuring Candidate Actions in the Database

The ATTACK_TYPE_CANDIDATE_ACTION table contains data about the possible services to activate in response to a particular type of attack.

The entry in the */opt/UMC/conf/thma/etc/<database>/data.sql* file contains the attributes in the format:

```
INSERT INTO ATTACK_TYPE_CANDIDATE_ACTION
VALUES ('<category>', '<subcategory>', '<serviceName>');
```

For example:

```
INSERT INTO ATTACK_TYPE_CANDIDATE_ACTION
VALUES ('DEFAULT', 'DEFAULT', 'BlockAttack');
```

Related Topics

- Overview of the Threat Mitigation Application on page 13
- For Information about configuring actions, see the *SRC PE Services and Policies Guide*
- Configuring a Database to Store Attack and Response Data on page 18
- Configuring Actions in the Database on page 21

ATTACK_TYPE_CANDIDATE_ACTION Attributes Used by the Threat Mitigation Application

The SRC TMP displays the configured attributes for the attack type and candidate actions.

category

- Category of the attack; displayed in the Attack Details page.
- Value—Text string
- Examples
  - DEFAULT
  - predefined

subcategory

- Subcategory of the attack; displayed in the Attack Type column.
- Value—Text string
- Examples
  - DEFAULT
  - FTP:USER:ROOT
  - ICMP:EXPLOIT:FLOOD

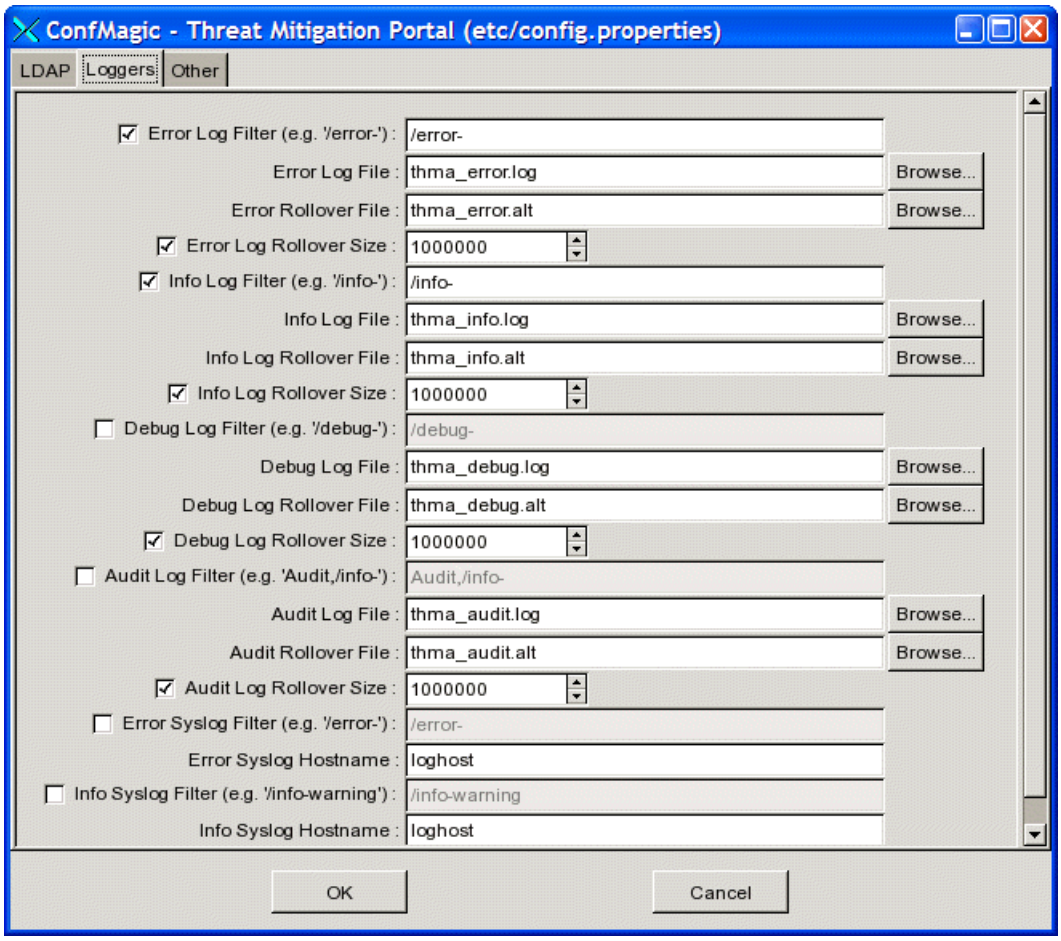
serviceName

- Service activated in response to an attack.
- Value—Text string
- Example—BlockAttacker

Configuring Logging

To configure logging for the Threat Mitigation Application:

- Edit or accept the default values for the fields in the Loggers tab.



For more information about logging, see the *SRC PE Monitoring and Troubleshooting Guide*.

Deploying the Threat Mitigation Application

The Threat Mitigation Application load script configures components (such as the J2EE application server, directory, and database) on the local host. However, depending on the components used, their installation host, and their configuration, you may need to manually configure some of the components or modify the configuration.

The Threat Mitigation Application load script automates the process of deploying the Threat Mitigation Application in JBoss (if it is installed locally) and completes these configuration tasks:

- Configures the *jbosscmp-jdbc.xml* file inside the */opt/UMC/conf/thma/webapp/thma.ear* file and the data source deployment descriptor based on the type of database specified and the database connection information.
- Installs the JDBC driver, data source deployment descriptor, and authentication configurations in JBoss (if JBoss is installed locally).

- Loads the Threat Mitigation Application sample data in the directory.
- Creates the database schema and loads sample database records. Follow the instructions at the end of the load script to complete the database configuration for the selected database. Some databases might require additional steps, such as creating a database user or enabling a remote TCP/IP connection.

To deploy the Threat Mitigation Application:

1. On the host, log in as root or as another authorized administrator.
2. Invoke the script by accessing the folder */opt/UMC/conf/thma/etc* and running the **load** command.

   ```
   cd /opt/UMC/conf/thma/etc
   ./load
   ```

3. Deploy the *thma.ear* file by using the procedure appropriate for your Web application server.

   If you are using JBoss, copy the file to the JBoss */default/deploy* directory. For example:

   ```
   cp /opt/UMC/conf/thma/webapp/thma.ear /opt/UMC/jboss/server/default/deploy
   ```

Related Topics

- Overview of the Threat Mitigation Application on page 13
- Sample Implementation on page 15
- Installing and Initially Configuring the Threat Mitigation Application on page 16
- Configuring Logging on page 24
- Applying Services to Manage Threats on page 26

Applying Services to Manage Threats

You can configure services to control problem traffic, such as limiting bandwidth or blocking traffic, in response to detection of malicious traffic. The Threat Mitigation Application passes the defining attribute values of the attack type to the service as parameters for possible use in the policies. The Threat Mitigation Application supports service activation on the JUNOS forwarding interface, the JUNOS provider edge interface, or the JUNOSe subscriber interface. You can configure only one of these interfaces as the service activation interface for the Threat Mitigation Application, but you can use an aggregate service to apply the policies on a combination of those interfaces.

The following example describes how to configure policies to decrease the amount of bandwidth available to the attacker and to block the attack or the attacker as implemented in the sample data. You can use any of these services or create your own services to define actions for the Threat Mitigation Application.

To configure services and policies to handle threats:

1. Create a policy that defines an action to be taken.

The sample data for each type of interface contains these policy groups:

- blockAttack—Blocks all traffic between the source and destination addresses for the specified protocol and ports. If the protocol or ports are not specified, then the default value is any protocol and any port.
- blockAttacker—Blocks all traffic coming from or going to the source address.
- default—Forwards traffic.
- slowAttacker—Limits the bandwidth available for all traffic coming from or going to the source address according to the specified rate.

For a policy folder that contains these policy groups for the JUNOS forwarding interface, see *ou=forwardingInterface, ou=thma, o=Policies, o=umc* in the sample data.

For a policy folder that contains these policy groups for the JUNOS provider edge interface, see *ou=peInterface, ou=thma, o=Policies, o=umc* in the sample data.

For a policy folder that contains these policy groups for the JUNOSe subscriber interface, see *ou=subrInterface, ou=thma, o=Policies, o=umc* in the sample data.

2. Create a new scope or use an existing scope for the services that define actions to be taken in response to attacks on different interfaces.

   For a sample scope that applies to the JUNOS forwarding interface, see *l=THMA-ForwardingInterface, o=Scopes, o=umc*.

   For a sample scope that applies to the JUNOS provider edge interface, see *l=THMA-PEnterface, o=Scopes, o=umc*.

   For a sample scope that applies to the JUNOSe subscriber interface, see *l=THMA-SubrInterface, o=Scopes, o=umc*.

3. For the scope used in Step 2:

   a. Create a service that defines actions to be taken in response to threats. You can create different types of services. For example, you can create aggregate services to apply the policies on these interfaces.

      The sample data contains normal services that specify the policy group configured in Step 1.

      For a sample service to block attacks on the forwarding interface, see *serviceName=BlockAttack, l=THMA-ForwardingInterface, o=Scopes, o=umc*.

   b. Assign the scope to a subscriber folder to make the service available to these subscribers.

      For a sample on the JUNOS forwarding interface, see *ou=routers, retailerName=SP-THMA, o=Users, o=umc*.

      For a sample on the JUNOS provider edge interface, see *ou=subscribers_peIf, retailerName=SP-THMA, o=Users, o=umc*.

For a sample on the JUNOSe subscriber interface, see *ou=subscribers_subrIf, retailerName=SP-THMA, o=Users, o=umc*.

4. Create service subscriptions for subscribers. In the sample data, we create a subscription at the folder level to allow all subscribers in the folder to inherit the subscription. Configure the subscriptions to manually activate the service through the SRC TMP.

   For a sample implementation, see *serviceName=BlockAttack, retailerName=SP-THMA, o=Users, o=umc* in the sample data.

   For information about configuring subscriptions, see Configuring Subscriptions (SRC CLI).

Related Topics

- Configuring Threat Mitigation on page 17
- Installing and Initially Configuring the Threat Mitigation Application on page 16
- Configuring Logging on page 24
- Examples: Classifying Subscribers and Interfaces for the Threat Mitigation Application on page 28

Examples: Classifying Subscribers and Interfaces for the Threat Mitigation Application

To apply policies to the forwarding interfaces, you configure additional entries in the subscriber classification and interface classification scripts.

Example: Subscriber Classification Scripts

In the subscriber classification script, threat mitigation requires the assignment of a subscriber profile for the forwarding interface and for any interface other than the forwarding interface (such as the provider edge interface on the router running JUNOS Software).

The Threat Mitigation Application needs to identify subscriber sessions in which to activate services persistently. These subscriber sessions should have a login name so that subscriber entries in the directory can be shared among the managed routers or interfaces. The login name must be unique. We recommend using the interface name and virtual router name to construct a unique login name. The login name must end in @<retailer's domain> and must not contain a / (slash) or another @ (at sign).

```
[routerName=commonRouterProfile,ou=routers,retailername=SP-THMA,o=Users,o=UMC?loginName=
   <-virtualRouterName.replace("@", "_")+"@thma"->??]
# host subscriber for JUNOS routers
interfaceName=="FORWARDING_INTERFACE"
```

This subscriber classification for the forwarding interface sets the virtual router name as the login name and thma as the service provider's domain name. The domain name must match the value of the Retailer Domain field specified when configuring the SRC TMP.

```
[uniqueID=DefaultTHMASubscriber,ou=subscribers,retailername=SP-THMA,o=Users,
   o=UMC?loginName=<-interfaceName.replace("@", "_").replace("/",
   "_")+"_"+virtualRouterName.replace("@", "_")+"@thma"->??]
```

```
# anything that is not the forwarding interface uses default subscriber
interfaceName!="FORWARDING_INTERFACE"
```

This subscriber classification for the provider edge interface sets the interface name as the login name.

To view the subscriber classifications referenced in this section, see *l=THMA, l=SAE, ou=staticConfiguration, ou=Configuration, o=Management, o=umc* in the sample data.

Example: Interface Classification Scripts for JUNOS Routing Platforms

An entry is needed in the interface classification script to specify the default policy for forwarding interfaces and provider edge interfaces on the routers running JUNOS Software. For example:

```
[policyGroupName=default,ou=forwardingInterface,ou=thma,o=Policies,o=UMC]
# manage router interface for mirroring
interfaceName=="FORWARDING_INTERFACE"
[policyGroupName=default,ou=peInterface,ou=thma,o=Policies,o=UMC]
# manage interfaces with an alias indicating
# an enterprise customer
interfaceName!="FORWARDING_INTERFACE"
```

To view the interface classifications referenced in this section, see the interface classification for the THMA<number> routers listed under *o=Network, o=umc* in the sample data.

Example: Interface Classification Scripts for JUNOSe Routers

An entry is needed in the interface classification script to specify the default policy for subscriber interfaces on the JUNOSe routers. For example:

```
# generic PPP users
[policyGroupName=default,ou=subrInterface,ou=thma,o=Policies,o=UMC]
pppLoginName!=""
# define DHCP interfaces here
[policyGroupName=DHCP,ou=junose,ou=sample,o=Policies,o=umc]
# all fastEthernet interfaces
interfaceName="fastEthernet*"
```

To view the interface classifications referenced in this section, see the interface classification for *orderedCimKeys=THMA_JUNOSE*, *o=Network, o=umc* in the sample data.

Related Topics

- Overview of the Threat Mitigation Application on page 13
- Overview of Classification Scripts

CHAPTER 3

Managing Threats with the SRC TMP

- Overview of the SRC TMP on page 31
- Overview of Configuring and Deploying the SRC TMP on page 33
- Accessing the SRC TMP on page 34
- Managing Attacks with the SRC TMP on page 36
- Fields in the thm.py File on page 41

Overview of the SRC TMP

The SRC TMP provided with the SRC software is designed to be used with the sample data for the Threat Mitigation Application. The SRC TMP is a Web application that lets you use a Web browser to manage threats.

Once you have configured and deployed the Threat Mitigation Application, you can use the SRC TMP to manage attack events.

When the NetScreen-Security Manager reports incidents to the SRC TMP, the SRC TMP:

- Provides a description of the incident, including source IP address, destination IP address, attack type, severity, time of first received record, time of last received record, count of repeated attacks, and possible actions.
- Allows the administrator to choose how to handle the threat in the appropriate manner by taking action, activating or deactivating a service, or managing an action already taken.
- Displays general information if the SRC software cannot collect information about an attack type because it is not defined in the ATTACK_TYPE table.

About the Record Servlet

The record servlet receives messages from the SRC **thm.py** script that runs in NetScreen-Security Manager. The **thm.py** script posts messages to a specified URL. The default pathname in the URL is /thmp/record. For information about changing the default pathname, see “Configuring Logging” on page 24.

NetScreen-Security Manager sends the following information from its XML schema to the record servlet for display in the SRC TMP.

- dayId—Date of the record as displayed in the Attack ID column to the left of the colon.
- recordId—Identifier for the record as displayed in the Attack ID column to the right of the colon.
- timeReceived—Time the attack event is received as displayed in the First Received Time and Last Received Time columns.
- subCategory—Subcategory of the attack as displayed in the Attack Type column.
- srcAddr—Source address of the attack as displayed in the Source column.
- dstAddr—Destination address of the attack as displayed in the Destination column.
- severity—Severity of the attack as displayed in the Severity column.
- repeatCount—Number of occurrences of the attack as displayed in the Repeat Count field.

The record servlet maps an attack ID with an attack type and its defining attributes (including protocol, source address, source port, destination address, destination port, user, application, uri). If the servlet receives more than one record for the same attack type with the same defining attribute values, the servlet stores the record with that attack ID once and increases the value of Repeat Count for that attack ID by one for each subsequent occurrence. The record servlet also records the highest severity of all attacks with the same defining attribute values and updates the last received timestamp.

If applicable, the SRC TMP displays the following information in the Attack Details page.

- category—Category of the attack; displayed in the Attack Type field.
- subCategory—Subcategory of the attack; displayed in the Attack Type field.
- srcAddr—Source address of the attack; displayed in the Source field.
- srcDns—The result of a reverse DNS lookup on the source address of the attack; displayed in the Source DNS field as a comma-separated list.
- srcPort—Source port of the attack; displayed in the Source Port field.
- dstAddr—Destination address of the attack; displayed in the Destination field.
- dstDns—The result of a reverse DNS lookup on the destination address of the attack; displayed in the Destination DNS field as a comma-separated list.
- dstPort—Destination port of the attack; displayed in the Destination Port field.
- protocol—Protocol of the attack; displayed in the Protocol field.

Related Topics

- Overview of Configuring and Deploying the SRC TMP on page 33
- Accessing the SRC TMP on page 34
- Installing and Initially Configuring the Threat Mitigation Application on page 16
- For information about the SRC **thm.py** script that runs in NetScreen-Security Manager, see Fields in the thm.py File on page 41

Overview of Configuring and Deploying the SRC TMP

The SRC TMP provided with the SRC software is designed to be used with the threat mitigation implementation in the sample data.

Using the NIC Resolver for the SRC TMP

The Threat Mitigation Application pushes policies to the interfaces from which the problem traffic enters the network. To do so, the SRC TMP must be able to map from a given attack source IP address to the SAEs managing the interfaces on the routers where that traffic enters the network. The Threat Mitigation Application uses the network information collector (NIC) to perform this mapping. Each service activation interface uses a different NIC configuration.

For information about the NIC configuration for each interface, see:

- JUNOS provider edge interface—"Configuring the NIC for Provider Edge Interfaces" on page 33
- JUNOS forwarding interface—"Configuring the NIC for Forwarding Interfaces" on page 33
- JUNOSe subscriber interface—"Configuring the NIC for Subscriber Interfaces" on page 34

For more information about configuring the service activation interface, see "Configuring Logging" on page 24.

Configuring the NIC for Provider Edge Interfaces

To configure the NIC to map the source IP address for a given attack to the SAEs managing the JUNOS subscriber-facing interfaces, use the OnePopStaticRouteIp configuration scenario and restart the NIC host. The OnePopStaticRouteIp configuration scenario resolves an assigned IP address for a subscriber whose traffic enters the network through an interface on a router running JUNOS Software to a reference for the SAE that manages the interface. The realm for this configuration accommodates the situation in which the network publisher component gathers interface information for the router running JUNOS Software. The resolution process takes a subscriber's IP address as a key and returns a reference to the SAE that manages the interface. For information about the NIC, see Locating Subscriber Management Information.

For information about associating an existing address pool with an interface, see Updating Information About Address Pools.

Configuring the NIC for Forwarding Interfaces

To configure the NIC to map the source IP address for a given attack to the SAEs managing the JUNOS forwarding interfaces, use the OnePop configuration scenario and restart the NIC host. The realm for the OnePop configuration scenario accommodates the situation in which IP address pools are configured locally on each VR. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value. For information about configuring the NIC, see Configuring the NIC (SRC CLI).

Configuring the NIC for Subscriber Interfaces

To configure the NIC to map the source IP address for a given attack to the SAEs managing the JUNOSe subscriber interfaces, use the OnePopAllRealms configuration scenario and restart the NIC host. The realm for the OnePopAllRealms configuration scenario accommodates the situations in which IP address pools are configured locally on each VR or IP address pools are shared by VRs in the same POP. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value. For information about configuring the NIC, see Configuring the NIC (SRC CLI).

If the IP address pools are shared across multiple VRs, you must also configure an external plug-in for the SAE plug-in agent in the NIC host as follows:

Plugin.nic.objectref=corbaname::<host>:<port>/NameService#nicsae/saePort

- <host> is the name or IP address of the COS name server
- <port> is the TCP port

For information about configuring the SAE for external plug-ins, see Configuring the SAE for External Plug-Ins (SRC CLI).

Related Topics

- Overview of the SRC TMP on page 31
- Accessing the SRC TMP on page 34
- Deploying the Threat Mitigation Application on page 25

Accessing the SRC TMP

To access the SRC TMP:

1. In your Web browser, enter the name or IP address of the host and the port number on which you installed the Threat Mitigation Application in the format:

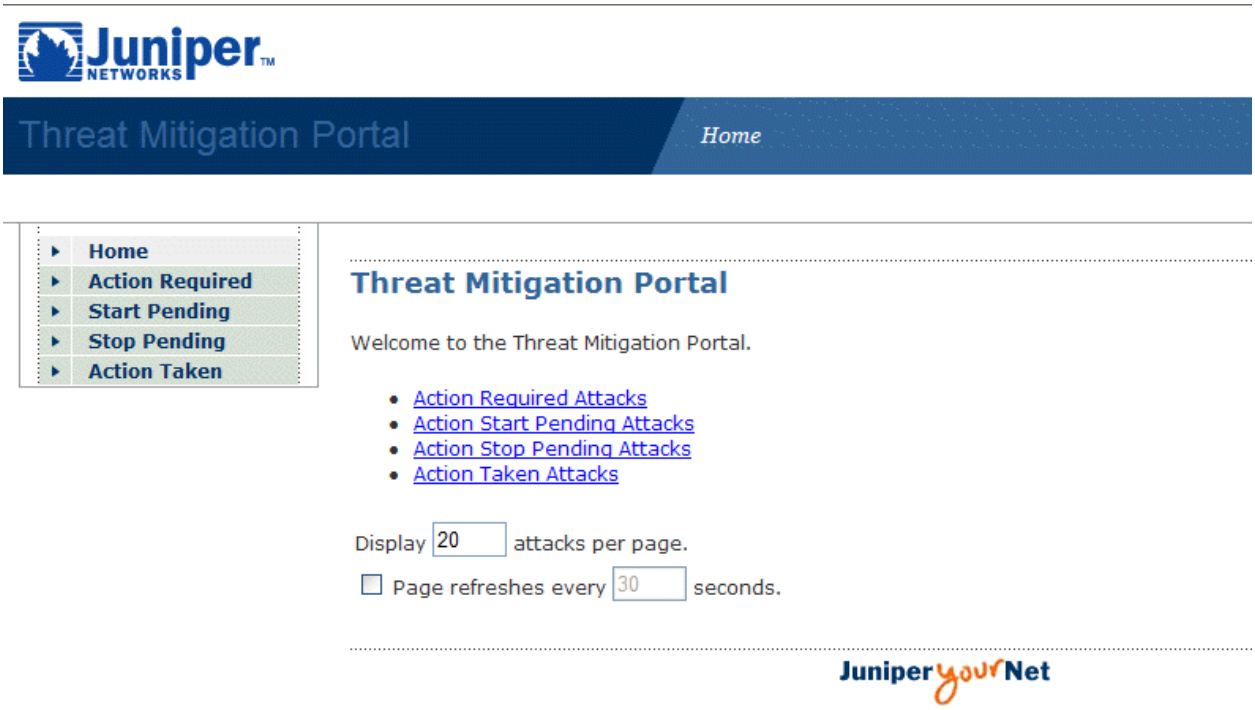
   http(s)://<host>:<port>/thmp

   A Connect to dialog box appears.

2. In the Connect to dialog box, enter your username and password, and click **OK**. The default values are:

   User name—admin
   Password—secret

   The Threat Mitigation Portal appears.



3. To modify the number of attacks displayed on each page from 20, enter the number in the Display attacks per page field.
4. To modify the page refresh rate, select the Page refreshes every 30 seconds check box, and enter the number of seconds in the text box.

You can manage the attacks that fall into these categories:

- Action Required—This page displays information about the attacks that require some action to be taken. See “Managing Attacks Requiring Action” on page 36.
- Start Pending—This page displays the attacks that are pending service activation. See “Managing Attacks Pending Service Activation” on page 37.
- Stop Pending—This page displays the attacks that are pending service deactivation. See “Managing Attacks Pending Service Deactivation” on page 38.
- Action Taken—This page displays the attacks for which some action was taken. See “Managing Attacks with Activated Services” on page 40.

The information provided about the attacks include attack ID, source and destination addresses, attack type, severity, first and last time the event was received, action that can be taken or action that was taken, and the time that the action was taken.

Related Topics

- Overview of the SRC TMP on page 31
- Overview of Configuring and Deploying the SRC TMP on page 33
- Configuring Logging on page 24

Managing Attacks with the SRC TMP

- Managing Attacks Requiring Action on page 36
- Managing Attacks Pending Service Activation on page 37
- Managing Attacks Pending Service Deactivation on page 38
- Managing Attacks with Activated Services on page 40

Managing Attacks Requiring Action

To manage attacks that require action to be taken:

1. In the Threat Mitigation Portal navigation pane, click **Action Required**.

   The Action Required page displays all attacks that require action.

   The Attack ID is linked to the Attack Details page, which displays more information about the attack record.

   The help button provides information about the possible actions that can be taken in response to an attack. For example, the Help could recommend blocking the attack, blocking the attacker, or slowing the attacker.

2. To sort the attacks by a different category, select another category from the **Sorted By** drop-down list, and click **Sort**.
3. To sort the attacks in a different order, select the order from the **Ordered By** drop-down list, and click **Sort**.
4. To take action, select the action from the **Action** drop-down list, and click **Take Action** to update the state of the attack in that row and activate the service that represents the action to be taken.

   If the attack is no longer in the same state as when you clicked **Take Action**, the action is aborted, and a message explains that the attack has been handled. Otherwise, the result depends on whether the service is activated.

   - If a service is activated, the attack is moved to the Action Taken page.

- If a service is waiting to be activated, the attack is placed in a pending state and appears in the Start Pending page.

5. To delete the attack, click **Delete** in the row for the attack.

Related Topics

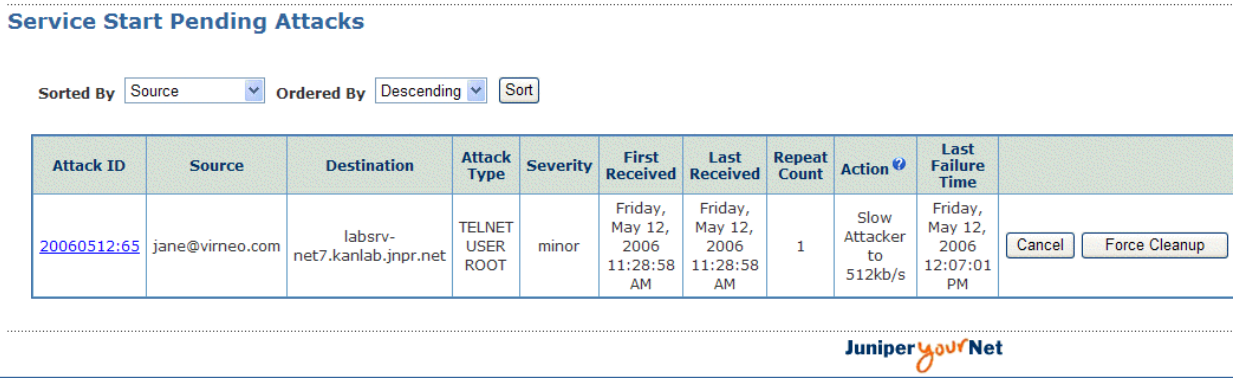
- Managing Attacks Pending Service Activation on page 37
- Managing Attacks Pending Service Deactivation on page 38
- Managing Attacks with Activated Services on page 40
- Configuring Attack Types in the Database on page 19
- Fields in the thm.py File on page 41

Managing Attacks Pending Service Activation

To manage attacks waiting for service activation:

1. In the Threat Mitigation Portal navigation pane, click **Start Pending**.

   The Start Pending page displays all attacks whose status is pending due to service activation.



   The Attack ID is linked to the Attack Details page, which displays more information about the attack record.

   The help button provides information about the possible actions that can be taken in response to an attack. For example, the Help could recommend blocking the attack, blocking the attacker, or slowing the attacker.

2. To sort the attacks by a different category, select another category from the **Sorted By** drop-down list, and click **Sort**.

3. To sort the attacks in a different order, select the order from the **Ordered By** drop-down list, and click Sort.

4. In the Service Start Pending Attacks table, you have the following options:

   - Click **Cancel** in a row to remove the attack from the Start Pending page and deactivate the service.

If the attack is no longer in the same state as when you clicked **Cancel**, the action is aborted, and a message explains that the attack has been handled. Otherwise, the result depends on whether the service is deactivated.

- If the service is deactivated, the attack is moved to the Action Required page.
- If the service is waiting to be deactivated, the attack is placed in a pending state and appears in the Stop Pending page. The Last Failure Time column indicates the time when the service deactivation was triggered.

- Click **Force Cleanup** in a row to delete the attack from the database.

  You are responsible for ensuring that the service is deactivated. The SRC TMP does not try to deactivate the service in this case.

- Click **Retry** in a row to manually reactivate the service.

  If the attack is no longer in the same state as when you clicked Retry, the action is aborted, and a message explains that the attack has been handled. Otherwise, the result depends on whether the service is activated.

  - If the service is activated, the attack is moved to the Action Taken page.
  - If the service is waiting to be activated, the attack stays in the same state and continues to appear in the Start Pending page. The Last Failure Time column indicates the time when the service activation was triggered.

  The SRC TMP automatically tries to reactivate the service according to the configuration properties (see “Configuring Logging” on page 24).

Related Topics

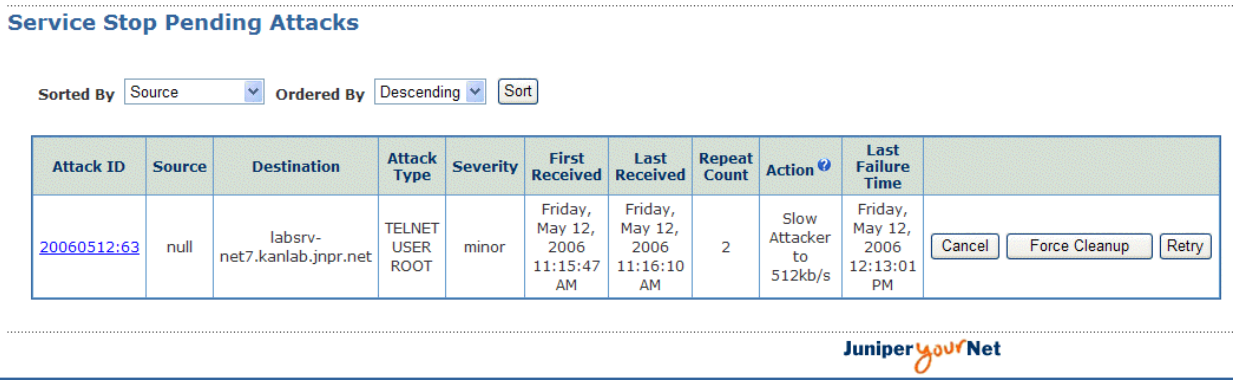
- Managing Attacks Requiring Action on page 36
- Managing Attacks Pending Service Deactivation on page 38
- Managing Attacks with Activated Services on page 40
- Configuring Attack Types in the Database on page 19
- Fields in the thm.py File on page 41

Managing Attacks Pending Service Deactivation

To manage attacks waiting for service deactivation:

1. In the Threat Mitigation Portal navigation pane, click **Stop Pending**.

   The Stop Pending page displays all attacks whose status is pending due to service deactivation.



   The Attack ID is linked to the Attack Details page, which displays more information about the attack record.

   The help button provides information about the possible actions that can be taken in response to an attack. For example, the Help could recommend blocking the attack, blocking the attacker, or slowing the attacker.

2. To sort the attacks by a different category, select another category from the **Sorted By** drop-down list, and click **Sort**.
3. To sort the attacks in a different order, select the order from the **Ordered By** drop-down list, and click **Sort**.
4. In the Service Stop Pending Attacks table, you have these options.

   - Click **Cancel** in a row to remove the attack from the Stop Pending page and activate the service.

     If the attack is no longer in the same state as when you clicked Cancel, the action is aborted, and a message explains that the attack has been handled. Otherwise, the result depends on whether the service is activated.

     - If the service is activated, the attack is moved to the Actions Taken page.
     - If the service is waiting to be activated, the attack record is placed in a pending state and appears in the Start Pending page. The Last Failure Time column indicates the time when the service activation was triggered.

   - Click **Force Cleanup** in a row to delete the attack from the database.

     You are responsible for ensuring that the service is deactivated. The SRC TMP does not try to deactivate the service in this case.

   - Click **Retry** in a row to try to manually deactivate the service again.

     If the attack is no longer in the same state as when you clicked **Retry**, the action is aborted, and a message explains that the attack has been handled. Otherwise, the result depends on whether the service is deactivated.

- If the service is deactivated, the attack is moved to the Action Required page.
- If the service is waiting to be deactivated, the attack record stays in the same state and continues to appears in the Stop Pending page. The Last Failure Time column indicates the time when the service deactivation was triggered.

The SRC TMP automatically tries to deactivate the service again according to the configuration properties (see “Configuring Logging” on page 24).

Related Topics

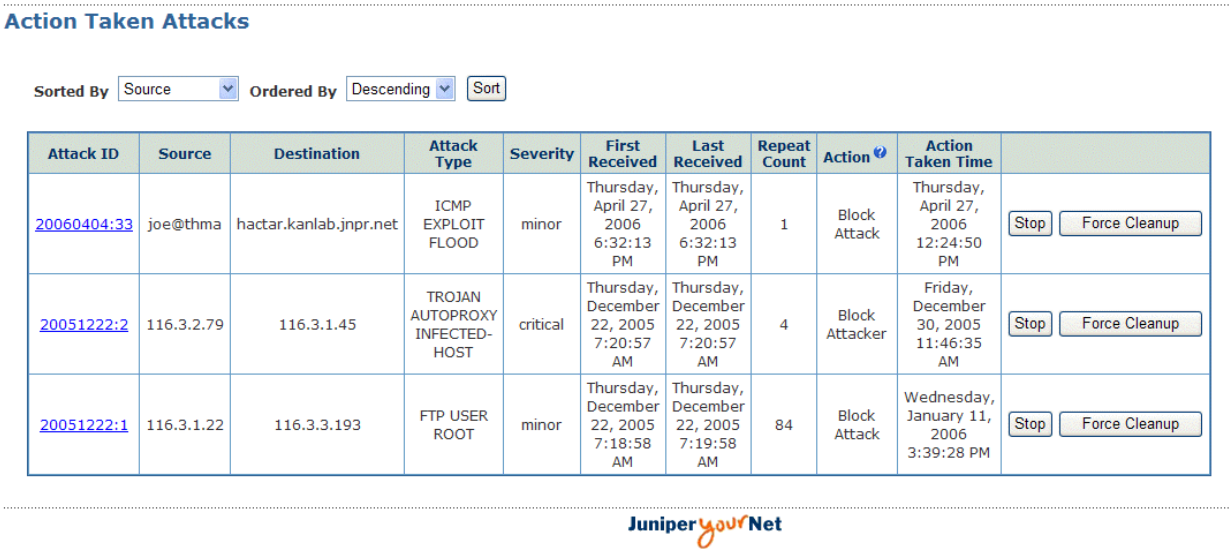
- Managing Attacks Requiring Action on page 36
- Managing Attacks Pending Service Activation on page 37
- Managing Attacks with Activated Services on page 40
- Configuring Attack Types in the Database on page 19
- Fields in the thm.py File on page 41

Managing Attacks with Activated Services

To manage attacks for which some action was taken:

1. In the Threat Mitigation Portal navigation pane, click **Action Taken**.

   The Action Taken page displays all attack records whose status is action taken.



The Attack ID is linked to the Attack Details page, which displays more information about the attack record.

The help button provides information about the possible actions that can be taken in response to an attack. For example, the Help could recommend blocking the attack, blocking the attacker, or slowing the attacker.

2. To sort the attacks by a different category, select another category from the **Sorted By** drop-down list, and click **Sort**.

3. To sort the attacks in a different order, select the order from the **Ordered By** drop-down list, and click **Sort**.
4. To cancel the action, click **Stop** in that row to update the state and deactivate the service that represents the action that was taken.

   If the attack is no longer in the same state as when you clicked **Stop**, the action is aborted, and a message explains that the attack has been handled. Otherwise, the result depends on whether the service is deactivated.

   - If a service is deactivated, the attack is moved to the Action Required page.
   - If a service is waiting to be deactivated, the attack record is placed in a pending state and appears in the Stop Pending page.
5. To delete the attack, click **Force Cleanup** in the row for the attack.

   You are responsible for ensuring that the service is deactivated. The SRC TMP does not try to deactivate the service in this case.

Related Topics

- Managing Attacks Requiring Action on page 36
- Managing Attacks Pending Service Activation on page 37
- Managing Attacks Pending Service Deactivation on page 38
- Configuring Attack Types in the Database on page 19
- Fields in the thm.py File on page 41

Fields in the thm.py File

The following list describes the fields in the thm.py file used by the SRC TMP.

RECORD URL

- URL of the record interface for the SRC TMP that stores information received from NetScreen-Security Manager. The interface records information about detrimental traffic in the ATTACK table in the database. The security rules configured in NetScreen-Security Manager determine the type of incidents recorded.
- Value—URL in the form " http(s)://<user>:<password>@<host>:<port>/thmp/record"
  - <user>—Client ID
  - <password>—Password associated with the client ID

  - <host>—Hostname or IP address of the server on which the SRC TMP runs
  - <port>—Port number used by the SRC TMP on the server
- Guidelines—Enclose the URL in quotation marks because this entry is a Python string.
- Default—RECORD_URL=“ http://admin:secret@127.0.0.1:8080/thmp/record”
- Example—RECORD_URL=“ https://admin:secret@192.0.2.25:8443/thmp/record”

FAIL_DIR

- Pathname to the directory that records incidents that were not successfully sent to the record URL.
- Value—Pathname in the form “ <pathname>”
- Guidelines—Enclose the pathname in quotation marks because this entry is a Python string.
- Default—FAIL_DIR = “ failedEvents”

FAIL_FILE_LIMIT

- Maximum number of events that will be recorded in the fail directory. If this number is exceeded, the oldest event is deleted to make room for the most recent event. If this number is 0, the script will not add any failed events, check the fail directory for failed events, or spawn the daemon process.
- Value—Integer in the range 0–2147483647
- Default—FAIL_FILE_LIMIT = 100

NUM_RETRIES

- Number of times the script (and daemon process) will retry sending an event to the record URL if the first attempt fails. If the retry limit is reached, the script gives up and writes the event to the fail directory. If the retry limit is reached by the daemon process, it stops trying to send failed events until its next interval. For example, if NUM_RETRIES is 2, then the script will try at most 3 times to send an event to the record URL.
- Value—Integer in the range 0–2147483647
- Default—NUM_RETRIES = 2

DAEMON_INTERVAL

- Amount of time that the daemon process will take between attempts to send events to the fail directory. When first started, the daemon process will wait this number of seconds before trying to send events recorded in the fail directory. If it fails to send any event in the fail directory, it will not try to send any more events for this amount of time.
- Value—Number of seconds in the range 0–604800 (1 week)
- Default—DAEMON_INTERVAL = 30

DEBUG

- Specifies whether or not to print debugging messages.
- Value
  - True—Print messages.
  - False—Do not print messages.
- Guidelines—Set this value to True only for troubleshooting. Set this value to False to minimize the effects on performance.
- Default—DEBUG = True

SEND_XML

- Specifies whether or not to send attack log events to the SRC TMP as an XML document.
- Value
  - True—The attack log event is sent to the SRC TMP as an XML document.
  - False—The script parses the XML document and posts the relevant data as individual request parameters.
- Guidelines—Set this value to True to minimize CPU resources consumed by this script. Set this value to False to minimize the CPU resources used by the SRC TMP in recording the attack. Setting this value to False will cause the script to consume approximately 60% more CPU resources.
- Default—SEND_XML = False

BACKGROUND_LOG_FILE

- Name of the file that logs messages for the process that retries sending attack log events. This file is created in the directory specified by FAIL_DIR.
- Value—Filename in the form " <filename>"
- Guidelines—Enclose the filename in quotation marks because this entry is a Python string. Set this value to None for no background logging.
- Default—BACKGROUND_LOG_FILE = " thm.log"

BACKGROUND_LOG_FILE_LIMIT

- Maximum size of the background log file. If this number is exceeded, a sequence number is appended to the filename and a new log file is started.
- Value—Number of bytes in the range 0–2147483647
- Default—BACKGROUND_LOG_FILE_LIMIT = 50000

PART 3

Managing Network Resources

- Providing Application-Level Session Tracking and QoS Control on page 47

CHAPTER 4

Providing Application-Level Session Tracking and QoS Control

- Overview of Application-Level Session Tracking and QoS Control on page 47
- Integration of the SRC Software and the Ellacoya DPI Platform on page 48
- Subscriber Login and Logout in a DPI Environment on page 52
- Service Activation and Deactivation in a DPI Environment on page 52
- Loading the Sample Data for the DPI on page 53
- Configuring the SRC Software for DPI Integration on page 54
- Configuring the Ellacoya DPI Platform for Integration on page 59

Overview of Application-Level Session Tracking and QoS Control

The SRC software has been integrated with the Ellacoya Networks Deep Packet Inspection (DPI) platform to provide a traffic management solution that combines the advanced traffic identification and reporting features of the Ellacoya DPI with the SRC software's intelligent service policy enforcement. With this solution, providers can identify, monitor, and control traffic on a per-application or per-subscriber basis.

Application traffic such as peer-to-peer file sharing or instant messaging, which in many cases originates or terminates outside a provider's network, can cause abusive or indiscriminate consumption of bandwidth and affect a provider's ability to deliver its own services. In particular, services that require higher, guaranteed levels of performance, such as voice over IP (VoIP) or video on demand, can be affected. Having visibility into applications that are transported over the network and their associated bandwidth consumption at various times is important, as is the ability to control those applications.

The DPI solution allows providers to implement service control policies on specific traffic flows quickly and effectively. Such policies include throttling back, capping volume, or even enhancing bandwidth or service quality for sanctioned peer-to-peer applications.

Benefits of Application-Level Session Tracking and QoS Control

By identifying and effectively controlling traffic at the application level, service providers can:

- Put usage controls on applications on a subscriber basis. For example, you can put a quota limit on the amount of peer-to-peer traffic that a subscriber can consume in a month.

  Once subscribers have used their quota, you can apply a policy that throttles back or blocks a subscriber's peer-to-peer traffic, bill the subscriber for additional usage, or allow the subscriber to purchase additional quota.
- Limit the total percentage of network resources that a specific type of traffic is allowed to consume.
- Provide higher or guaranteed levels of performance for premium services by applying QoS control to application sessions. For example, two subscribers start an Xbox Live session. The Ellacoya DPI platform detects activity for this application, and sends application usage counters to the SRC software. The SRC software pushes policies that deliver a specific level of QoS for this application session to a router or other network device.
- Charge subscribers based on their usage of premium content-based services.
- Offer and charge for tiered Internet services based on both speed and application.
- Better support network planning functions by gaining an in-depth understanding of traffic flows and patterns on a per-subscriber and per-application basis.

Related Topics

- Integration of the SRC Software and the Ellacoya DPI Platform on page 48
- Subscriber Login and Logout in a DPI Environment on page 52
- Loading the Sample Data for the DPI on page 53
- Configuring the Script Service on page 56

Integration of the SRC Software and the Ellacoya DPI Platform

This topic describes the integration of the SRC software with the Ellacoya DPI platform.

Ellacoya Networks DPI Platform

The SRC software is integrated with the Ellacoya IP Service Control System (IPSCS), which delivers comprehensive monitoring tools and extensive reporting that gives providers network visibility into and control over their subscribers, network, and service offerings (see Figure 1 on page 49). The IPSCS includes the following components:

- IPSCS e30 Switch, which provides application session and usage information based on real-time deep inspection of network traffic.
- Service logic engine (SLE), which is the control component of the IPSCS software. The SLE includes the Usage Quota Management System (UQMS) applications programming interfaces (API) that the SRC software can call to collect usage data and to indicate that subscriber sessions have started, changed, or stopped. Calls for the API are carried between the SLE and the SRC software by the Remote Method Invocation (RMI) protocol.

The IPSCS allows the creation of Usage Quota Management Systems, which allows third-party systems, such as the SRC software, to be integrated with the IPSCS. The UQMS is a provisioning system that determines business rules and communicates them to the IPSCS and other network components.

Service providers provision business rules on the UQMS. Based on the configuration instructions from the UQMS, the IPSCS collects usage statistics on specified intervals and applications, and then forwards the data to the UQMS for processing. The UQMS sends authentication and configuration information for subscribers to the IPSCS.

Juniper Networks Platforms

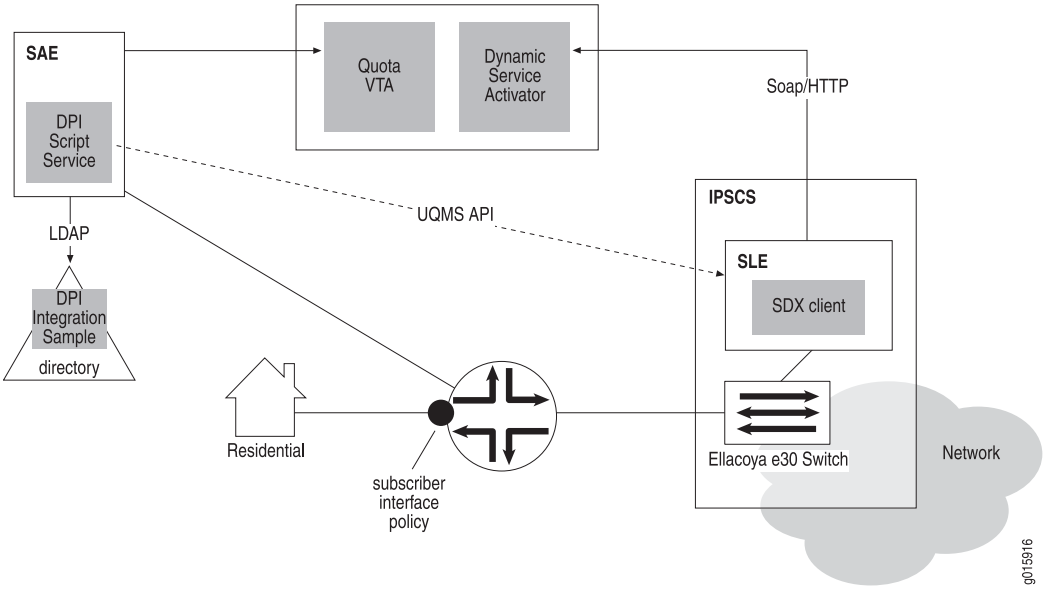
As shown in Figure 1 on page 49, the SRC software is integrated with the Ellacoya DPI platform through an SAE script service that works as a UQMS to the IPSCS. The script service sends event messages to and receives messages from the SLE through the UQMS API. The script service translates subscriber logins and logouts and service activation and deactivation to corresponding RMI function calls in the SLE. The script service collects accounting data from the SLE and maps the data to SRC service session usage data. The SRC software is integrated with the Ellacoya platform in a way that makes services implemented by the e30 Switch just like any other SRC service.

The SRC software supports the DPI solution on routers running JUNOSe or JUNOS Software, and on PacketCable Multimedia Specification (PCMM) policy servers and cable modem termination system (CMTS) devices.

You can integrate the SLE with the SRC software through a SOAP interface to the NIC and the SAE.

Figure 1 on page 49 shows the components involved in the SRC-DPI integration.

Figure 1: DPI Integration Overview



IPSCS Service Offers and Service Bundles

The Ellacoya IPSCS has the concept of service offers that contain a collection of service bundles. Service bundles are applied to a subscriber and contain a set of policy rules that define the patterns needed to recognize application sessions between specified source and destination networks. The service bundle includes the actions to take on application sessions, including marking specified streams with a DiffServ code point. Each rule in a service bundle can also have an activation period to control the time it is active.

A service offer also allows the association of traffic-accounting profiles (TAPs) with service bundles. TAPs are used to interface with external accounting systems. If a TAP is applied to a service bundle for a subscriber, then the counters for that service bundle are accumulated in accounting records indexed by TAP name.

Mapping Service Offers and Service Bundles to SRC Concepts

A service offer applied to a subscriber in the IPSCS corresponds to the SRC concept of the set of services active for a subscriber. A service bundle in an applied service offer corresponds to an SRC service activation.

A TAP applied to a service bundle roughly corresponds to accounting being enabled on an SRC policy.

The DPI integration maps SRC service activations and deactivations to changing the service offer applied to the subscriber so that it contains the corresponding set of service bundles. For example, a service provider offers three SRC services—Web access, e-mail, and peer-to-peer file sharing—to its subscribers. Each service can be dynamically enabled or disabled. The Ellacoya DPI system provides eight service offers that accommodate all possible combinations of the three SRC services.

Synchronization Between the SRC Software and the Ellacoya System

Table 6 on page 50 outlines the interactions that must be synchronized between the SRC software and the Ellacoya system.

Table 6: Synchronization Between the SRC Software and the Ellacoya System

Object	Source	Description
Service offer	Provisioned on the Ellacoya system	Contains the definition of the profile set in terms of network policies. For example, rate limits and ToS marking.
Subscriber	Dynamic	Subscribers are dynamically created in the Ellacoya database as the SRC software provisions them through the UQMS API.
Subscriber/Service offer binding	Dynamic	The binding of the subscriber to a service offer is done dynamically when the SRC software provisions the subscriber through the UQMS API.

Table 6: Synchronization Between the SRC Software and the Ellacoya System *(continued)*

Object	Source	Description
Subscriber/IP address binding	Dynamic	The binding between the subscriber and IP address is a transient binding communicated from the SRC software when the subscriber session starts.

Collecting Accounting Data

The IPSCS has a bulk statistics architecture where usage counters are collected for all active subscribers and services in bulk, and is controlled by a global accounting interval. The DPI script service polls for new usage data through the UQMS API based on the SLE global accounting interval. The SLE returns a list of file Uniform Resource Identifiers (URIs) of bulkstats files that have not been processed by the DPI script service.

After the DPI script service collects accounting data from the SLE, it maps the accounting data to usage data for an SRC service, and provides the usage data to the SAE. Because the SLE reports accounting data in the last accounting interval and the SAE expects service usage data since the service started, the script service adds the usage data in each accounting interval since the script service started. The script service keeps the counters in memory and also stores them so they can be recovered if there is an SAE failover.

The script service also stores a timestamp for the last time new data was reported to the SAE. This way, if there is a failover, the script service can determine when the doneWithFile method can be called for a file and whether or not a piece of usage data has been processed. The new SAE (after failover) requests the accounting data for sessions where the data in the file was reported to the previous SAE. At this time, the script service can compare timestamps (records in the file are also timestamped) to recognize that the data was previously reported and to recognize when all of the data in a file has been processed.

The IPSCS provides only one accounting record per accounting interval, per service offer, and per subscriber. This behavior means that if a DPI script service is stopped and restarted during one IPSCS accounting interval (usually 15 minutes), service usage of the two service sessions within that interval are allocated to one of the service sessions—the first service session for which the SAE requests an accounting update.

Related Topics

- Overview of Application-Level Session Tracking and QoS Control on page 47
- Provisioning the IPSCS on page 59
- Subscriber Login and Logout in a DPI Environment on page 52
- Service Activation and Deactivation in a DPI Environment on page 52
- SRC Script Services for DPI on page 54

Subscriber Login and Logout in a DPI Environment

Login events are not directly handled by the SAE, and there is no need for subscriber tracking plug-ins. When a script service is activated, the script service determines the Ellacoya service offer name based on the SRC services that are to be activated on login for the subscriber. Then, the script service calls the startSubscriberRadiusSession method of the UQMS API, which starts a subscriber session, and provides the subscriber's IP address, ID, and service offer name to the SLE.

When a script service is deactivated because the subscriber logs out, the script service notifies the SLE to remove the subscriber session, and it ignores all subsequent service deactivations for the subscriber.

Related Topics

- Overview of Application-Level Session Tracking and QoS Control on page 47
- Service Activation and Deactivation in a DPI Environment on page 52
- SRC Script Services for DPI on page 54
- Configuring Subscriptions to DPI Services on page 59
- Loading the Sample Data for the DPI on page 53

Service Activation and Deactivation in a DPI Environment

When the SAE activates the first DPI script service or deactivates the last DPI script service, the script service calls the startSubscriberRadiusSession() or stopSubscriberRadiusSession() methods (which start and stop subscriber sessions) in the UQMS API.

You can set up the script service so that when the first service that a script service implements is activated at login, the script service starts all other activate-on-login and persistently activated services for the subscriber that are also implemented by the script service. The combination of all activate-on-login and persistent services maps to one service offer in the SLE.

When there is a change to the subscriber's services that triggers a service activation—for example, the subscriber activates a service with a portal, the service provider adds new services, or the subscriber's quota is replenished—the script service calls the SLE changeSubscriberSession() method (which changes a subscriber session) with the new service offer name that corresponds to the combination of active DPI script services of the subscriber.

When deactivation of script services is triggered by changes to a subscriber's services—for example, the scheduled time for a service has ended, the service provider removes a service, or a subscriber exceeds the quota—the DPI script service calls the SLE changeSubscriberSession() method with the new service offer name that corresponds to the combination of active DPI script service of the subscriber.

The SAE expects the script service to return its final usage data when it stops the service. However, the SLE does not return the final accounting record synchronously when the

changeSubscriberSession() method is called. When a DPI script service is stopped, the script service signals to the SAE that accounting data is to be collected at a later time.

Related Topics

- Integration of the SRC Software and the Ellacoya DPI Platform on page 48
- Subscriber Login and Logout in a DPI Environment on page 52
- SRC Script Services for DPI on page 54
- Configuring Subscriptions to DPI Services on page 59
- Adding a Service Scope on page 54
- Configuring the Script Service on page 56

Loading the Sample Data for the DPI

The DPI sample data includes directory entries for service and policy configurations that are provided as a guide for how to set up an IPSCS-based service to be used as a quota service for the SRC VTA.

To load the sample data for the DPI feature:

1. Go to the dpi folder. The default folder is:

   /opt/UMC/sae/var/scriptservice/dpi

2. Enter the following command, and respond to the prompts.

   ```
   # ./loadsample
   Please enter the authentication dn [cn=umcadmin,o=umc]:
   Please enter the authentication password [admin123]:
   Please enter the directory host address [127.0.0.1]:
   Please enter the directory port [389]:
   ```

The software adds the following sample data:

```
adding new entry l=dpi, o=Scopes, o=UMC

adding new entry serviceName=service1, l=dpi, o=Scopes, o=UMC
adding new entry serviceName=service2, l=dpi, o=Scopes, o=UMC
adding new entry serviceName=service3, l=dpi, o=Scopes, o=UMC
adding new entry retailername=SP-DPI, o=Users, o=umc
adding new entry ou=local, retailername=SP-DPI, o=Users, o=umc
adding new entry uniqueID=jane,ou=local, retailername=SP-DPI, o=Users, o=umc
adding new entry serviceName=service3,ou=local, retailername=SP-DPI, o=Users, o=umc
adding new entry serviceName=service1,ou=local, retailername=SP-DPI, o=Users, o=umc
adding new entry serviceName=service2,ou=local, retailername=SP-DPI, o=Users, o=umc
```

Related Topics

- Integration of the SRC Software and the Ellacoya DPI Platform on page 48
- Provisioning the IPSCS on page 59
- Subscriber Login and Logout in a DPI Environment on page 52
- SRC Script Services for DPI on page 54

- Configuring a Virtual Router Object for DPI on page 58

Configuring the SRC Software for DPI Integration

To set up the SRC software for the DPI solution, perform the following tasks:

- SRC Script Services for DPI on page 54
- Adding a Service Scope on page 54
- Creating a DPI Script Service on page 55
- Configuring the Script Service on page 56
- Configuring a Virtual Router Object for DPI on page 58
- Configuring Subscriptions to DPI Services on page 59

SRC Script Services for DPI

Every subscriber must have at least one DPI script service active during subscriber login. This DPI script service corresponds to a service bundle in the SLE that provides the default policy for processing subscriber traffic. To match the service bundle with the script service, the service bundle must have the same name as the DPI script service and be prefixed by SDX_. For example, if the script service is called dpiService1, the service bundle must be called SDX_dpiService1.

You can create a service scope to hold your script services. Putting a script service in a service scope allows the service to be used in different regions of the network. (See “Configuring a Virtual Router Object for DPI” on page 58 for more information.) The DPI sample data contains a service scope called DPI. You can model your script services after the sample data.

After you configure a service scope, you need to configure script services for the scope and configure parameters for the script service. For more information about configuring service scopes, script services, and parameters, see Configuring Service Scopes (SRC CLI).

Related Topics

- Integration of the SRC Software and the Ellacoya DPI Platform on page 48
- Provisioning the IPSCS on page 59
- Adding a Service Scope on page 54
- Creating a DPI Script Service on page 55
- Configuring the Script Service on page 56
- Loading the Sample Data for the DPI on page 53

Adding a Service Scope

To add a service scope:

1. Add a service scope with the SRC CLI or the C-Web interface.

See Configuring Service Scopes (SRC CLI) and Configuring Service Scopes (C-Web Interface)

2. Optionally, configure a precedence for the service scope.

Related Topics

- SRC Script Services for DPI on page 54
- Service Activation and Deactivation in a DPI Environment on page 52
- Configuring Subscriptions to DPI Services on page 59
- Creating a DPI Script Service on page 55
- Configuring the Script Service on page 56

Creating a DPI Script Service

To create a script service within the service scope:

1. Within the service scope, create a script service with the SRC CLI or the C-Web interface.

   See Customizing Service Implementations or Configuring Script Services (C-Web Interface)

2. Specify the values for the following options for the service.

Script Type

- Type of script that the script service uses.
- Value—You must use URL for DPI script services

Class Name

- Name of the class that implements the ScriptService SPI. The SAE instantiates this class when it starts the script service.
- Value—You must enter net.juniper.smgt.dpi.sle.EllacoyaScriptService for DPI script services

File/URL

- URL to the Java archive (JAR) files that are associated with the script.
- Value—For a DPI script service, you must include all of the following JAR files.
  - *file:///opt/UMC/sae/var/scriptservice/dpi/lib/dpi-ss.jar,*
  - *file:///opt/UMC/sae/var/scriptservice/dpi/lib/sle-client.jar,*
  - *file:///opt/UMC/sae/var/scriptservice/dpi/lib/jbossall-client.jar,*
  - *file:///opt/UMC/sae/var/scriptservice/dpi/lib/jdom.jar,*
  - *file:///opt/UMC/sae/var/scriptservice/dpi/lib/jnet.jar,*
  - *file:///opt/UMC/sae/var/scriptservice/dpi/lib/log4j.jar*

NOTE: All JAR files except dpi-ss.jar are Ellacoya SLE files, and must be compatible with the version of the SLE that is being used. Check the release notes for the version of the SLE that the JAR files packaged with the SRC software are from. If in doubt, replace the JAR files from the SLE installation, which can be found in *<SLE install dir>/lib* (for example: */opt/ellacoya/lib*).

Related Topics

- SRC Script Services for DPI on page 54
- Service Activation and Deactivation in a DPI Environment on page 52
- Configuring Subscriptions to DPI Services on page 59
- Adding a Service Scope on page 54
- Configuring the Script Service on page 56

Configuring the Script Service

To configure the script service, you provide parameter substitutions with the values of the configuration parameters that are in the service definitions. To do so:

1. Configure parameters for the script service. See Customizing Service Implementations or Configuring Script Services (C-Web Interface)
2. Configure all the parameters listed in Table 7 on page 56.

Table 7: Parameter Definitions for DPI Services

Parameter Name	Description
dpi_sle_connection_servers	IP address or hostname of the Ellacoya SLE server. If there is more than one server, separate entries with a comma. Enter the SLE addresses in priority order, with the primary server first. When the script service makes a connection to an SLE, it tries the addresses in the order that you entered them until it succeeds in connecting. If the connection is broken, the script service attempts to connect again, beginning with the first address in the list. Once the script connects to an SLE, it does not test the connection to the primary server and fail back to the primary server if the server recovers from a failure.

Table 7: Parameter Definitions for DPI Services *(continued)*

Parameter Name	Description
dpi_sle_subscriber_nameattr	Attribute that is used to identify a subscriber. This parameter instructs the SLE how to identify the subscriber. Usually subscribers are identified by their login name. You can also use the MAC address, primary user, NAS-Port ID, or interface name. The values for the parameter definition are: • loginname • primaryusername • macaddr • interfacename • nasportid
dpi_sle_subscriber_ipv6supported	Specifies whether or not IPv6 is supported for the DPI service. Must currently be false because IPv6 is not supported on Ellacoya systems.
dpi_sle_connection_domainName	Name of the domain that contains the service bundles for this service. The Ellacoya software configures service bundles by domain. This parameter value should match the domain that is configured in the Ellacoya Service Creation Manager.
dpi_sle_connection_username	Name used for authentication with the SLE. It must match the username that is configured in the Ellacoya Service Creation Manager.
dpi_sle_connection_password	Password used for authentication with the SLE.
dpi_sle_accountingmgr_storesize	Size limit of usage and accounting data that is stored. (A script services loads usage data from the SLE.) All script services in a virtual router share the same store. When the SAE requests usage data more slowly than the script service loads it from the SLE, the data accumulates in the store. When the store is filled, the script service suspends loading usage data from the SLE. However, as the SAE requests usage data, space in the store is freed, allowing the script service to resume loading data from the SLE.

Table 7: Parameter Definitions for DPI Services *(continued)*

Parameter Name	Description
dpi_sle_activate_inoneshot	The values for the parameter are true or false. The default value is true. If true, when a DPI script service is activated as an AOL subscription during subscriber login, the script service finds all AOL subscriptions and persistent sessions of the subscriber. The script service assumes that the SAE will activate all the subscriptions and sessions, and it starts an SLE subscriber session with a service offer that corresponds to the combination of the AOL subscriptions and persistent sessions. This optimization avoids repeatedly changing the subscriber's service offer while multiple AOL subscription or persistent sessions are activated one by one. Set this parameter to false if an AOL subscription or persistent session is not guaranteed to be activated during subscriber login; for example, an authorization plug-in or mutex group could prevent the activation of the services.
dpi_sle_subscriber_prefix	Set this parameter to match the subscriber prefix configured in the SLE. The DPI script service adds this prefix to subscriber IDs used in SLE subscriber sessions that the DPI script service starts. The default value is BBIP, which is also the default value used in the SLE. You can change the SLE subscriber prefix in the SubscriberPrefix field in the Accounting Service window of the SLE Configuration Console.

Related Topics

- SRC Script Services for DPI on page 54
- Service Activation and Deactivation in a DPI Environment on page 52
- Configuring Subscriptions to DPI Services on page 59
- Adding a Service Scope on page 54
- Creating a DPI Script Service on page 55

Configuring a Virtual Router Object for DPI

You must create a virtual router object in the router to which the subscriber logs in, and attach the DPI service scope to the virtual router.

Each time a DPI script service is invoked by an activation, deactivation, or interim accounting request, the script service checks the service involved to see if the configuration in the service's parameter substitutions is different or newer than the parameters in the script service's configuration. If so, the script service instance updates its configuration according to the service. This functionality means you can change the configuration of the script service by updating the parameter substitutions in the service.

If you want to use the same DPI configuration throughout your network, make sure that all DPI script services have the same configuration. If you have several SLEs in your network that are responsible for the traffic coming from different sets of routers, you must ensure that the DPI script service gets the appropriate configuration on a per router basis. You can accomplish this by putting the script services into a scope and attaching the scope to the router for an SLE. Every service in the scope must have the same configuration so that the correct SLE handles all subscribers on routers that use the scope.

To attach a service scope to a virtual router configuration, configure the name of the DPI service scope for the DPI for the Scope field option of the virtual router configuration. See *SRC PE Network Guide*.

Related Topics

- Integration of the SRC Software and the Ellacoya DPI Platform on page 48
- Subscriber Login and Logout in a DPI Environment on page 52
- Service Activation and Deactivation in a DPI Environment on page 52
- SRC Script Services for DPI on page 54
- Loading the Sample Data for the DPI on page 53

Configuring Subscriptions to DPI Services

You need to configure subscriptions to the DPI services. You can set up the subscriptions to activate immediately on login.

Related Topics

- Service Activation and Deactivation in a DPI Environment on page 52
- Subscriber Login and Logout in a DPI Environment on page 52
- SRC Script Services for DPI on page 54
- Configuring the Script Service on page 56
- Creating a DPI Script Service on page 55
- *SRC PE Subscribers and Subscriptions Guide*

Configuring the Ellacoya DPI Platform for Integration

To set up the SRC software to for the DPI solution, you need to perform the following tasks:

- Provisioning the IPSCS on page 59
- Configuring the SLE on page 60
- Synchronizing System Clocks on page 61

Provisioning the IPSCS

For integration with the SRC software, the IPSCS must be provisioned in a specific way. All IPSCS service objects, including application signatures, service elements, TAPs, service bundles, and service offers, must be created.

Before you introduce a new service that will be implemented by a DPI script service, you must create the service bundle and service offers. Before you remove a service, you must deactivate all sessions for the service before removing the corresponding service offers and bundles from the IPSCS.

See “Integration of the SRC Software and the Ellacoya DPI Platform” on page 48.

Service Bundles

Enable usage collection on service bundles by checking Collect Start/Stop Usage Data and the Collect Transmit and Receive Byte Counts checkboxes in Service Creation Manager.

Every DPI script service must have a corresponding ISPCS service bundle. To match the service bundle with the script service, the service bundle must have the same name as the DPI script service, but be prefixed by SDX_. For example, if the script service is called dpiService1, the service bundle must be called SDX_dpiService1.

Service Offers

Enable usage collection on service offers by checking the Enable Usage Collection checkbox in Service Creation Manager.

For every combination of implemented services that could be active at the same time, there must be a service offer with the name prefixed by “ SDX_” configured in the IPSCS. The service offer must contain the service bundles that correspond to the services.

When a script service activates or deactivates a service session, it checks the list of services that will be active after the activation or deactivation. The script service then searches for a service offer with the right set of service bundles. The script service keeps a local cache of all service offer names and which bundles they contain. If the cache does not contain a service offer with the right set of bundles, the script service reloads the service offers from the SLE. If necessary, the script service queries the SLE for all service offers in the configured domain. If the appropriate offer is not found after a reload, the service activation or deactivation fails with an exception.

Traffic-Accounting Profiles

For every service bundle, there must be a TAP with the same name as the service bundle.

Related Topics

- Overview of Application-Level Session Tracking and QoS Control on page 47
- Configuring the SLE on page 60
- Synchronizing System Clocks on page 61

Configuring the SLE

In Service Creation Manager, configure the following:

1. Create passive authentication configuration, and enable subscriber learning.
2. Create a switch configuration, and apply the passive authentication configuration to the switch configuration.

3. Create a switch that corresponds to your e30 Switch, and apply the switch configuration *to the switch.

You also need to enable accounting in the SLE. To do so, log in to the SLE Configuration Console console, and check the AccountingEnabled service attribute in the AccountingService window.

Related Topics

- Provisioning the IPSCS on page 59
- Integration of the SRC Software and the Ellacoya DPI Platform on page 48
- Configuring a Virtual Router Object for DPI on page 58

Synchronizing System Clocks

The system clock in the e30 Switch must keep synchronized with the SAE. Make sure that the system clock of the Ellacoya switch does not use daylight savings mode.

Related Topics

- Provisioning the IPSCS on page 59
- Integration of the SRC Software and the Ellacoya DPI Platform on page 48

PART 4

Controlling Volume Usage with the SRC VTA

- Overview of Controlling Volume Usage with the SRC VTA on page 65
- Installing and Initially Configuring the SRC VTA on page 81
- Configuring the SRC VTA with VTA Configuration Manager on page 103
- Managing Subscriber Accounts with VTA Portals on page 179
- Example of a Bucket VTA on page 193

CHAPTER 5

Overview of Controlling Volume Usage with the SRC VTA

- Overview of the SRC VTA on page 65
- SRC VTA Architecture and Connections to SRC Components on page 68
- How the SRC VTA Works on page 68
- SRC VTA Operation on page 73
- Managing VTA Accounts and Sessions on page 74
- Example: Limiting Subscriber Access Based on Account Balances on page 78

Overview of the SRC VTA

The SRC Volume-Tracking Application (SRC VTA) allows service providers to track and control the network usage of subscribers and services. You can control volume and time usage on a per-subscriber or per-service basis. This level of control means that service providers can offer tiered services that use volume as a metric, while also controlling abusive subscribers and applications.

When a subscriber or service exceeds bandwidth limits (or quotas), the SRC VTA can take actions, including directing the subscriber to a portal to activate additional services or purchase additional bandwidth, imposing rate limits on traffic, sending an e-mail notification, or charging extra for additional bandwidth consumed.

If you use the SRC VTA with the deep packet inspection (DPI) feature, you can control the volume of traffic for specific applications, such as peer-to-peer file sharing.

Types of VTAs

SRC software releases earlier than Release 6.3.x supported two types of VTAs—quota and threshold. You can now configure the quota VTA to provide the same functionality as the threshold VTA. (See “Example of a Bucket VTA” on page 193.)

Terminology

Table 8 on page 66 defines terms that are used in the SRC VTA documentation and sample data.

Table 8: SRC VTA Terms

Term	Definition
Behaving service	Service that a VTA activates for subscribers when the SRC VTA is not restricting their rates of data transfer.
Bought quota	Allowance of data volume that subscribers purchase and can transfer (upload or download) at any time.
Bought account	Record that details a subscriber's use of bought quota.
Bucket account	Account that is periodically measured and refilled depending on the usage of the account.
Misbehaving service	Service that a VTA activates for subscribers when the SRC VTA is restricting their rates of data transfer.
Periodic quota	Allowance of data volume that a service provider allocates to subscribers on a recurrent basis. Subscribers use this allowance to upload or download data.
Periodic account	Record that tracks a subscriber's use of periodic quota.
Quota service	Service for which a VTA monitors usage. The SRC VTA activates the service for subscribers when they have a positive balance in their VTA accounts, and deactivates the service when the VTA account has a negative balance.
VTA account	Record of credit and debit entries that track a subscriber's use of a particular network resource.
VTA session	Period of activity between a VTA subscriber and a VTA.

VTA Service and Subscriber Accounts

A VTA account represents the resources available to a service or a subscriber. You can configure VTA accounts and then charge a particular service or subscriber's usage against the account. Each subscriber or service can have a different quota, or allowance of data volume.

You can set up the way the VTA charges accounts and how account balances are updated.

You can also configure actions in response to changes in account balances. Available actions include stopping a service, starting a service, updating an account balance, sending an e-mail, and running a script. For example, if account A is emptied, the action might be to stop services X and Y, and start service Z.

The SRC VTA requires a relational database to store information about accounts. The SRC VTA installation includes sample schemas and configurations for the MySQL and Oracle databases.

VTA Sessions

The SRC VTA tracks subscriber activity through VTA sessions. A VTA session does not necessarily correspond to an individual subscriber session or service session. For example, a single service session can correspond to multiple VTA sessions if the service session covers multiple billing periods.

The SRC VTA not only can track the volume and time of a service session, it can track any state of a subscriber derived from SAE plug-in events and respond to the state change.

The SRC VTA requires a relational database to store information about sessions. The SRC VTA installation includes sample schemas and configurations for the MySQL and Oracle databases.

Managing Subscriber Accounts with Portals

We provide two sample portals that manage subscriber accounts. One is an administrator portal that administrators can use to manage VTA subscriber accounts. The second is a subscriber portal that subscribers can use to manage their own accounts. Before you can use these portals, you need to configure the Web applications for the SRC VTA.

The suggested billing model for services managed by the SRC VTA is one in which subscribers pay for services when they select them through a Web portal.

Volume-Based Services

The SRC VTA lets you set triggers at multiple levels to provide flexible and extensive volume-based services. For example:

- When the volume level reaches 300 MB, turn on the internet-256 service, turn off the internet-512 service, and send an e-mail to the subscriber.
- When the volume level reaches 100 MB, send an e-mail warning to the subscriber.
- When the volume level is 0 MB, turn on the continue-TCP-only service, turn off the internet-256 service, send an e-mail to the subscriber, and notify the accounting server.
- When the volume level is –100 MB, turn off the continue-TCP-only service, send an e-mail to subscriber, and notify the accounting server.

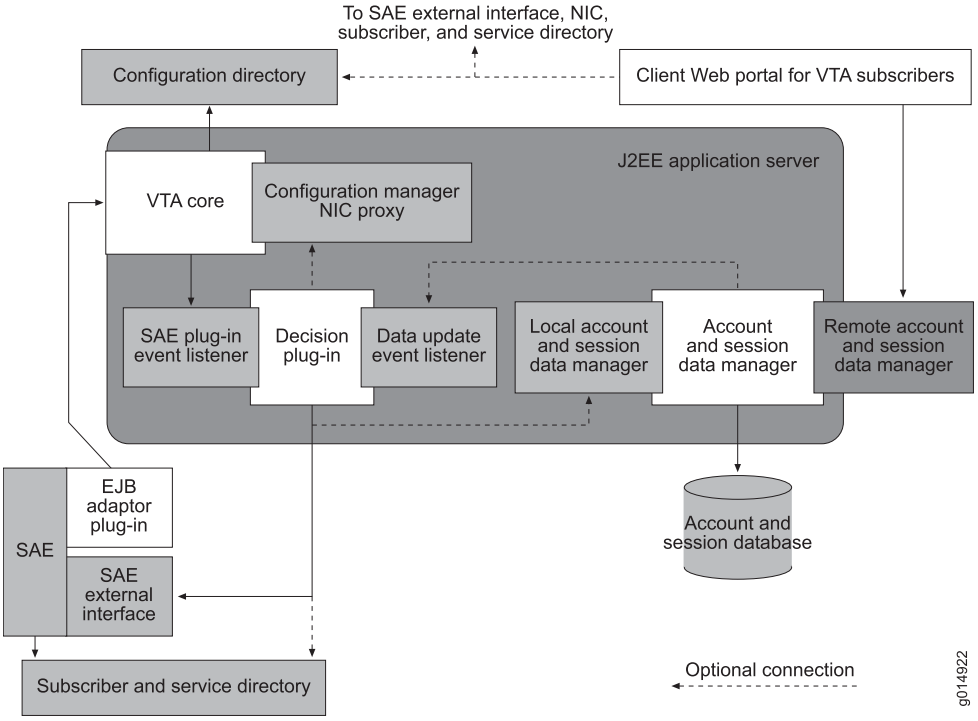
Related Topics

- How the SRC VTA Works on page 68
- SRC VTA Operation on page 73
- Before You Install the SRC VTA on page 81
- Accessing the Subscriber Portal on page 188
- Installing the SRC VTA and Running the Configuration Script on page 82
- SRC VTA Architecture and Connections to SRC Components on page 68

SRC VTA Architecture and Connections to SRC Components

Figure 2 on page 68 shows the SRC VTA architecture and the position of a VTA in the SRC network.

Figure 2: SRC VTA Architecture and Position in the SRC Network



Related Topics

- Overview of the SRC VTA on page 65
- How the SRC VTA Works on page 68
- SRC VTA Operation on page 73
- Installing the SRC VTA and Running the Configuration Script on page 82

How the SRC VTA Works

The SRC VTA manages subscriber accounts using a rule-driven event-processing system that can prioritize the actions taken for certain conditions. The SRC VTA is triggered by events, such as the logging in of subscribers, the use of network services, or the changing of account balances. These events can cause actions, such as updating account balances, starting or stopping network services, or running scripts to perform external actions.

A VTA processes external events based on its configuration. A VTA configuration is made up of:

- Event handlers

- Actions
- Processors

Events

Each VTA event corresponds to one subscriber and contains some attributes. The SRC VTA supports the following types of events:

- Service- and subscriber-tracking events from the SAE; for example, start or stop tracking events.
- Account update events triggered by updating database accounts.
- Callback events triggered by a method call.

Event Attributes

Each event carries attributes. Table 9 on page 69 describes the types of attributes that are available for each type of event.

Table 9: Event Attributes

Event Type	Available Attributes
Service and subscriber tracking events from the SAE	• Plug-in attributes, such as PA_SERVICE_NAME or PA_LOGIN_NAME, associated with an SAE plug-in event. For a list of SAE plug-in events, see the SAE CORBA plug-in SPI online documentation on the Juniper Networks Web site at **http://www.juniper.net/techpubs/software/management/sdx/api-index.html** or in the SDK+AppSupport+Demos+Samples.tar.gz file on the Juniper Web site at https://www.juniper.net/support/csc/swdist-erx/src.html. • currentTime attribute—Time since January 1, 1970 UTC when the SRC VTA receives the event. The value is the number of milliseconds in the range 0–9223372036854775807. • subscriberId—Defines how to calculate the subscriber ID based on attributes of a service or a subscriber-tracking event. The subscriberId event attribute is a result of the calculation. It identifies the subscriber of the corresponding VTA event.

Table 9: Event Attributes *(continued)*

Event Type	Available Attributes
Account update events	• old_status_<accountname>—Returns the old status of the account. • new_status_<accountname>—Returns the new status of the specified account. • old_lastUpdateTime_<accountname>—Returns the old last update time of the account. • new_lastUpdateTime_<accountname>—Returns the new last update time of the account. • old_balance_<accountname>—Returns the old balance of the account. • new_balance_<accountname>—Returns the new balance of the specified account • The currentTime attribute, which is the time since January 1, 1970 UTC, when the SRC VTA receives the event. The value is the number of milliseconds in the range 0–9223372036854775807. • subscriberId—Defines how to calculate the subscriber ID based on attributes of a service or a subscriber-tracking event. The subscriberId event attribute is a result of the calculation. It identifies the subscriber of the corresponding VTA event.
Callback events	• callId—Includes all parameters provided with the callback. • currentTime—Time since January 1, 1970 UTC, when the SRC VTA receives the event. The value is the number of milliseconds in the range 0–9223372036854775807. • subscriberId—Defines how to calculate the subscriber ID based on attributes of a service or a subscriber-tracking event. The subscriberId event attribute is a result of the calculation. It identifies the subscriber of the corresponding VTA event.

Event Handlers

An event handler defines how the SRC VTA processes an event. VTA event handlers consist of:

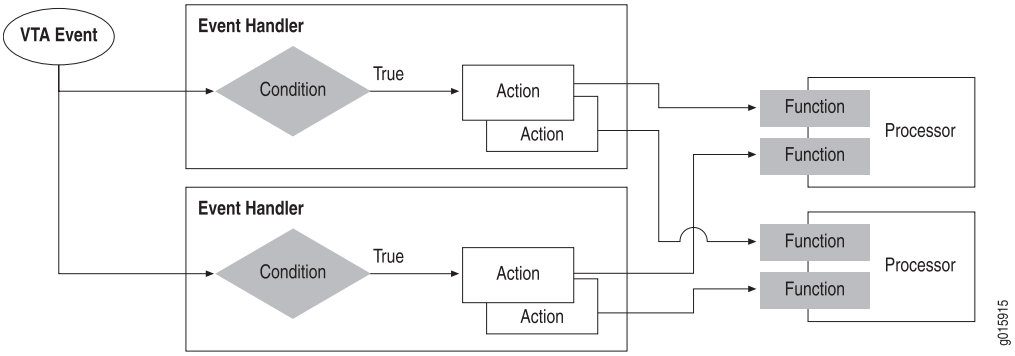
- Type of event—Tracking, update, or callback event; for example, a subscriber start–tracking event or a service start–tracking event.
- Priority—Priority at which event handlers are evaluated and executed. Event handlers are evaluated and executed from high to low priority.
- Condition—Condition that the event handler evaluates to determine whether the event handler should handle the event.
- Actions—List of actions to be performed by the event handler.

You can set up multiple event handlers to process events. For example, the first event handler could retrieve the balance for a quota account, and the next event handler could refill the quota account, depending on whether the condition of the second event handler is met.

Figure 3 on page 71 shows the event handler model. The SRC VTA processes an event as follows:

1. The event handler with the highest priority receives the event, determines whether the event's type is the same as the event type of the event handler, and determines whether the event satisfies the condition of the event handler.
2. If the condition is met, the SRC VTA performs the corresponding actions based on the event attributes. An action invokes a function and provides the parameters required by that function to the processor.
3. When an event handler finishes processing an event, the next applicable event handler according to the priority of the event handler processes the event.

Figure 3: VTA Event Handler Model



Actions

You can specify actions that the SRC VTA takes in response to an event; for example, updating an account balance, starting a service, or stopping a service. The actions are performed if the event is of the specified type and matches the specified condition. An action is modeled as a call to a function.

An action can update event attributes or add new attributes to an event for subsequent processing of the same event by another event handler.

An action configuration includes:

- Function—Function that the action invokes
- Parameter—Parameters and corresponding values to be passed to the function

Processors

Processors receive and process events. The SRC VTA has four processors:

- Database engine processor—Acts as a proxy to a database.
- Mail processor—Sends e-mail notifications when certain events occur.
- SAE proxy processor—Acts as a proxy to the SAE.
- Script runner processor—Runs external scripts or JavaScript programs.

Database Engine Processor

The database engine processor is a proxy to a database. You can use the database engine processor to:

- Calculate the use of network resources for a service.
- Calculate the interim accounting interval for each service based on a subscriber's remaining resources and use of the service.
- Update VTA accounts with a JavaScript program.
- Terminate a VTA session. This feature is usually used at the end of a billing period so that you can finish collecting data for the current billing period and start a new VTA session for the new billing period.

Mail Processor

You use the mail processor to set up the SRC VTA to send e-mail notifications to subscribers when certain events occur.

SAE Proxy Processor

You can use the SAE proxy processor to:

- Set an interim interval for a service.
- Set a service session timeout for a subscription.
- Set a session timeout for a subscriber.
- Start a subscription to a service. You can specify the parameter substitutions to use when the service is started.
- Stop a subscription to a service. You can include a reason for the subscription's being stopped. When the service is stopped, the reason is sent to the billing system so it can differentiate between service stops.

The SAE proxy processor is a proxy to the SAE external interface that resolves the subscriber interface based on the event types to which functions are applied.

- If a function is applied to SAE subscriber-tracking or service-tracking events, the processor finds the SAE reference in the event message.

  There is an exception if the CurrentSubscriberOnly parameter is set to false. In this case, the function finds subscribers in all SAEs with the NIC.

- If a function is applied to other events, the processor uses subscriber's ID in the event as the key for the NIC to find the SAE reference.

Script Runner Processor

The Script runner processor can invoke external executable scripts or JavaScripts. We recommend using JavaScript, where possible, for better performance.

- External scripts are executable programs, such as shell scripts, that are available on the SRC VTA's host. Each external script can perform a task and return a value. If the script returns a value, the value can be added to the current event as an event attribute.
- JavaScript programs are used to process attributes of a VTA event. For example, it can convert a VTA event attribute in a timestamp to a date string and add it to the event as a new attribute. The attribute can then be used for subsequent actions, such as sending an e-mail notification to the subscriber. The JavaScript program can refer to any attributes of the event being processed, and it must return a value.

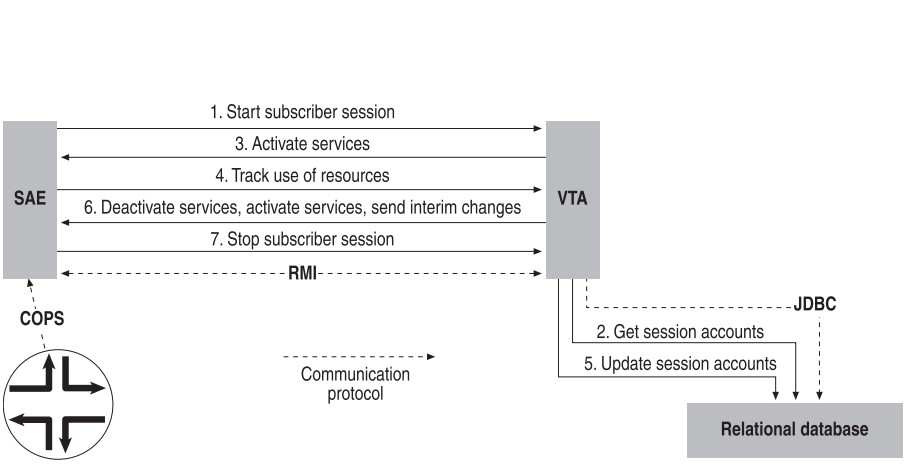
Related Topics

- Overview of the SRC VTA on page 65
- SRC VTA Operation on page 73
- Installing the SRC VTA and Running the Configuration Script on page 82
- Configuring Events on page 154
- Configuring Event Handlers on page 156
- SRC VTA Architecture and Connections to SRC Components on page 68

SRC VTA Operation

Figure 4 on page 73 illustrates the SRC VTA operation process.

Figure 4: Operation of the SRC VTA



The SRC VTA operates as follows:

1. When an event that activates a service occurs (for example, a subscriber logs in), the SAE sends a session start event to the SRC VTA through the SAE external plug-in interface.
2. Optionally, the SRC VTA queries a database for the subscriber's current set of sessions and accounts.
3. Depending on the configuration of the SRC VTA, it may activate or deactivate services based on the subscriber's use of resources.

4. The SRC VTA tracks the subscriber's use of resources for a period of time.
5. The SRC VTA updates sessions and account balances in the database.
6. The SRC VTA sends to the SAE changes in the interim accounting interval and takes action to limit excessive bandwidth use or to allow increased bandwidth use.
7. When an event that deactivates a service occurs (for example, the subscriber logs out), the SRC VTA updates the VTA session, closes the VTA session, and may update the account balances.

Events received through the remote interface can also cause the SRC VTA to activate services, deactivate services, or change the accounting interval.

Related Topics

- Overview of the SRC VTA on page 65
- How the SRC VTA Works on page 68
- Installing the SRC VTA and Running the Configuration Script on page 82
- SRC VTA Architecture and Connections to SRC Components on page 68

Managing VTA Accounts and Sessions

The SRC VTA allows service providers to manage accounts and sessions by:

- Identifying Subscribers, SAEs, and Sessions on page 74
- Managing VTA Accounts and Sessions on page 75
- Using SRC VTA Keys to Manage Accounts and Sessions on page 75
- Managing Subscriber Sessions and Service Sessions on page 76
- Using SRC VTA Keys to Manage Subscriber and Server Sessions on page 77

Identifying Subscribers, SAEs, and Sessions

The SRC VTA must be able to identify each subscriber by a unique identifier. The SRC VTA uses the identifier to manage:

- VTA accounts and sessions
- Subscriber and service sessions

You can configure the SRC VTA to use data keys to identify corresponding data values for these management tasks. The data keys depend on the subscriber's identifier and comprise one or more plug-in attributes.

Some identifiers are suitable for residential subscribers and some for enterprise subscribers. Because the SRC software supports only enterprise subscribers on routers running JUNOS Software, you cannot use some identifiers.

Related Topics

- Overview of the SRC VTA on page 65
- Managing VTA Accounts and Sessions on page 75
- Managing Subscriber Sessions and Service Sessions on page 76

- Locating the SAE That Manages a Subscriber for the SRC VTA on page 98
- Configuring Identifiers for Subscribers and Sessions on page 161

Managing VTA Accounts and Sessions

Depending on the information that identifies subscribers in your SRC configuration, you can configure the SRC VTA to use several types of plug-in attributes as data keys to identify accounts and sessions in the VTA database. If you use a NIC with the VTA portals, the SRC VTA can also use some of these plug-in attributes to construct a data key that the NIC can use to determine which SAE manages a subscriber. When the NIC identifies an SAE, the SRC VTA can also obtain a key to identify the subscriber session that the SAE is managing for the subscriber.

You configure the data keys that identify accounts and sessions in the SRC VTA Subscriber ID field. See "Configuring Identifiers for Subscribers and Sessions" on page 161 .

Related Topics

- Overview of the SRC VTA on page 65
- Managing Subscriber Sessions and Service Sessions on page 76
- Configuring a Database to Store Account and Session Data on page 84
- Configuring the SRC VTA to Manage Database Accounts on page 111
- Using SRC VTA Keys to Manage Accounts and Sessions on page 75
- Example: Limiting Subscriber Access Based on Account Balances on page 78

Using SRC VTA Keys to Manage Accounts and Sessions

Table 10 on page 76 shows the keys that you can specify for the SRC VTA to query the VTA database, NIC, and SAE. For the SRC VTA to use a subscriber identifier, the plug-in event must include the corresponding attribute(s) that are listed in the subscriber identifier row (attributes start with PA_). For more information about plug-in attributes, see the documentation for the SAE CORBA plug-in on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/management/src/api-index.html

Table 10: Keys That the SRC VTA Constructs to Manage Accounts and Sessions

Subscriber's Identifier	Database Key	Corresponding NIC Key	Corresponding SAE Key
Accounting ID	PA_ACCOUNTING_ID	PA_ACCOUNTING_ID	IP address that is returned from the NIC lookup
Subscriber DN	PA_USER_DN	PA_USER_DN	PA_USER_DN
Interface alias	PA_INTERFACE_ALIAS	None	None
Interface alias & VR	PA_INTERFACE_ALIAS@ PA_ROUTER_NAME	PA_ROUTER_NAME	None
Interface name & VR	PA_INTERFACE_NAME@ PA_ROUTER_NAME	PA_ROUTER_NAME	PA_INTERFACE_NAME
Login name	PA_LOGIN_NAME	PA_LOGIN_NAME	PA_LOGIN_NAME (default)
MAC address	PA_USER_MAC_ ADDRESS	None	None
PPP login name or public DHCP name	PA_PRIMARY_USER_ NAME	None	PA_PRIMARY_USER_ NAME
Port on router	PA_PORT_ID@ PA_ROUTER_NAME	None	None

Related Topics

- Overview of the SRC VTA on page 65
- Identifying Subscribers, SAEs, and Sessions on page 74
- Managing VTA Accounts and Sessions on page 75
- Configuring the SRC VTA to Manage Subscriber Accounts on page 115
- Using SRC VTA Keys to Manage Subscriber and Server Sessions on page 77

Managing Subscriber Sessions and Service Sessions

When the SRC VTA receives plug-in events, it may need to start or stop a subscriber session or service session. The plug-in events identify the SAE that manages a subscriber; however, the SRC VTA must construct a data key from one or more plug-in attributes to identify the subscriber session or service session. Depending on the information that identifies subscribers in your SRC configuration, you must configure the SRC VTA to use the keys shown in "Using SRC VTA Keys to Manage Subscriber and Server Sessions" on page 77. You configure these data keys in the SAE Subscriber Lookup field. See "Configuring Identifiers for Subscribers and Sessions" on page 161 .

Related Topics

- Overview of the SRC VTA on page 65
- Identifying Subscribers, SAEs, and Sessions on page 74
- Using SRC VTA Keys to Manage Accounts and Sessions on page 75
- Example: Limiting Subscriber Access Based on Account Balances on page 78

Using SRC VTA Keys to Manage Subscriber and Server Sessions

Depending on the information that identifies subscribers in your SRC configuration, you must configure the SRC VTA to use the keys shown in Table 11 on page 77.

Table 11: Keys That the SRC VTA Constructs to Manage Subscriber and Service Sessions

Subscriber's Identifier	SAE Key
DN	PA_USER_DN
Interface name on router	A combination of the following plug-in attributes: • PA_INTERFACE_NAME • PA_ROUTER_NAME
IP address	A combination of the following plug-in attributes: • PA_USER_IP_ADDRESS • PA_EVENT_TIME
IP address of the subscriber on an interface	A combination of the following plug-in attributes: • PA_USER_IP_ADDRESS • PA_INTERFACE_NAME • PA_ROUTER_NAME
Login name	PA_LOGIN_NAME
PPP login name or public DHCP name	PA_PRIMARY_USER_NAME

Related Topics

- Overview of the SRC VTA on page 65
- Identifying Subscribers, SAEs, and Sessions on page 74
- Managing VTA Accounts and Sessions on page 75
- Configuring the SRC VTA to Manage Subscriber Accounts on page 115
- Using One VTA Account for Multiple Subscriber Sessions on page 163
- Using SRC VTA Keys to Manage Accounts and Sessions on page 75

Example: Limiting Subscriber Access Based on Account Balances

The sample data provides an example called Quota that limits a subscriber's access rate based on the balances of accounts that record the subscriber's use of network resources. Subscribers receive a quota of transfer (upload and download) volume in two ways:

- Periodic quota—Volume that is periodically added to a subscriber's account. For example, a subscriber may receive a 25-MB periodic quota each month. The periodic quota is tracked in the periodic account.
- Bought quota—Additional volume that a subscriber can purchase and use at any time. For example, a subscriber may purchase 25 MB of bought quota in January, and use the bought quota between January and March. Bought quota is tracked in a bought account.

As a subscriber consumes volume, the SRC VTA debits the accounts, using first the periodic quota and, if no periodic quota is available, the bought quota.

Subscribers managed by a VTA require a subscription to quota services—services for which a VTA monitors and manages usage. You must configure these subscriptions to be activated when the subscriber logs in. When a subscriber logs in to the SAE, the SAE tries to activate the quota services. However, if neither the periodic account nor the bought account has a positive balance, the SRC VTA deactivates the quota services, and the SAE applies to the subscriber either the default policy or another policy that implements a service with a lower bandwidth.

The units of the accounts depend on the formula that you define to determine the use of network resources. With this formula, the SRC VTA calculates the change to the accounts and the interim accounting interval.

The Quota configuration example provides a VTA that operates as follows:

1. When a service session for the quota service starts, the SAE sends a start event to the SRC VTA.
2. The SRC VTA starts a VTA session that has the same identifier as the service session and a qualifier of zero.
3. When the SRC VTA receives the first interim update from the SAE in a VTA session, it records a balance change that details the use of resources and the event time for the VTA session. When the SRC VTA receives interim updates in the VTA session, it updates the use of resources and the event time in the balance change recorded previously.
    - If the periodic account contains sufficient resources to cover the balance change, the SRC VTA changes only the balance of that account.
    - If the periodic account does not contain sufficient resources for the change, the SRC VTA records one balance change for the resources available in that account and records another balance change for the difference in the bought account. In this case, the SRC VTA records subsequent balance changes to the bought account.

- If neither account has sufficient resources, the SRC VTA deactivates the quota service.

4. If the SAE session ends, the VTA session ends. When a new service session starts, Steps 1 to 3 recur. However, a service session may last for several VTA sessions. In this case, the SAE and SRC VTA continue the process described in Step 3.
5. When an administrator replenishes the periodic quota, the SRC VTA ends the VTA session, finalizes all balance changes for the session, and records a credit to the periodic account.
6. When the subscriber buys additional volume, the SRC VTA ends the VTA session, finalizes all balance changes for the session, and records a credit to the bought account.
7. When the SRC VTA next receives an interim update event from the SAE, it starts a new VTA session. The SRC VTA obtains the start time for the VTA session from the SAE event, and records debits to the accounts as described in Step 3.

   The SRC VTA always ends the VTA session when an administrator replenishes periodic quota or the subscriber buys volume. However, a service session may last for several billing periods. In this case, when the SRC VTA starts a new VTA session, it continues to assign the SAE session identifier to the VTA session and increments the qualifier by one. Keeping the VTA session within a billing period allows the SRC VTA to finalize balance changes.

Related Topics

- Overview of the SRC VTA on page 65
- Identifying Subscribers, SAEs, and Sessions on page 74
- Managing VTA Accounts and Sessions on page 75
- Managing Subscriber Sessions and Service Sessions on page 76

CHAPTER 6

Installing and Initially Configuring the SRC VTA

- Before You Install the SRC VTA on page 81
- Installing the SRC VTA and Running the Configuration Script on page 82
- Using JavaScript Programs in VTA Configurations on page 83
- Additional SRC VTA Configuration Script Tasks on page 83
- Configuring a Database to Store Account and Session Data on page 84
- Configuring the J2EE Application Server on page 85
- Creating Deployment Descriptors on page 86
- Example: Configuring the Oracle Database as the VTA Database on page 87
- Troubleshooting Database Deadlocks on page 88
- Configuring VTA Services and Policies on page 88
- Configuring Subscribers and Subscriptions to VTA Services on page 89
- Accessing the J2EE Application Server's Client Libraries on page 90
- Specifying How the SRC VTA Loads Configurations from the Directory on page 90
- Configuring the SAE to Send Tracking Events to the SRC VTA on page 92
- Specifying Tracking Plug-Ins for Enterprise Subscribers on JUNOS Routing Platforms on page 94
- Configuring the Event Queue on page 95
- Using NICs with the SRC VTA on page 98
- Renaming a VTA on page 100
- Modifying the VTA Renaming Rules on page 100

Before You Install the SRC VTA

Because VTAs rely on other components in the SRC network, you must complete several tasks before you install the SRC VTA software. After you install the software, you must also complete several configuration tasks before the application can function correctly.

Before you install and configure a VTA, you must:

1. Deploy a working SRC network.

   To support the SRC VTA, you must install SAEs to manage the routers or other devices through which subscribers connect to the network. You must also install and configure the directory in which you will store the SRC data.

2. Install a J2EE application server on the host that supports the SRC VTA.

   For information about installing JBoss (the J2EE application server provided with the SRC software), see “Installing Web Applications Inside JBoss” on page 7.

NOTE: If you are installing a J2EE application server other than JBoss, you need to remove the JBoss client libraries from the opt/UMC/sae/lib/plugins/ejb directory on every SAE and replace them with the relevant client libraries for the application server you are using.

3. On the host that supports the SRC VTA, install a relational database to store the data that the SRC VTA tracks.

Related Topics

- Overview of the SRC VTA on page 65
- How the SRC VTA Works on page 68
- SRC VTA Operation on page 73
- Installing the SRC VTA and Running the Configuration Script on page 82
- SRC VTA Architecture and Connections to SRC Components on page 68

Installing the SRC VTA and Running the Configuration Script

For information about the SRC VTA software, see “SRC Application Library Software” on page 3.

To install each VTA:

1. Install the Solaris packages for the SRC VTA (see “Installing SRC Application Packages” on page 5) on hosts that support JBoss.
2. Run the VTA configuration script on each SRC VTA host. The script configures JBoss for the SRC VTA and loads sample data. To invoke the script, access the folder */opt/UMC/conf/vta*, and run the **load** command.

   ```
   cd /opt/UMC/conf/vta
   ./load
   ```

3. On each host, restart JBoss and the SAE.
4. Configure the SRC VTA and associated components (see “Additional SRC VTA Configuration Script Tasks” on page 83 and “Running VTA Configuration Manager” on page 104).

Related Topics

- Overview of the SRC VTA on page 65

- Before You Install the SRC VTA on page 81
- Configuring VTA Services and Policies on page 88
- Installing VTA Configuration Manager on page 104

Using JavaScript Programs in VTA Configurations

You can use JavaScript programs in your VTA configuration for such tasks as calculating a usage metric or an interim accounting interval, specifying an event condition, updating event attributes in processors, and writing scripts to update accounts.

You can include referenced variables in a JavaScript program, such as plug-in attributes, event attributes, or account balances.

When a variable is referenced or updated in the JavaScript program, enclose it in angle brackets (<>) so that the JavaScript program retrieves only the necessary information. Within the JavaScript program, only one instance of the referenced variable must be enclosed in angle brackets.

You define formulas in the JavaScript scripting language (see http://wp.netscape.com/eng/mozilla/3.0/handbook/javascript/index.html). The Quota VTA executes the script in the Rhino JavaScript implementation (see http://www.mozilla.org/rhino).

Related Topics

- How the SRC VTA Works on page 68
- Configuring JavaScript Programs on page 143
- Configuring VTA Actions to Run Scripts on page 150
- Loading and Importing VTA Configurations on page 105
- Installing the SRC VTA and Running the Configuration Script on page 82

Additional SRC VTA Configuration Script Tasks

The VTA configuration script configures the components that it finds and accesses on each host. However, depending on your configuration and the components you use, such as the type of database and J2EE application server, you may need to manually configure some of the components or the configuration. The following topics describe the configuration procedures related to the SRC VTA; each topic explains which procedures the VTA configuration script completes. Topics include:

- Configuring a Database to Store Account and Session Data on page 84
- Configuring the J2EE Application Server on page 85
- Configuring VTA Services and Policies on page 88
- Configuring Subscribers and Subscriptions to VTA Services on page 89
- Accessing the J2EE Application Server's Client Libraries on page 90

- Specifying How the SRC VTA Loads Configurations from the Directory on page 90
- Configuring the SAE to Send Tracking Events to the SRC VTA on page 92
- Specifying Tracking Plug-Ins for Enterprise Subscribers on JUNOS Routing Platforms on page 94

Related Topics

- Overview of the SRC VTA on page 65
- How the SRC VTA Works on page 68
- Installing the SRC VTA and Running the Configuration Script on page 82
- Configuring Scripts That Update Accounts on page 114

Configuring a Database to Store Account and Session Data

The SRC VTA requires a relational database to store accounts and session data. For information about databases that we have tested for use with the SRC VTA, see the *SRC Application Library Release Notes*.

To configure a database:

1. For each VTA, create a database that uses the schema for the SRC VTA.

   To view the database schemas, see the following files, which are created when you install the Solaris package for the SRC VTA.

   - */opt/UMC/conf/vta/quota/vtadata.sql*
   - */opt/UMC/conf/vta/bucket/vtadata.sql*
   - */opt/UMC/conf/vta/quota/vtadata_oracle.sql*
   - */opt/UMC/conf/vta/bucket/vtadata_oracle.sql*

2. Configure access to the database for an administrator by using the SRC VTA to monitor and manage subscribers.

   The access parameters that you configure must match the access parameters that you configure for the data sources created in the J2EE application server (see “Configuring the J2EE Application Server” on page 85).

Related Topics

- Overview of the SRC VTA on page 65
- Managing VTA Accounts and Sessions on page 75
- Configuring the SRC VTA to Manage Database Accounts on page 111
- Troubleshooting Database Deadlocks on page 88
- Using SRC VTA Keys to Manage Accounts and Sessions on page 75
- Deleting Information from the VTA's Database on page 186

Configuring the J2EE Application Server

Before configuring the J2EE application server, install the JDBC driver that allows your database to connect to the J2EE application server, and restart the J2EE application server. See the documentation for the database to determine the required JDBC driver. For example, for JBoss, copy the driver to */opt/UMC/jboss/server/default/lib*.

NOTE: If you use JBoss, the VTA configuration script configures the J2EE application server, and you do not need to complete these tasks.

To configure the J2EE application server to support the SRC VTA:

1. For each VTA, create a data source with a Java Naming and Directory Interface (JNDI) name that matches the name of the data source (see Table 12 on page 85 for the default names).

   The load script copies default data-source deployment descriptors. These descriptors are appropriate for an environment in which you use JBoss with the MySQL database and have both of these applications running on the same host. The descriptor files begin with *mysql-*. For more information about these files, see the documentation for the version of JBoss included with the SRC software.

   To modify the names of the VTA deployment descriptors, see "Creating Deployment Descriptors" on page 86.

2. Set up a Java Message Service (JMS) connection factory, and link it to the resource environment reference jms/QueueConnectionFactory.

   The way you link the JMS connection factory to the resource environment reference depends on the J2EE application server. See the documentation for the J2EE application.

3. For each VTA, create a JMS queue for the ConnectionFactory class with the appropriate name for the SRC VTA (see Table 12 on page 85).

4. Create a role called VTA-Admin, and configure the administrator profiles so that administrators can access the VTA administration portal with this role.

5. Specify an authentication mechanism, and access parameters (such as a username and password) by which administrators can access the data source through the portals.

   The access parameters that you configure must match the access parameters that you configure for the database (see "Configuring a Database to Store Account and Session Data" on page 84).

Table 12: Names for Data Sources and JMS Queues

Type of VTA	Default Name for Data Sources	Name for JMS Queue
Quota	Quota/MySQLS	queue/Quota/SAEEventQueue

Table 12: Names for Data Sources and JMS Queues *(continued)*

Type of VTA	Default Name for Data Sources	Name for JMS Queue
Bucket	Bucket/MySQLS	queue/Bucket/SAEEventQueue

Related Topics

- Accessing the J2EE Application Server's Client Libraries on page 90
- SRC VTA Architecture and Connections to SRC Components on page 68
- Using SRC VTA Keys to Manage Subscriber and Server Sessions on page 77

Creating Deployment Descriptors

The enterprise archives (EAR files) for a VTA contain several modules that require deployment descriptors for the J2EE application server. In these EAR files are sample deployment descriptors for several J2EE application servers. See the *Release Notes* for information about the J2EE servers that we have tested with the SRC VTA.

To determine the names of the files that contain the descriptors and how to edit them, see the documentation for your J2EE application server.

To deploy the SRC VTA in a J2EE application server that we have not tested, you may need to develop your own deployment descriptors. In this case, you may be able to use the samples we provide as a guide.

To create a deployment descriptor:

1. Create a folder for the VTA on a host.

   mkdir vta

2. Copy the EAR file for the VTA from the archive file to the folder that you created in Step 1.
3. From the EAR file, extract the following files into the folder you created:
   - *vtacore.jar*
   - *datamgr.jar*
   - *quotadp.jar*

     For example:

     cd vta
     jar xvf quotavta.ear datamgr.jar

4. For each JAR file you extracted, extract the file that defines the deployment descriptors for the J2EE application server. For example, for JBoss:

   jar xvf datamgr.jar META-INF/jboss.xml

5. Edit the file that defines the deployment descriptors for the J2EE application server.

6. Replace in the JAR file the file that defines the deployment descriptors for the J2EE application server. For example:

   jar uvf datamgr.jar META-INF/jboss.xml

7. Replace the JAR file in the EAR file. For example:

   jar uvf quotavta.ear datamgr.jar

Related Topics

- How the SRC VTA Works on page 68
- SRC VTA Operation on page 73
- Configuring the J2EE Application Server on page 85
- Accessing the J2EE Application Server's Client Libraries on page 90

Example: Configuring the Oracle Database as the VTA Database

To connect VTA with an Oracle database:

1. For each VTA, create a database that uses the schema for the SRC VTA.

   To view the database schemas, see the following files, which are created when you install the Solaris package for the SRC VTA.

   - */opt/UMC/conf/vta/quota/vtadata_oracle.sql*
   - */opt/UMC/conf/vta/bucket/vtadata_oracle.sql*

2. Install the JDBC driver that allows your database to connect to the J2EE application server. For example, for JBoss, copy the driver to */opt/UMC/jboss/server/default/lib*.
3. Modify the Oracle data source configuration file with the access parameters (connection-url, user-name and password) and copy it to */opt/UMC/jboss/server/default/deploy*.

   - For quota VTA—*quota/oracle-quotavta-ds.xml*
   - For bucket VTA—*bucket/oracle-bucketvta-ds.xml*

4. Change the data source name and mapping in the EAR file for the VTA (*quotavta.ear* or *bucketvta.ear*).

   a. Extract *datamgr.jar* from the EAR file.

      For example:

      jar xvf quotavta.ear datamgr.jar

   b. Extract *META-INF/jbosscmp-jdbc.xml* from *datamgr.jar*.

      jar xvf datamgr.jar META-INF/jbosscmp-jdbc.xml

   c. In the *jbosscmp-jdbc.xml* file, change the data source name in the <jbosscmp-jdbc><defaults></defaults></jbosscmp-jdbc> XML element.

- For quota VTA—java:/Quota/OracleDS
- For bucket VTA—java:/Bucket/OracleDS

d. In the *jbosscmp-jdbc.xml* file, change the data source mapping in the <jbosscmp-jdbc><defaults></defaults></jbosscmp-jdbc> XML element.

- For Oracle8—Oracle 8
- For Oracle 9i or higher—Oracle9i

e. Replace *META-INF/jbosscmp-jdbc.xml* in *datamgr.jar*.

jar uvf datamgr.jar META-INF/jbosscmp-jdbc.xml

f. Replace *datamgr.jar* in the EAR file.

For example:

jar uvf quotavta.ear datamgr.jar

Related Topics

- Configuring a Database to Store Account and Session Data on page 84
- Configuring the SRC VTA to Manage Database Accounts on page 111

Troubleshooting Database Deadlocks

Problem The JBoss application server logs the following error when the database reports a deadlock—a condition in which the database operation cannot continue because two processes are both waiting for the other process to be completed before they proceed.

java.sql.SQLException: General error, message from server: "Deadlock found when trying to get lock; Try restarting transaction

Solution Deadlocks can occur for a variety of reasons in normal database operation. The SRC VTA resolves deadlocks in the database, and you should ignore this message.

Related Topics

- Installing Web Applications Inside JBoss on page 7
- Configuring a Database to Store Account and Session Data on page 84
- Configuring the SRC VTA to Manage Database Accounts on page 111
- Deleting Information from the VTA's Database on page 186

Configuring VTA Services and Policies

You do not need to complete the tasks in this topic if you used the VTA configuration script to load the sample data. The sample data includes services for the SRC VTA and policies for the services, and it configures the services to generate tracking events.

Only the SRC VTA should activate and deactivate services that the SRC VTA controls, and you must ensure that these services are not visible on a portal for subscribers to

control manually. You can use other services with a VTA if you design the policies and priorities for those services to work together.

For example, if you manage subscribers with a VTA, you can allow subscribers to manually activate a service that overrides the quota service, and consequently prevents charges in the periodic and bought accounts. You would account for use of this service through RADIUS rather than a VTA, and subscribers would incur an extra cost for using the service. In this case, you configure the overriding service with a higher precedence than the quota service.

To configure services for the SRC VTA:

1. Create services for which a VTA monitors and manages usage.
2. Configure policies that specify ingress and egress accounting rules consistent with the usage formula.

   For information about configuring accounting rules for a policy, see Policy Management Overview.

Related Topics

- How the SRC VTA Works on page 68
- Managing Subscriber Sessions and Service Sessions on page 76
- Configuring Subscribers and Subscriptions to VTA Services on page 89
- Installing the SRC VTA and Running the Configuration Script on page 82

Configuring Subscribers and Subscriptions to VTA Services

You need to add retailers and subscribers to the directory. If you are using the SRC VTA for testing or demonstration purposes, you can use the retailers and subscribers that are provided in the sample data. (For information about loading sample data, see “Installing the SRC VTA and Running the Configuration Script” on page 82.) If you do not load the sample data, you need to create at least one shared subscriber.

To configure subscribers to VTA services:

1. Add a retailer and at least one shared subscriber.

   See Adding Retailers (SRC CLI).
2. For all subscribers managed by the SRC VTA, create an individual or a group subscription to services for which a VTA monitors and manages usage.

   See Configuring Subscriptions (SRC CLI).
3. For the service, configure the subscriptions to automatically activate the service when the subscribers log in.

Related Topics

- Overview of the SRC VTA on page 65
- Automatic Login of Subscribers on page 180
- Locating the SAE That Manages a Subscriber for the SRC VTA on page 98

- Configuring VTA Services and Policies on page 88
- Configuring the SRC VTA to Manage Subscriber Accounts on page 115
- Accessing the Subscriber Portal on page 188

Accessing the J2EE Application Server's Client Libraries

If you use JBoss 3.2.6 on the host, the client libraries are packed with the SAE and located in the /opt/UMC/sae/lib/plugins/ejb directory, and you do not need to complete the following tasks. However, if you are using an application server other than JBoss, you need to remove the JBoss client libraries from this directory, and replace them with the relevant client libraries for the application server you are using by completing the following tasks on every SAE.

Each SAE that interacts with the SRC VTA requires access to the J2EE application server's client libraries. To provide this access:

1. Refer to the documentation for the J2EE application server to determine the locations and names of the files for the client libraries.

   When you install the SAE, the files for the JBoss client libraries are placed in the folder */opt/UMC/sae/lib/plugins/ejb*.
2. Delete the JBoss client libraries from the */opt/UMC/sae/lib/plugins/ejb* direcory.
3. Copy the files for the application server client libraries to the */opt/UMC/sae/lib/plugins/ejb* directory on each host that supports an SAE.
4. Using the **shared sae group name configuration plug-ins name ejb-adaptor** statement, set **classpath** option with the correct filename for the new libraries.
5. Restart the SAE on each host to which you copied the client libraries.

Related Topics

- Using JavaScript Programs in VTA Configurations on page 83
- Configuring the J2EE Application Server on page 85
- Configuring JavaScript Programs on page 143
- Specifying How the SRC VTA Loads Configurations from the Directory on page 90
- Loading a Configuration from a Directory on page 106

Specifying How the SRC VTA Loads Configurations from the Directory

Bootstrap properties specify how the SRC VTA loads configurations from the directory. If you install the directory on a different host than the J2EE application server, you must modify the bootstrap properties to specify the directory host.

To configure the bootstrap properties for each VTA:

1. Create a folder for the VTA on a host.

```
mkdir vta
```

2. Copy the EAR file for the VTA from the archive file to the folder that you created in Step 1.
3. From the EAR file, extract the file *vtacore.jar* into the folder you created.

```
cd vta
jar xvf quotavta.ear vtacore.jar
```

4. From the file *vtacore.jar*, extract the file *META-INF/ejb-jar.xml*.

```
jar xvf vtacore.jar META-INF/ejb-jar.xml
```

5. In the folder that you created in Step 1, edit the *META-INF/ejb-jar.xml* file.

   See “Properties in ejb-jar.xml file” on page 91 for information about the properties in this file.
6. Replace the file *META-INF/ejb-jar.xml* in the file *vtacore.jar.*

```
jar uvf vtacore.jar META-INF/ejb-jar.xml
```

7. Replace the file *vtacore.jar* in the EAR file.

```
jar uvf quotavta.ear vtacore.jar
```

Related Topics

- Loading and Importing VTA Configurations on page 105
- Loading a Configuration from a Directory on page 106
- Importing a VTA Configuration from a Local File on page 109
- Committing a VTA Configuration to a Directory on page 175
- Exporting a VTA Configuration to a Local File on page 176

Properties in ejb-jar.xml file

This topic describes the properties in the ejb-jar.xml file.

Config.java.naming.provider.url, Config.java.naming.security.principal, Config.java.naming.security.credentials, Config.java.naming.security.protocol

- Standard JNDI properties.
- Value—See Configuring Initial Directory Eventing Properties for SRC Components.

Config.net.juniper.smgt.lib.config.staticConfigDN

- Root of the static configuration properties for the SRC VTA.
- Value—DN
- Default—*l=VTA, ou=staticConfiguration, ou=Configuration, o=Management, o=umc*

Config.net.juniper.smgt.lib.config.dynamicConfigDN

- Root of the dynamic configuration properties for the SRC VTA.
- Value—DN
- Default—*ou=dynamicConfiguration, ou=Configuration, o=Management, o=umc*

Config.net.juniper.smgt.des.<propertySuffix>

- Defines how the SRC VTA monitors values that it reads from the directory.
- Value—See Configuring Initial Directory Eventing Properties for SRC Components.

vta.namespace

- Root namespace of the SRC VTA.
- Value—Path, relative to the root of the static configuration properties, that defines where the VTA's configuration is stored
- Example—/Applications/Quota

Configuring the SAE to Send Tracking Events to the SRC VTA

The SRC VTA communicates with the SAE through the Enterprise JavaBean (EJB) adapter plug-in. This plug-in is an SAE plug-in and performs the following functions:

- Filters SAE plug-in events for the SRC VTA.
- Adapts internal SAE events to EJB-compatible methods.
- Sends SAE tracking plug-in events to the SRC VTA.

To configure the EJB adapter plug-in:

1. From configuration mode, access the EJB adapter plug-in configuration. In this sample procedure, the EJB adapter plug-in called QuotaVTA is configured in the nw-area SAE group.

   user@host# edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adaptor

2. Configure the class name of the J2EE application server's JNDI service provider.

   [edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adaptor]
   user@host# set jndi-service-provider *jndi-service-provider*

3. Configure the URL of J2EE application server that is running JNDI service.

[edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adaptor]
user@host# set application-server-url *application-server-url*

4. Configure the JNDI name of the SAEEventListener EJB of the peer VTA.

   [edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adaptor]
   user@host# set jndi-sae-event-listener *jndi-sae-event-listener*

5. (Optional) Configure the LDAP filter that determines the subscriber and service events that the EJB adapter plug-in sends to the VTA. If you specify plug-in attributes in this field, you must include the same attributes in the attributes option.

   [edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adaptor]
   user@host# set event-admitter *event-admitter*

   Table 13 on page 93 lists the values that you can use for LDAP filter strings.

Table 13: Settings for Filter Strings

Filter String	Action
()	Matches no objects
(*)	Matches all objects
List of <attribute>= <value> pairs <attribute>—Name of a property or attribute <ldapAttributeName> <value>—One of the following: • * (asterisk) • Explicit string • String that contains an * **Note:** To define a special character (* & , ! | \ ) in a string, precede it with the backslash symbol (\).	• If <value> is *, checks for any value. • If <value> is an explicit string, checks whether any value of the property matches the string, regardless of case. • If <value> is a string that contains a *, checks whether any value of the property contains the string, regardless of case.
(&<filter><filter>...)	True if all filters match
(|<filter><filter>...)	True if at least one filter matches
(!<filter>)	True if the filter does not match

The variables in the filter include the names of plug-in attributes and a PluginEventType variable. The value of this variable is the name of the type of event, such as PE_START_SERVICE. For names of plug-in attributes and plug-in event types, see the SAE CORBA plug-in documentation on the Juniper Networks Web site at http://www.juniper.net/techpubs/software/management/src/api-index.html or in the SDK+AppSupport+Demos+Samples.tar.gz file on the Juniper Web site at https://www.juniper.net/support/csc/swdist-erx/src.html.

6. (Optional) Specifies whether or not the J2EE application server uses load balancing to determine the location that manages requests to the VTA.

   [edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adaptor]
   user@host# set use-ejb-cluster

7. Configure load-balancing scheme of the J2EE application server that hosts the VTA. See the documentation for the J2EE application server to determine which load-balancing scheme it supports.

   [edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adaptor]
   user@host# set ejb-clustering-strategy (EJBObjectClustering | EJBHomeClustering | JNDIClustering)

8. (Optional) Configure the plug-in attributes that the EJB adapter plug-in sends to the VTA listener. If you do not define a list of attributes, the EJB adapter plug-in sends all plug-in attributes to the VTA. Sending unnecessary plug-in attributes can adversely affect the performance of SRC components.

   [edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adaptor]
   user@host# set attributes [(host | router-name | interface-name | ...)...]

   Specify at least the following plug-in attributes: router-name, session-id, login-name, user-ip-address, ssp-host, domain, service-name, event-time, session-time, in-octets, out-octets, in-packets, out-packets, session-timeout, downstream-bandwidth, upstream-bandwidth, service-session-name, subscription-name. You may need to add attributes if you use them for the event admitter.

Related Topics

- How the SRC VTA Works on page 68
- Configuring the SAE Proxy Processor on page 136
- Configuring Actions for the SAE Proxy Processor on page 137
- Configuring the J2EE Application Server on page 85
- Specifying Tracking Plug-Ins for Enterprise Subscribers on JUNOS Routing Platforms on page 94

Specifying Tracking Plug-Ins for Enterprise Subscribers on JUNOS Routing Platforms

When user-tracking plug-ins are attached to the retailer on routers running JUNOS Software, login names are needed to trigger the user-tracking plug-in and generate user-tracking events. Because enterprise subscribers do not have a login name, the SRC VTA cannot get the required user-tracking events.

To allow enterprise subscribers on routers running JUNOS Software to use retailer-attached user-tracking plug-ins, configure the EJB adapter plug-in to filter SAE plug-in events for the SRC VTA and send SAE tracking events to the SRC VTA.

To use the EJB adapter plug-in to send events for a specific retailer:

1. Configure the event admitter of the EJB adapter plug-in (see “Configuring the SAE to Send Tracking Events to the SRC VTA” on page 92).

   Specify the PA_USER_DN event attribute with the retailer’s relative distinguished name (RDN). For example, the following event admitter matches events from subscribers in the SP-Quota retailer:

   ```
   PA_USER_DN=*SP-Quota*
   ```

2. Configure the EJB adapter plug-in as the global subscriber-tracking plug-in for the SAE. See Configuring Tracking Plug-Ins (SRC CLI).

   In the sample procedure, the EJB adapter plug-in called QuotaVTA is configured as the subscriber-tracking plug-in in the nw-area SAE group.

   ```
   [edit shared sae group nw-area configuration plug-ins event-publishers]
   user@host#  set subscriber-tracking QuotaVTA
   ```

Related Topics

- Types of Internal Plug-Ins
- Properties in ejb-jar.xml file on page 91
- Configuring Administrative Information for Enterprise Subscribers (SRC CLI)

Configuring the Event Queue

You can configure event queues that hold plug-in events from the SAE until the VTA processes them.

- Setting the Size of the Event Queue on page 95
- Calculating the Size of the Nonpersistent Event Queue on page 96
- Specifying the Type of Event Queue on page 96
- Configuring the Event Queue Size on page 97

Setting the Size of the Event Queue

The VTA has an event queue that holds plug-in events from the SAE until the VTA processes them. There are two types of event queues:

- With a persistent event queue, no events are lost even if the VTA or the J2EE application server fails and is restarted. If you are using a persistent event queue, set the size of the VTA event queue to about the number of events your VTA can process in 60 seconds.
- With a nonpersistent (in memory) event queue, events can be lost if the VTA or the J2EE application server fails. However, for performance reasons you may want to configure a nonpersistent event queue.

If you configure a nonpersistent event queue, the event queue size is the maximum number of events that can be lost if the application server or the VTA fails. If the SAE sends events faster than the VTA can process them, the event queue fills, and the VTA signals the SAE

to stop sending events for 60 seconds. Your VTA instances should be deployed on a host or cluster with sufficient throughput to handle events above the average rate generated by the SAEs in normal operation. This way, the event queue is sufficient to buffer peak event generation rates.

Related Topics

- Calculating the Size of the Nonpersistent Event Queue on page 96
- Specifying the Type of Event Queue on page 96
- Configuring the Event Queue Size on page 97
- Configuring Events on page 154

Calculating the Size of the Nonpersistent Event Queue

If communication between the SAE and VTA is lost for an extended period, a large backlog of events can build up in the SAE fail queue. This backlog can result in the SAE sending events far in excess of the average rate for an extended period. Note, however, that smaller event queue sizes affect recovery time if the connection between the SAE and the VTA is disrupted. When the VTA event queue fills, the VTA signals the SAE to stop sending events, and the SAE does not send any events for 60 seconds. If the VTA event queue is limited to less than the number of events the VTA can process in 60 seconds, it will empty the queue and be idle until the SAE starts sending it events again.

For example, if you set the VTA event queue size to the number of events the VTA can process in 30 seconds, after an SAE-VTA communication disruption the SAE rapidly sends events that it has stored in its local fail queue. After the VTA has collected the events that take it 30 seconds to process, it signals the SAE to stop sending events, and the SAE stops for 60 seconds. The VTA will complete processing the events in its queue in 30 seconds and then will be idle for 30 further seconds before the SAE starts to send events again. In this example, the VTA could clear the backlog of events in the SAE's fail queue twice as fast if the VTA's event queue was large enough to hold the number of events it can process in 60 seconds.

Contact the Juniper Technical Assistance Center or Juniper Professional Services if you need further advice about sizing the VTA event queue.

Related Topics

- Setting the Size of the Event Queue on page 95
- Specifying the Type of Event Queue on page 96
- Configuring the Event Queue Size on page 97
- Configuring Events on page 154

Specifying the Type of Event Queue

To specify whether the event queue is persistent or nonpersistent:

1. Create a folder for the VTA on a host.

   mkdir vta

2. Copy the EAR file for the VTA from the archive file to the folder that you created in Step 1.
3. From the EAR file, extract the file *vtacore.jar* into the folder you created.

   cd vta
   jar xvf quotavta.ear vtacore.jar

4. From the file *vtacore.jar*, extract the file *META-INF/ejb-jar.xml*.

   jar xvf vtacore.jar META-INF/ejb-jar.xml

5. In the folder that you created in Step 1, edit the *META-INF/ejb-jar.xml* file.

   See "Event Queue Property" on page 97 for information about the property that specifies the type of event queue.

6. Replace the file *META-INF/ejb-jar.xml* in the file *vtacore.jar.*

   jar uvf vtacore.jar META-INF/ejb-jar.xml

7. Replace the file *vtacore.jar* in the EAR file.

   jar uvf quotavta.ear vtacore.jar

Related Topics

- Setting the Size of the Event Queue on page 95
- Calculating the Size of the Nonpersistent Event Queue on page 96
- Configuring the Event Queue Size on page 97
- Configuring Events on page 154

Event Queue Property

This topic describes the event queue property.

PersistentQueue

- Type of queue—persistent or nonpersistent.
- Value—true or false
- Default—false

Configuring the Event Queue Size

To configure the VTA event queue size for JBoss, edit the MaxDepth attribute in the file */opt/UMC/conf/vta/quota/mq-quotavta-service.xml*, and then deploy this file in your JBoss deployment directory. For other application servers, consult the documentation for information about how to configure message queues.

Related Topics

- Setting the Size of the Event Queue on page 95
- Calculating the Size of the Nonpersistent Event Queue on page 96
- Specifying the Type of Event Queue on page 96

- Configuring Events on page 154

Using NICs with the SRC VTA

You use the Network Information Collector (NIC) to locate the SAE that manages a subscriber for an SRC VTA:

- Locating the SAE That Manages a Subscriber for the SRC VTA on page 98
- Configuring a NIC on page 99
- Configuring NIC Proxies for the VTA on page 99

Locating the SAE That Manages a Subscriber for the SRC VTA

You can use NIC proxies if the SRC VTA software needs to locate the SAE that manages a particular subscriber. For example, if the VTA receives an account update event and determines that it needs to reconfigure the corresponding SAE session, the VTA must find the SAE that is managing the session. The VTA can do this through the NIC.

You can also use the NIC with the SRC VTA to allow the following:

- Automatically log in subscribers to the VTA Web portals—The NIC maps the subscriber's IP address to the subscriber's login name, DN, or name of the interface and VR to which the subscriber connects. This scenario is for subscribers who connect to the SRC network through a JUNOSe router.
- Immediately activate subscriptions to quota services—The VTA immediately activates a subscriber's quota service when a deposit is made to the subscriber's account. In this case, the NIC maps the subscriber's identifier to the SAE reference. This scenario is for subscribers who connect to the network through routers running JUNOSe or JUNOS Software.

  If you do not set up a NIC for this purpose or you use an identifier that the NIC cannot map to an SAE reference, subscribers must log out and log in again before the VTA can activate their quota services when deposits are made to their accounts.
- Allow subscribers to log in with their IP addresses. The NIC maps the subscriber's IP address to the identifier that you use for subscribers in the VTA database. To use the sample VTA portals, you must implement this type of NIC. If you do not implement this NIC, you can provide another way for subscribers to log in, such as a central Web page on which subscribers can enter their usernames and passwords. This scenario is for subscribers who connect to the SRC network through a JUNOSe router.

Related Topics

- Identifying Subscribers, SAEs, and Sessions on page 74
- Configuring Subscribers and Subscriptions to VTA Services on page 89
- Configuring a NIC on page 99
- Configuring NIC Proxies for the VTA on page 99

Configuring a NIC

For demonstrations and installations with few subscribers, you can configure the VTA to use a NIC proxy stub, which explicitly defines a set of data mappings. However, for standard installation with a significant number of subscribers and multiple SAEs, you must set up a full NIC configuration.

To configure a NIC for the VTA management portals:

1. Use the OnePopLogin configuration scenario (see NIC Configuration Scenarios).
2. Plan and configure the NIC hosts. See Configuring the NIC (SRC CLI).
3. Add the NIC SAE agents to each SAE configuration as external plug-ins. Specify these plug-in attributes: router-name, session-id, user-type, login-name, user-ip-address.

   For information about configuring SAE plug-ins, see Configuring the SAE for External Plug-Ins (SRC CLI).
4. (Optional) Configure a NIC proxy stub. See Configuring NIC Test Data (SRC CLI) for information about configuring the NIC proxy stub.
5. Configure a NIC proxy for the VTA. See “Configuring NIC Proxies for the VTA” on page 99.

Related Topics

- Before You Configure the NIC
- Locating the SAE That Manages a Subscriber for the SRC VTA on page 98
- Starting the NIC (SRC CLI)
- Configuration Statements for the NIC

Configuring NIC Proxies for the VTA

For information about NIC proxies, see Overview of NIC Proxy Configuration.

To configure NIC proxies, select the NIC proxy that you want to configure.

- If subscribers connect to the network through a JUNOSe router, you can configure a NIC proxy that passes the subscriber’s IP address and receives the identifier that you configured for the subscriber. This NIC allows customers to log in through the Web portals.

  You must also specify the namespace of the NIC proxy in the *CONSTANTS.incl* file of the Web applications for the VTA portals (see “Properties for VTA Portals” on page 181).
- You can configure a NIC proxy that passes the subscriber’s identifier to a NIC resolver and receives the corresponding SAE reference. This NIC allows the VTA to immediately activate a subscriber’s quota service when a deposit is made to the subscriber’s account. This feature is available for subscribers who connect to the network through routers running JUNOSe or JUNOS Software.

For information about the parameters that you can configure for NIC proxies, see Configuration Statements for NIC Proxies.

Related Topics

- Before You Configure a NIC Proxy
- Locating the SAE That Manages a Subscriber for the SRC VTA on page 98
- Configuring a NIC on page 99

Renaming a VTA

When multiple VTA configurations are deployed on the same application server, each VTA must have a unique JNDI name. You can use a script to change the JNDI name of a VTA by generating a new EAR file in the */opt/UMC/conf/vta* directory based on the renaming rules specified in the */opt/UMC/conf/vta/rules.xml* file.

To rename a VTA, specify the absolute path of the EAR file by executing the **rename** command in the */opt/UMC/conf/vta* folder:

```
./rename <earFilePathName> [<rulesFilePathName>]
```

where <earFilePathName> is the absolute pathname of the EAR file that you want to rename, and <rulesFilePathName> is the absolute pathname of the renaming rules file.

You must specify <rulesFilePathName> if you change the default location from the */opt/UMC/conf/vta* file or you are using a different rules file. By default, the renaming rules rename the Quota VTA called *quotavta.ear* to *bucketvta.ear*.

```
cd /opt/UMC/conf/vta
./rename /opt/UMC/conf/vta/quotavta.ear
```

Related Topics

- Overview of the SRC VTA on page 65
- SRC VTA Operation on page 73
- Installing the SRC VTA and Running the Configuration Script on page 82
- Additional SRC VTA Configuration Script Tasks on page 83
- Modifying the VTA Renaming Rules on page 100

Modifying the VTA Renaming Rules

To modify the rules used to rename a VTA, edit the */opt/UMC/conf/vta/rules.xml* file. For example, you can substitute different oldString and newString attributes for the <replace> element values and save the new values to a new file that you specify as the renaming rules file for the **rename** command.

For information about modifying the renaming rules, see the */opt/UMC/conf/vta/rules.xml* file. This file contains details about the replacement values for the VTA names.

Related Topics

- Overview of the SRC VTA on page 65
- SRC VTA Operation on page 73

- Renaming a VTA on page 100
- Installing the SRC VTA and Running the Configuration Script on page 82

CHAPTER 7

Configuring the SRC VTA with VTA Configuration Manager

- Installing VTA Configuration Manager on page 104
- Running VTA Configuration Manager on page 104
- Loading and Importing VTA Configurations on page 105
- Loading a Configuration from a Directory on page 106
- Connecting to the Directory Fields on page 108
- Importing a VTA Configuration from a Local File on page 109
- Accessing the VTA Configuration on page 110
- Configuring the SRC VTA to Manage Database Accounts on page 111
- Configuring Scripts That Update Accounts on page 114
- Configuring the SRC VTA to Manage Subscriber Accounts on page 115
- Configuring a Usage Metric for Service Accounts on page 117
- Sample Formulas for Usage Metrics for the SRC VTA on page 120
- Configuring an Interim Accounting Interval for Service Accounts on page 121
- Adjusting the Interim Accounting Interval for a Service on page 123
- Service Variables on page 124
- Configuring Actions for the Database Engine Processor on page 126
- Setting Up the SRC VTA to Send E-Mail Notifications on page 130
- Configuring the SRC VTA to Send E-Mail Notifications on page 132
- Configuring the SAE Proxy Processor on page 136
- Configuring Actions for the SAE Proxy Processor on page 137
- Configuring the SRC VTA to Run Scripts on page 143
- Configuring JavaScript Programs on page 143
- JavaScript Fields on page 146
- Configuring External Scripts on page 147
- External Script Fields on page 149
- Configuring VTA Actions to Run Scripts on page 150

- Configuring Events on page 154
- Configuring Event Handlers on page 156
- Configuring Identifiers for Subscribers and Sessions on page 161
- Using One VTA Account for Multiple Subscriber Sessions on page 163
- Logging Event Messages for the SRC VTA on page 165
- Logging Events Messages to a Text File on page 166
- Logging Events Messages to a System Logging Server on page 171
- Validating VTA Configurations on page 174
- Committing a VTA Configuration to a Directory on page 175
- Exporting a VTA Configuration to a Local File on page 176

Installing VTA Configuration Manager

VTA Configuration Manager is a Web application that lets you use a Web browser to configure the SRC VTA. To install VTA Configuration Manager:

1. Copy the *vtaconfig.war* file from the archive file to the *jboss/server/default/deploy* folder.
2. If you have not run the VTA configuration script, copy the *roles.properties* and *users.properties* files from */opt/UMC/conf/vta* to your *jboss/server/default/conf* folder.
3. In the *roles.properties* file, add the following property.

   admin=VTA_Admin

   VTA_Admin is used as the login username for VTA Configuration Manager. You can configure a different login username; however, you cannot change " =VTA_Admin."
4. In the *users.properties* file, add the following property.

   admin=secret

   where secret is the password used to log in to VTA Configuration Manager.

Related Topics

- How the SRC VTA Works on page 68
- Configuring Web Applications for the SRC VTA on page 180
- Installing the SRC VTA and Running the Configuration Script on page 82
- Running VTA Configuration Manager on page 104

Running VTA Configuration Manager

To begin using VTA Configuration Manager:

1. In your Web browser, enter the name or IP address of the VTA host and the port number for VTA Configuration Manager in the format:

http://<host>:8080/vtaconfig

A Connect to dialog box appears

2. In the Connect to dialog box, enter your username and password, and click **OK**. The default values are:

   User name—admin
   Password—secret

   The main VTA Configuration Manager page appears.

Related Topics

- How the SRC VTA Works on page 68
- Installing the SRC VTA and Running the Configuration Script on page 82
- Installing VTA Configuration Manager on page 104
- Loading and Importing VTA Configurations on page 105
- Accessing the VTA Configuration on page 110

Loading and Importing VTA Configurations

You can store your VTA configurations in a directory or in a file on the local host:

- Directory—If you store the configurations in a directory, you need to load a configuration before you work on it in VTA Configuration Manager. When you are finished with a configuration, you need to commit the configuration to the directory.

  If you loaded the VTA sample data, it is in the following files in the directory:

  - */Applications/Bucket*
  - */Applications/HostCheck*
  - */Applications/Quota*

See “Loading a Configuration from a Directory” on page 106 .

- Local host—If you store the configurations in files on the local host, you need to import a configuration file before you work on the configuration in VTA Configuration Manager. When you are finished with a configuration, you need to export the configuration to a file.

  See “Importing a VTA Configuration from a Local File” on page 109 .

Related Topics

- Specifying How the SRC VTA Loads Configurations from the Directory on page 90
- Installing the SRC VTA and Running the Configuration Script on page 82
- Accessing the VTA Configuration on page 110
- Committing a VTA Configuration to a Directory on page 175
- Exporting a VTA Configuration to a Local File on page 176

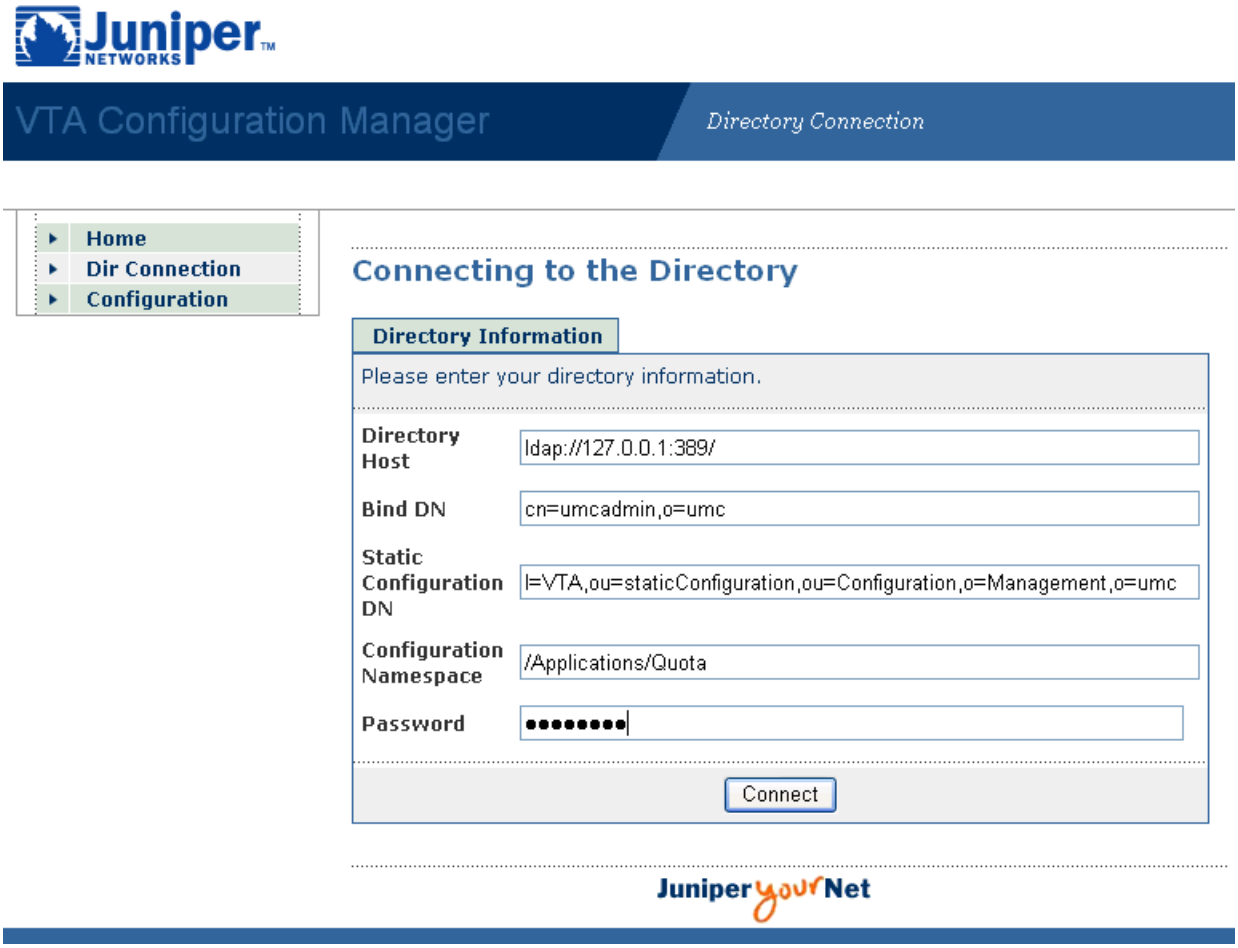
Loading a Configuration from a Directory

If your VTA configuration is stored in a directory, you need to connect to the directory before you load configurations into VTA Configuration Manager.

To connect to a directory and load a configuration from the directory:

1. In the VTA Configuration Manager navigation pane, select **Dir Connection**.

   The Connecting to the Directory page appears.



2. Edit or accept the default values for the fields, and click **Connect**.

   See "Connecting to the Directory Fields" on page 108.

3. In the VTA Configuration Manager navigation pane, select **Load**.

   The software loads the configuration that you specified in the Configuration Namespace field, and the Edit Configuration page appears with your configuration. From this page, you can view or change your VTA configuration.

Related Topics

- Loading and Importing VTA Configurations on page 105
- Importing a VTA Configuration from a Local File on page 109
- Committing a VTA Configuration to a Directory on page 175
- Specifying How the SRC VTA Loads Configurations from the Directory on page 90
- Accessing the VTA Configuration on page 110

Connecting to the Directory Fields

Use the fields described in this section to configure the connection to the VTA configuration in the directory. In VTA Configuration Manager, you can edit the following fields in the Connecting to the Directory screen.

Directory Host

- Location of the directory in which VTA configurations are stored.
- Value—ldap://<IP address or hostname>:<port number>/
- Default—ldap://127.0.0.1:389/

Bind DN

- DN used to authenticate access to the directory.
- Value—DN
- Default—*cn=umcadmin, o=umc*

Static Configuration DN

- Subtree in the directory in which the VTA configuration is stored.
- Value—<DN>
- Default—*l=VTA, ou=staticConfiguration, ou=Configuration, o=Management, o=umc*

Configuration Namespace

- Name and path of the configuration file that you want to load from the directory or that you want to commit to the directory.
- Value—Directory path and name of the configuration file. The sample data is in the following files:
  - */Applications/Bucket*
  - */Applications/HostCheck*
  - */Applications/Quota*
- Default—*/Applications/Quota_Test*

Password

- Password used to connect to the directory.
- Value—Valid directory password
- Default—admin123

Importing a VTA Configuration from a Local File

If your VTA configuration is stored in a local file, you need to import the file into VTA Configuration Manager. To do so:

1. In the VTA Configuration Manager navigation pane, select **Import**.

   The Import Configuration page appears.

2. Enter the filename and path in the From file field, or click **Browse** and select a file from the local host.
3. Click **Finish**.

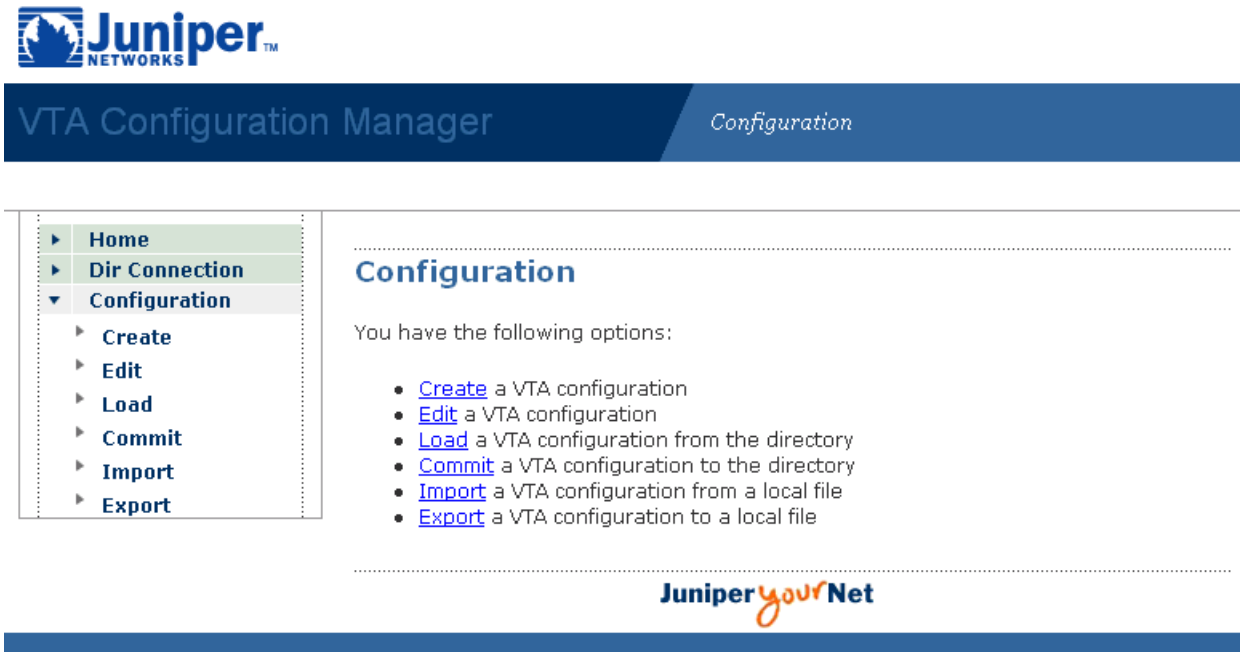
Related Topics

- Loading a Configuration from a Directory on page 106
- Loading and Importing VTA Configurations on page 105
- Committing a VTA Configuration to a Directory on page 175
- Specifying How the SRC VTA Loads Configurations from the Directory on page 90
- Accessing the VTA Configuration on page 110

Accessing the VTA Configuration

To configure the SRC VTA, select Configuration in the VTA Configuration Manager navigation pane.

CAUTION: If you have previously loaded or imported a configuration, and you select Create and create a new configuration, the configuration that you loaded or imported will be overwritten by the new configuration when you commit or export the configuration.



From this screen you can perform the following tasks:

- Configuring the SRC VTA to Manage Database Accounts on page 111
- Setting Up the SRC VTA to Send E-Mail Notifications on page 130
- Configuring the SAE Proxy Processor on page 136
- Configuring the SRC VTA to Run Scripts on page 143
- Configuring Events on page 154
- Configuring Event Handlers on page 156
- Configuring Identifiers for Subscribers and Sessions on page 161
- Using One VTA Account for Multiple Subscriber Sessions on page 163
- Logging Event Messages for the SRC VTA on page 165

- Validating VTA Configurations on page 174
- "Committing a VTA Configuration to a Directory" on page 175

Related Topics

- Before You Install the SRC VTA on page 81
- Installing the SRC VTA and Running the Configuration Script on page 82
- Installing VTA Configuration Manager on page 104
- Loading and Importing VTA Configurations on page 105
- Exporting a VTA Configuration to a Local File on page 176

Configuring the SRC VTA to Manage Database Accounts

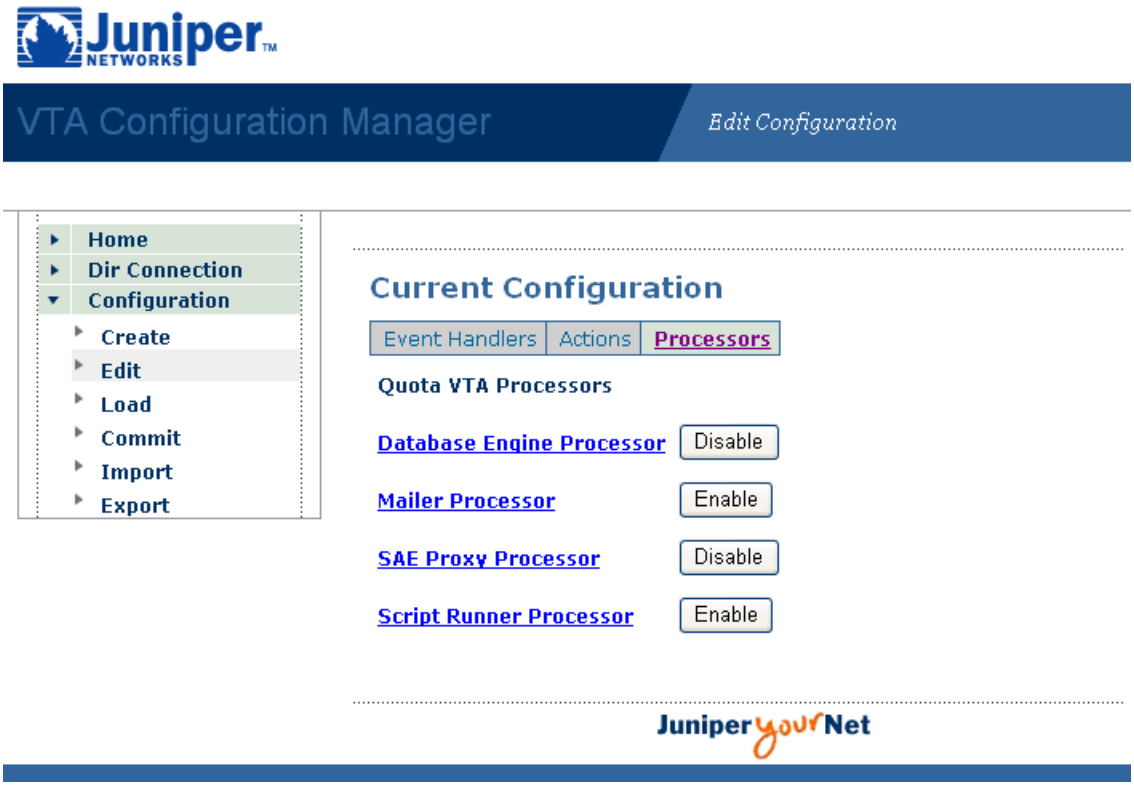
The SRC VTA uses the database engine processor to update database accounts. The database engine processor works as a proxy to a database. It can calculate usage, update account balances, get account and active service session data, and set initial balances of subscriber accounts. You can also use it to dynamically adjust interim accounting intervals based on a service or based on a subscriber's remaining resources and use of the network for that service.

The database engine processor consists of account update scripts, subscriber accounts, service accounts, and actions.

To configure the database engine processor:

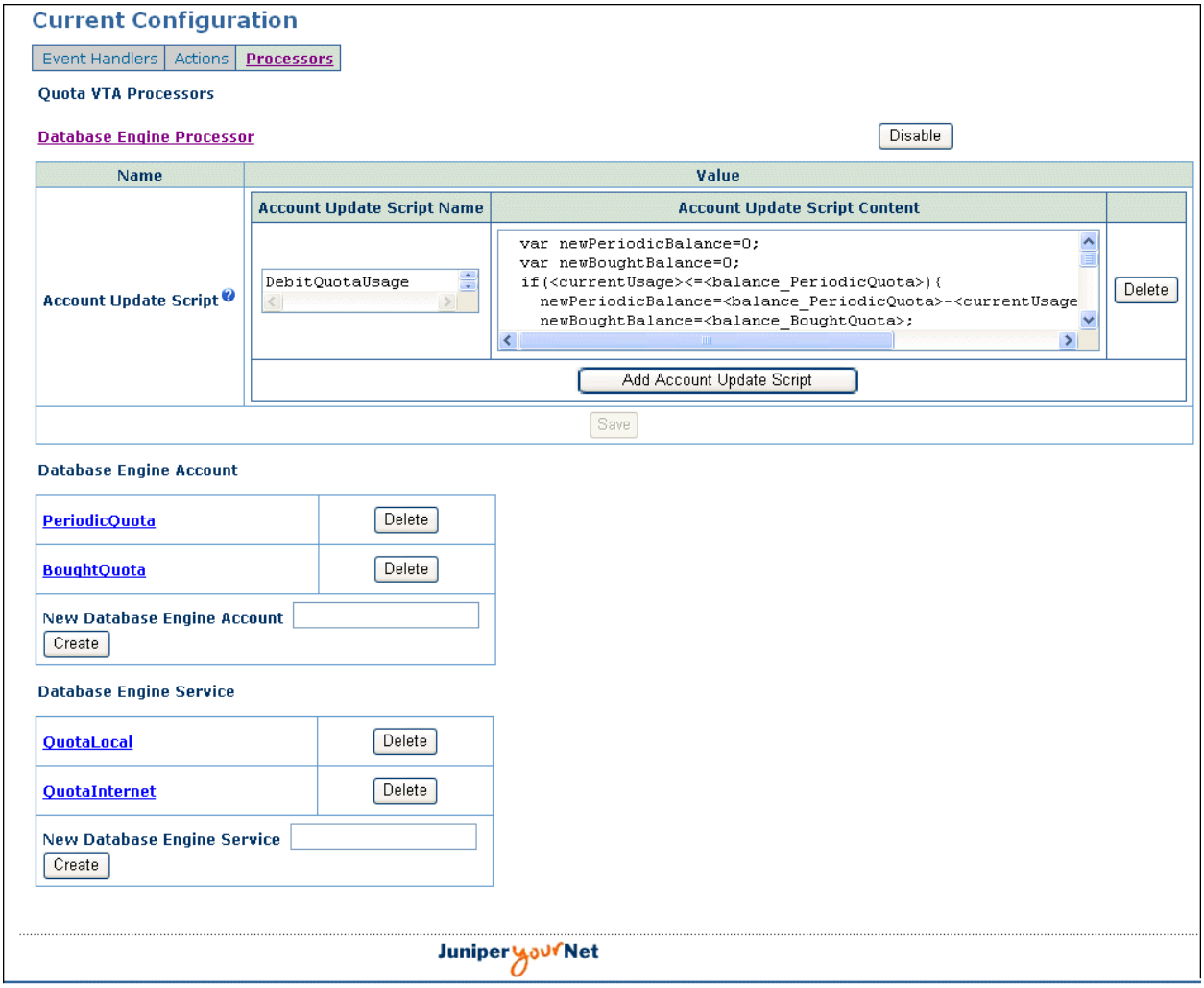
1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, select **Processors**.

A list of processors appears.



3. Click **Enable** to enable the processor and display the configuration page for the database engine processor, or click **Database Engine Processor**.

   The Database Engine Processor screen appears.



4. You can enable or disable the processor and perform the following tasks:
   - Configuring Scripts That Update Accounts on page 114
   - Configuring the SRC VTA to Manage Subscriber Accounts on page 115
   - Configuring a Usage Metric for Service Accounts on page 117
   - Adjusting the Interim Accounting Interval for a Service on page 123
   - "Configuring Actions for the Database Engine Processor" on page 126

Related Topics

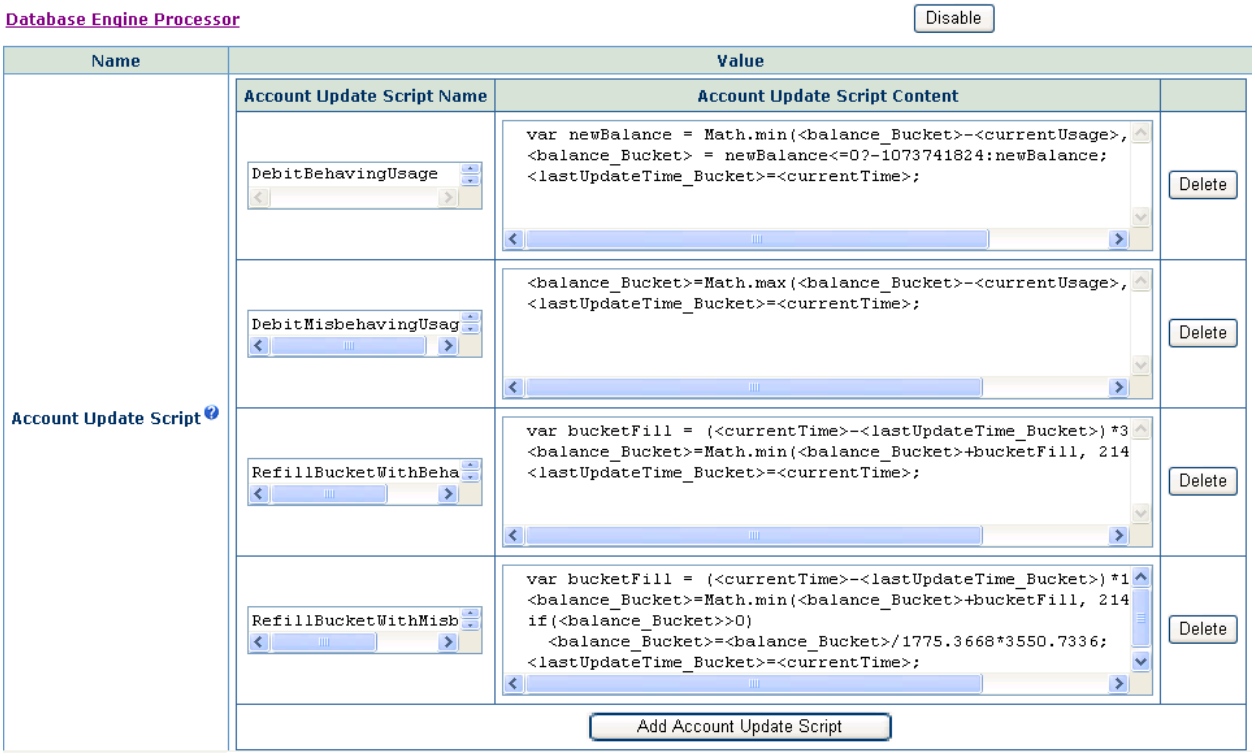
- Database Engine Processor Fields on page 122
- Accessing the VTA Configuration on page 110
- Troubleshooting Database Deadlocks on page 88

Configuring Scripts That Update Accounts

You can set up scripts to update balances in the accounts from which the usage of a service is charged and update accounts by assigning values to variables for the account balances.

You can create or delete account update scripts:

- To add a script, click **Add Account Update Script**, enter a name for the script in the Add Account Update Script field that appears, and click **Create**.
- To delete a script, click **Delete** next to the script that you want to delete.



Related Topics

- Configuring the SRC VTA to Run Scripts on page 143
- Configuring the SRC VTA to Manage Database Accounts on page 111
- Example of a Bucket VTA on page 193

Account Update Script Fields

Account Update Script Name

- Name of the script.
- Value—Name of a script
- Default—No value

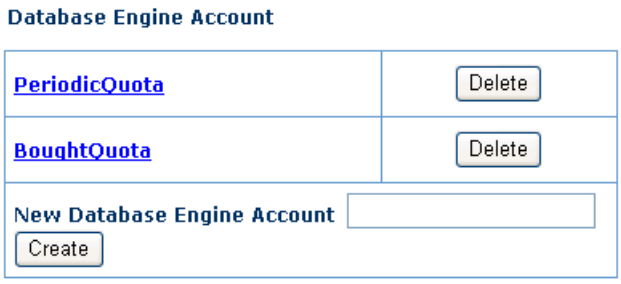
Account Update Script Content

- JavaScript program that updates a subscriber's account. The script can refer to the attribute name of any attributes in the event being processed.
- Value—Fields of an account can be updated by assigning values to the following parameters:
  - balance_<accountName>—Values written to this parameter are put in the balance field of the account.
  - status_<accountName>—Values written to this parameter are put in the status field of the account.
  - lastUpdateTime_<accountName>—Values written to this parameter are put in the last_update_time field of the account.
- Example—<balance_PeriodicQuota>=<balance_PeriodicQuota>-\ <currentUsage>;<lastUpdateTime_PeriodicQuota>=<currentTime>;

Configuring the SRC VTA to Manage Subscriber Accounts

A subscriber account is a record of credit and debit entries in the database that track a subscriber's use of a particular network resource.

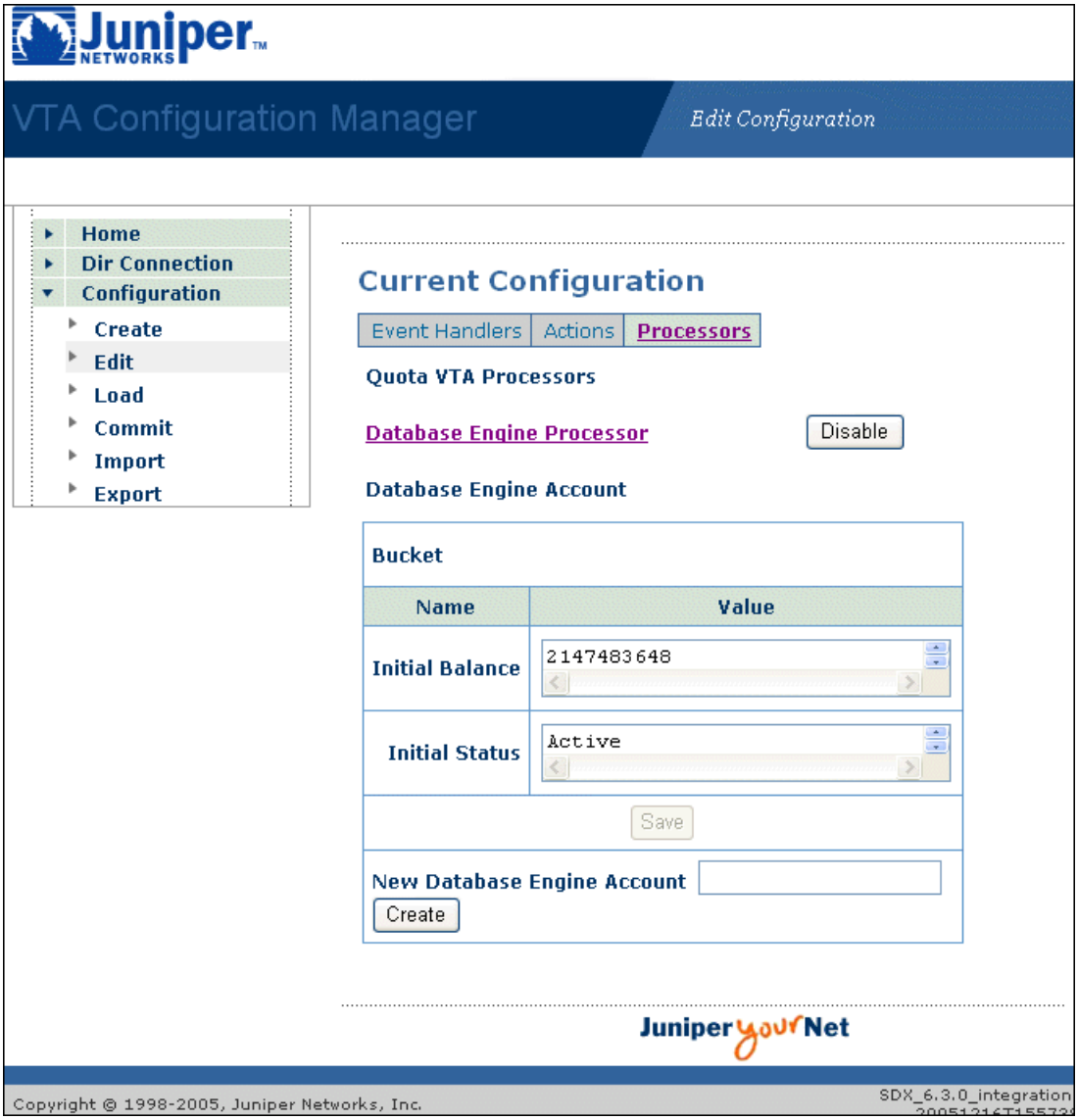
You configure subscriber accounts in the Database Engine Account section of the Database Engine Processor configuration page. For example:



You can create, delete, or modify accounts:

- To add an account, enter a name for the account in the New Database Engine Account field, and click **Create**.
- To delete an account, click **Delete** next to the account that you want to delete.
- To modify an account, select the account that you want to modify.

If you create or modify an account, the account configuration screen appears.



Related Topics

- Configuring the SRC VTA to Manage Database Accounts on page 111
- Configuring VTA Services and Policies on page 88
- Configuring Subscribers and Subscriptions to VTA Services on page 89

Database Engine Processor Buckets

Initial Balance

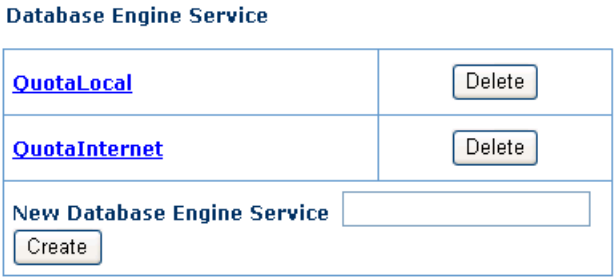
- Initial balance for the subscriber account.
- Value—Integer in the range –9223372036854775807 through 9223372036854775807
- Default—No value

Initial Status

- Initial status for the subscriber account.
- Value—Text string
- Example—active

Configuring a Usage Metric for Service Accounts

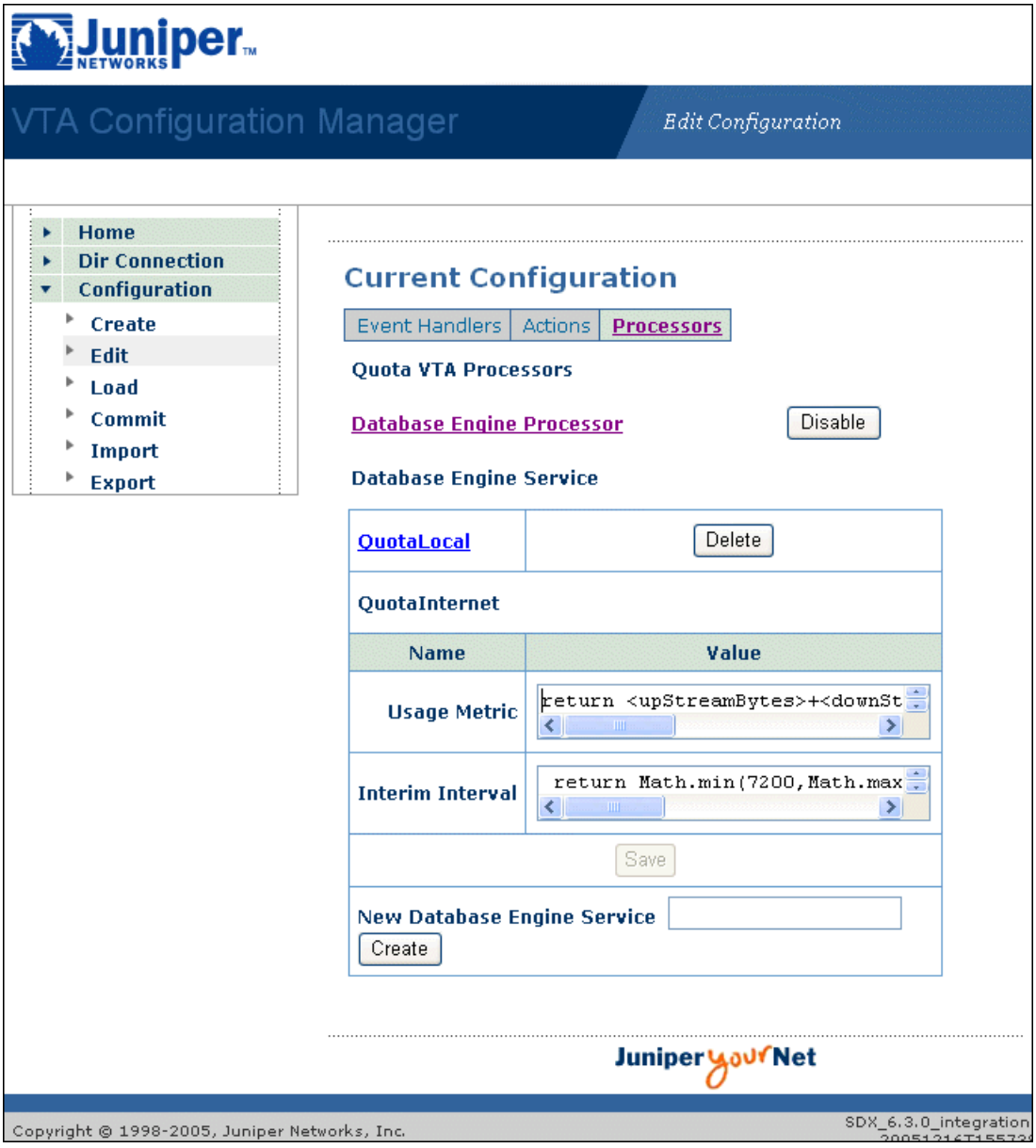
You configure VTA service accounts in the Database Engine Service section of the Database Engine Processor configuration page. For example:



You can create, delete, or modify accounts:

- To add an account, enter a name for the account in the New Database Engine Service field, and click **Create**.
- To delete an account, click **Delete** next to the account that you want to delete.
- To modify an account, select the account that you want to modify.

If you create or modify an account, the account configuration screen appears.



Related Topics

- Defining a Formula for Determining Network Resource Usage That the SRC VTA Evaluates on page 119
- Configuring an Interim Accounting Interval for Service Accounts on page 121
- Viewing Information About the Account on page 189
- Sample Formulas for Usage Metrics for the SRC VTA on page 120

Defining a Formula for Determining Network Resource Usage That the SRC VTA Evaluates

In the Usage Metric box in the VTA Configuration Manager, you define a formula that determines the use of network resources for a service. Each service in a VTA can use a different formula. You can configure the SRC VTA software to evaluate this formula for every accounting event it receives from the SAE for each quota service. It can then debit the result from the accounts. Use the variables described in this section to define the formula.

downStreamBytes

- Amount of data that the subscriber downloaded from the network since the last accounting event.
- Value—Number of bytes in the range 0–9223372036854775807

downStreamPackets

- Number of data packets that the subscriber downloaded from the network since the last accounting event.
- Value—Integer in the range 0–9223372036854775807
- Guidelines—Do not use downStreamPackets in a usage formula and maxUsageRate in the interim interval formula for the same service at the same time.

interimTime

- Time since the last accounting event.
- Value—Number of seconds in the range 0–2147483647
- Guidelines—Generally, this value equals the interim accounting interval; however, it may exceed the interim accounting interval if an accounting event is lost. Similarly, the value may be less than the interim accounting interval if a stop event occurs in the middle of an accounting interval.

upStreamBytes

- Amount of data that the subscriber uploaded to the network since the last accounting event.
- Value—Number of bytes in the range 0–9223372036854775807

upStreamPackets

- Number of data packets that the subscriber uploaded to the network since the last accounting event.
- Value—Integer in the range 0–9223372036854775807
- Guidelines—Do not use upStreamPackets in a usage formula and maxUsageRate in the interim interval formula for the same service at the same time.

Sample Formulas for Usage Metrics for the SRC VTA

Table 14 on page 120 provides examples of usage formulas.

Table 14: Examples of Formulas That Calculate Use of Network Resources

Formula	Description	Function
return <upStreamBytes> + <downStreamBytes>	Number of bytes sent and received by the subscriber.	Tracks volume of data that the subscriber transfers.
return 2*<upStreamBytes> + <downStreamBytes>	Twice the number of sent bytes plus the number of received bytes.	Allows higher charges for subscribers who are operating servers.
return <interimTime>	Time the subscriber is connected.	Tracks time that the subscriber connects rather than volume of data transfer.
return <downStreamBytes>/<interimTime>	Rate of downstream data transfer.	Allows higher charges for higher transfer rates.
QuotaInternet formula: return <upStreamBytes> + <downStreamBytes>– (<upStreamPackets> + <downStreamPackets>)*20 QuotaLocal formula: return (<upStreamBytes> + <downStreamBytes>– (<upStreamPackets> + <downStreamPackets>)*20)/2	Formulas for separate, complementary services in a single VTA. The following expression returns the total number of bytes in the IP headers of packets uploaded and downloaded by the service, and as such is not subscriber data. It is not counted as usage. (<upStreamPackets>+ <downStreamPackets>)*20	Provides support for two services: QuotaInternet for Internet service and QuotaLocal for local service. Allows higher charges for Internet service than for local service. By allocating a fixed usage limit for both services to each subscriber, encourages subscribers to access local resources due to decreased cost.

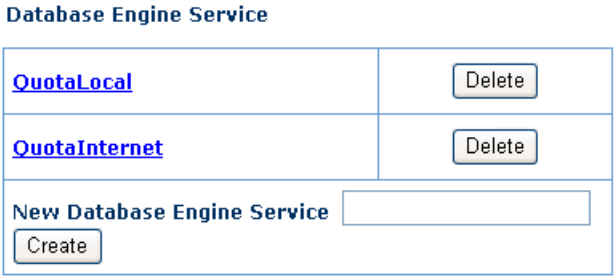
Related Topics

- Configuring a Usage Metric for Service Accounts on page 117

- Defining a Formula for Determining Network Resource Usage That the SRC VTA Evaluates on page 119

Configuring an Interim Accounting Interval for Service Accounts

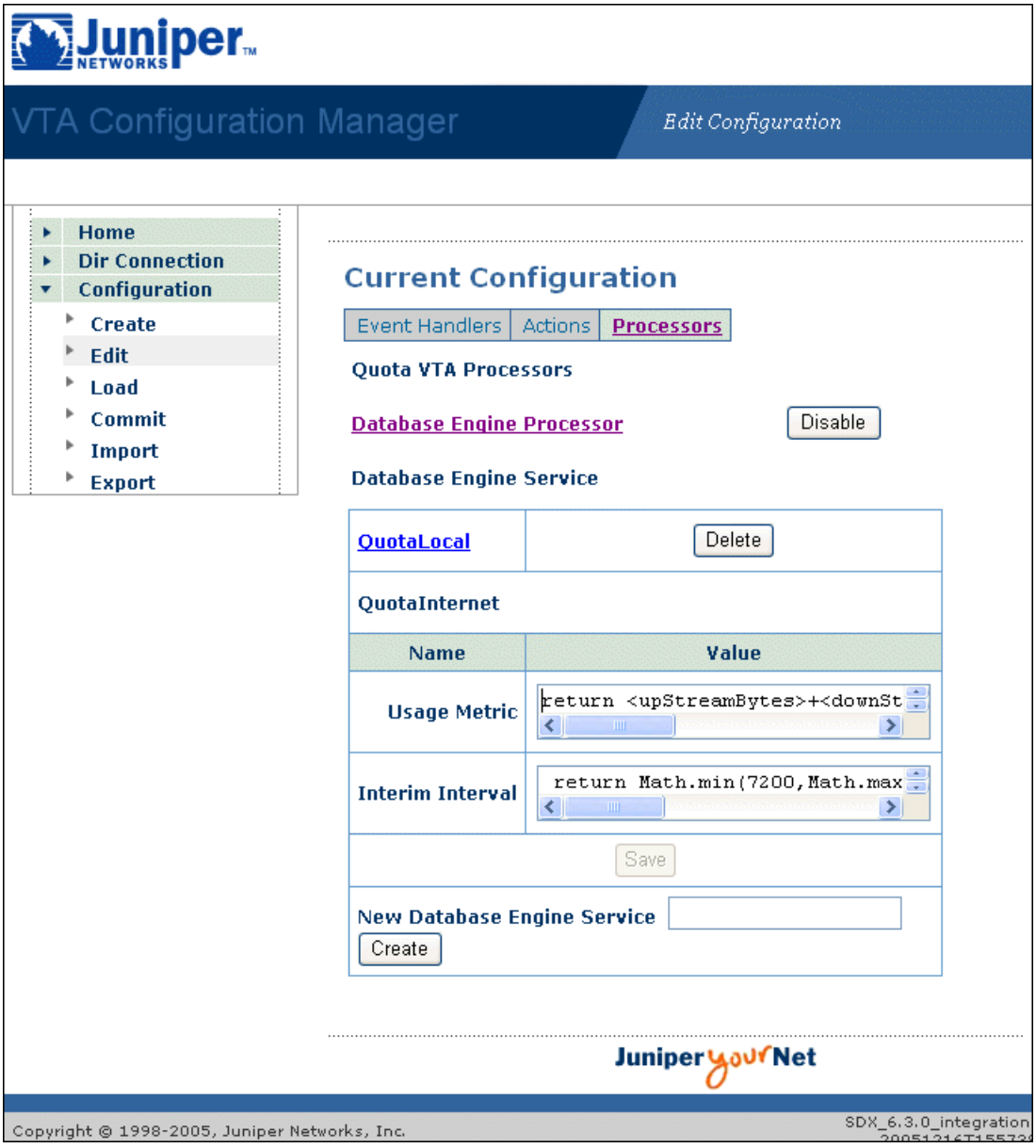
You configure VTA service accounts in the Database Engine Service section of the Database Engine Processor configuration page. For example:



You can create, delete, or modify accounts:

- To add an account, enter a name for the account in the New Database Engine Service field, and click **Create**.
- To delete an account, click **Delete** next to the account that you want to delete.
- To modify an account, select the account that you want to modify.

If you create or modify an account, the account configuration screen appears.



Related Topics

- Configuring a Usage Metric for Service Accounts on page 117
- Configuring VTA Services and Policies on page 88
- Adjusting the Interim Accounting Interval for a Service on page 123
- Viewing Information About the Account on page 189
- Database Engine Processor Fields on page 122

Database Engine Processor Fields

New Database Engine Service

- Name of the service account.
- Value—Text string
- Default—No value

Usage Metric

- Formula that calculates usage based on an accounting event for the specified service. The formula is in the form of a JavaScript program, and it can specify variables.

  For information about these variables, see Defining a Formula for Determining Network Resource Usage That the SRC-VTA Evaluates .

- Example—return (<upStreamBytes>+<downStreamBytes>-(<upStreamPackets>+<downStreamPackets>)*20)/2;
- Default—No value

Adjusting the Interim Accounting Interval for a Service

In the Interim Interval box, you define a formula to dynamically adjust the interim accounting interval for each service based on the subscriber's remaining resources and use of the network for that service. Each service in the SRC VTA can use a different formula. You can configure the SRC VTA software to evaluate the formula to obtain the accounting intervals. Depending on the result, the SRC VTA performs the following functions:

- If the result is zero, the SRC VTA disables interim accounting.
- If the result is a negative number, the SRC VTA does not change the interim accounting interval.
- If the result is a positive number, the SRC VTA changes the interim accounting interval to this value.

The variables are categorized as:

- Current service—Provides session data of the service for the current service-tracking event.
- Other service—Provides service session usage information for another subscriber service for the current service-tracking event. For example, if a subscriber has two quota services, QuotaLocal and QuotaInternet, the interim formula for QuotaLocal can provide usage information to QuotaInternet.
- Account balance—Provides the balance in the account.

Related Topics

- Configuring an Interim Accounting Interval for Service Accounts on page 121
- Configuring VTA Services and Policies on page 88
- Service Variables on page 124

Service Variables

Use the variables described in this section to define formulas.

Current Service Variables

Use the variables described in this section to define a formula for the current service.

lastInterimTime

- Last interim time interval.
- Value—Number of seconds in the range 1–2147483647

sessionLength

- Length of the current session.
- Value—Number of seconds in the range 0–2147483647; value is 0 when the SRC VTA is calculating the interim time of start events. For other events, value is set by the PA_SESSION_TIME attribute.

maxUsageRate

- Maximum rate at which the subscriber can use network resources according to the formula described in Table 15 on page 126.
- Value—Integer in the range 0–9223372036854775807
- Guidelines—This formula corresponds to the usage formula for the same service as the interim formula.

The maxUsageRate variable is calculated for a service by means of the following values for the variables in the corresponding usage formula:

- upStreamBytes=PA_UPSTREAM_BANDWIDTH
  - downStreamBytes=PA_DOWNSTREAM_BANDWIDTH
  - interimTime=lastInterimTime
  - upStreamPackets=0
  - downStreamPackets=0

If you use the parameters upStreamPackets (PA_IN_PACKETS) and downStreamPackets (PA_OUT_PACKETS) in the usage formula and at the same time maxUsageRate in the interim interval formula, the maxUsageRate is not accurate, because the values for maximum upStreamPackets and downStreamPackets are unknown.

averageUsageRate

- Average rate at which the subscriber is consuming volume in units per second. The unit can be a value such as dollars, bytes, or packets. The type of unit depends on the value specified in the formula. Measurement begins when the service starts.
- Value—Integer in the range 0–9223372036854775807; the value is 0 when the SRC VTA is calculating the interim time of start events.

For other events, the value is the usage formula divided by PA_SESSION_TIME. The usage formula is calculated from PA_IN_PACKETS, PA_OUT_PACKETS, PA_OUT_OCTETS, PA_IN_OCTETS, and PA_SESSION_TIME.

latestUsageRate

- Rate of service usage since the last usage report.
- Value—Integer in the range 0–9223372036854775807; the value is 0 when the SRC VTA is calculating the interim time of start events.

The value is calculated by using the result of the usage formula divided by the length of the service session since the previous usage report for the same service.

Other Service Variables

Use the variables described in this section to define a formula for another service.

System requirements to calculate service usage, in the form of the averageUsageRate and the sessionLength variables, can affect system performance. Using a longer interim interval means that there are fewer interim events to process, which requires fewer system resources.

averageUsageRate_<serviceName>

- Average rate at which the service is consuming volume in units per second. The unit can be a value such as dollars, bytes, or packets. The type of unit depends on the value specified in the formula. Measurement begins when the service starts.
- Value—Integer in the range 0–9223372036854775807; the value is 0 when the SRC VTA is calculating the interim time of start events.
- Guidelines—Service names can contain alphanumeric characters and dashes (–).

sessionLength_<serviceName>

- Length of a service session for the service.
- Value—Integer in the range 0–2147483647; the value is 0 when the SRC VTA is calculating the interim time of start events.
- Guidelines—Service names can contain alphanumeric characters and dashes (–).

Account Balance Variable

Use the variable described in this section to provide balance information from each of the subscriber's accounts.

balance_<accountName>

- Balance for the specified account before the new usage value is applied.
- Value—Integer in the range 0–9223372036854775807
- Example—balance_PeriodicQuota refers to the balance for the PeriodicQuota account.

Sample Formulas for Interim Accounting Interval

Table 15 on page 126 provides examples of formulas to dynamically adjust the interim accounting interval for a service.

Table 15: Examples of Interim Accounting Interval

Formula	Description
return 900	Accounting interval is fixed at 900 seconds (15 minutes).
return (<balance_Periodic> + <balance_Bought>) /<maxUsageRate>	Minimum time required for the subscriber to empty the periodic and bought accounts.
return <sessionLength> >= 60*15 ? (<balance_Periodic> + <balance_Bought>) /<averageUsageRate>/2 : (<balance_Periodic> + <balance_Bought>) /<maxUsageRate>	Half the time required for the subscriber to empty the accounts at the current average rate, or the minimum time if the session is shorter than 15 minutes. Because the average rate may not be representative early in the session, check when the account is half empty.

Related Topics

- Adjusting the Interim Accounting Interval for a Service on page 123
- Defining a Formula for Determining Network Resource Usage That the SRC VTA Evaluates on page 119

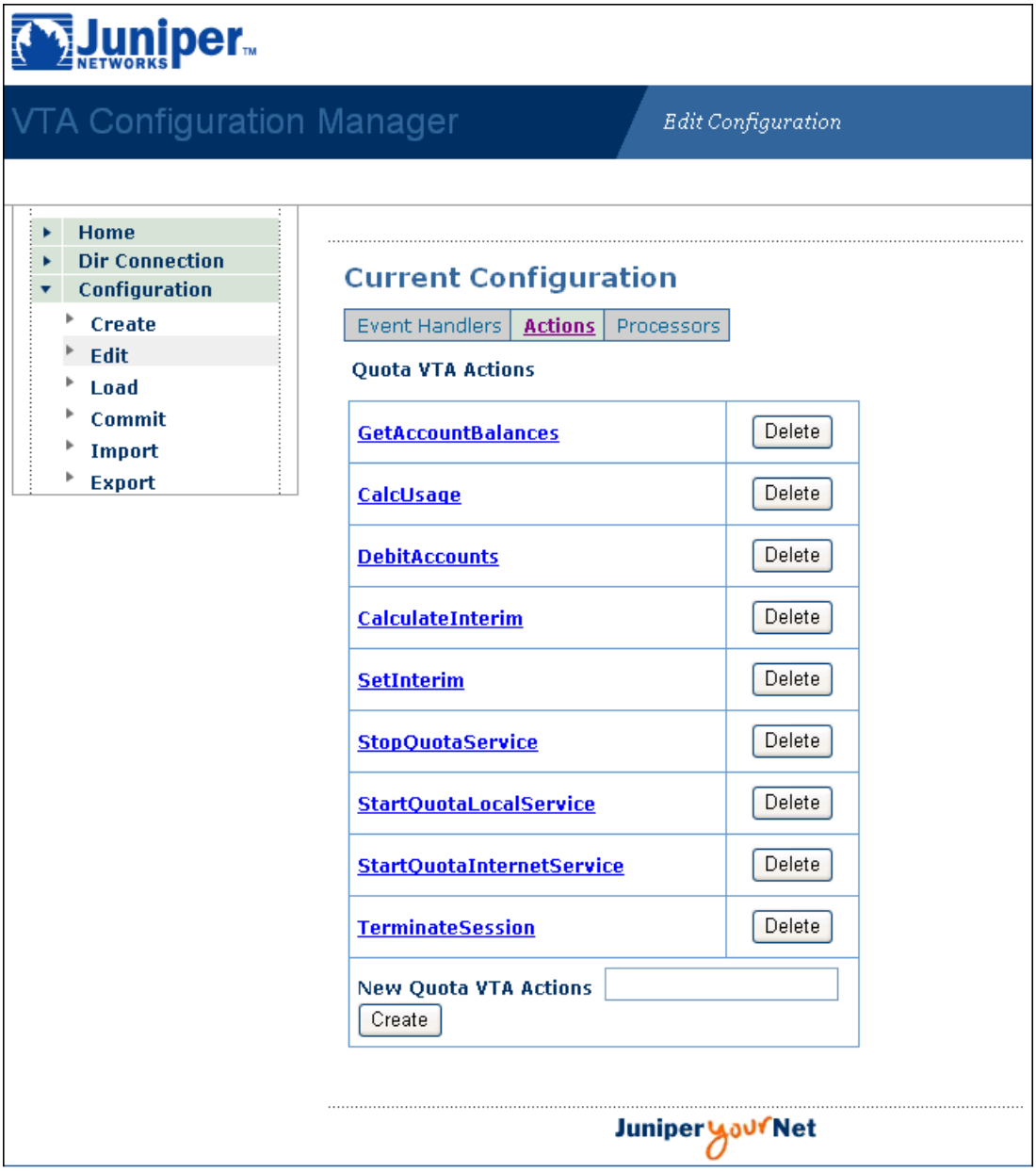
Configuring Actions for the Database Engine Processor

You can configure actions that the database engine processor performs on events. For example, you can set up an action to calculate usage in a service-tracking event by using the usage metric that you configured for a service.

To configure actions for the database engine processors:

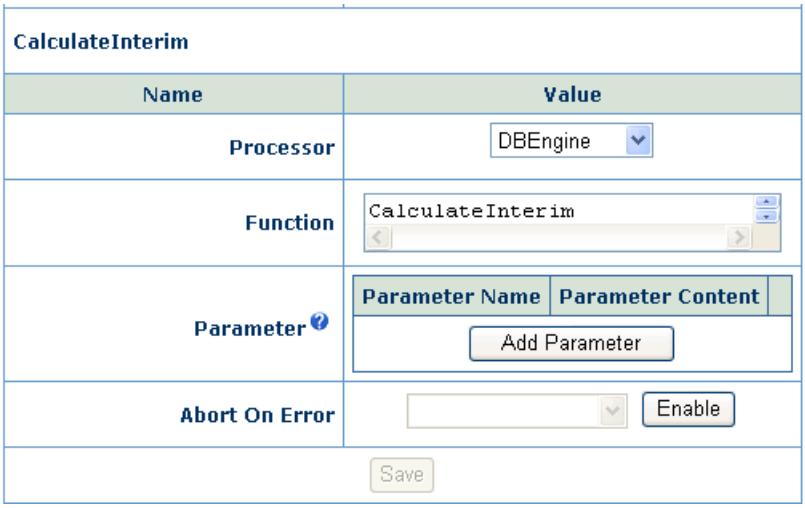
1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, select **Actions**.

   A list of actions appears. For example:



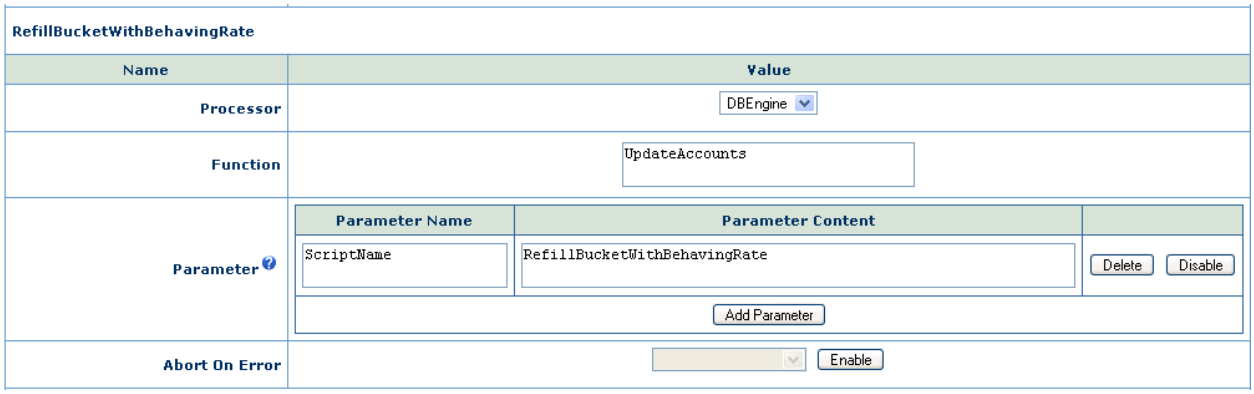
3. To add an action, enter a name for the action in the New Quota VTA Actions field, and click **Create**.

   The action configuration screen appears.



4. Select DBEngine in the Processor field, and click **Save**. (If DBEngine does not appear in the drop-down list, enable the database engine processor.)

   An expanded configuration screen for the action appears.



5. Edit the action fields.

   See “Action Fields for the Database Engine Processor” on page 129.

6. If you are finished configuring the SRC VTA, save the configuration to a directory or local file.

   See “Committing a VTA Configuration to a Directory” on page 175.

Related Topics

- Database Engine Processor Fields on page 122
- Configuring a Usage Metric for Service Accounts on page 117
- Configuring VTA Actions to Run Scripts on page 150
- Configuring Actions for the SAE Proxy Processor on page 137

Action Fields for the Database Engine Processor

In VTA Configuration Manager, you can edit the following fields in the Quota VTA Actions screen.

Processor

- Processor for which you are configuring the action.
- Value—DBengine
- Default—No value

Function

- Function calls that the database engine processor invokes. These functions are variables that you can use to update database accounts.
- Value
  - CalculateInterim—Calculates the interim interval in the service-tracking event by using the interim interval configured for the service. It has no parameters. It adds the following attribute to the event after the function is executed:
    - interimInterval—Interim interval of the service
  - CalculateUsage—Calculates usage in the service-tracking event by using the usage metric configured for the service. It has no parameters. It adds the following attributes to the event after the function is executed:
    - currentUsage—Usage since the previous usage report
    - sessionSinceLastReport—Session length since the previous usage report
  - GetAccounts—Gets account data for the corresponding subscriber for the event. Subsequent event handlers of the event can use the retrieved data. It has no parameters. It adds the following attributes to the event after the function is executed:
    - balance_<accountName>—Balance for the account.
    - lastUpdateTime_<accountName>—Last update time in milliseconds since January 1, 1970 UTC for the account.
    - status_<accountName>—Status of the account.
  - TerminateSession—Closes active VTA sessions that have a status of Start or Interim. It does not stop the corresponding services in the SAE. You can use this function to stop a service at the end of a billing period. Usage data collected after the VTA session is stopped is stored in new VTA session records.
  - UpdateAccounts—Runs an account update script that changes the account balances of the corresponding subscriber for the event. It adds the following attributes to the event after the function is executed:
    - balance_<accountName>—Balance for the account after it is updated.

- lastUpdateTime_<accountName>—Last update time in milliseconds since January 1, 1970 UTC for the account after it is updated.
- status_<accountName>—Status of the account after it is updated.

- Default—No value

Parameter Name

- Parameters to pass to the function.
- Value
  - Event attribute name; the event attribute name is replaced by the attribute's value wherever it appears.
  - For the UpdateAccounts function, you can configure the following parameter:
    - scriptName—Name of the account update script defined in the database engine processor
- Example—ScriptName

Parameter Content

- Values for the parameters.
- Value
  - For the scriptName parameter—Name of the account update script defined in the database engine processor
- Example—DebitQuotaUsage

Abort On Error

- Disables or enables and sets the processing in response to an error.
- Value
  - Break—Stop processing the current event.
  - Continue—Continue with the next action, if any, in the same event handler.
  - Next Event Handler—Continue with the next event handler (if any).
- Default—No value

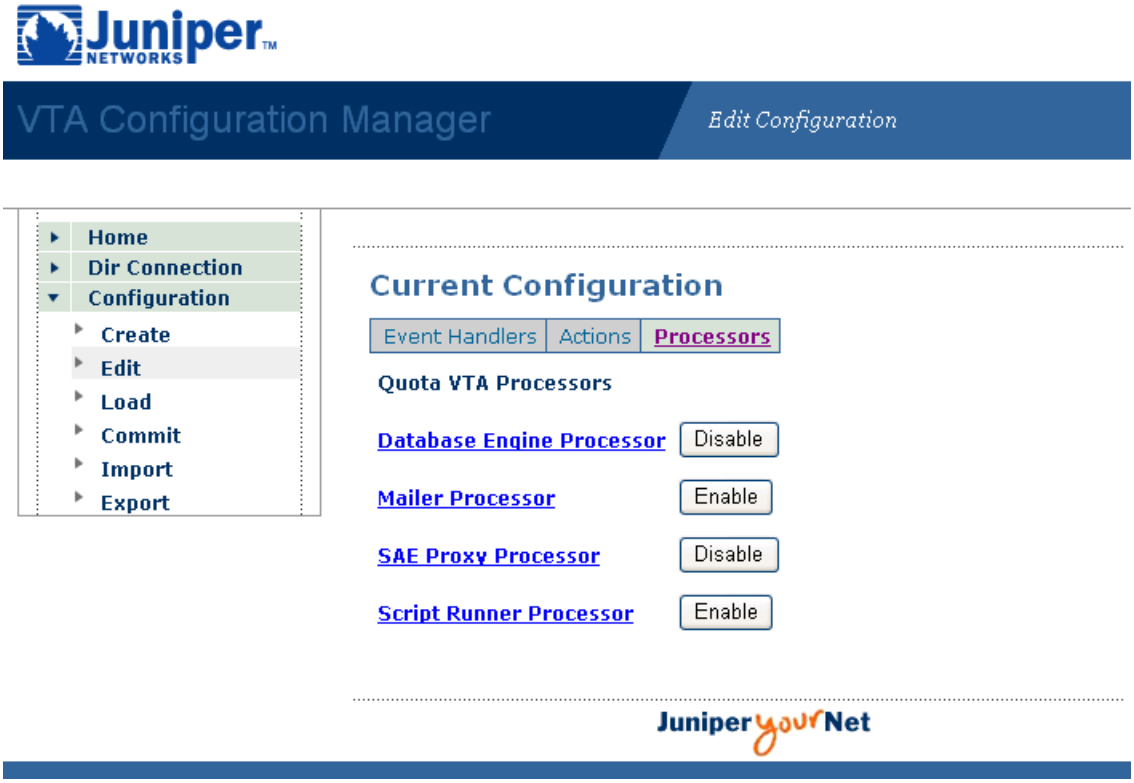
Setting Up the SRC VTA to Send E-Mail Notifications

Use the mailer processor to specify the SMTP server to use for e-mail messages that the SRC VTA sends to subscribers.

To configure the mailer processor:

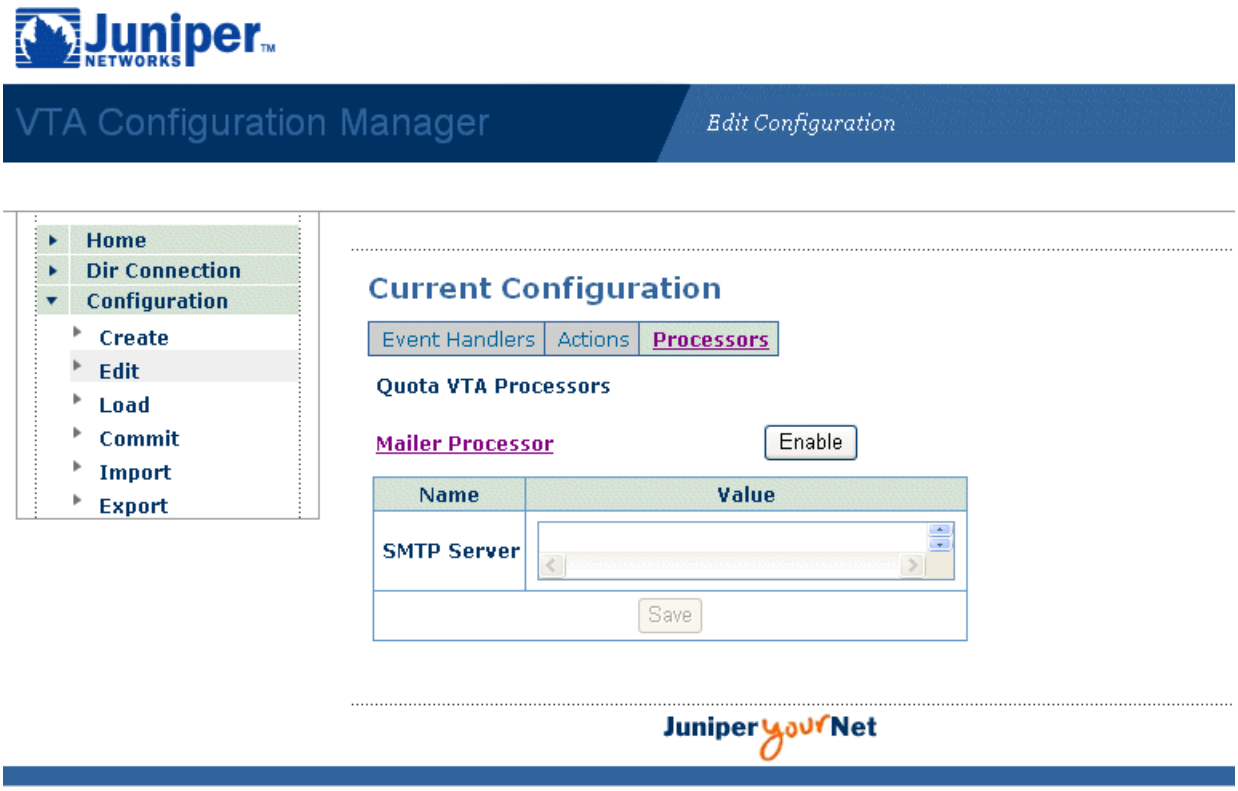
1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, select **Processors**.

   A list of processors appears.



3. Click **Enable** to enable the processor and display the configuration page for the mailer processor, or click **Mailer Processor**.

   The Mailer Processor screen appears.



4. Fill in the SMTP server box, and click **Save**.
5. Configure an action for the mailer processor. See “Configuring the SRC VTA to Send E-Mail Notifications” on page 132.

Related Topics

- Configuring the SAE Proxy Processor on page 136
- Configuring the SRC VTA to Manage Subscriber Accounts on page 115
- Database Engine Processor Fields on page 122

Mail Processor Field

SMTP Server

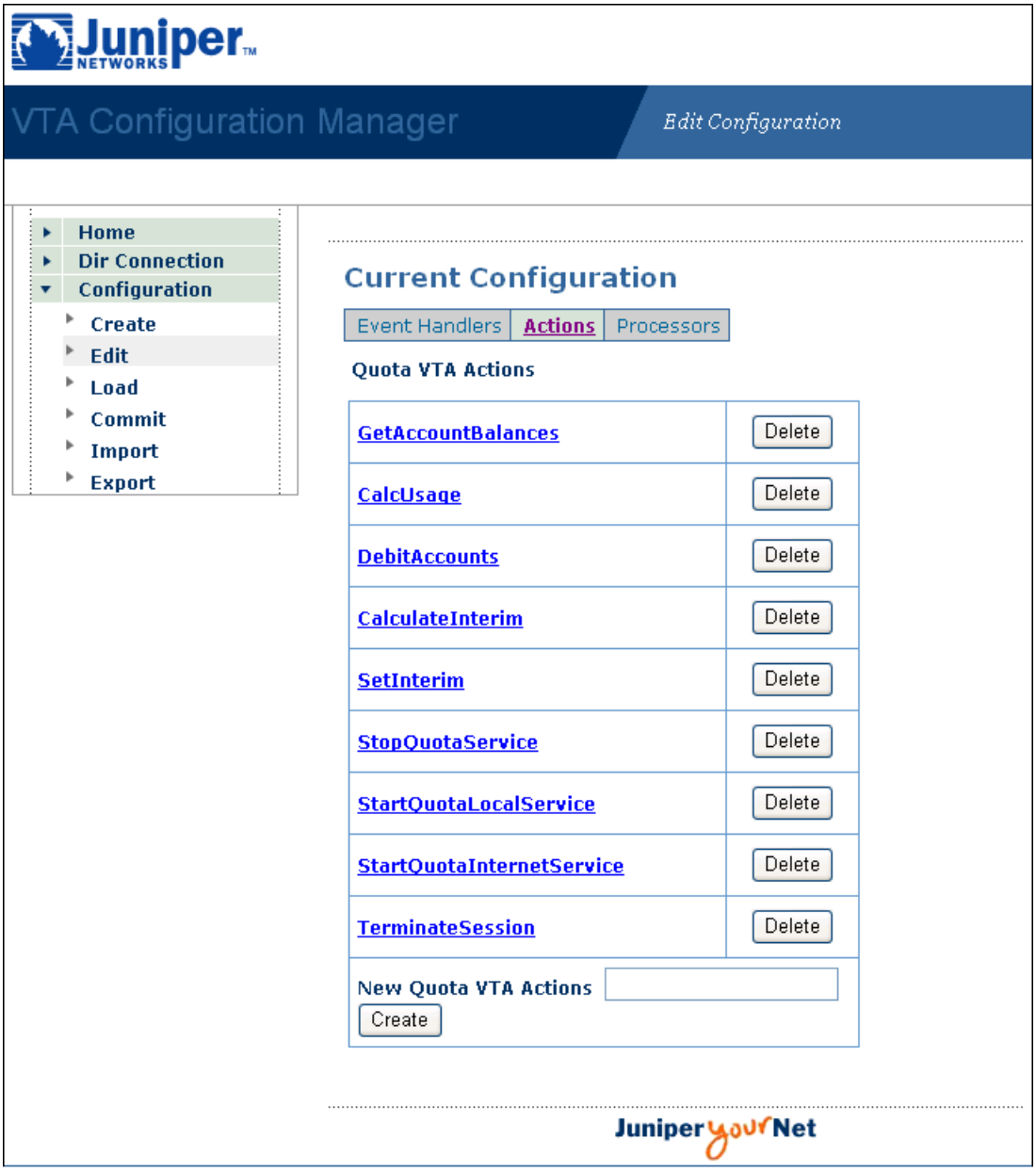
- SMTP server for outgoing e-mail.
- Value—Hostname or IP address of the SMTP server
- Default—No value

Configuring the SRC VTA to Send E-Mail Notifications

You can configure actions for the mailer processor that cause the SRC VTA to send e-mail notifications. To configure actions to the mailer processor:

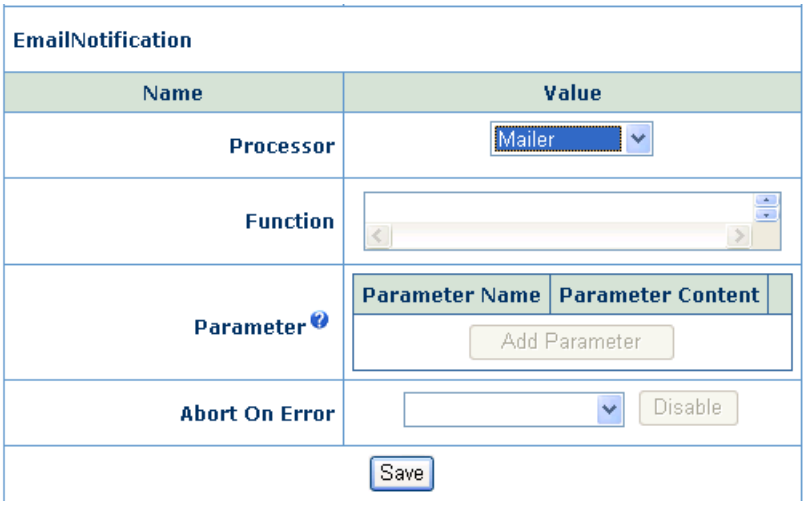
1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, select **Actions**.

   A list of actions appears. For example:



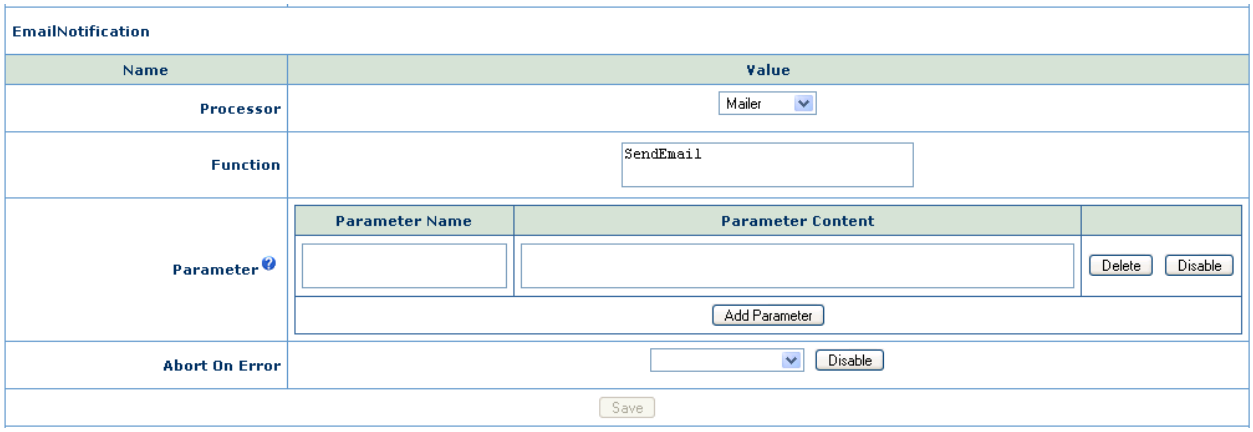
3. To add an action, enter a name for the action in the New Quota VTA Actions field, and click **Create**.

   The action configuration screen appears.



4. Select **Mailer** in the Processor field, and click **Save**. (If Mailer does not appear in the drop-down list, enable the mailer processor.)

   An expanded configuration area for the action appears.



5. Edit the action fields.

Related Topics

- Setting Up the SRC VTA to Send E-Mail Notifications on page 130
- Configuring Actions for the Database Engine Processor on page 126
- Installing VTA Configuration Manager on page 104
- Example of a Bucket VTA on page 193

E-Mail Notification Fields

Processor

- Processor for which you are configuring the action.
- Value—Mailer (If Mailer does not appear in the drop-down list, enable the mailer processor.)
- Default—No value

Function

- Function calls that the mail processor invokes.
- Value
  - SendEmail—Sends e-mail messages. It provides no output.
- Default—No value

Parameter Name

- Parameters to pass to the processor.
- Value
  - " Recipient" —Address of the e-mail recipient
  - " From" —Address of the e-mail sender
  - " Subject" —Subject of the e-mail
  - " Text" —Text of the e-mail
- Default—No value

Parameter Content

- Values for the parameters.
- Value
  - Recipient—Address of the e-mail recipient
  - From—Address of the e-mail sender
  - Subject—Subject of the e-mail
  - Text—Text of the e-mail
- Default—No value

Abort On Error

- Disables or enables and sets the processing in response to an error.
- Value
  - Break—Stop processing the current event.
  - Continue—Continue with the next action, if any, in the same event handler.

- Next Event Handler—Continue with the next event handler (if any).

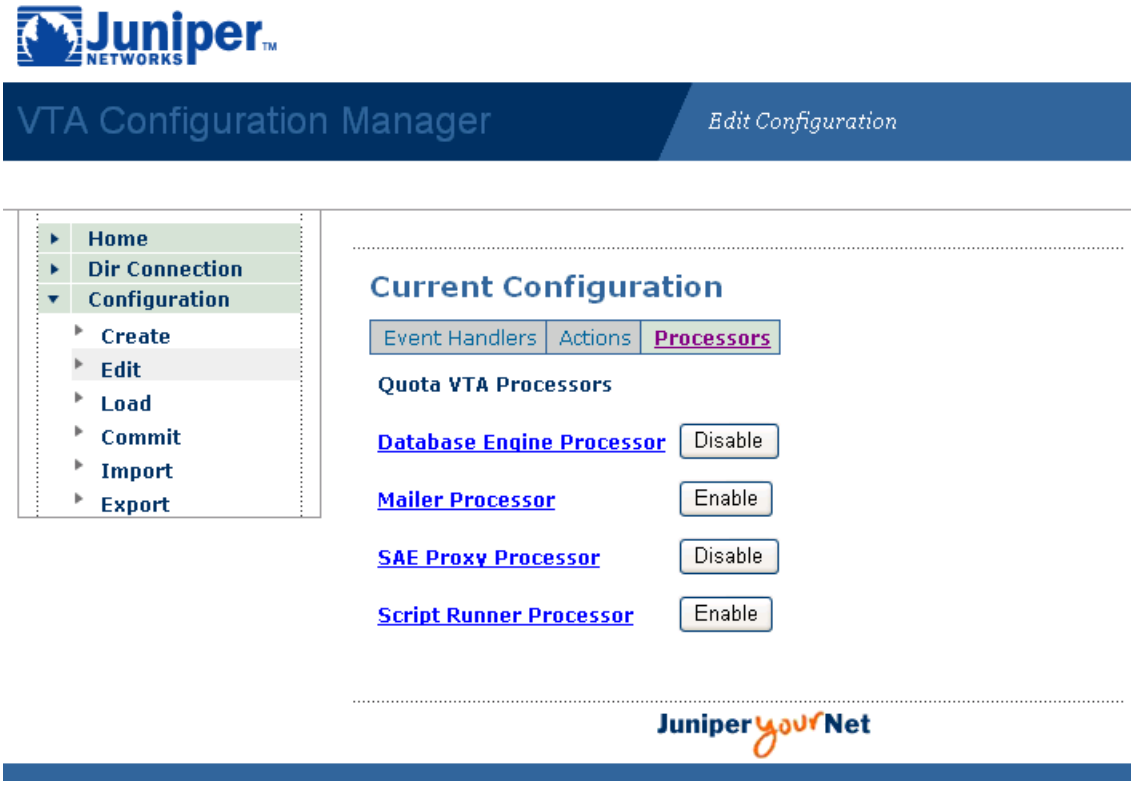
- Default—No value

Configuring the SAE Proxy Processor

To configure the SAE proxy processor:

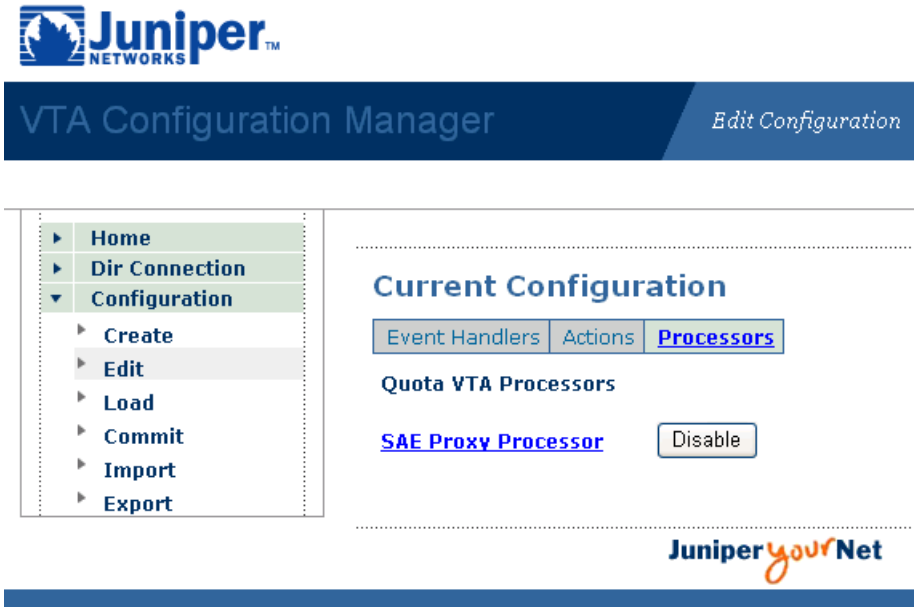
1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, select **Processors**.

   A list of processors appears.



3. Click **Enable** next to SAE Proxy Processor to enable the processor.

   The SAE Proxy Processor area appears.



4. There are no fields to configure for the SAE proxy processor, but you must configure actions for the processor. See “Configuring Actions for the SAE Proxy Processor” on page 137.

Related Topics

- Configuring VTA Actions to Run Scripts on page 150
- Example of a Bucket VTA on page 193

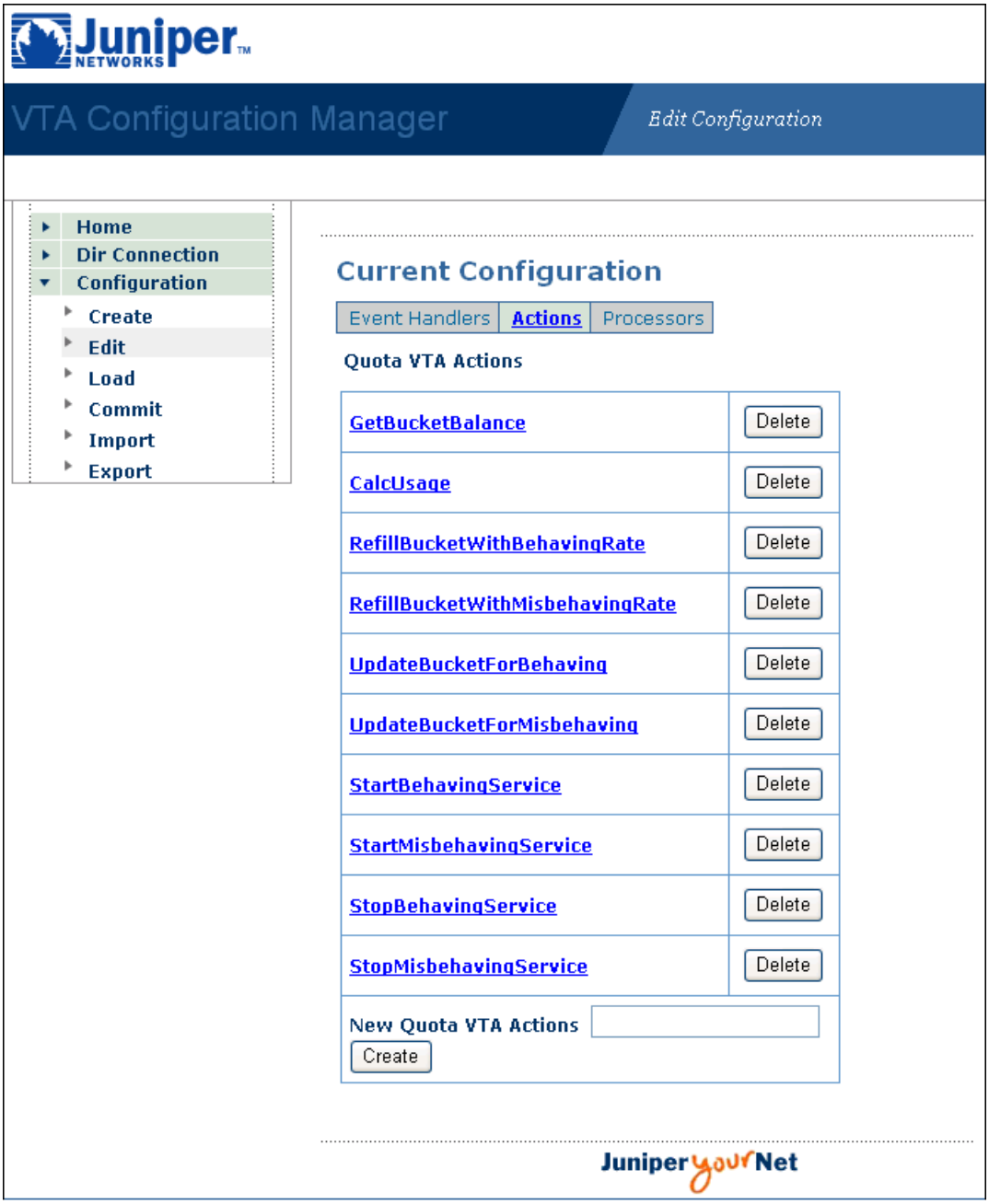
Configuring Actions for the SAE Proxy Processor

You can configure actions that the SAE proxy processor performs on events. For example, you can set up an action to start or stop a service.

To configure actions for the SAE proxy processor:

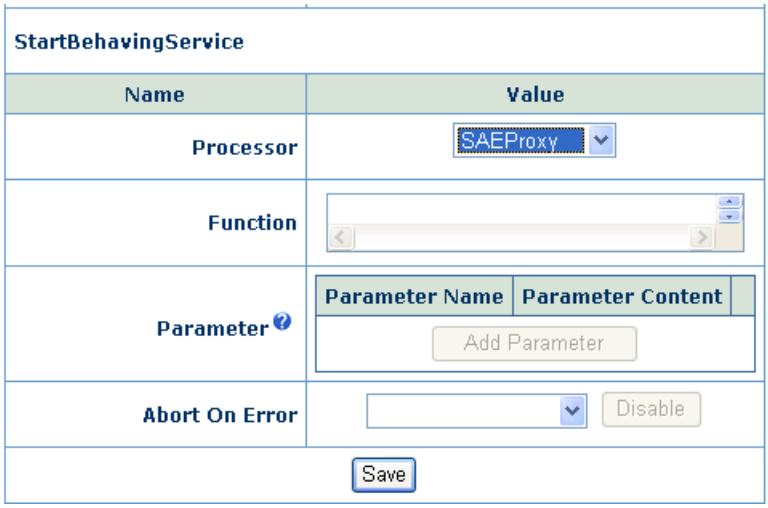
1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, select Actions.

   A list of actions appears. For example:



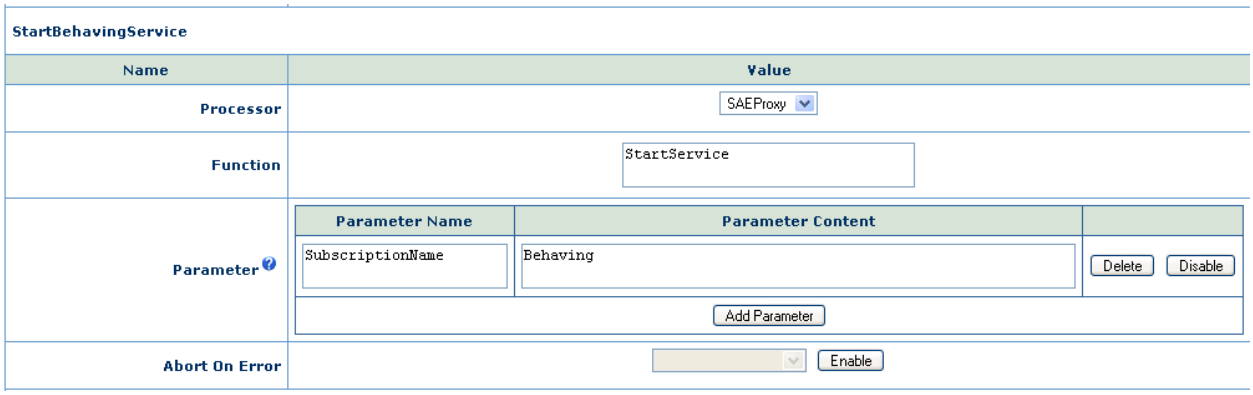
3. To add an action, enter a name for the action in the New Quota VTA Actions field, and click **Create**.

   The action configuration area appears.



4. Select **SAEProxy** in the Processor field, and click **Save**. (If SAEProxy does not appear in the drop-down list, enable the SAE proxy processor.)

   An expanded configuration area for the action appears.



5. Fill in the fields described in this section.

Related Topics

- Configuring the SAE Proxy Processor on page 136
- Configuring the SAE to Send Tracking Events to the SRC VTA on page 92
- Example of a Bucket VTA on page 193

StartBehavingService Fields

Processor

- Processor for which you are configuring the action.
- Value—SAEproxy (If SAEProxy does not appear in the drop-down list, enable the SAE proxy processor.)
- Default—No value

Function

- Function calls that the SAE proxy processor invokes.
- Value
  - SetInterimInterval—Sets the interim interval for the service by using the interimInterval attribute in the event. Before this function is called, the CalculateInterim function of the database engine processor must be called.
  - SetServiceTimeout—Sets the service session timeout for the subscription.
  - SetUserTimeout—Sets the subscriber session timeout of the current subscriber to the corresponding VTA event.
  - StartService—Starts the specified subscription to the service with the specified substitutions.
  - StopService—Stops the specified subscription to the specified service.
- Default—No value

Parameter Name

- Parameters to pass to the processor.
- Value
  - For the SetServiceTimeout function:
    - SubscriptionName—Name of the subscription. Default subscriptions have the same name as the service. This parameter is optional when a service-tracking event is being processed. If this parameter is omitted, the current service session is stopped.
    - SessionName—Name of the service session. If the subscriptionName is omitted, this parameter is ignored. If this parameter is omitted, the default service session is used.
    - SessionTimeout—Length of the service session timeout. When the session timeout expires, the service session is stopped.
    - CurrentSubscriberOnly—Specifies whether the function is applied to the current subscriber only or to all subscribers who have the same subscriber ID.
  - For the SetUserTimeout function:
    - SessionTimeout—Length of timeout. When the session timeout expires, the subscriber is logged out.

- CurrentSubscriberOnly—Specifies whether the function is applied to the current subscriber only or to all subscribers who have the same subscriber ID.

- For the StartService function:
  - SubscriptionName—Name of the subscription. Default subscriptions have the same name as the service.
  - SessionTimeout—Length of the service session timeout. If this parameter is omitted, the default is no timeout.
  - SessionName—Name of the service session. If this parameter is omitted, the default service session is used.
  - CurrentSubscriberOnly—Specifies whether the function is applied to the current subscriber only or to all subscribers who have the same subscriber ID.
  - Substitution.<substitutionName>—Name of a substitution to use when starting the service. If this parameter is omitted, the service is started without substitutions.
  - Persistent—Specifies whether a service session is persistent. If you use the SRC VTA to activate a service with the persistent option, this service is subsequently activated by the SAE every time the subscriber connects to the network. This option provides efficiency because the SRC VTA does not need to make a decision to activate the service on subsequent logins and because applications such as deep packet inspection (DPI) can more efficiently activate a group of services at login.

- For the StopService function:
  - SubscriptionName—Name of the subscription. Default subscriptions have the same name as the service. If this parameter is omitted, the current service session is stopped.
  - SessionName—Name of the service session. If the subscriptionName parameter is omitted, this parameter is ignored. If this parameter is omitted, the default service session is used.
  - Reason—Reason for the termination. When the service is stopped, the termination cause is sent to the billing system so it can differentiate between service stops. If this parameter is omitted, no termination cause is provided to the billing system.
  - CurrentSubscriberOnly—Specifies whether the function is applied to the current subscriber only or to all subscribers who have the same subscriber ID.
  - Persistent—Specifies whether a service session is persistent. If you use the SRC VTA to activate a service with the persistent option, this service is subsequently activated by the SAE every time the subscriber connects to the network. This option provides efficiency because the SRC VTA does not need to make a decision

to activate the service on subsequent logins and because applications such as DPI can more efficiently activate a group of services at login.

- Example—Quota

Parameter Content

- Values for the parameters.
- Value
  - For the subscriptionName parameter:
    - Name of the subscription in the format <serviceName>%<subscriptionId>.
  - For the sessionName parameter:
    - Name of the service session.
  - For the sessionTimeout parameter:
    - Length of the service session timeout in seconds. If the session timeout is set to 0, the service session is stopped immediately.
  - For the substitution parameter:
    - Name of a substitution to use.
  - For the reason parameter:
    - Integer that identifies the termination cause. Possible values are defined in RFC 2866—RADIUS Accounting (June 2000).
  - For the CurrentSubscriberOnly parameter:
    - Set to true to apply to current subscriber only. If you do not set this parameter, true is the default behavior.
    - Set to false to apply to all subscribers who have the same subscriber ID.
  - For the persistent parameter:
    - Set to true to cause the session to be persistent. We recommend that you set this value to true for service sessions that are started in the DPI environment.
    - Set to false to specify that the session is not persistent and the service is activated or deactivated for the current subscriber session only.
- Example—Quota

Abort On Error

- Disables or enables and sets the processing in response to an error.
- Value
  - Break—Stop processing the current event.

  - Continue—Continue with the next action, if any, in the same event handler.
  - Next Event Handler—Continue with the next event handler (if any).
- Default—No value

Configuring the SRC VTA to Run Scripts

The script runner processor can invoke external executable scripts or JavaScript programs. We recommend using JavaScript for better performance.

- External scripts are executable programs, such as shell scripts, that are available on the VTA's host. Each external script can perform a task and return a value. If the script returns a value, the value can be added to the current event as an event attribute.
- JavaScript programs are used to process attributes of a VTA event. For example, it can convert a VTA event attribute in a timestamp to a date string and add it to the event as a new attribute. The attribute can then be used for subsequent actions, such as sending an e-mail notification to the subscriber. The JavaScript program can refer to any attributes of the event being processed, and it must return a value.

To configure the script runner processor:

1. Configure a JavaScript program or an external script. See:
   - Configuring JavaScript Programs on page 143
   - Configuring External Scripts on page 147
2. Configure an action for the script runner processor. See:
   - Configuring VTA Actions to Run Scripts on page 150

Related Topics

- Configuring Events on page 154
- JavaScript Fields on page 146
- External Script Fields on page 149

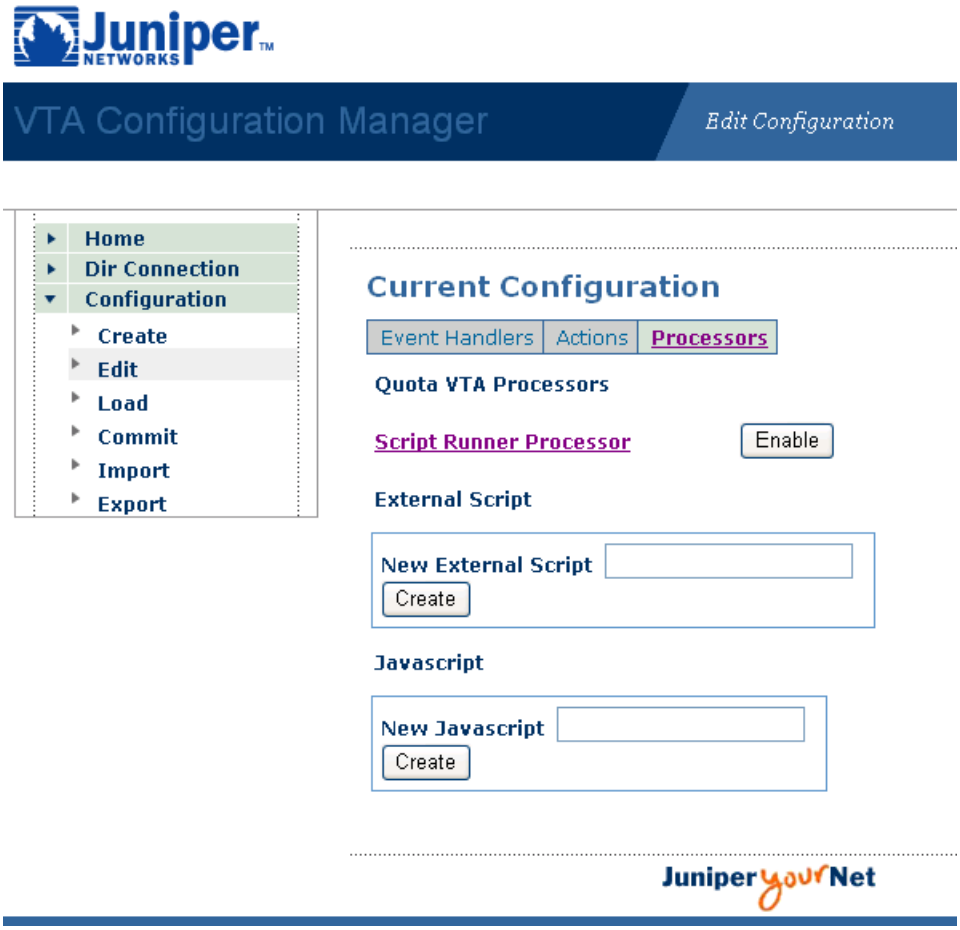
Configuring JavaScript Programs

To configure a JavaScript program:

1. In VTA Configuration Manager, select **Edit** or **Create** in the navigation pane.

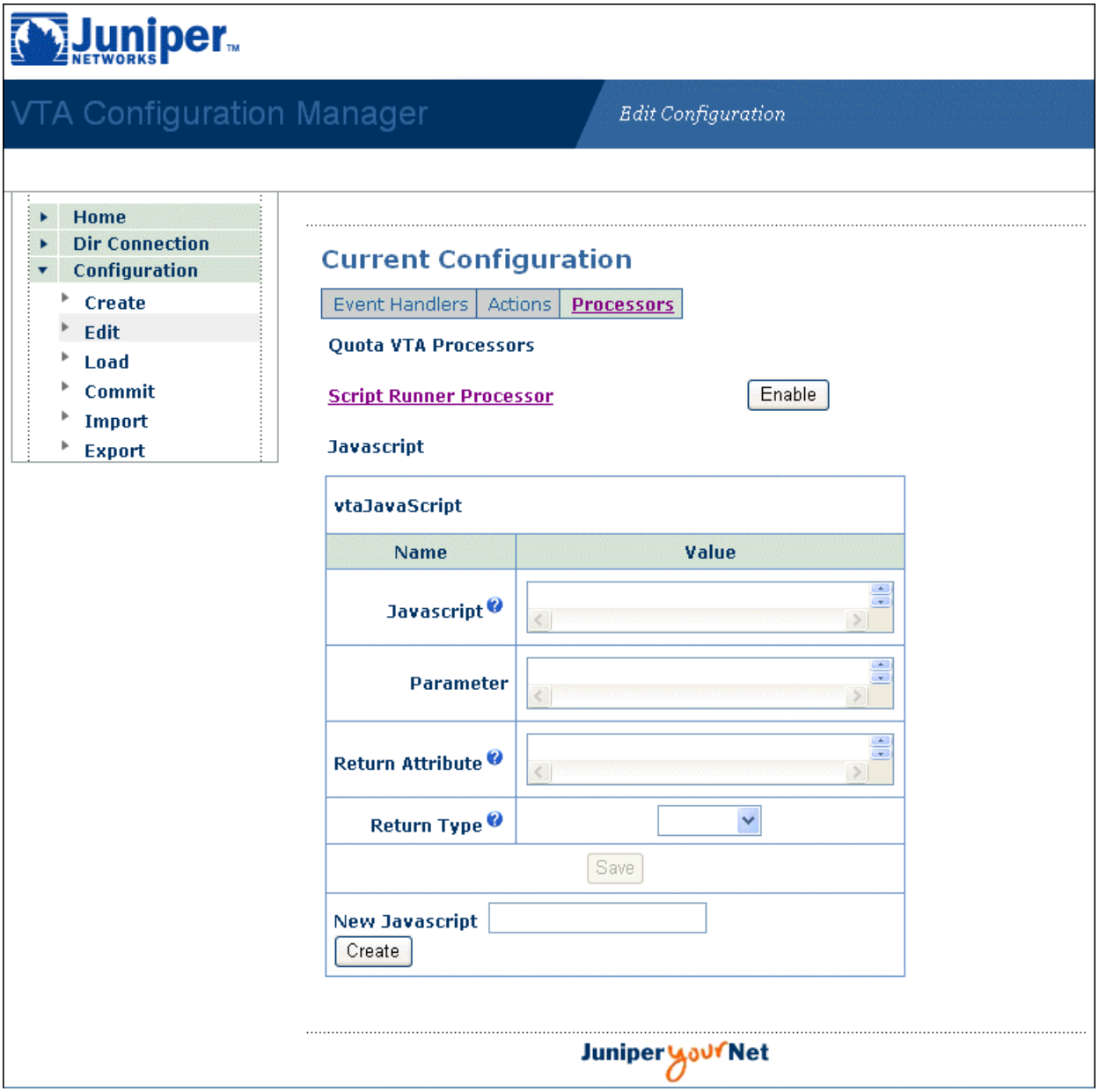
   The Current Configuration window appears.
2. Select **Processors**.
3. Select **Script Runner Processor**.

   The Script Runner Processor configuration appears.



4. If the script runner processor is disabled, click **Enable** to enable it.
5. Enter the name of the script in the New Javascript box, and click **Create**.

   The JavaScript configuration appears.



6. Edit the fields for the JavaScript.

   See “JavaScript Fields” on page 146 .

7. Configure a VTA action for the script.

   See “Configuring VTA Actions to Run Scripts” on page 150 .

Related Topics

- Using JavaScript Programs in VTA Configurations on page 83
- Configuring the SRC VTA to Run Scripts on page 143

JavaScript Fields

This section describes the fields used to configure JavaScript programs.

New Javascript

- Name of the JavaScript program.
- Value—Text string
- Default—No value

Javascript

- A function body in JavaScript.
- Value—Path name. To refer to the event attributes being processed, include the attribute name delimited by angle brackets (<and>). The JavaScript can verify whether the event has the referred attribute. If a referred attribute does not exist in the event, the attribute's value is null. The JavaScript must return a value.
- Default—No value

Parameter

- Parameters required by the script. When invoking the script from an action, the action must supply values for these parameters.
- Value—Comma-separated list of parameters
- Default—No value
- Example—fullLoginName

Return Attribute

- Attribute that provides the return value of the script as a valid Java identifier that subsequent actions and event handlers can refer to.
- Value—Attribute name
- Guidelines—JavaScripts must return an attribute value. The name of a return attribute cannot start with an underscore (_), because these event attributes are reserved for internal use.
- Default—No value

Return Type

- Java type of the return attribute.
- Value
  - Integer
  - Long

- Float
- Double
- String
- Boolean

Configuring External Scripts

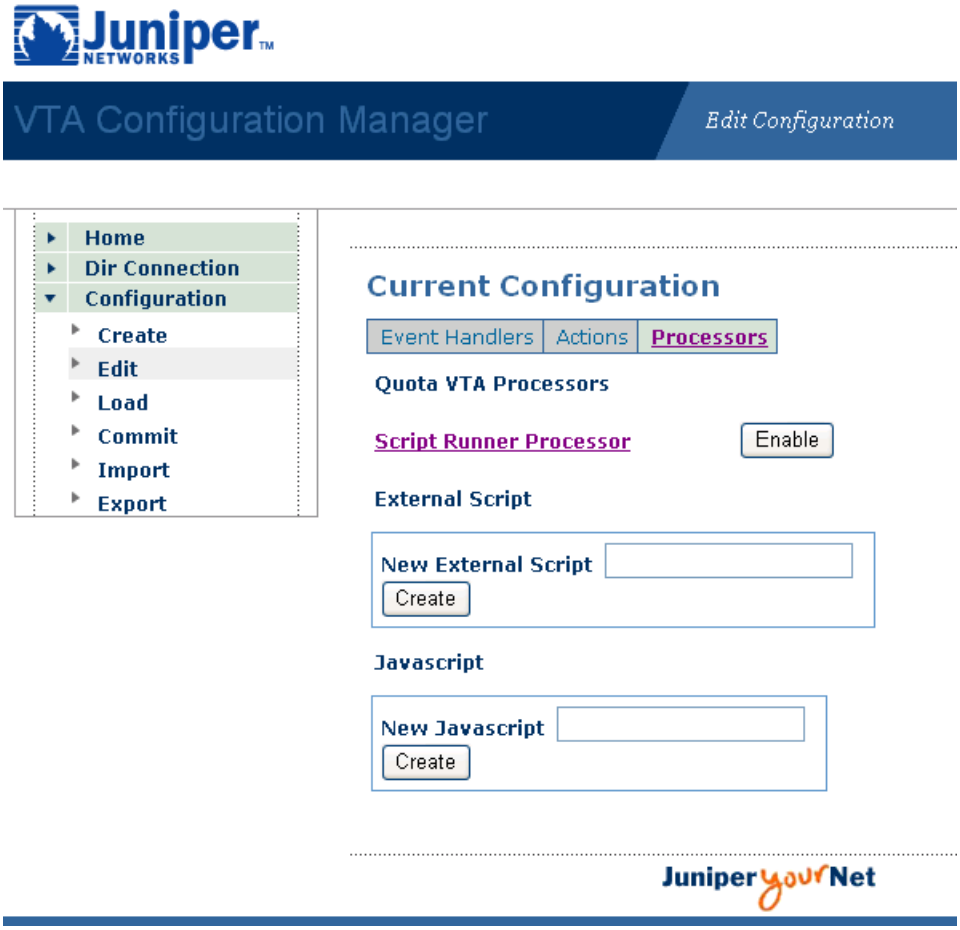
To configure an external script:

1. In VTA Configuration Manager, select **Edit** or **Create** in the navigation pane.

   The Current Configuration window appears.

2. Select **Processors**.
3. Select **Script Runner Processor**.

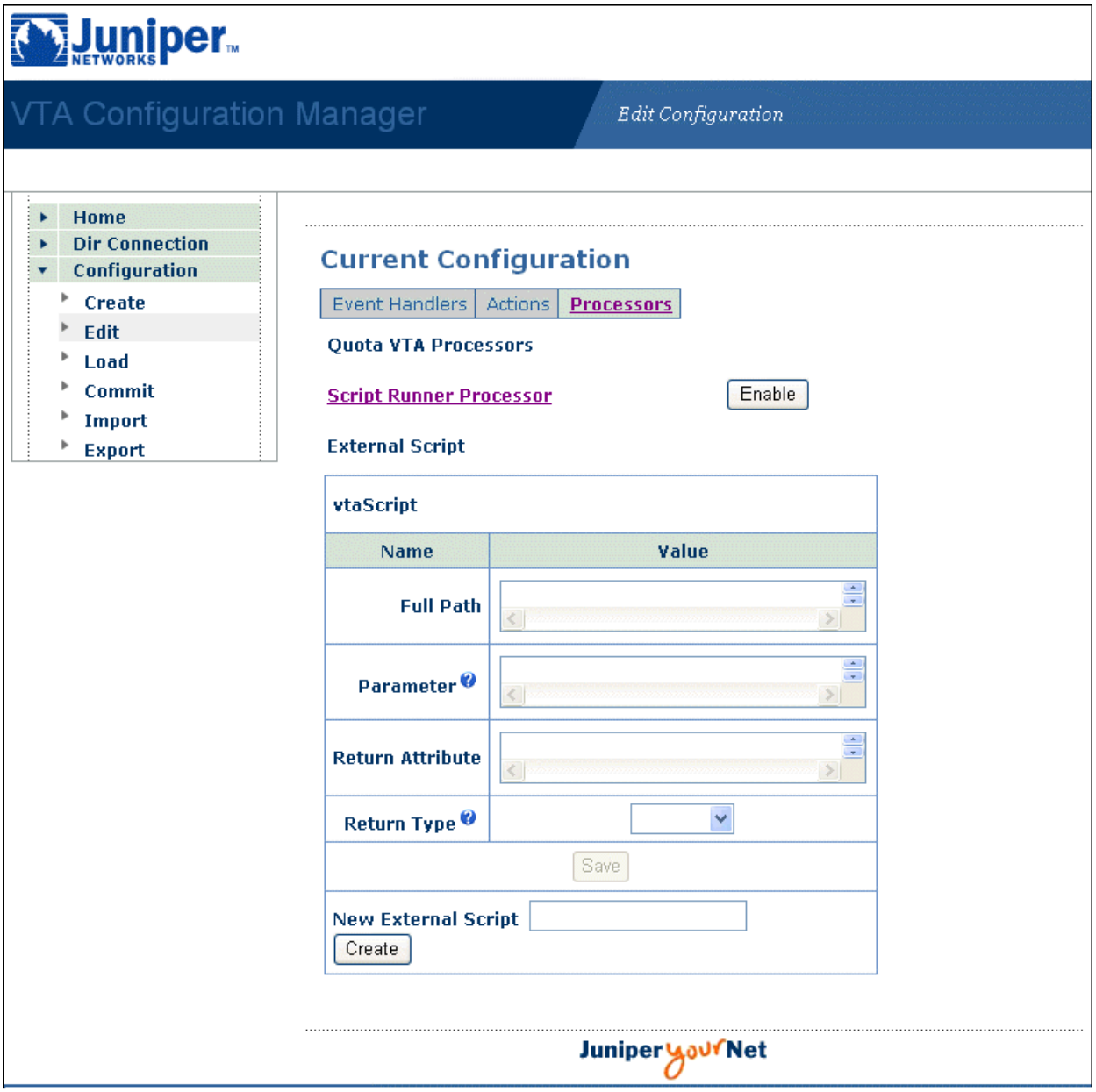
   The Script Runner Processor configuration appears.



4. If the Script Runner processor is disabled, click **Enable** to enable it.

5. Enter the name of the script in the New External Script box, and click **Create**.

   The External Script configuration appears.



6. Edit the fields for the external script.

   See “External Script Fields” on page 149 .

7. Configure a VTA action for the script.

   See “Configuring VTA Actions to Run Scripts” on page 150 .

Related Topics

- Configuring Scripts That Update Accounts on page 114
- Configuring the SRC VTA to Run Scripts on page 143

External Script Fields

This section describes the fields used to configure external scripts.

New External Script

- Name of the external script.
- Value—Text string
- Default—No value

Full Path

- Full pathname of the external script.
- Value—Pathname
- Default—No value

Parameter

- Parameters required by the script. When an action invokes the script, the action must supply values for these parameters.
- Value—Comma-separated list of parameters
- Guidelines—External scripts require parameters.
- Default—No value
- Example—fullLoginName

Return Attribute

- Attribute that provides the return value of the script as a valid Java identifier that subsequent actions and event handlers can refer to. If this attribute is not set, the return value is not used by the event. The external script returns the value by printing to standard output.
- Value—Attribute
- Guidelines—The name of a return attribute cannot start with an underscore (_), because these event attributes are reserved for internal use.
- Default—No value

Return Type

- Java type of the return attribute.
- Value
  - Integer
  - Long
  - Float
  - Double
  - String
  - Boolean
- Default—No value

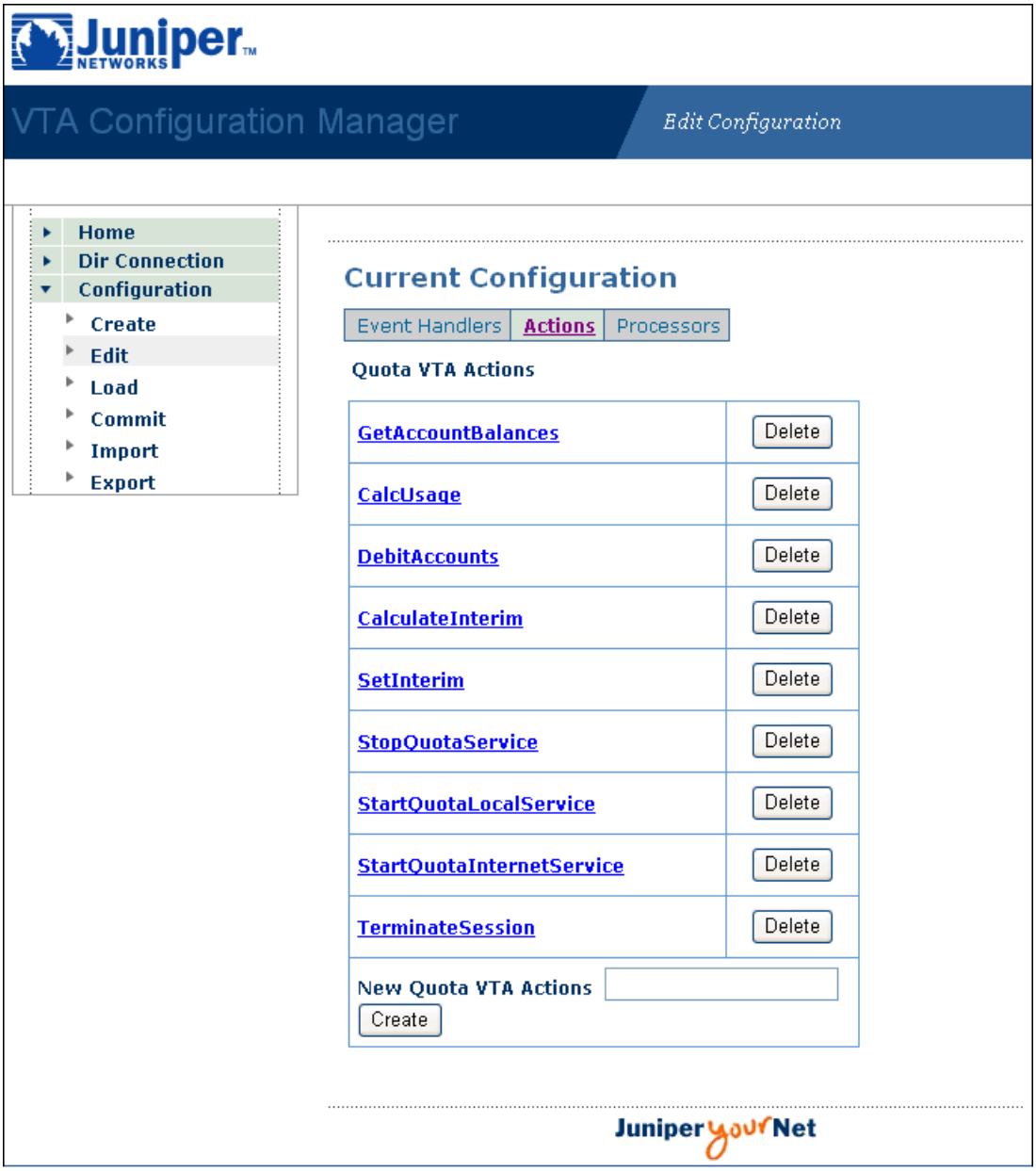
Configuring VTA Actions to Run Scripts

You can configure actions that the script runner processor performs on events. For example, you can set up an action to run either an external script or a JavaScript program.

To configure actions for the script runner processor:

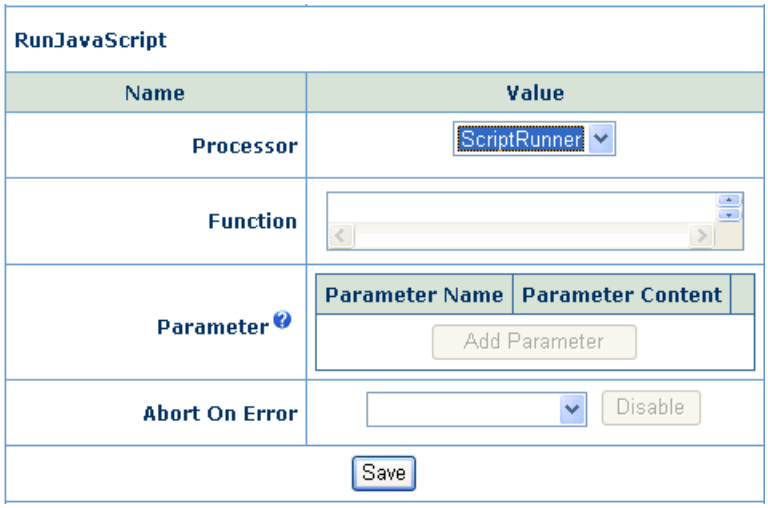
1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, click **Actions**.

   A list of actions appears. For example:



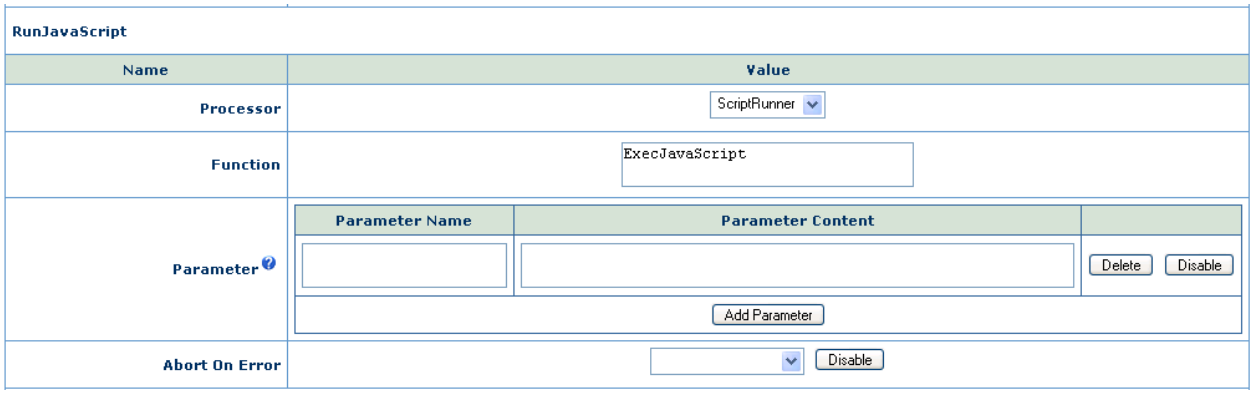
3. To add an action, enter a name for the action in the New Quota VTA Actions field, and click **Create**.

   The action configuration screen appears.



4. Select ScriptRunner in the Processor field, and click **Save**. (If ScriptRunner does not appear in the drop-down list, enable the script runner processor.)

   An expanded configuration screen for the action appears.



5. Fill in the fields described in this section.

Related Topics

- Configuring the SRC VTA to Run Scripts on page 143
- Configuring JavaScript Programs on page 143
- Configuring External Scripts on page 147

RunJavaScript Fields

Processor

- Processor for which you are configuring the action.
- Value—ScriptRunner (If ScriptRunner does not appear in the drop-down list, enable the script runner processor.)
- Default—No value

Function

- Function calls that the script runner processor invokes.
- Value
  - ExecExtScript—Runs an external script defined in the processor. It provides any defined ReturnAttribute attribute for the corresponding script as output by adding the attribute to the event.
  - ExecJavaScript—Runs a JavaScript defined in the processor. It provides any defined ReturnAttribute attribute for the corresponding script as output by adding the attribute to the event.
- Default—No value

Parameter Name

- Parameters to pass to the processor.
- Value
  - For external scripts:
    - ScriptName—Name of the script
    - Parameters that you declared in the script
  - For JavaScripts:
    - ScriptName—Name of the script

Parameter Content

- Parameter values to pass to the processor.
- Value
  - For ScriptName parameters—Name of the script
  - For other parameters—Values of parameters that you declared in the script

Abort On Error

- Disables or enables and sets the processing in response to an error.
- Value
  - Break—Stop processing the current event.
  - Continue—Continue with the next action, if any, in the same event handler.

    - Next Event Handler—Continue with the next event handler (if any).

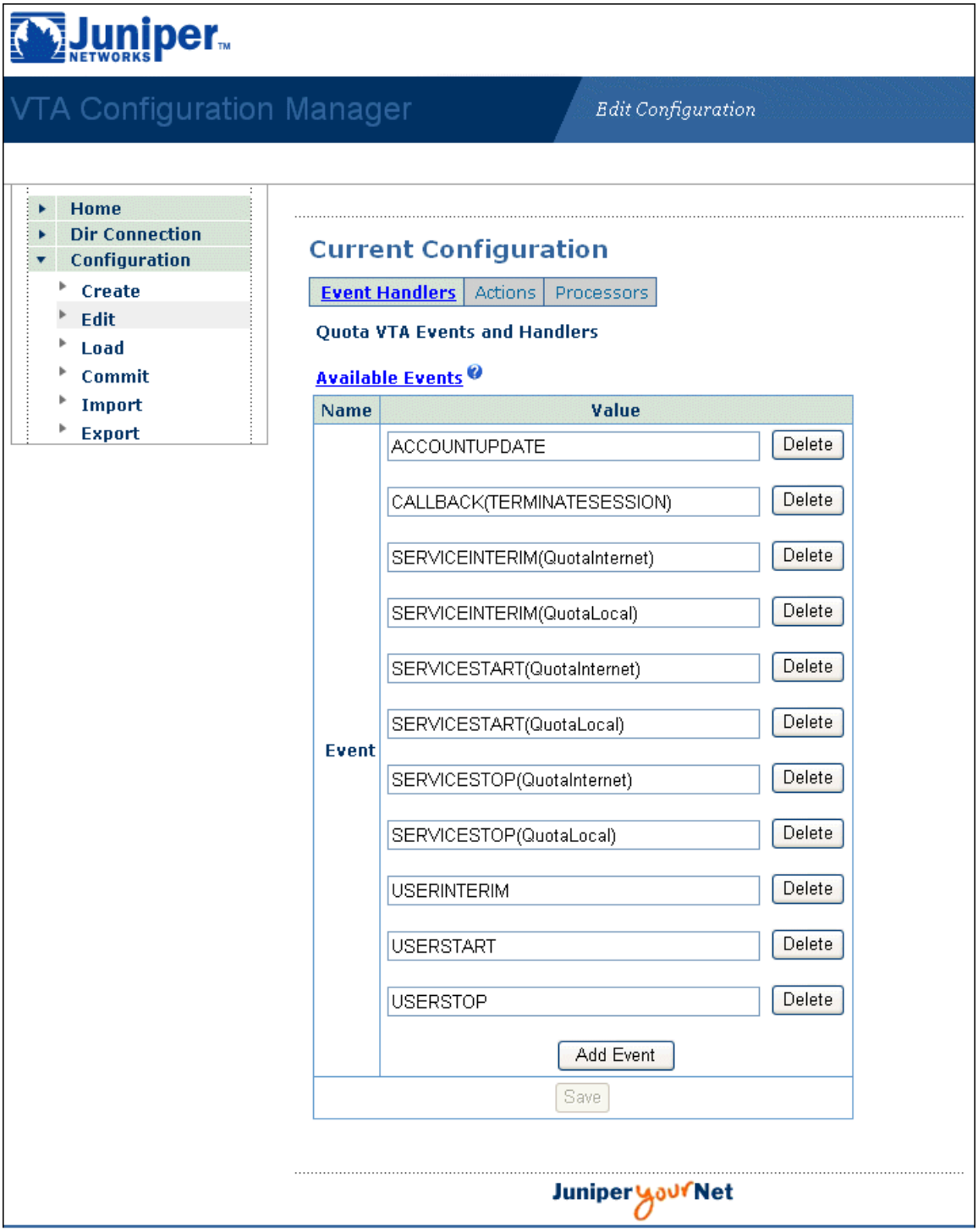
- Default—No value

Configuring Events

To configure events:

1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Select **Event Handlers**.

   A list of configured events appears. For example:



3. Click **Add Event**.

A blank field appears at the top of the event list.

4. Fill in the Value box, and then click **Save**.

Related Topics

- How the SRC VTA Works on page 68
- Configuring Event Handlers on page 156
- Event Handler Fields on page 160
- Example of a Bucket VTA on page 193

Available Events Field

Value

- Value of the event.
- Value
  - ACCOUNTUPDATE—Database update event
  - CALLBACK(<callId>)—External callback event for the specified call
  - SERVICEINTERIM(<serviceName>)—Service interim–tracking event for the specified service
  - SERVICESTART(<serviceName>)—Service start–tracking event for the specified service
  - SERVICESTOP(<serviceName>)—Service stop–tracking event for the specified service
  - USERINTERIM—User interim–tracking event
  - USERSTART—User start–tracking event
  - USERSTOP—User stop–tracking event
- Example—SERVICESTART(QuotaInternet), CALLBACK(TERMINATESESSION)

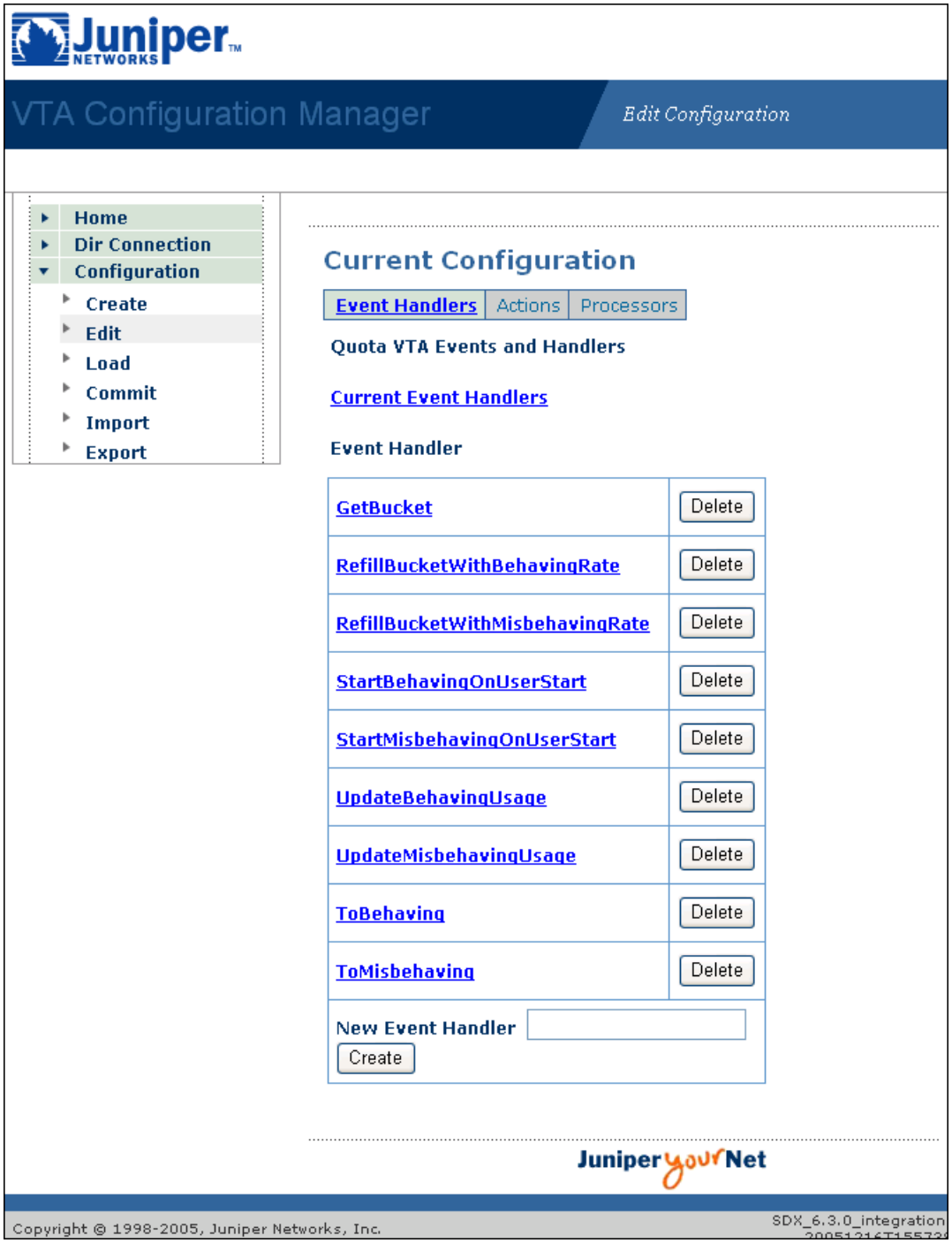
Configuring Event Handlers

Event handlers are defined by events, condition, priority, and actions. When an event is received, the corresponding event type and condition of the handler are evaluated based on the priority. When a condition is met, the corresponding actions are performed according to the event attributes. The action can update or add event attributes to the events for subsequent processing of the same event. An action invokes a function provided by a processor and must provide the parameters required by that function. After an event handler has completed its processing, event processing continues with the next applicable event handler.

To configure event handlers:

1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Select **Current Event Handlers**.

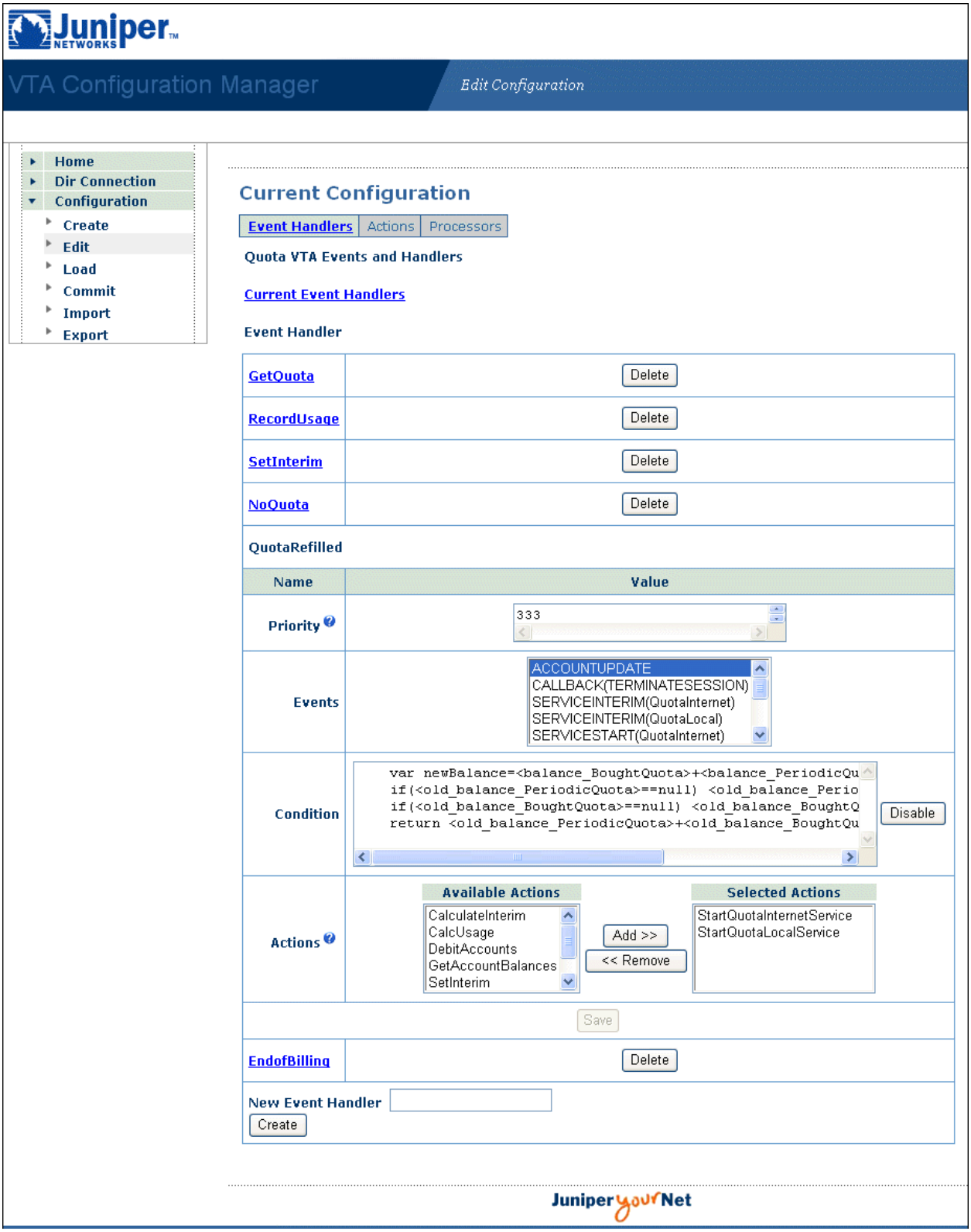
   A list of configured event handlers appears. For example:



You can create, delete, or modify event handlers.

- To add an event handler, enter a name for the action in the New Event Handler box, and click **Create**.
- To delete an event handler, click **Delete** next to the event handler that you want to delete.
- To modify an event handler, select the event handler that you want to modify.

If you create or modify an event handler, the event handler configuration screen appears. For example:


Juniper NETWORKS
VTA Configuration Manager
Edit Configuration
Home
Dir Connection
Configuration
Create
Edit
Load
Commit
Import
Export
Current Configuration
Event Handlers
Actions
Processors
Quota VTA Events and Handlers
Current Event Handlers
Event Handler
GetQuota
Delete
RecordUsage
Delete
SetInterim
Delete
NoQuota
Delete
QuotaRefilled
Name
Value
Priority
333
Events
ACCOUNTUPDATE
CALLBACK(TERMINATESESSION)
SERVICEINTERIM(QuotaInternet)
SERVICEINTERIM(QuotaLocal)
SERVICESTART(QuotaInternet)
Condition
var newBalance=<balance_BoughtQuota>+<balance_PeriodicQu
if(<old_balance_PeriodicQuota>==null) <old_balance_Perio
if(<old_balance_BoughtQuota>==null) <old_balance_BoughtQ
return <old_balance_PeriodicQuota>+<old_balance_BoughtQu
Disable
Actions
Available Actions
CalculateInterim
CalcUsage
DebitAccounts
GetAccountBalances
SetInterim
Add >>
<< Remove
Selected Actions
StartQuotaInternetService
StartQuotaLocalService
Save
EndofBilling
Delete
New Event Handler
Create
JuniperyourNet

Related Topics

- How the SRC VTA Works on page 68
- Configuring Events on page 154
- Event Handler Fields on page 160
- Example of a Bucket VTA on page 193

Event Handler Fields

In VTA Configuration Manager, you can edit the following fields in the Event Handler screen.

Priority

- Priority for evaluating and running this event handler.
- Value—Integer in the range 1–1000; the smaller number has the higher priority
- Default—No value

Events

- List of events.
- Value—Select from the list of configured events. Use the Ctrl or Shift keys to select multiple events.
- Default—No value

Condition

- Condition that the event handler evaluates to determine whether the event handler should handle the event.
- Value—You can specify the condition as a formula, as a JavaScript program, or as one of the following values returned by the JavaScript program:
  - true—Event handler should handle the event.
  - false—Event handler should not handle the event.
- Guidelines—If no condition is specified, true is returned as the value. If a referenced attribute does not exist in the event, the referenced attribute's value is null.
- Default—true
- Example—The following example is a condition that must be met to refill a quota:

```
var newBalance=<balance_BoughtQuota>+<balance_PeriodicQuota>;
if(<old_balance_PeriodicQuota>==null)
   <old_balance_PeriodicQuota>=<balance_PeriodicQuota>;
if(<old_balance_BoughtQuota>==null)
   <old_balance_BoughtQuota>=<balance_BoughtQuota>;
return
   <old_balance_PeriodicQuota>+<old_balance_BoughtQuota><=0&&newBalance>0;
```

Actions

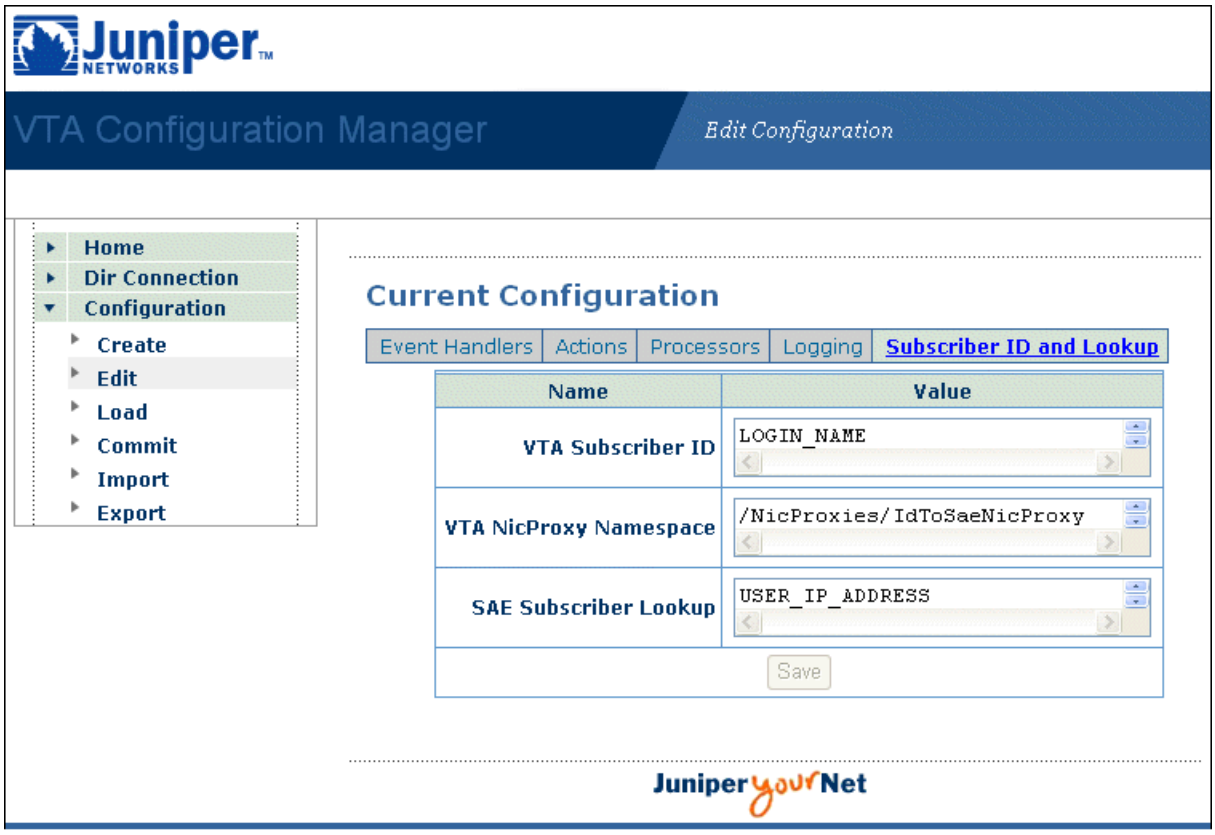
- Actions that the event handler performs in response to an event.
- Value—Use the Add and Remove buttons to create a list of selected actions. Multiple actions are performed in the order specified in the list.
- Default—No value

Configuring Identifiers for Subscribers and Sessions

To configure identifiers for subscribers and sessions:

1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Select **Subscriber ID and Lookup**.

   The Subscriber ID and Lookup screen appears.



3. Edit or accept the default values for the fields, and click **Save**.

   See “Subscriber ID and Lookup Fields” on page 162.

4. If you are finished configuring the SRC VTA, save the configuration to a directory or local file.

See “Committing a VTA Configuration to a Directory” on page 175.

For more information about this topic, see “Identifying Subscribers, SAEs, and Sessions” on page 74.

Related Topics

- Overview of the SRC VTA on page 65
- Using One VTA Account for Multiple Subscriber Sessions on page 163
- Automatic Login of Subscribers on page 180

Subscriber ID and Lookup Fields

In VTA Configuration Manager, you can edit the following fields in the Subscriber ID and Lookup screen.

VTA Subscriber ID

- Data key that identifies subscriber accounts and sessions in the VTA database. Some settings also provide information that the NIC and the SAE use to identify subscribers. For more information, see Table 10 on page 76.
- Value—One of the following data keys:
  - LOGIN_NAME— Login name (JUNOSe routers only), which is the data key for the VTA database, the NIC, and the SAE
  - USER_DN—Subscriber DN, which is the data key for the VTA database, the NIC, and the SAE
  - INTERFACE_NAME@ROUTER_NAME—Interface name and virtual router
    - Provides the ROUTER_NAME data key for NIC
    - Provides the INTERFACE_NAME data key for SAE
  - INTERFACE_ALIAS—Interface alias
  - INTERFACE_ALIAS@ROUTER_NAME—Interface alias and virtual router
  - USER_MAC_ADDRESS—Subscriber MAC address
  - PRIMARY_USER_NAME—Primary login name
  - PORT_ID@ROUTER_NAME —NAS port ID and virtual router (JUNOSe router only)
  - ACCOUNTING_ID—Accounting ID, which is the data key for the VTA database and the NIC. The NIC uses ACCOUNTING_ID to look up the SAE, and it returns SAE

references and subscriber IP address. The SRC VTA uses the returned IP address as the SAE data key.

- Guidelines—LOGIN_NAME, USER_DN, and INTERFACE_NAME@ROUTER_NAME also provide data keys for the NIC and the SAE; the other settings do not.
- Default—LOGIN_NAME

VTA NicProxy Namespace

- If you are using a NIC to map subscriber identifiers to an SAE, and you select a VTA subscriber ID value that provides a data key for the NIC, specify the NIC proxy that uses that data key.
- Value—Location of the NIC proxy configuration relative to the configuration properties for the SRC VTA.
- Guidelines—To cause the NIC to look up the SAE using the accounting ID, configure */NicProxies/AcctIdToSaeNicProxy*.
- Default—*/Nic/Proxies/IdToSaeNicProxy*

SAE Subscriber Lookup

- Data key that uniquely identifies the subscriber in your SRC configuration. The SRC VTA uses this data key to identify a subscriber session or service session when it receives a plug-in event. For more information, see Table 11 on page 77.
- Value—One of the following data keys:
  - USER_IP_ADDRESS—Subscriber IP address (JUNOSe routers only)
  - USER_DN—Subscriber DN
  - LOGIN_NAME —Login name (JUNOSe router only)
  - INTERFACE_NAME@ROUTER_NAME—Interface name and virtual router
  - PRIMARY_USER_NAME—Primary login name
- Default—Subscriber IP address

Using One VTA Account for Multiple Subscriber Sessions

The SRC VTA allows multiple subscriber sessions to share the same VTA account. The SRC VTA debits usage from all the subscriber sessions from the account. When the account is empty, service sessions for all subscribers are stopped. When the account is refilled, the SRC VTA starts services for all subscriber sessions that share the account.

To use this feature, you use the subscriberId event attribute to map a group of subscribers to the VTA account. You then use the ACCOUNTING_ID plug-in attribute as the VTA subscriber ID. You also set up the NIC to use the accounting ID to look up the SAE that manages a subscriber.

To set up the SRC software to use one VTA account for multiple subscribers:

1. In the subscriber classifier script, assign a value to the accountingUserId attribute. For example, you could assign it to the userName, interfaceName, loginName, or a combination of classification criteria. The purpose of the assignment is to allow the SRC VTA to identify subscribers by many different subscriber attributes using accountingUserId as a wrapper.

   For example, the following subscriber classifier script assigns the value of the userName to the accountingUserId attribute:

   [<-retailerDn->?accountingUserId=<-userName->?sub?(uniqueID=<-userName->)]

2. Configure the SAE to publish the PA_ACCOUNTING_ID plug-in attribute in subscriber tracking events to the NIC SAE agent plug-in.

   See Configuring Internal Plug-Ins (SRC CLI).

3. Configure the NIC to use the OnePopAcctId NIC scenario.

   See Configuring the NIC (SRC CLI).

4. In VTA Configuration Manager, configure the following in the Subscriber ID and Lookup screen:

   - In the VTA Subscriber ID box, enter ACCOUNTING_ID.
   - In the VTA NicProxy Namespace field, enter */NicProxies/AcctIdToSaeNicProxy*, which causes the NIC to look up the SAE using the accounting ID.

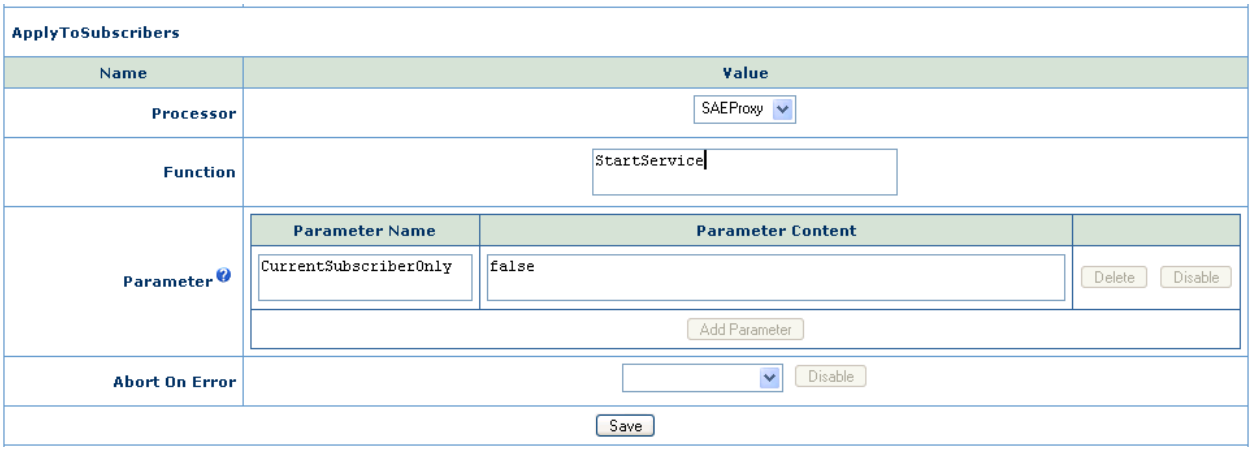
     See “Configuring Identifiers for Subscribers and Sessions” on page 161.

5. If you are using the VTA portals to manage subscriber accounts, configure the following property in the CONSTANTS.incl file:

   NIC_PROXY_NAMESPACE=/NicProxies/IpToAcctIdNicProxy

   See “Managing Subscriber Accounts with the Administrator Portal” on page 184.

6. (Optional) Set up an action for the SAE proxy processor to apply functions to all subscriber sessions that share the same VTA account. For example, the following action starts services for all subscriber sessions that have the same subscriber ID.



Related Topics

- Overview of the SRC VTA on page 65
- Managing VTA Accounts and Sessions on page 75
- Managing Subscriber Sessions and Service Sessions on page 76
- Configuring Subscribers and Subscriptions to VTA Services on page 89
- Configuring the SRC VTA to Manage Subscriber Accounts on page 115

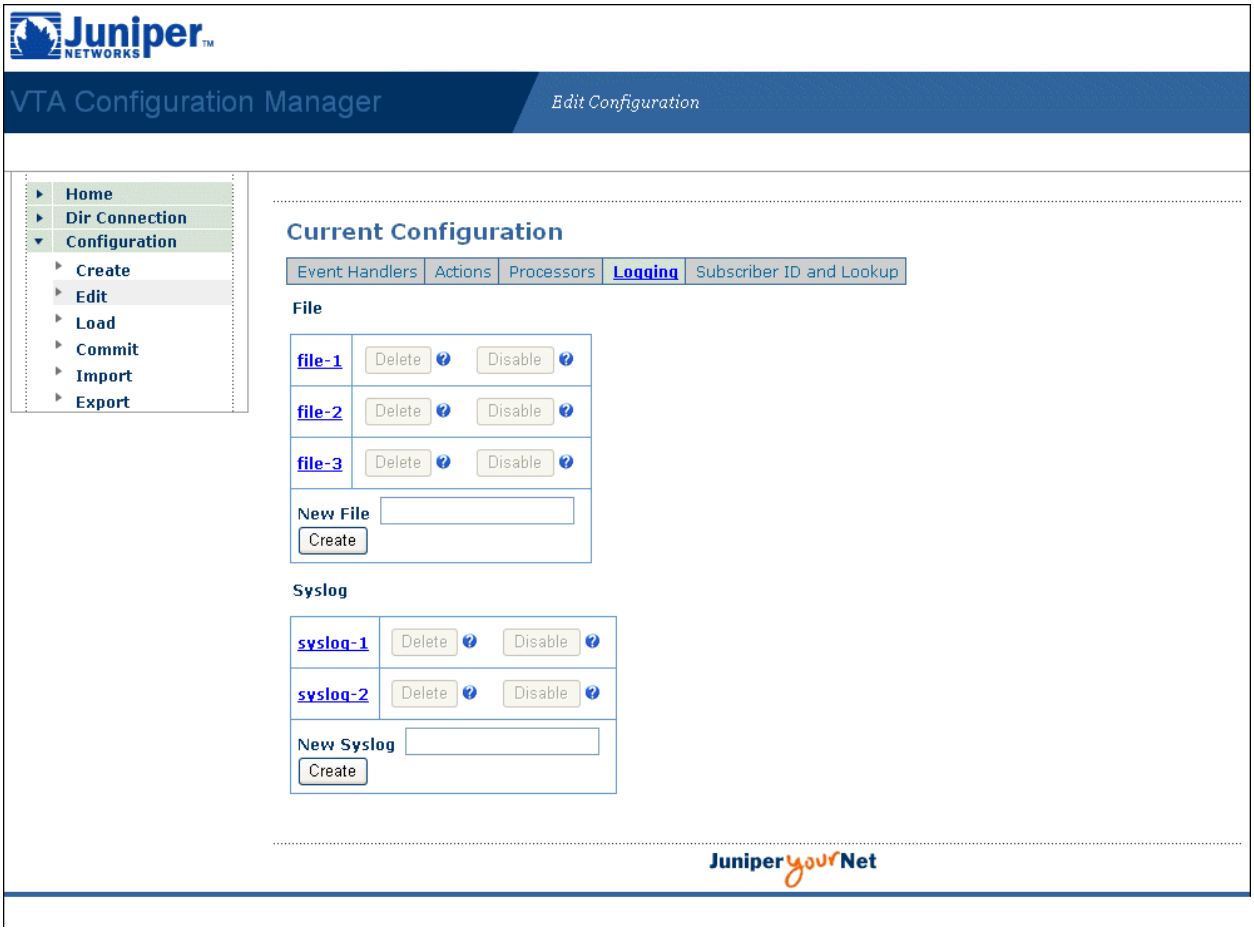
Logging Event Messages for the SRC VTA

The SRC VTA generates event messages that you can save in logs—either by writing the messages to text files or by using the system log (syslog) facilities.

To access the logging configuration:

1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, select **Logging**.

   The current File and Syslog configurations appear.



3. You can now configure a text file or syslog configuration. See:
   - "Logging Events Messages to a Text File" on page 166.
   - "Logging Events Messages to a System Logging Server" on page 171.

Related Topics

- How the SRC VTA Works on page 68
- File Logging Fields on page 168
- System Logging Fields on page 172
- Example of a Bucket VTA on page 193

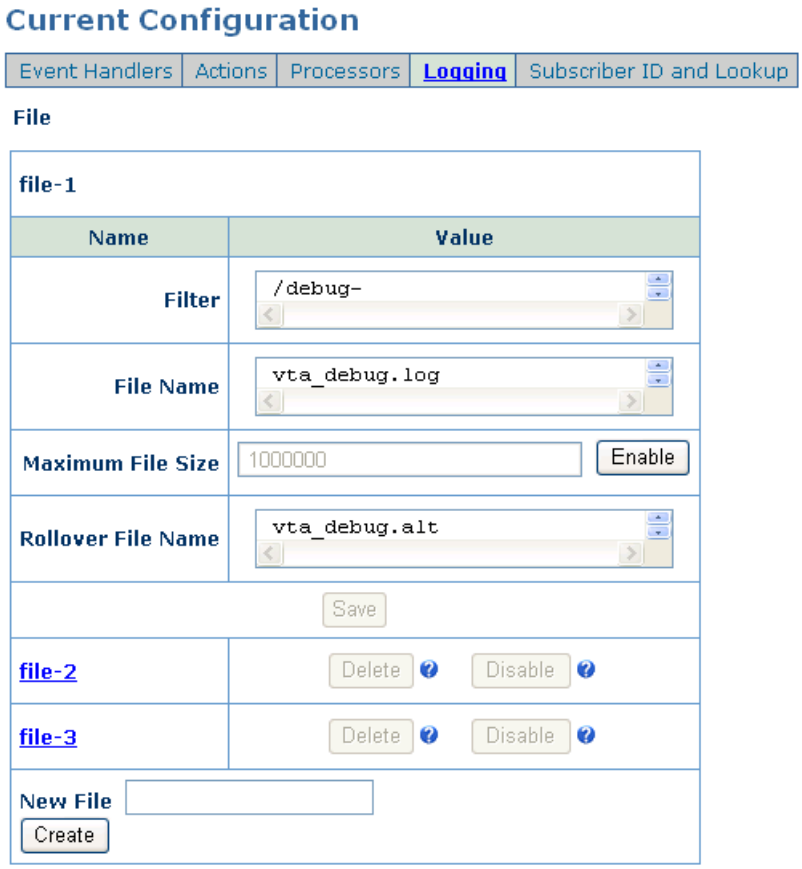
Logging Events Messages to a Text File

To set up the SRC VTA to save event messages in text files:

1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, select **Logging**.

3. In the File section, select a file logging configuration to modify, or type a new file name in the New File box, and click **Create**.

   The current file logging configuration appears.



4. Edit the file logging fields.

   See “File Logging Fields” on page 168 .

5. If you are finished configuring the SRC VTA, save the configuration to a directory or local file.

   See “Committing a VTA Configuration to a Directory” on page 175 .

Related Topics

- Logging Event Messages for the SRC VTA on page 165
- Logging Events Messages to a System Logging Server on page 171
- System Logging Fields on page 172
- Example of a Bucket VTA on page 193

File Logging Fields

In VTA Configuration Manager, you can modify the following fields in the File section of the Logging screen.

Filter

- Filter that determines the type of messages that this log file contains.
- Value—Expression. The software filters events by evaluating each subexpression from left to right. When the software finds a match, it logs or ignores the message accordingly. You can specify an unlimited number of subexpressions. The order in which you specify the subexpressions affects the result. Expressions have the format:

  singlematch [,singlematch]

  where

  singlematch—[!] (<category> | ([<category>]/[<severity>] | [<minimumSeverity>]-[<maximumSeverity>]))

  - !—Do not log matching events
  - <category>—SRC component that generated the event message. To log only events in a specific category, you can define the category, which is a text string that matches the name of a category. The text string is not case sensitive. For the names of categories, view a log file for a default filter. Juniper Networks Technical Assistance Center (JTAC) can also provide category names.
  - [<severity>] | [<minimumSeverity>]-[<maximumSeverity>] —Name or number in the range 1–127. A higher number indicates a higher severity level. Table 16 on page 168 shows common severity levels that you can specify by name.

Table 16: Named Severity Levels

Name	Severity Level
logmin	1
debug	10
info	20
notice	30
warning	40
error	50
crit	60
alert	70

Table 16: Named Severity Levels *(continued)*

Name	Severity Level
emerg	80
panic	90
logmax	127

Enabling debug log messages has a negative affect on system performance. Do not enable debug log messages unless JTAC instructs you to do so.

You can define a severity level as follows:

- Specify an explicit severity. For example:

  warning—Defines only warning messages

- Specify a minimum severity and a maximum severity. For example:

  info-warning—Defines messages of minimum severity level of info and a maximum severity level of warning

- Accept the default minimum (logmin) or maximum (logmax) severity by omitting the minimum or maximum severity. For example:

  info-—Defines messages of minimum severity level info and maximum severity level logmax

-warning—Defines messages of minimum severity level logmin and maximum severity level warning

- Specify no severity to log all event messages.

- Default—No value
- Example—Table 17 on page 170 shows some examples of filters.

Table 17: Examples of Filters for Event Messages

Syntax	Event Messages Saved
/	All event messages
/info-	Event messages of level info and above from all categories
vta/debug	Debug events from the VTA category only
!vta,/debug	All debug events except those from the VTA category
!VtaMsg/info-,vtaMsg,vta	All messages from the VTA category, except those from VtaMsg category with level less than info

Filename

- Filename that contains the current logs.
- Value—Path and filename of the current log file in the format:

  <pathname>/<filename.log>

- Guidelines—Make sure you have write access to the folder.
- Default—By default, SRC components and applications write log files in the folder where the application is started. If you are using the version of JBoss packaged with the SRC software, the pathname is */opt/UMC/jboss/server/default/log/.*
- Example—*/opt/UMC/jboss/server/default/log/vta_debug.log*

Maximum File Size

- Disables or enables and sets the maximum size of the log file and the rollover file.
- Value—Number of kilobytes in the range 0–4294967295
- Guidelines—Do not set the maximum file size to a value greater than the available disk space.
- Default—1000000

Rollover Filename

- Path and filename of the rollover log file. When the log file reaches the maximum size, the software closes the log file and renames it with the name you specify for the rollover file. If a previous rollover file exists, the software overwrites it. The software then reopens the log file and continues to save event messages in it.
- Value—Path and filename of the rollover log file in the format <pathname>/<filename.alt>
- Default—No value
- Example—vta_debug.alt

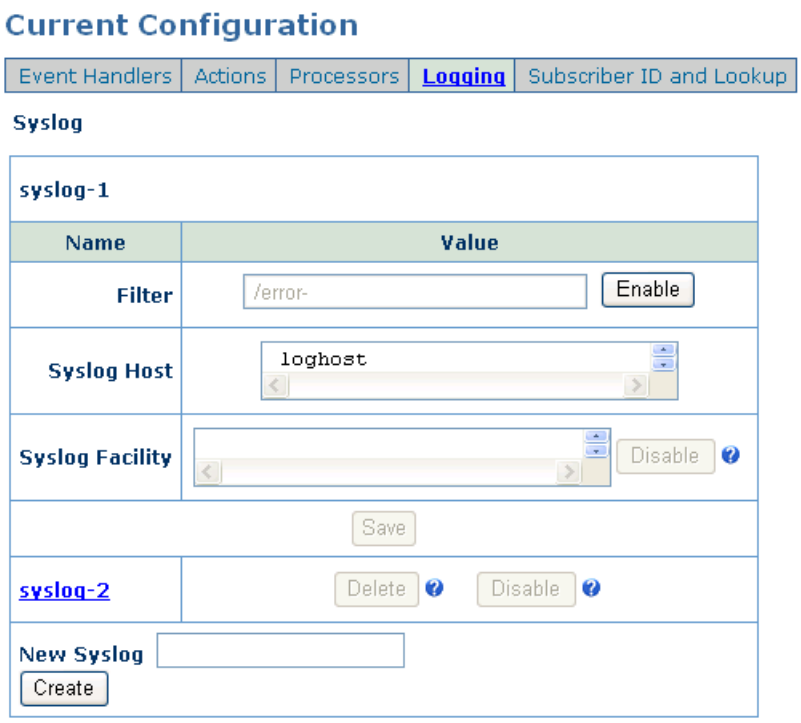
Logging Events Messages to a System Logging Server

To configure the SRC VTA to save event messages on a system logging server:

1. In the VTA Configuration Manager navigation pane, select **Edit**.
2. Under Current Configuration, select **Logging**.

   The current logging configuration appears.
3. To add a system log configuration, enter a name for the configuration in the New Syslog box, and click **Create**.

   The Syslog configuration screen appears.



4. Edit the fields.

See “System Logging Fields” on page 172 .

5. If you are finished configuring the SRC VTA, save the configuration to a directory or local file.

   See “Committing a VTA Configuration to a Directory” on page 175.

Related Topics

- Logging Event Messages for the SRC VTA on page 165
- Logging Events Messages to a Text File on page 166
- File Logging Fields on page 168
- Example of a Bucket VTA on page 193

System Logging Fields

In VTA Configuration Manager, you can edit the following fields in the syslog section of the Logging screen.

Filter

- Filter that determines the type of messages that this log file contains.
- Value—Expression. The software filters events by evaluating each subexpression from left to right. When the software finds a match, it logs or ignores the message accordingly. You can specify an unlimited number of subexpressions. The order in which you specify the subexpressions affects the result. Expressions have the format:

  singlematch [,singlematch]

  where

  singlematch—[!] (<category> | ([<category>]/[<severity>] | [<minimumSeverity>]-[<maximumSeverity>]))

  - !—Do not log matching events
  - <category>—SRC component that generated the event message. To log only events in a specific category, you can define the category, which is a text string that matches the name of a category. The text string is not case sensitive. For the names of categories, view a log file for a default filter. Juniper Networks Technical Assistance Center (JTAC) can also provide category names.
  - [<severity>] | [<minimumSeverity>]-[<maximumSeverity>] —Name or number in the range 1–127. A higher number indicates a higher severity level. Table 18 on page 172 shows common severity levels that you can specify by name.

Table 18: Named Severity Levels

Name	Severity Level
logmin	1
debug	10

Table 18: Named Severity Levels *(continued)*

Name	Severity Level
info	20
notice	30
warning	40
error	50
crit	60
alert	70
emerg	80
panic	90
logmax	127

Enabling debug log messages has a negative affect on system performance. Do not enable debug log messages unless JTAC instructs you to do so.

You can define a severity level as follows:

- Specify an explicit severity. For example:

  warning—Defines only warning messages

- Specify a minimum severity and a maximum severity. For example:

  info-warning—Defines messages of minimum severity level of info and a maximum severity level of warning

- Accept the default minimum (logmin) or maximum (logmax) severity by omitting the minimum or maximum severity. For example:

  info-—Defines messages of minimum severity level info and maximum severity level logmax

-warning—Defines messages of minimum severity level logmin and maximum severity level warning

- Specify no severity to log all event messages.

- Guidelines—This field is mandatory.
- Default—No value
- Example—Table 19 on page 174 shows some examples of filters.

Table 19: Examples of Filters for Event Messages

Syntax	Event Messages Saved
/	All event messages
/info-	Event messages of level info and above from all categories
vta/debug	Debug events from the VTA category only
!vta,/debug	All debug events except those from the VTA category
!VtaMsg/info-,vtaMsg,vta	All messages from the VTA category, except those from VtaMsg category with level less than info

Syslog Host

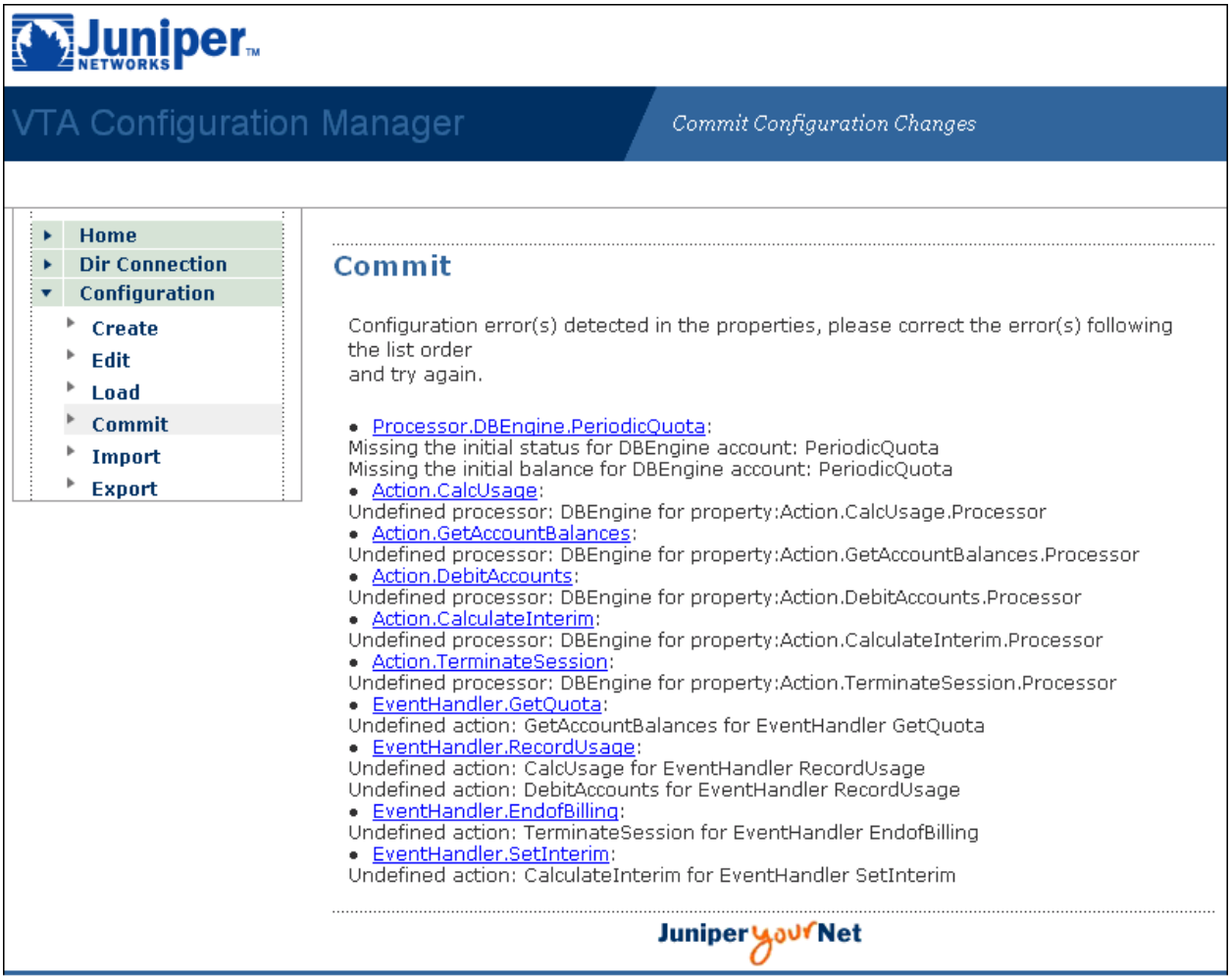
- IP address or name of a host that collects event messages with a standard system logging daemon.
- Value—IP address or text string
- Default—No value

Syslog Facility

- Type of system log in accordance with the system logging protocol.
- Value—Integer in the range 0–23; each integer corresponds to the standard number for a system logging client. See The syslog Protocol—draft-ietf-syslog-protocol-16.txt (July 2006 expiration).
- Default—No value

Validating VTA Configurations

When you commit a configuration to a directory or export the configuration to a file, VTA Configuration Manager validates the VTA configuration. If there are errors in the configuration, a screen detailing the errors and containing links to configuration objects that failed validation appears. For example:



If you click on a link, the VTA Configuration Manager displays the page that contains the error and lists errors for that page at the bottom of the screen.

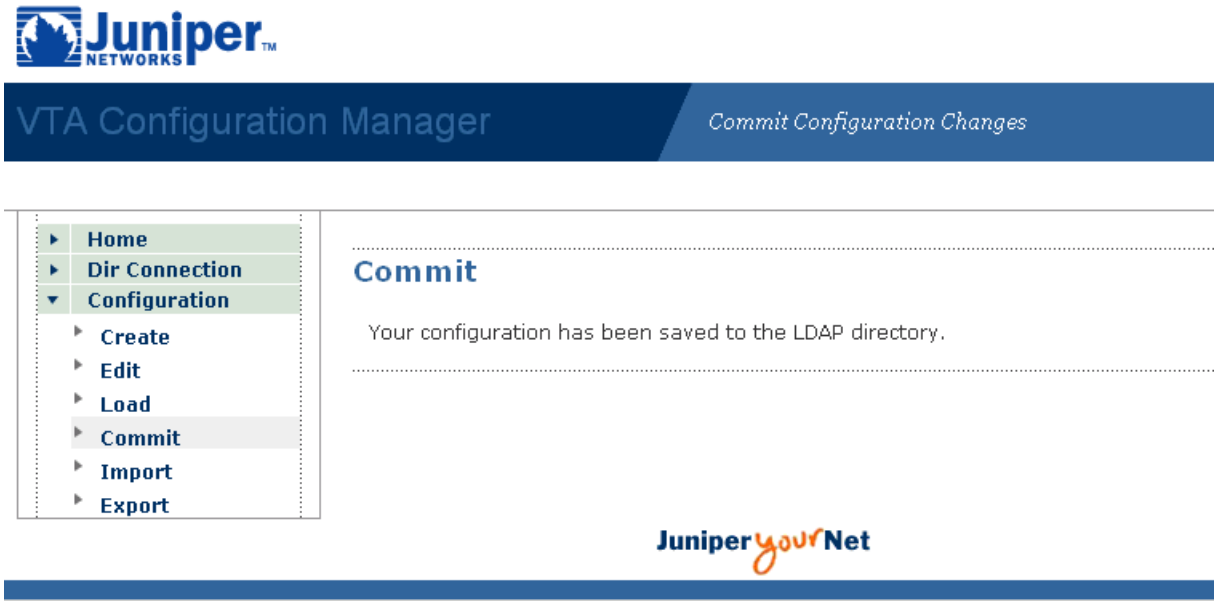
Related Topics

- Overview of the SRC VTA on page 65
- Installing the SRC VTA and Running the Configuration Script on page 82
- Accessing the VTA Configuration on page 110
- Testing the VTA Configuration on page 187
- Committing a VTA Configuration to a Directory on page 175

Committing a VTA Configuration to a Directory

Before you commit your VTA configuration to a directory, you need to connect to the directory and specify the name and location of your configuration in the Configuration Namespace field of the Connecting to the Directory page. See “Loading a Configuration from a Directory” on page 106.

To commit your VTA configuration to a directory, in the navigation pane, select Commit. The software copies your configuration to the directory and the Commit page.



Related Topics

- Loading and Importing VTA Configurations on page 105
- Importing a VTA Configuration from a Local File on page 109
- Exporting a VTA Configuration to a Local File on page 176
- Specifying How the SRC VTA Loads Configurations from the Directory on page 90

Exporting a VTA Configuration to a Local File

To export a VTA configuration to a local file:

1. In the VTA Configuration Manager navigation pane, select **Export**.

   The Export The Current Configuration page appears.

2. Right-click the word **here**, and then use your browser to save the file.

Related Topics

- Accessing the VTA Configuration on page 110
- Loading and Importing VTA Configurations on page 105
- Loading a Configuration from a Directory on page 106
- Importing a VTA Configuration from a Local File on page 109
- Committing a VTA Configuration to a Directory on page 175

CHAPTER 8

Managing Subscriber Accounts with VTA Portals

- Overview of Managing Subscriber Accounts with VTA Portals on page 179
- Automatic Login of Subscribers on page 180
- Configuring Web Applications for the SRC VTA on page 180
- Managing Subscriber Accounts with the Administrator Portal on page 184
- Accessing the Administrator Portal on page 184
- Viewing Subscriber Accounts on page 185
- Replenishing Periodic Accounts on page 186
- Deleting Information from the VTA's Database on page 186
- Testing the VTA Configuration on page 187
- Allowing Subscribers to Manage Their Accounts with the Subscriber Portal on page 187
- Accessing the Subscriber Portal on page 188
- Viewing Information About the Account on page 189
- Purchasing a Periodic Account on page 189
- Suspending a Periodic Account on page 190
- Purchasing Extra Bandwidth on page 191

Overview of Managing Subscriber Accounts with VTA Portals

We provide two sample portals that manage subscriber accounts. One is an administrator portal that administrators can use to manage VTA subscriber accounts. The second is a subscriber portal that subscribers can use to manage their own accounts. Before you can use these portals, you need to configure the Web applications for the SRC VTA (see “Configuring Web Applications for the SRC VTA” on page 180).

The sample portals run inside the J2EE application server and use the VTA core’s NIC proxy.

To customize the portals to operate outside the VTA’s J2EE application server, you must:

- Develop a way to connect them to the VTA’s remote account and session manager interfaces.

- If the software automatically logs in subscribers with their current IP addresses (JUNOSe only), configure NIC proxies for the client (see “Configuring NIC Proxies for the VTA” on page 99).

Automatic Login of Subscribers

By using a NIC with the SRC VTA, you can enable subscribers who connect to the SRC network through a JUNOSe router to log in through the VTA Web portals. The NIC must map the subscriber’s IP address to one of the following values:

- Subscriber’s login name
- Subscriber’s DN
- Name of the interface and VR to which the subscriber connects

The value to which this NIC maps the subscriber’s IP address depends on the information that identifies subscribers in your SRC configuration.

If subscribers connect to the network through a router running JUNOS Software or you choose not to implement this NIC with a router running JUNOSe Software, you can provide subscribers with access to this information through another method, such as a central login page where subscribers can enter their usernames and passwords.

Related Topics

- Locating the SAE That Manages a Subscriber for the SRC VTA on page 98
- Specifying Tracking Plug-Ins for Enterprise Subscribers on JUNOS Routing Platforms on page 94
- Configuring Subscribers and Subscriptions to VTA Services on page 89
- Viewing Subscriber Accounts on page 185
- Accessing the Subscriber Portal on page 188

Configuring Web Applications for the SRC VTA

To configure Web applications for the SRC VTA:

1. If you have not already done so, create a folder for the SRC VTA on a host, and copy the EAR file for the SRC VTA to the VTA folder.

   mkdir vta
   cp <file> /quotavta.ear vta

2. From the EAR file, extract the file *<vtaName>CustCare.war* into the folder you created.

   cd vta
   jar xvf quotavta.ear quotaCustCare.war

3. From the file *<vtaName>CustCare.war*, extract the file *CONSTANTS.incl*.

   jar xvf quotaCustCare.war CONSTANTS.incl

4. Edit the properties in the file CONSTANTS.incl.

   See "Properties for VTA Portals" on page 181 for a description of the properties in the CONSTANTS.incl file.

5. Replace the file *CONSTANTS.incl* into the file *<vtaName>CustCare.war*.

   jar uvf quotaCustCare.war CONSTANTS.incl

6. Replace the file *<vtaName>CustCare.war* into the EAR file.

   jar uvf quotavta.ear quotaCustCare.war

Related Topics

- Installing SRC Web Applications on page 7
- Installing Web Applications Inside JBoss on page 7
- Managing Subscriber Accounts with the Administrator Portal on page 184

Properties for VTA Portals

This section describes the properties in the CONSTANTS.incl file.

PERIODIC_QUOTA

- Quota initially deposited into a subscriber's periodic account when the subscriber activates the account through the subscriber portal. This value must match the value that the administrator specifies when replenishing periodic accounts. (See "Replenishing Periodic Accounts" on page 186).
- Value—Integer in the range 1–9223372036854775807 followed by the letter L or calculation in Java code
- Default—5000000000L

SUBSCRIBER_ACTIVATE_DESCRIPTION

- Description recorded for a balance change in the periodic account when the subscriber activates the account through the subscriber portal. This description records the change in the account status; however, the actual change in the balance is zero.
- Value—Text string
- Default—Activated by subscriber

PERIODIC_QUOTA_INITIAL_TOP_UP_DESCRIPTION

- Description recorded for a balance change in the periodic account when the subscriber activates the account through the subscriber's portal. This change in the balance is the value of the PERIODIC_QUOTA property.
- Value—Text string
- Default—Periodic replenishment to 5,000 MB

SUBSCRIBER_DEACTIVATE_DESCRIPTION

- Description for a balance change in the periodic account when the subscriber deactivates the account through the subscriber's portal. This description records the change in the account status; however, the actual change in the balance is zero. When the periodic account is inactive but the balance is nonzero, the SRC VTA continues to use the account until the administrator invalidates inactive accounts through the administrator's portal.
- Value—Text string
- Default—Deactivated by subscriber

PERIODIC_QUOTA_INACTIVE_TOP_UP_DESCRIPTION

- Description for a balance change in inactive periodic accounts when the administrator replenishes periodic accounts through the administrator's portal. This action sets the balance of inactive periodic accounts to zero.
- Value—Text string
- Default—Expiration of deactivated periodic account's balance

SUBSCRIBER_PURCHASE_DESCRIPTION

- Description recorded for a balance change in the bought account when the subscriber buys bandwidth through the subscriber portal.
- Value—Text string
- Default—Subscriber purchased bandwidth

NIC_PROXY_NAMESPACE

- Namespace of the NIC proxy. This NIC proxy maps the subscriber's IP address to the identifier that the SRC VTA uses as the key to the subscriber's records in the VTA database.
- Value—Path, relative to the root of the static configuration properties, that defines where the NIC proxy configuration is stored.
- Guidelines—If you are using the accounting ID to identify the SAE that manages a subscriber, set this property to /NicProxies/AcctIdToSaeNicProxy
- Default—/NicProxies/IpToIdNicProxy

BOUGHT_QUOTA_ACCOUNT_NAME

- Name of account that stores bought quota.
- Value—Text string
- Guidelines—If you change this value from the default, you must also update the sample Quota VTA configuration to use the same account names and statuses.
- Default—BoughtQuota

PERIODIC_QUOTA_ACCOUNT_NAME

- Name of account that stores periodic quota.
- Value—Text string
- Guidelines—If you change this value from the default, you must also update the sample Quota VTA configuration to use the same account names and statuses.
- Default—PeriodicQuota

ACTIVE_STATUS

- Status string.
- Value—Text string
- Guidelines—If you change this value from the default, you must also update the sample Quota VTA configuration to use the same account names and statuses.
- Default—Active

INACTIVE_STATUS

- Status string.
- Value—Text string
- Guidelines—If you change this value from the default, you must also update the sample Quota VTA configuration to use the same account names and statuses.
- Default—Inactive

SSPORTAL_SIGNIN_URL

- URL of the portal login page for the Quota VTA to authenticate users, primarily users with addresses that are assigned externally whose sessions are not detected automatically by the SAE.
- Value
  - <URL>—URL in the format "http://<server>:<port>/"

  - null—Keyword not enclosed in quotation marks

- Guidelines—If the URL is specified, the SRC VTA redirects the subscriber to the portal login page when the SRC VTA cannot resolve a subscriber by IP address. If the value is null and the SRC VTA cannot resolve the subscriber by IP address, an unknown subscriber message appears.

- Default—null

Managing Subscriber Accounts with the Administrator Portal

We provide a sample administrator portal for the Quota VTA that manages VTA subscriber accounts. You can use this portal to demonstrate and test the SRC VTA. In a production environment, you must integrate your customer interface with these portals. For more information, contact Juniper Networks Professional Services.

The sample portal is designed to be used with the Quota configuration example in the sample data. As a result, the portals expect the Periodic and Bought accounts.

Related Topics

- Overview of the SRC VTA on page 65
- Managing VTA Accounts and Sessions on page 75
- Configuring the SRC VTA to Manage Subscriber Accounts on page 115
- Accessing the Administrator Portal on page 184
- Viewing Subscriber Accounts on page 185

Accessing the Administrator Portal

To access the sample administrator portal:

1. Enter the following URL in your Web browser.

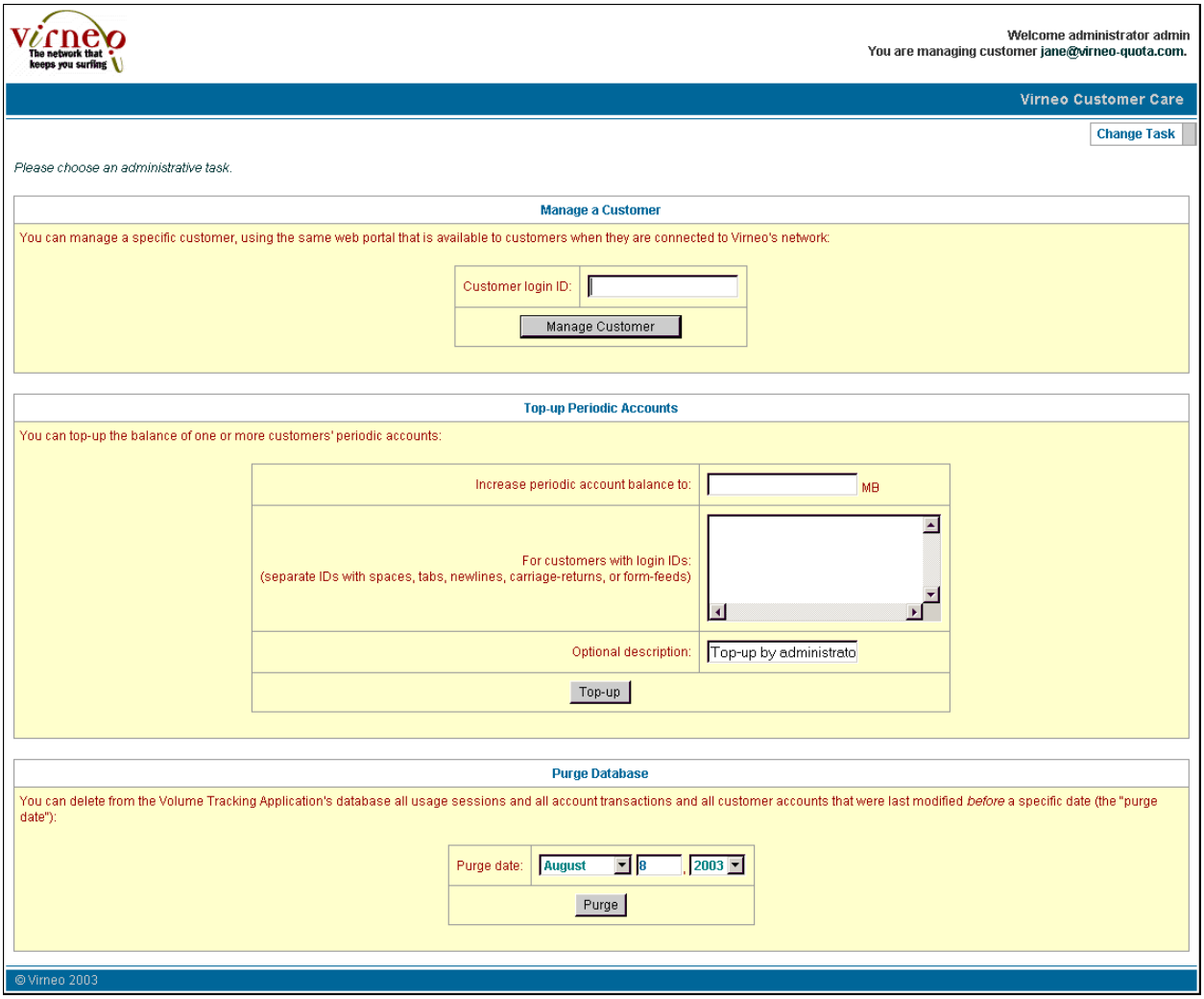
   http://<host>:<port>/quotaCustCare/admin.jsp

   - <host>—IP address or name of the host on which you installed the SRC VTA
   - <port>—HTTP port for the J2EE application server.

2. When prompted, enter the username and password configured for the J2EE application server (see “Accessing the J2EE Application Server’s Client Libraries” on page 90).

NOTE: If you are using JBoss and ran the configuration script, the script creates the username admin and the password secret for demonstration purposes.

   The administrator portal appears.



Related Topics

- Configuring Web Applications for the SRC VTA on page 180
- Properties for VTA Portals on page 181
- Managing Subscriber Accounts with the Administrator Portal on page 184
- Allowing Subscribers to Manage Their Accounts with the Subscriber Portal on page 187

Viewing Subscriber Accounts

Purpose To view a subscriber account:

1. On the administrator portal, click Manage a Customer.
2. In the Customer login ID box, enter the subscriber's login ID.
3. Click Manage Customer.

The subscriber portal appears. For information about using the subscriber's portal, see "Allowing Subscribers to Manage Their Accounts with the Subscriber Portal" on page 187.

Related Topics

- Configuring the SRC VTA to Manage Subscriber Accounts on page 115
- Replenishing Periodic Accounts on page 186
- Viewing Information About the Account on page 189

Replenishing Periodic Accounts

A periodic account is an account into which volume is periodically added to the subscriber's account. The periodic quota is tracked in the periodic account.

To replenish subscriber periodic accounts:

1. On the administrator portal, scroll to the Top-up Periodic Accounts area.
2. In the Increase periodic account balance to box, enter the value to which the customer's account should be replenished.
3. In the For customers with login IDs box, enter the login IDs of the subscribers whose accounts you wish to replenish.
4. (Optional) In the Optional description box, enter a summary of your action, which will appear in the subscriber accounts.
5. Click **Top-up**.

Related Topics

- Purchasing a Periodic Account on page 189
- Suspending a Periodic Account on page 190
- Viewing Information About the Account on page 189
- Viewing Subscriber Accounts on page 185

Deleting Information from the VTA's Database

To delete information about subscriber accounts, account transactions, and session histories:

1. On the administrator portal, scroll to the Purge Database area.
2. In the Purge date box, enter the date before which all data should be deleted from the database.
3. Click **Purge**.

Related Topics

- Configuring a Database to Store Account and Session Data on page 84
- Configuring the SRC VTA to Manage Database Accounts on page 111

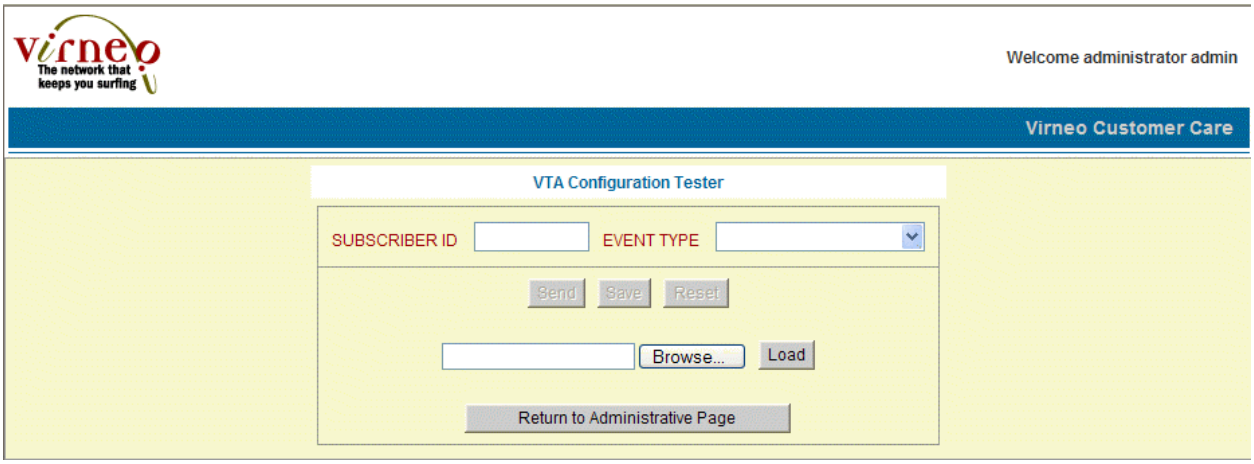
- Troubleshooting Database Deadlocks on page 88

Testing the VTA Configuration

To test the VTA configuration:

1. On the administrator portal, scroll to the Test VTA Configuration area.
2. Click **Test** to generate a simulated event for testing.

   The VTA Configuration Tester page appears.



3. Enter the subscriber ID and event type. The corresponding Web page for the specified event type appears.
4. Enter the event attributes to test, click **Send** to process the event in the Quota VTA, and print the results into VTA log files. You can also load an event from a file by selecting the file with **Browse** and clicking **Load**.
5. (Optional) Save the event created in the VTA Configuration Tester to a file by clicking **Save**.

Related Topics

- Validating VTA Configurations on page 174
- Installing the SRC VTA and Running the Configuration Script on page 82
- Installing VTA Configuration Manager on page 104
- Accessing the VTA Configuration on page 110
- Loading and Importing VTA Configurations on page 105
- Committing a VTA Configuration to a Directory on page 175

Allowing Subscribers to Manage Their Accounts with the Subscriber Portal

We provide a sample subscriber portal for the Quota VTA that subscribers can use to manage their accounts. You can use this portal to demonstrate and test the SRC VTA.

In a production environment, you must integrate your customer interface with these portals. For more information, contact Juniper Networks Professional Services.

The sample portal is designed to be used with the Quota configuration example in the sample data. As a result, the portals expect the Periodic and Bought accounts.

The Quota VTA creates an account with zero balance and inactive status the first time that the subscriber logs in. Typically, you direct subscribers to the subscriber portal when they log in for the first time. The subscriber can activate the periodic account and buy bandwidth. Subscribers are billed according to the choices they make. When an administrator replenishes periodic accounts, the subscriber receives a new quota of bandwidth and is billed for that quota.

Subscribers can deactivate their periodic accounts. Subscribers who do so can use the balance of their periodic quotas before you replenish periodic accounts. At this time, inactive accounts expire, and their balances are set to zero. Subscribers can reactivate their accounts at any time and will be billed for the new periodic quota.

Related Topics

- Overview of the SRC VTA on page 65
- Managing Subscriber Accounts with the Administrator Portal on page 184
- Accessing the Subscriber Portal on page 188
- Replenishing Periodic Accounts on page 186
- Example of a Bucket VTA on page 193

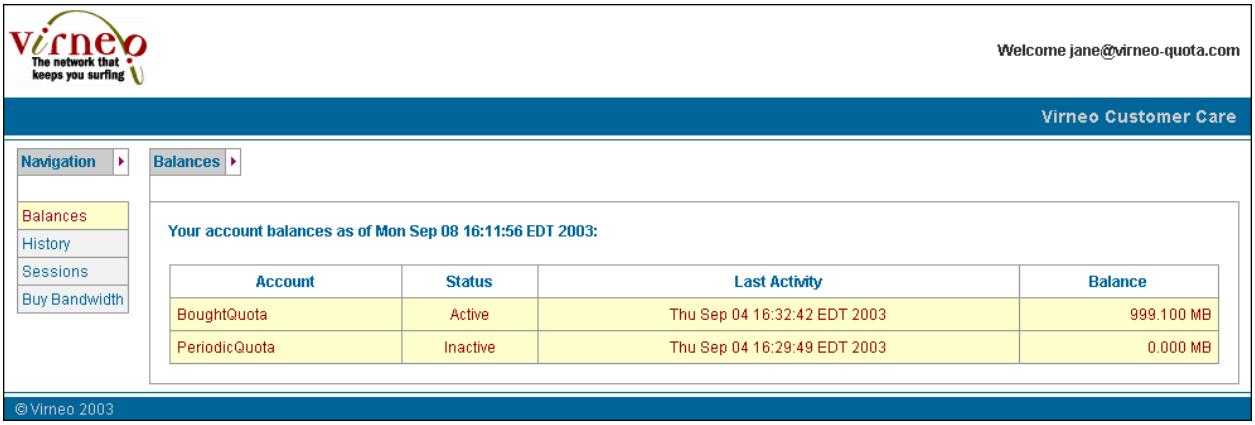
Accessing the Subscriber Portal

To access the sample subscriber portal, enter the following URL in your Web browser:

http://<host>:<port>/quotaCustCare/custcare.jsp

- <host> is the IP address or name of the host on which you installed the SRC VTA.
- <port> is the HTTP port for the J2EE application server.

The subscriber portal appears.



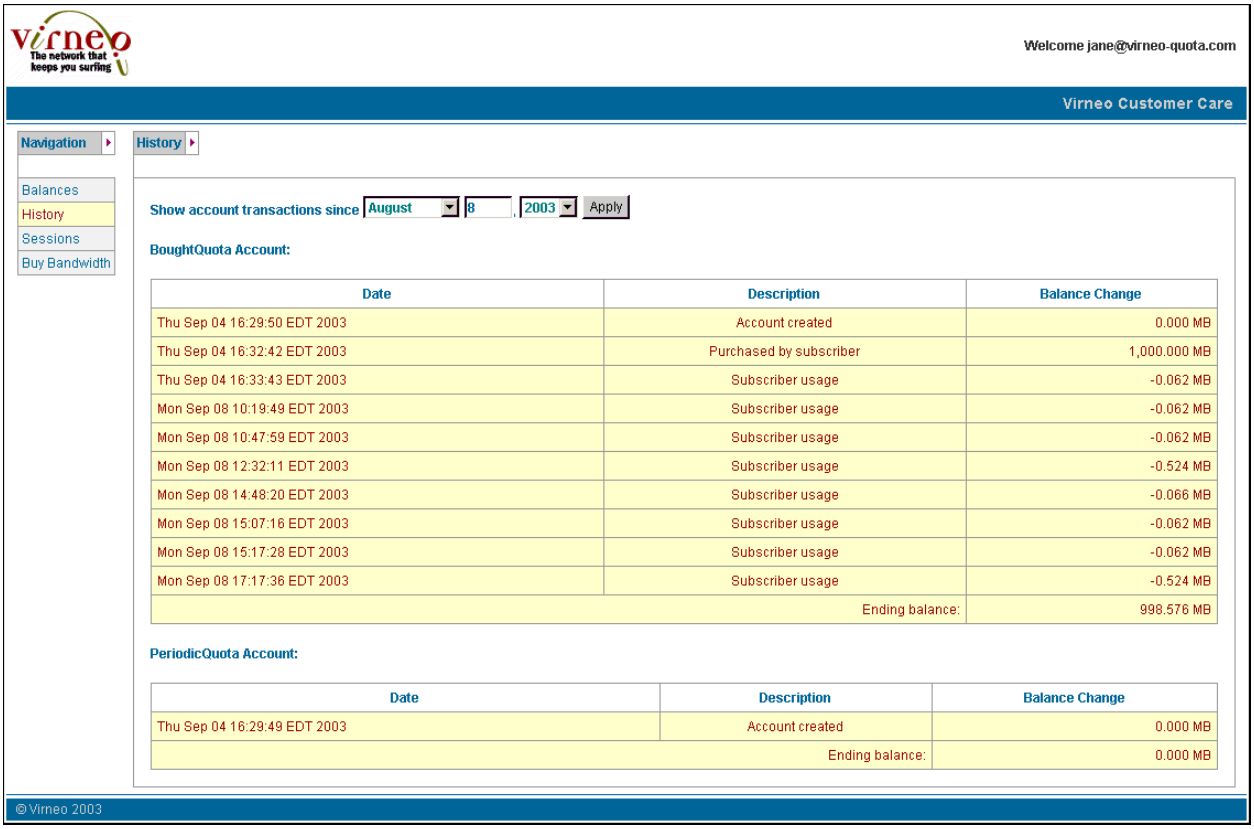
Related Topics

- Overview of the SRC VTA on page 65
- Allowing Subscribers to Manage Their Accounts with the Subscriber Portal on page 187
- Managing Subscriber Accounts with the Administrator Portal on page 184
- Viewing Subscriber Accounts on page 185

Viewing Information About the Account

Purpose View account balances, account history, or current sessions.

Action Click the appropriate button in the navigation area of the subscriber portal. The corresponding Web page appears.



Related Topics

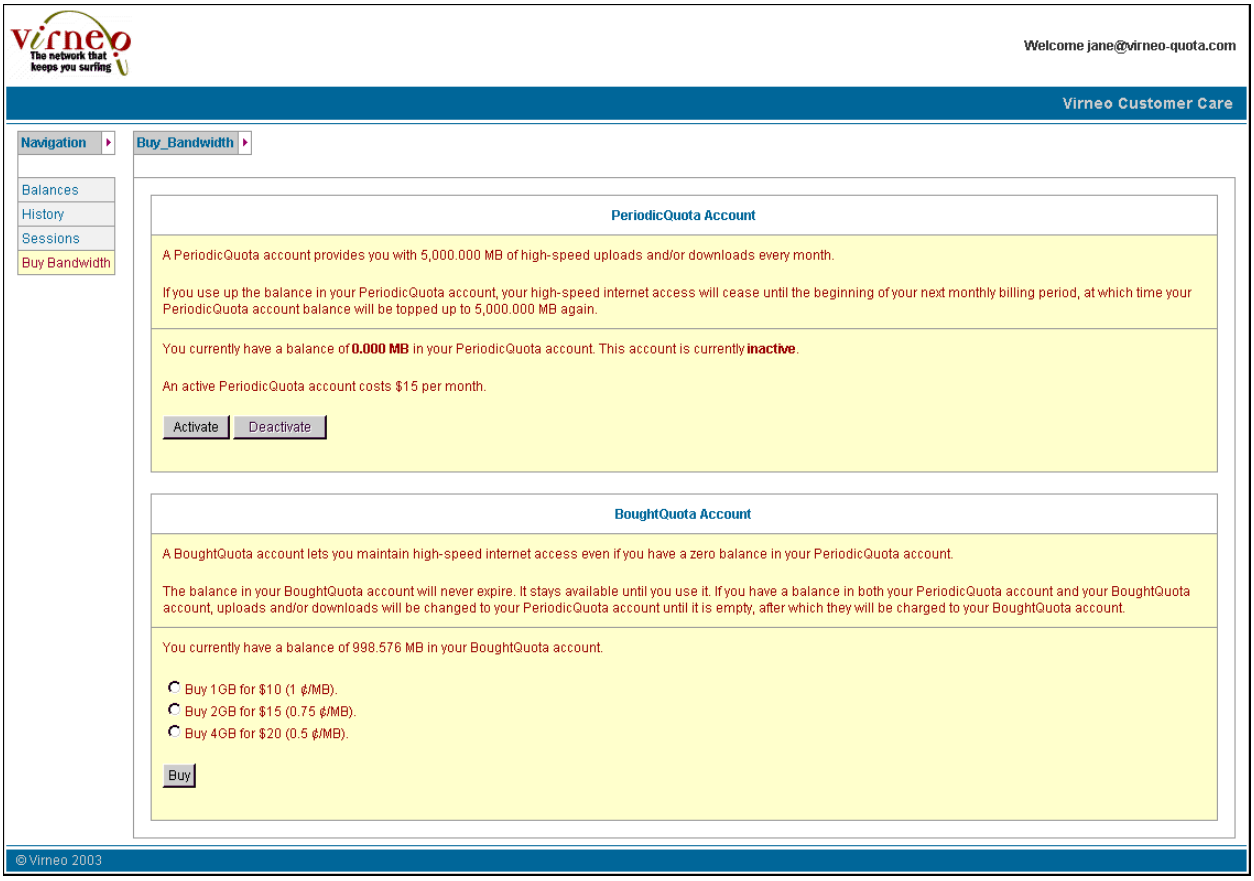
- Configuring a Database to Store Account and Session Data on page 84
- Configuring Subscribers and Subscriptions to VTA Services on page 89
- Viewing Subscriber Accounts on page 185

Purchasing a Periodic Account

A periodic account is an account into which the service provider periodically adds volume. The periodic quota is tracked in the periodic account.

To purchase an account that your service provider replenishes periodically:

1. In the navigation pane, click **Buy Bandwidth**.
2. Navigate to the PeriodicQuota Account area.
3. Click **Activate**.



Related Topics

- Suspending a Periodic Account on page 190
- Replenishing Periodic Accounts on page 186
- Purchasing Extra Bandwidth on page 191
- Viewing Information About the Account on page 189

Suspending a Periodic Account

To suspend an account that your service provider replenishes periodically:

1. In the navigation pane, click **Buy Bandwidth**.
2. Navigate to the PeriodicQuota Account area.
3. Click **Deactivate**.

Related Topics

- Purchasing a Periodic Account on page 189
- Replenishing Periodic Accounts on page 186
- Purchasing Extra Bandwidth on page 191
- Viewing Information About the Account on page 189

Purchasing Extra Bandwidth

To purchase extra bandwidth:

1. In the navigation pane, click **Buy Bandwidth**.
2. Navigate to the BoughtQuota Account area.
3. Select the option you want to purchase.
4. Click **Buy**.

   If this is the first time you have purchased extra bandwidth, your bought account will be activated. The bandwidth you purchased will be added to your bought account.

Related Topics

- Purchasing a Periodic Account on page 189
- Viewing Information About the Account on page 189

CHAPTER 9

Example of a Bucket VTA

- Example of a Bucket VTA on page 193

Example of a Bucket VTA

- Overview of Bucket VTA Example on page 193
- Events for Bucket VTA on page 193
- Event Handlers for Bucket VTA on page 194
- Database Engine Processor for Bucket VTA on page 197
- SAE Proxy Processor for Bucket VTA on page 199
- Actions for Bucket VTA on page 201

Overview of Bucket VTA Example

In this example each subscriber has a bucket account; that is, an account that is periodically measured and refilled depending on the usage of the account.

In this example, there are two bucket sizes:

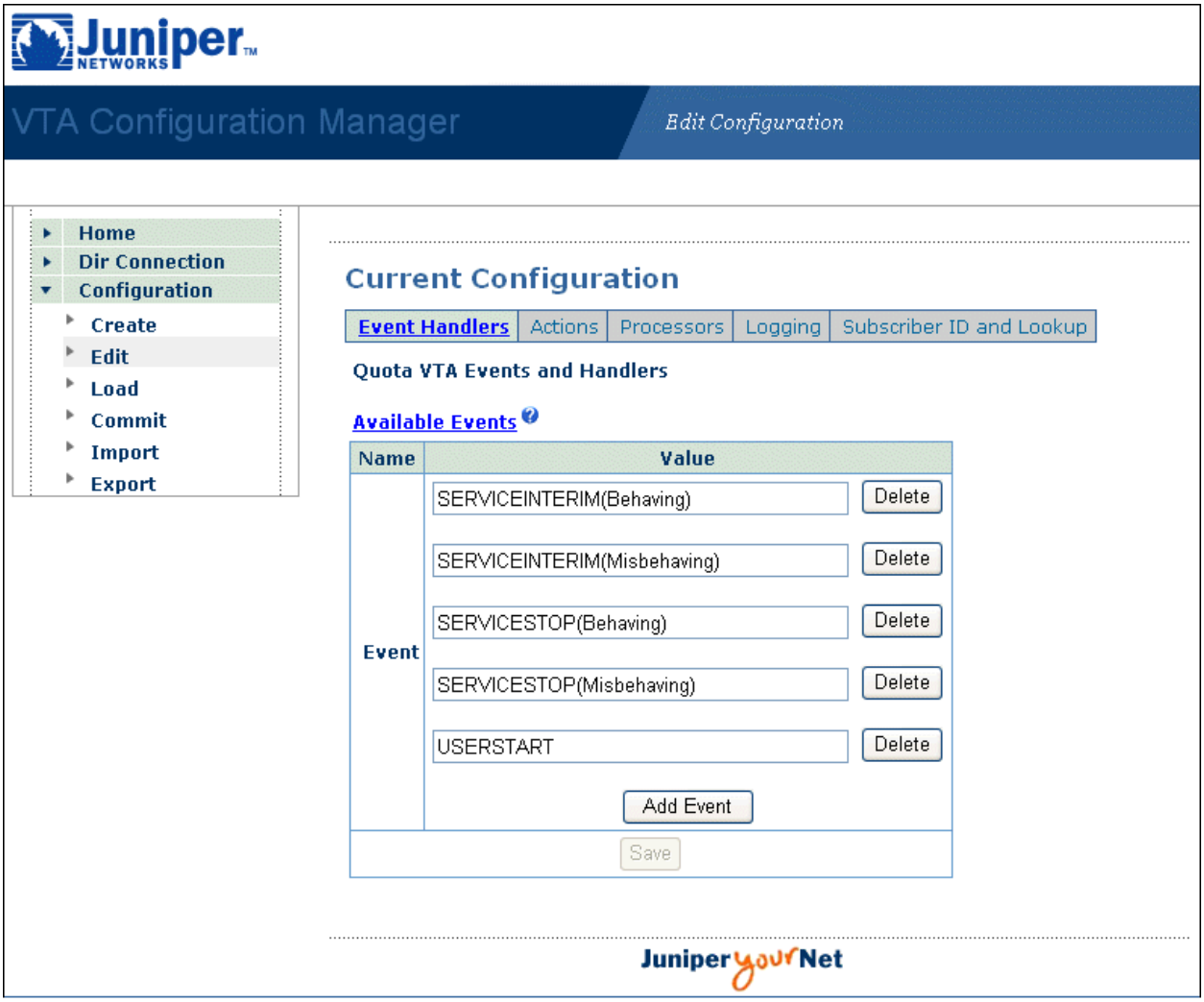
- Behaving bucket size—2 GB
- Misbehaving bucket size—1 GB

The rate at which the buckets are refilled depends on usage. If the subscriber is staying within the behaving rate, the bucket account is refilled at the rate of 2 GB per week. If the subscriber is exceeding the behaving rate, the bucket account is refilled at the rate of 1 GB per week:

- Behaving refill rate—3550 bps, or 2 GBs per 7 days
- Misbehaving refill rate —1775 bps, or 1 GB per 7 days

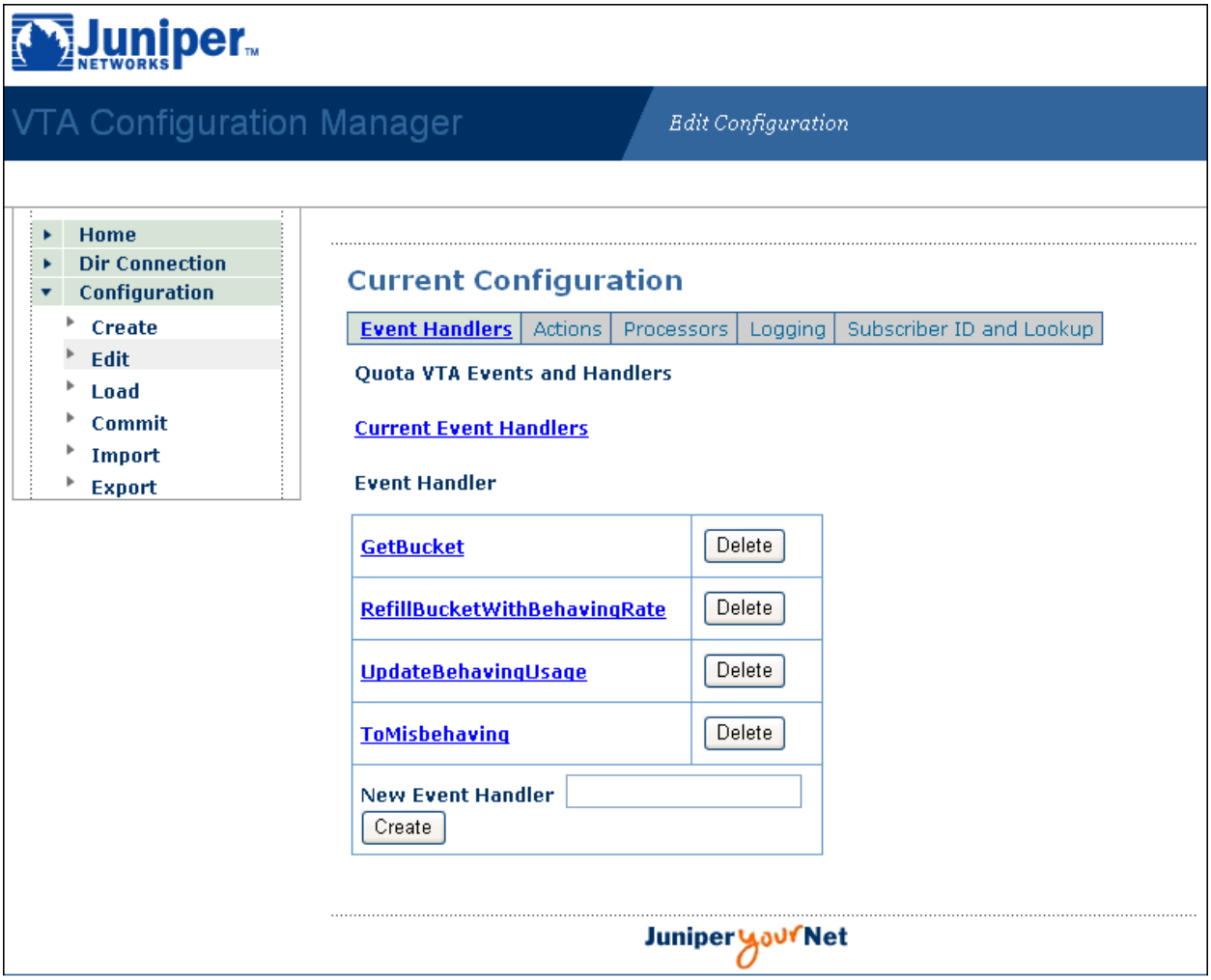
Events for Bucket VTA

The bucket VTA example has the following events:



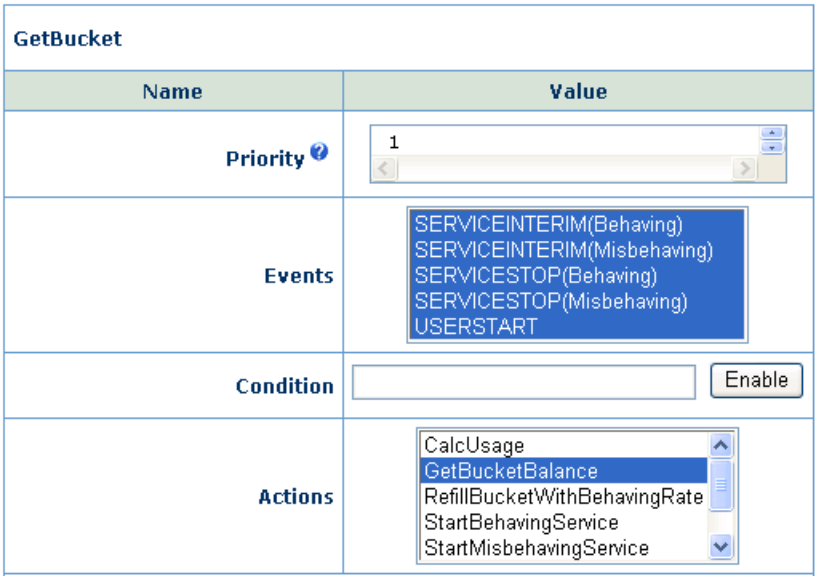
Event Handlers for Bucket VTA

The bucket VTA includes the following event handlers.



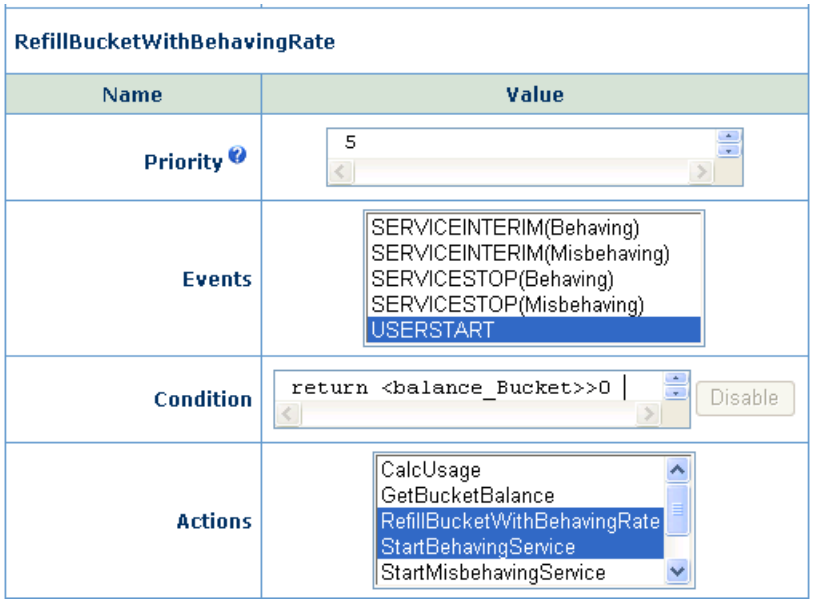
GetBucket Event Handler

The GetBucket event handler retrieves the balance of the bucket account when the subscriber logs in.



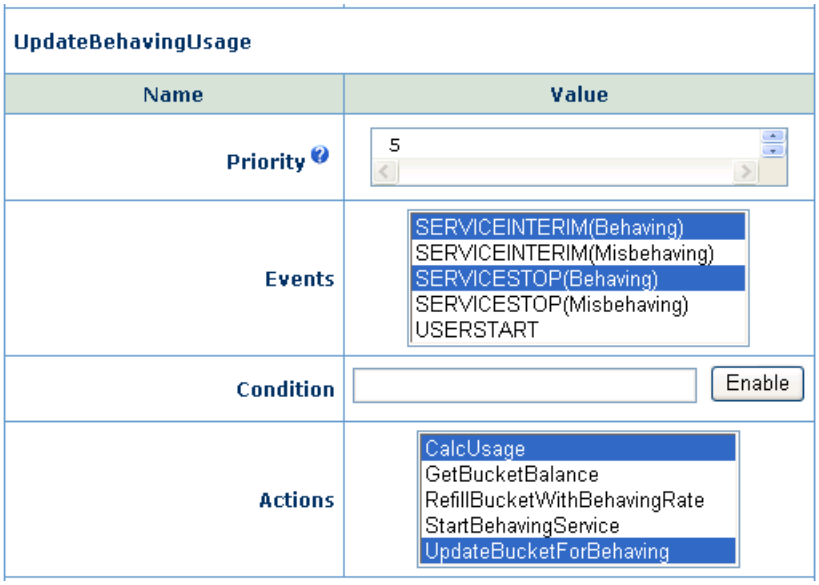
RefillBucketWithBehavingRate Event Handler

The RefillBucketWithBehavingRate event handler refills the bucket with the behaving rate and starts the behaving service if the bucket account has a positive balance.



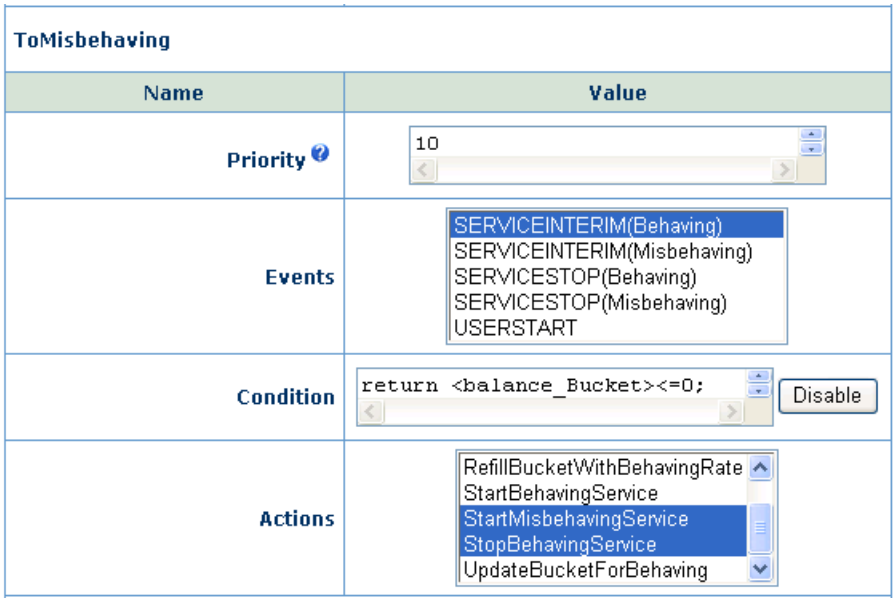
UpdateBehavingUsage Event Handler

The UpdateBehavingUsage event handler updates the bucket account with the usage from the behaving service.



ToMisbehaving Event Handler

The ToMisbehaving event handler changes the subscriber to misbehaving mode when the bucket has zero or negative balance.



Database Engine Processor for Bucket VTA

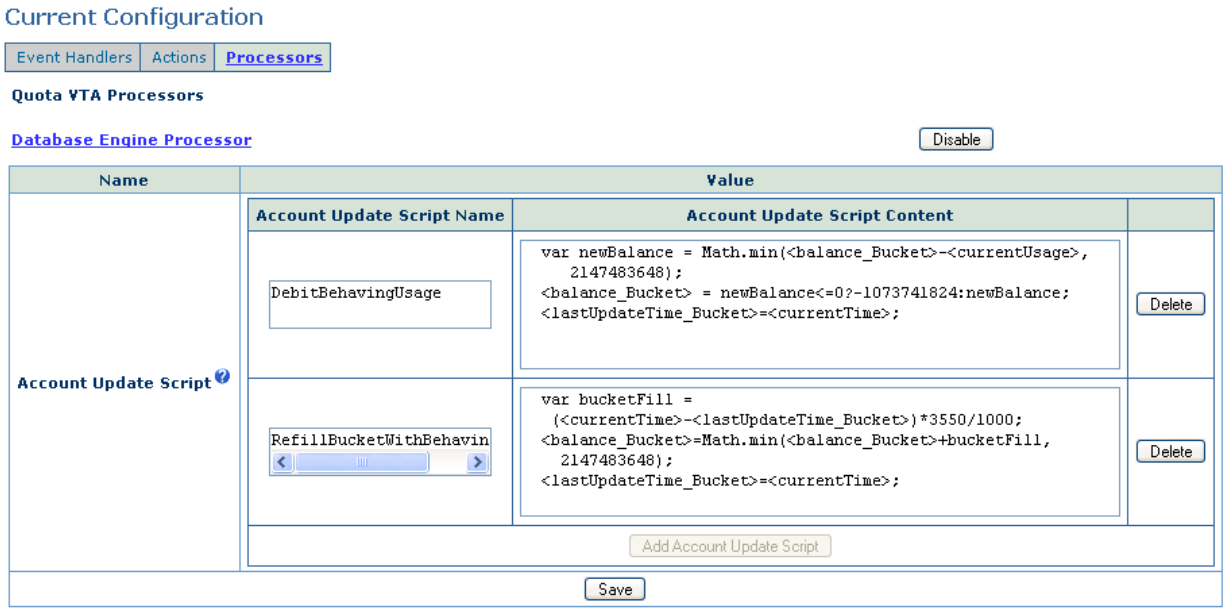
The bucket VTA contains a database engine processor that is configured as follows.

Account Update Scripts

The database engine processor contains two account update scripts:

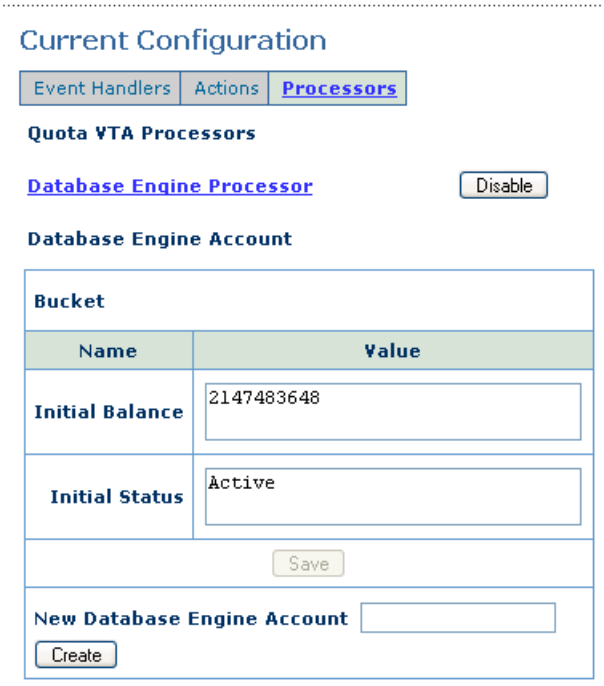
- DebitBehavingUsage—Debits the usage of the behaving service from the bucket account and resets the balance of the bucket account based on the new balance.

- RefillBucketWithBehavingRate—Refills the bucket account with the behaving rate.



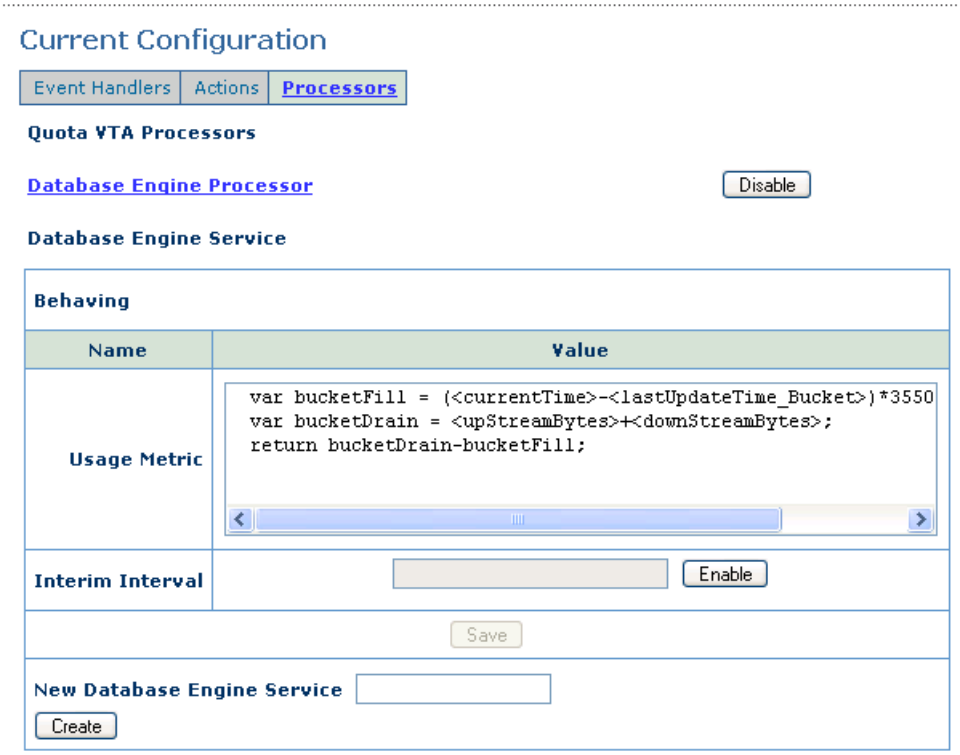
Subscriber Account

The subscriber account sets the initial balance of the account to 2147483648 and sets the initial status of the account to active.



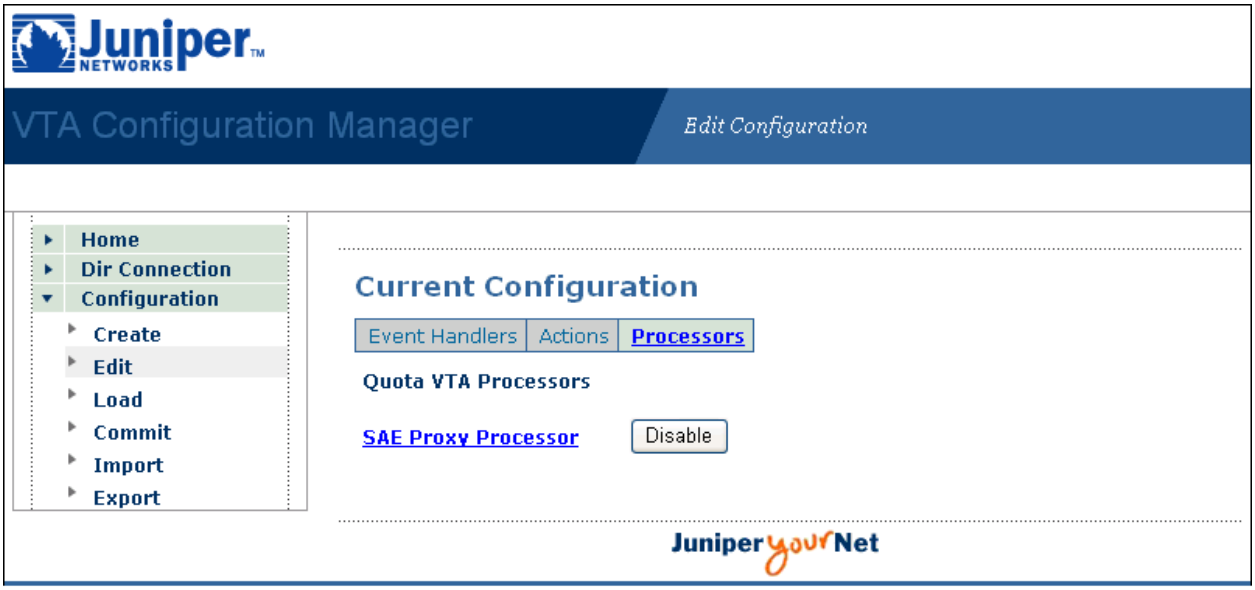
Service Accounts

The service account has the following usage metric:



SAE Proxy Processor for Bucket VTA

The SAE proxy processor is set to enabled.

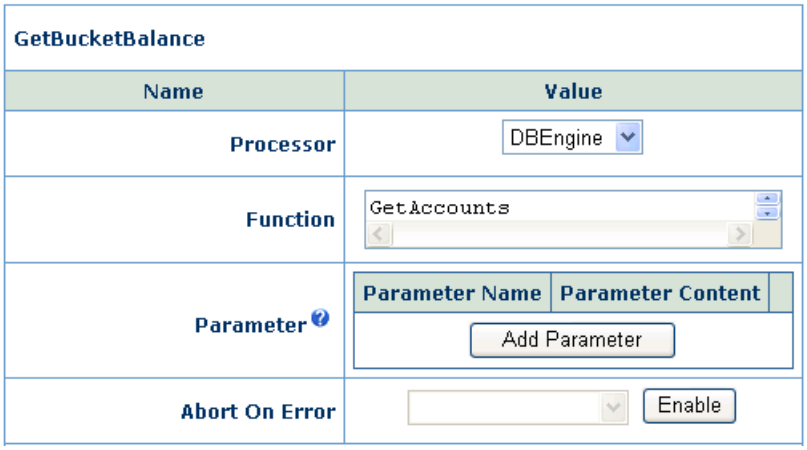


Actions for Bucket VTA

The bucket VTA includes the following actions:

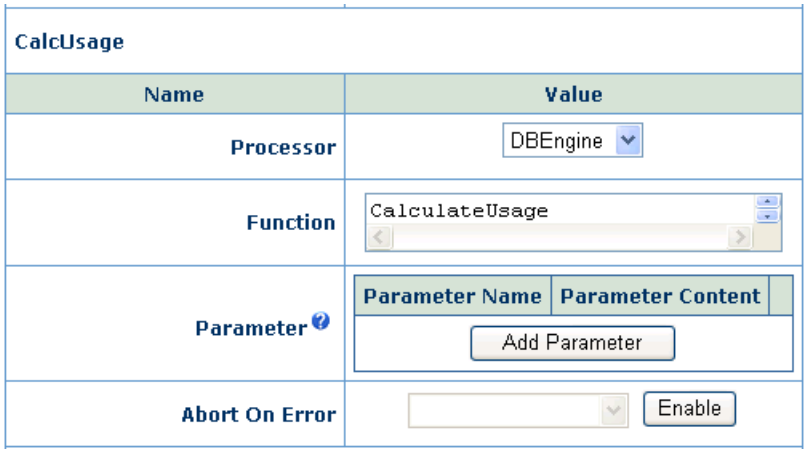
GetBucketBalance Action

The GetBucketBalance action retrieves account balances.



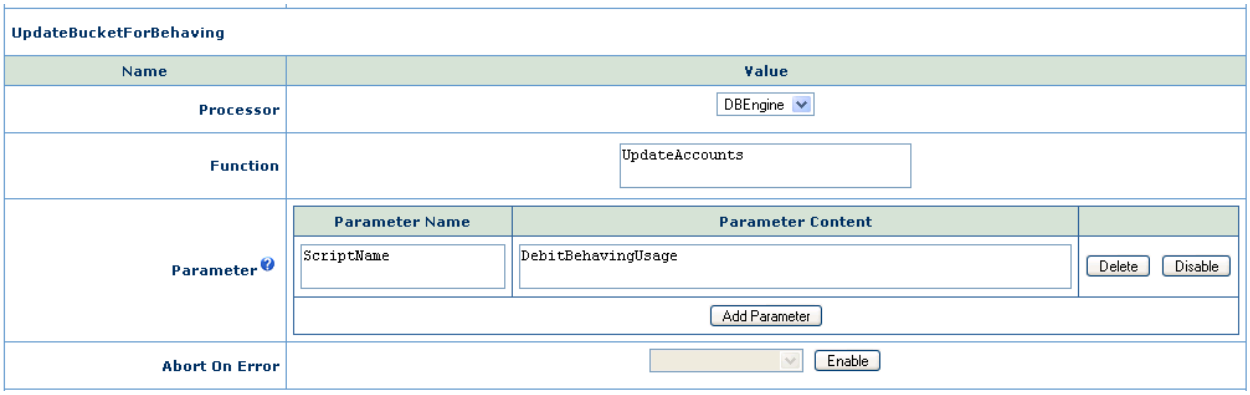
CalcUsage Action

The CalcUsage action calculates usage in the service-tracking event by using the usage metric configured for the service.



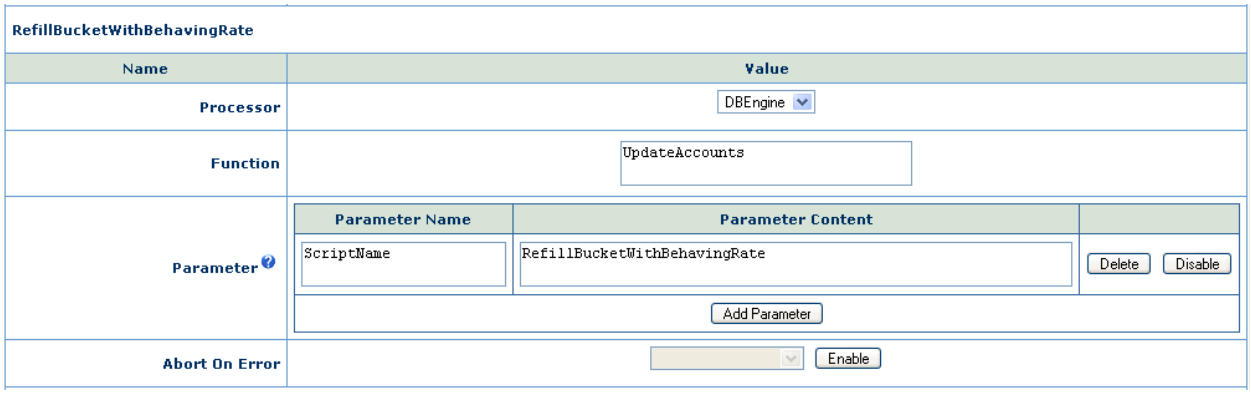
UpdateBucketForBehaving Action

The UpdateBucketForBehaving action debits usage of the behaving service from the bucket account.



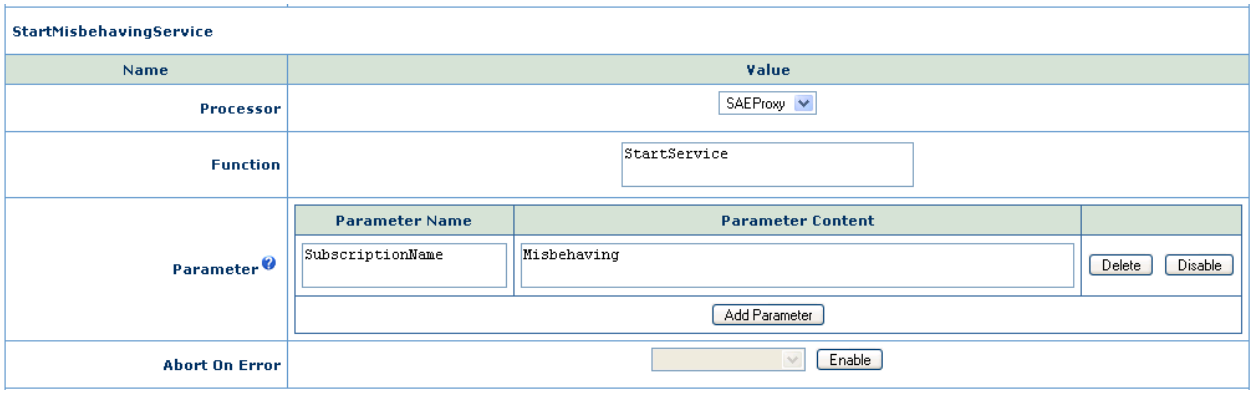
RefillBucketWithBehavingRate Action

The RefillBucketWithBehavingRate action refills the bucket with the behaving rate.



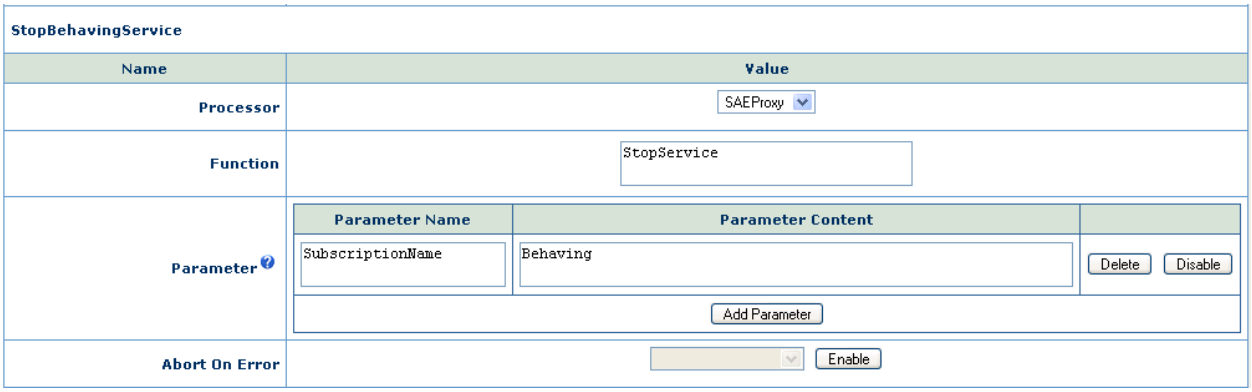
StartMisbehavingService Action

The StartMisbehavingService action starts the misbehaving service.



StopBehavingService Action

The StopBehavingService action stops the behaving service.



Related Topics

- Overview of the SRC VTA on page 65
- How the SRC VTA Works on page 68

- SRC VTA Operation on page 73

PART 5

Index

- Index on page 207

Index

A

admintool command....6
application-level tracking and QoS control. *See* DPI
attacks
  pending service activation, managing....37
  pending service deactivation, managing....38
  requiring action, managing....36
  with activated services, managing....40

C

conventions
  notice icons....xix
  text....xix
customer support....xxi
  contacting JTAC....xxi

D

deep packet inspection. *See* DPI
documentation
  comments on....xxi
DPI (deep packet inspection)
  AOL subscriptions, activating....58
  benefits....47
  collecting accounting data....50
    size limit, setting....57
  configuring
    Ellacoya DPI platform....59
    IPSCS, provisioning....59
    parameters....54
    script services....54
    service logic engine (SLE)....60
    SLE server, IP address or hostname....56
    SRC software....54
    subscriber prefix....58
    subscriptions....59
    virtual router....58
  Ellacoya Networks DPI components....48
    e30 Switch....48
    IP Service Control System (IPSCS)....48
    Remote Method Invocation (RMI) protocol....48
    service logic engine (SLE)....48
  integration overview....48
  loading sample data....53
  overview....47
  script service....48
    JAR files....54
  service activation and deactivation....52
  service bundles....50
    domain....57
    provisioning....59
  service offers....50
    provisioning....59
  subscriber login and logout....52
  subscriber/IP address binding....51
  subscriber/service offer binding....50
  subscribers
    authentication username....57
    identifying....57
    password....57
  synchronization
    SRC software and Ellacoya system....50
    system clocks....61
  traffic-accounting profiles (TAPs)....50, 59
  UQMS (Usage Quota Management System)....48

E

Ellacoya Networks, deep platform inspection. *See* DPI

F

feature sets....5
firewall ports for SRC-related components....8
folders for installed software....5

I

installing
  Solaris packages....5
  SRC ACP....4

Tomcat....5
Web applications....7
IP Service Control System (IPSCS). *See* DPI

J
JBoss
installing Web applications inside....7
removing Web applications from....8

M
manuals
comments on....xxi

N
notice icons....xix

P
packages, Solaris. *See* Solaris packages
patches
Solaris....4
ports
SRC-related components....8

Q
QoS (quality of service)
providing at application level. *See* DPI

R
record servlet for SRC TMP....31
removing
Solaris packages....6
Web applications....8

S
sample data
deep packet inspection (DPI)....53
script service
DPI integration....54
scripts
thm.py....31
service logic engine (SLE). *See* DPI
Solaris packages
installing....5
removing....6
Solaris patches....4
SRC ACP (SRC Admission Control Plug-In)
installing....4
SRC Threat Mitigation Portal. *See* SRC TMP
SRC TMP (SRC Threat Mitigation Portal)....31
configuring....33
deploying....25, 33
managing attacks....31, 34
pending service activation....37
pending service deactivation....38
requiring action....36
with activated services....40
running....34
SRC Volume-Tracking Application. *See* SRC VTA
SRC VTA (SRC Volume-Tracking Application)
accounts
calculating interim interval....129
calculating usage....129
deleting information with administrator portal....186
description....66
getting balances....129
initial balance, setting....115
interim accounting interval, setting....119
managing with portals....65
managing with VTA Configuration Manager....111
purchasing with subscriber portal....187
replenishing with administrator portal....186
service....65
subscriber....65
suspending with subscriber portal....187
updating with actions....126
usage metric, setting....117
viewing with administrator portal....184
actions....71
running scripts....150
updating accounts....126
architecture....68
behaving service....66
bought account....66
bought quota....66
example....78
bucket account....66
example....193
configuring. *See* VTA Configuration Manager
connections to SRC components....68
database engine processor
configuring....111
e-mail notifications, sending....130
event attributes....68

event handlers....68
configuring....156
event queues....95
events....68
account update....70
callback....70
configuring....154
tracking events from SAE....69
example
bucket VTA....193
limiting subscriber access....78
how it works....68
initially configuring....82
installing....82
interval accounting interval, setting....119
JavaScript programs....83
mail processor
configuring....130
misbehaving service....66
network resource usage, determining....119
NIC with the SRC VTA....98
operation process....73
overview....65
periodic account....66
periodic quota....66
example....78
processors....68
providing volume-based services....65
quota service....66
activation upon deposit....98
related configuration tasks....83
deployment descriptors....86
identifying subscribers, SAEs, and sessions....74
J2EE application server....85
J2EE application server client libraries....90
loading configurations from directory....90
NIC....99
NIC proxies....99
services and policies....88
subscribers and subscriptions....89
tracking plug-ins, enterprise subscribers....94
renaming VTAs....100
SAE events....69
SAE proxy processor
configuring....136
script runner processor
configuring....143
scripts
external....143, 147
JavaScript programs....143
running....150
sessions....65
closing....129
subscriber login with IP address....98
testing VTA configuration with administrator portal....186
troubleshooting database deadlocks....88
types....65
usage metric, configuring....117
validating configurations....174
SRC_APLIB.tar.gz file....3
support, technical *See* technical support
swmtool command....6

T

technical support
contacting JTAC....xxi
text conventions defined....xix
threat mitigation
managing attacks with the SRC TMP....34
pending service activation....37
pending service deactivation....38
requiring action....36
with activated services....40
Threat Mitigation Application
database configuration....18
deploying....25
initially configuring....16
installing....16
prerequisites....14
record servlet....31
related configuration tasks....17
sample implementation....15
services in response to incidents....26
thm.py script....31
Threat Mitigation Portal. *See* SRC TMP
threat-mitigation-database-candidate-actions....23
Tomcat, installing....5

U

uninstalling. *See* removing

V

Volume-Tracking Application. *See* SRC VTA

VTA Configuration Manager....104
available configuration tasks....110
committing configurations to directory....175
connecting to directory....106
database accounts....111
exporting configurations to a local file....176
importing configurations from a local file....109
installing....104
loading configurations from directory....106
login name....104
password....104
running....104
validating configurations....174
VTA portals
administrator portal....184
accessing....184
deleting information from VTA database....186
replenishing periodic accounts....186
testing VTA configuration....186
viewing subscriber accounts....184
automatic login....180
configuring properties....180
managing subscriber accounts....179
subscriber portal....187
accessing....187
purchasing extra bandwidth....187
purchasing periodic accounts....187
suspending periodic accounts....187
VTA. *See* SRC VTA

W

WAR files....7
Web applications
installing....7
removing....8