Chapter 8

Using JUNOS Routing Platforms in the SRC Network with a Solaris Platform

This chapter describes how to set up the SRC software on a Solaris platform with the SRC configuration applications that run only on Solaris platforms. It also shows how to set up JUNOS routing platforms so that the routing platforms can be used the SRC network. It includes information about how to monitor the interactions between the SAE and JUNOS routing platforms and how to troubleshoot SRC problems on JUNOS routing platforms.

You can also use the CLI that runs on Solaris platforms and the C-series platform to configure the SRC system to work with JUNOS routing platforms. See *Chapter 7, Using JUNOS Routing Platforms in the SRC Network with the SRC CLI*.

Topics in this chapter include:

- Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 164
- Troubleshooting SRC Problems on JUNOS Routing Platforms on page 164

BEEP Connection Between JUNOS Routing Platforms and the SAE

For information about which JUNOS routing platforms and releases a particular SRC release supports, see the SRC *Release Notes*.

The SAE interacts with a JUNOS software process, referred to as the SRC software process in this documentation, on the JUNOS routing platform. The SAE and the SRC software process communicate using the Blocks Extensible Exchange Protocol (BEEP). You can secure the BEEP connection by using Transport Layer Security (TLS).

When the SRC software process establishes a BEEP session for the SAE, the SAE configures an interface on the JUNOS routing platform. The SAE builds the configuration for an interface using the policies stored in the directory. If the policies are subsequently modified, the SAE builds a new configuration and reconfigures the interface on the JUNOS routing platform.The JUNOS routing platform stores data about interfaces and services that the SAE manages in a configuration group called sdx. You must create this configuration group on the JUNOS routing platform.

Adding JUNOS Routing Platforms and Virtual Routers

On JUNOS routing platforms, the SAE manages interfaces. The SRC software associates a virtual router called default with each JUNOS routing platform. Each JUNOS routing platform in the SRC network and its associated virtual router (VR) called default must appear in the directory. The VRs are not actually configured on the JUNOS routing platform; the VR in the directory provides a way for the SAE to manage the interfaces on the JUNOS routing platform.

There are two ways to add routers to the directory:

- Use SDX Admin to detect operative routers in the SRC network and add them to the directory. This operation creates a VR called default in the directory for each detected JUNOS routing platform.
- Add each router and VR individually. You need to add routers and VRs individually if you use an LDAP client other than SDX Admin or if you want to add inoperative routers.

NOTE: You must define connected SAEs for each router in the virtual router object of the directory. This step is required for the SAE to work with the router. See *Specifying the SAEs That Can Manage the Router* on page 143.

Adding Operative JUNOS Routing Platforms

To add routers that are currently operative and have an operating SNMP agent:

1. In the SDX Admin navigation pane, select **o = Network**, and right-click.
2. Select **Discover Network**.

   The Discover Network dialog box appears.
3. Enter the IP address, the prefix of the network, and the SNMP community string.
4. Click **OK**.

   For each JUNOS routing platform, the software creates one VR called default. You can modify the configuration of these objects. For information about configuring these objects, see *Adding Routers Individually* on page 137 and *Adding Virtual Routers Individually* on page 139.

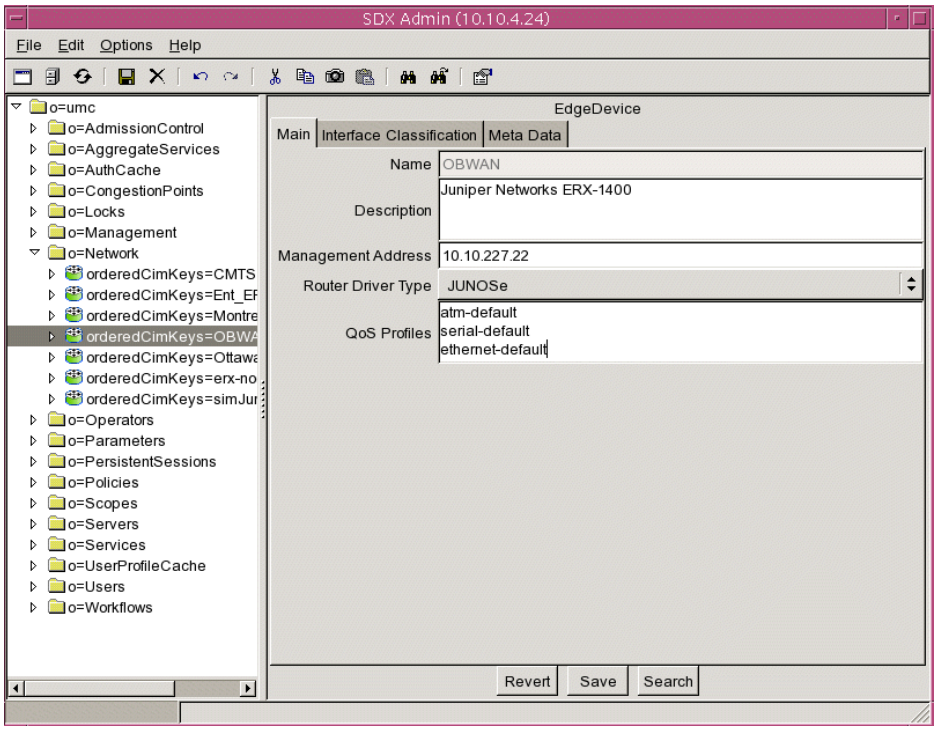
Adding Routers Individually

To add a single router with SDX Admin:

1. In the navigation pane, right-click the Network folder, and select **New > EdgeDevice**.

   The New EdgeDevice dialog box appears.
2. Enter the name of the router exactly as it is configured in the JUNOS software, and click **OK**.

   The new device appears in the navigation pane, and the Main tab of the EdgeDevice pane appears.



3. Edit or accept the default values for the router fields.

   See *Router Fields* on page 138.

4. Click **Save**.

Router Fields

In SDX Admin, you can modify the following fields in the content pane for a router (*orderedCimKeys = < EdgeDeviceName >, o = network, o = umc)*.

Description

- Information about this device; keywords that the SRC find utility uses.
- Value—Text string
- Example—ERX-1400 router located in Ottawa

Management Address

- IP address of the router or CMTS device. If you add a router using the discover network feature, the software automatically adds the IP address of the first SNMP agent on the router to respond to the discover request.
- Value—IP address
- Example—192.0.1.1

Router Driver Type

- Type of device that this directory object will be used to manage.
- Value
  - JUNOSe—JUNOSe router
  - JUNOS—JUNOS routing platform
  - PCMM—CMTS device
- Default—No value

QoS Profiles

- For JUNOSe routers, specifies quality of service (QoS) profiles that are configured on the router.
- Value—List of QoS profiles on separate lines
- Guideline—This field applies to JUNOSe routers only
- Example—atm-default

Adding Virtual Routers Individually

To add a VR with SDX Admin:

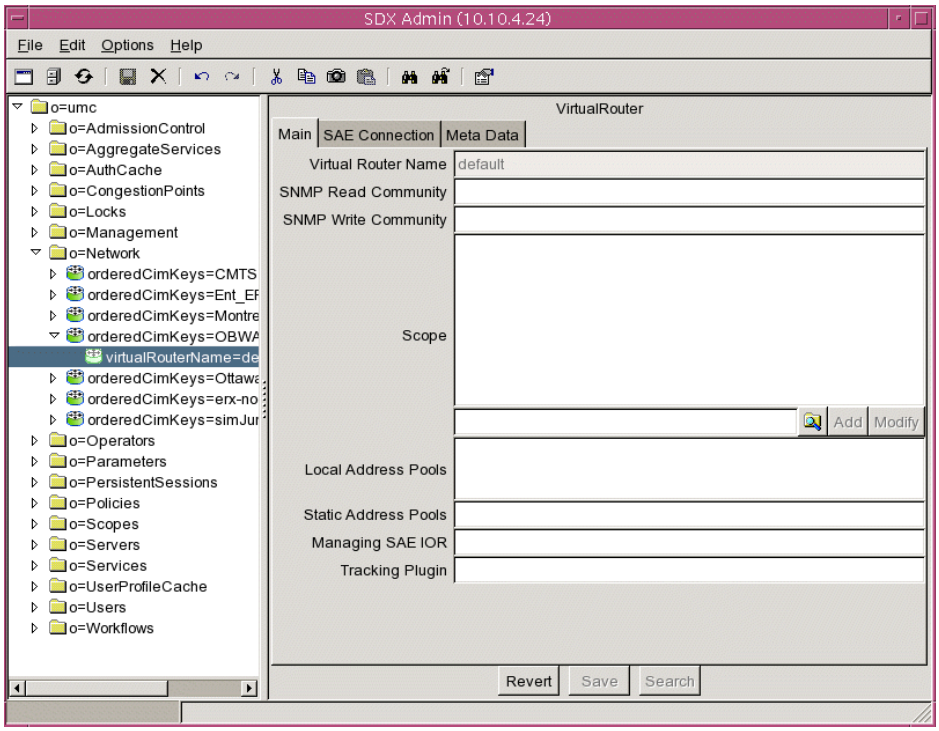
1. In the navigation pane, right-click the device to which you want to add the VR, and select **New > VirtualRouter**.

   The New VirtualRouter dialog box appears.

2. Enter the name of the VR, and click **OK**.

   - For JUNOSe routers, the name of the VR, which is case sensitive, must exactly match the name of the VR configured on the router.
   - For JUNOS routing platforms and CMTS devices, use the name default.

   The new VR appears in the navigation pane, and the Main tab of the VirtualRouter pane appears.



3. Enter or accept the default values for the virtual router fields.

   See *Virtual Router Fields* on page 140.

4. Select the **SAE Connection** tab in the VirtualRouter pane, and add SAEs that are connected to the router. See *Specifying the SAEs That Can Manage the Router* on page 143.

NOTE: This step is required for the SAE to work with the router.

5. Click **Save**.

Virtual Router Fields

In SDX Admin, you can modify the following fields in the content pane for a virtual router (*virtualRouterName* = < *virtualRouterName* > , *orderedCimKeys* = < *EdgeDeviceName* > , *o* = *network, o* = *umc*).

SNMP Read Community

- SNMP community name associated with SNMP read-only operations for this VR.
- Value—Text string
- Example—admin

SNMP Write Community

- SNMP community name associated with SNMP write operations for this VR.
- Value—Text string
- Example—public

Scope

- Service scopes assigned to this VR.
- Value—Text string
- Example—POP-Westford

Local Address Pools

- List of IP address pools that a JUNOSe VR currently manages and stores.
- Value—You can specify an unlimited number of ranges of local IP address pools for JUNOSe VRs. You can specify either the first and last addresses in a range or the first IP address and a factor that indicates the start of the range. You can also specify IP addresses to exclude. Use spaces in the syntax only to separate the first and last explicit IP addresses in a range.

  The IP pool syntax has the format:

  ([<ipAddressStart> <ipAddressEnd>] |
  {<ipBaseAddress>/(<mask> | <digitNumber>)(,<ipAddressExclude>)*})

  where:

  - < ipAddressStart > —First IP address (version 4 or 6) in a range
  - < ipAddressEnd > —Last IP address (version 4 or 6) in a range
  - < ipBaseAddress > —Network base address
  - < mask > —IP address mask
  - < digitNumber > —Integer specifying the number of significant digits of the first IP address in the range
  - < ipAddressExclude > —List of IP addresses to be excluded from the range
  - |—Choice of expression; choose either the expression to the left or the expression to the right of this symbol
  - *—Zero or more instances of the preceding group
- Guidelines—Configure this field on JUNOSe VRs only. If you do not configure the **PoolPublisher** router initialization scripts for a JUNOSe router, configure this field for the JUNOSe VR.
- Default—No value

- Example—This example shows four ranges for the IP address pool.

  ([10.10.10.5 10.10.10.250]
  {10.20.20.0/24}
  {10.21.0.0/255.255.0.0}
  {10.20.30.0/24,10.20.30.1})

  - The first range (a simple range) specifies all the IP addresses between the two IP addresses 10.10.10.5 and 10.10.10.250.
  - The second range specifies all the IP addresses in the range 10.20.20.0 to 10.20.20.255.
  - The third range uses a network mask to specify all the IP addresses in the range 10.21.0.0 to 10.21.255.255.
  - The fourth range specifies all the addresses of the network 10.20.30.0 to 10.20.30.255, excluding the address 10.20.30.1.

Static Address Pools

- List of IP address pools that a JUNOSe VR manages but does not store. You can configure these address pools only in the SRC software.
- Value—See the field Local Address Pools.
- Guidelines—Configure this field on JUNOSe and CMTS VRs only.
- Default—No value
- Example—([10.10.10.5 10.10.10.250] {10.20.20.0/24})

Managing SAE IOR

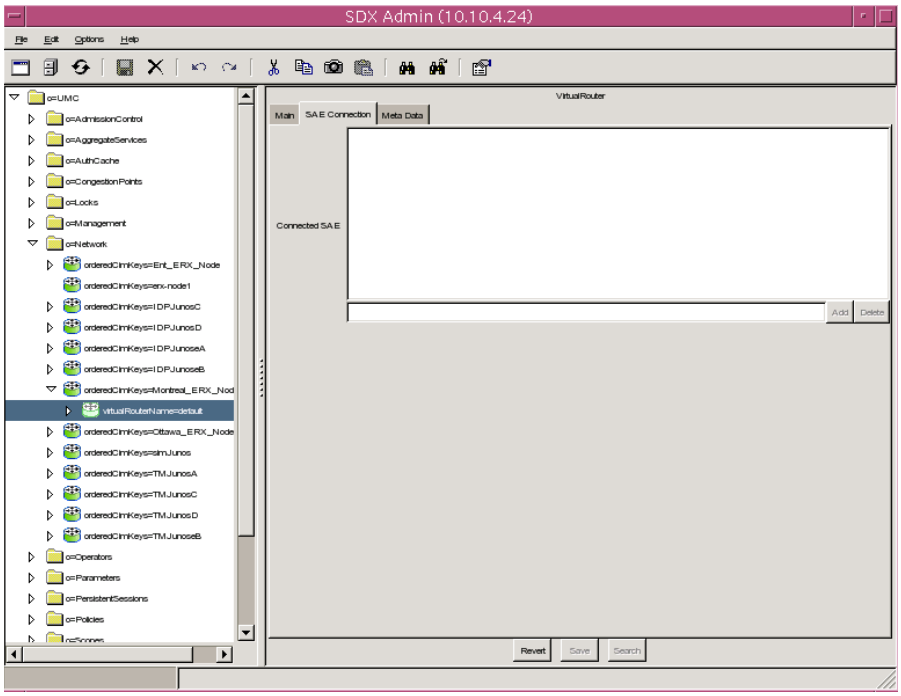
- Common Object Request Broker Architecture (CORBA) reference for the SAE managing this VR.
- Value—One of the following items:
  - The actual CORBA reference for the SAE
  - The absolute path to the interoperable object reference (IOR) file
  - A corbaloc URL in the form corbaloc:: < host > :8801/SAE
    - < host > is the name or IP address of the SAE host.
- Default—No value
- Guidelines—The **PoolPublisher** and **IorPublisher** router initialization scripts provide this information when the router connects to the SAE. If you do not select one of these router initialization scripts, enter a value in this field.
- Example—One of the following items:
  - Absolute path—*/opt/UMC/sae/var/run/sae.ior*
  - corbaloc URL—corbaloc::boston:8801/SAE
  - Actual IOR—IOR:000000000000002438444C3A736D67742E6A756E697...

Tracking Plug-in

- Plug-ins that track interfaces that the SAE manages on this VR. The SAE calls these plug-in instances for every interface it manages. The SAE calls these plug-ins after an interface comes up, when new policies are installed on the interface, and when the interface goes down.
- Value—Comma-separated list of plug-in instances
- Guidelines—Enter plug-in instances and network information collector (NIC) SAE plug-in agents that are specific to this VR.
- Default—No value
- Example—nicsae, flexRadius

Specifying the SAEs That Can Manage the Router

You must add the addresses of SAEs that can manage this router. This step is required for the SAE to work with the router. To add the SAEs, select the SAE Connection tab in the VirtualRouter pane.



Adding an SAE

To add an SAE:

1. Type the IP address of the SAE in the field below the Connected SAE box.
2. Click **Add**.

Modifying an SAE Address

To modify an SAE address:

1. Click the IP address of the SAE in the Connected SAE box.
2. Modify the IP address in the field below the Connected SAE box.
3. Click **Modify**.

Deleting an SAE Address

To delete an SAE address:

1. Click the IP address of the SAE in the Connected SAE box.
2. Remove the IP address from the field below the Connected SAE box.
3. Click **Delete**.

Connected SAE

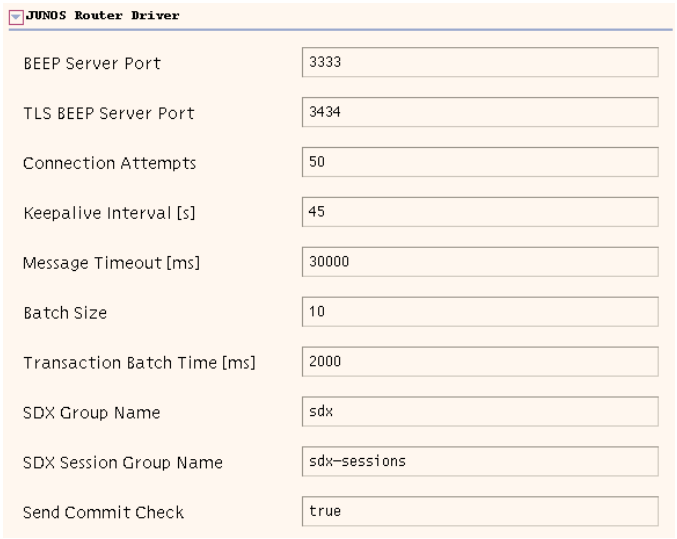
- SAEs that are connected to the router or CMTS device.
- Value—IP addresses
- Default—No value

Configuring the SAE to Manage JUNOS Routing Platforms

A JUNOS routing platform interacts with the SAE by using a JUNOS software process called sdxd. When the sdxd process establishes a TCP/IP connection to the SAE, the SAE begins to manage the router. The JUNOS router driver configuration defines parameters related to the interactions between the SAE and the sdxd process.

To use SDX Configuration Editor to configure a JUNOS router driver:

1. In the navigation pane, select a configuration file for the SAE that you want to configure.
2. Select the Router tab, and expand the **JUNOS Router Drive**r section.



3. Edit or accept the default values in the fields.

   See *JUNOS Router Driver Fields* on page 145.

4. Select **File > Save**.

5. Right-click the configuration file, select **SDX System Configuration > Export to LDAP Directory**.

JUNOS Router Driver Fields

In SDX Configuration Editor, you can edit the following fields in the JUNOS Router Driver section of the Router pane in an SAE configuration file.

BEEP Server Port

- TCP port number that is used to communicate with the sdxd process on JUNOS routing platforms. This port number must match the port number configured in the sdxd process on the router.
- Value—TCP port number; if this value is set to zero and the TLS BEEP server port is set, only TLS connections will be accepted.
- Guidelines—If you change this port number, you need to restart the SAE before the change takes effect.
- Default—3333
- Property name—Router.junos.server_port

TLS BEEP Server Port

- TLS port number that is used for TLS connections to the JUNOS routing platform.

  See *Configuring Secure Connections Between the SAE and JUNOS Routing Platforms* on page 148.
- Value—TLS port number. If the number is set to 0, the SAE does not accept TLS connections.
- Guidelines—If you change this port number, you need to restart the SAE before the change takes effect.
- Default—3434
- Property name—Router.junos.server_tls_port

Connection Attempts

- Number of socket connection attempts that are accepted while the SAE creates sockets before new attempts are dropped.
- Value—Positive value greater than 0; if the value is equal to or less than 0, the default value is used
- Default—50
- Property name—Router.junos.backlog_connections

Keepalive Interval [s]

- Interval between keepalive messages sent from the router. The sdxd process on the router monitors the connection to the SAE by sending keepalive messages at one-third the specified interval. If the sdxd process does not receive the expected keepalive answer within the specified timeout, it closes the connection. A short interval results in a high load on the BEEP interface. A long interval results in a long time before a connection failure is detected.
- Value—Number of seconds in the range 0–2147483647. A value of 0 means that timeout is disabled.
- Default—45
- Property name—Router.junos.keepalive

Message Timeout [ms]

- Amount of time that the router driver waits for a response from the sdxd process. Under a high load the router may not be able to respond fast enough to requests. Change this value only if a high number of timeout events appear in the error log.
- Value—Number of milliseconds in the range 0–2147483647
- Default—30000
- Property name—Router.junos.message_timeout

Batch Size

- Minimum number of service configuration transactions that are committed at the same time. If any of the transactions in a batch fails, all transactions are aborted, and the associated service activations or deactivations fail.
- Value—Integer in the range 0–2147483647
- Guidelines—To control maximum latency for a job when services are activated in parallel, specify 120% of the number of CORBA threads as the batch size.
- Default—10
- Property name—Router.junos.batch_size

Transaction Batch Time [ms]

- Maximum time to collect configuration transactions in a batch. The batch is completed if either the batch size or the batch time is reached.
- Value—Number of milliseconds in the range 0–2147483647
- Guidelines—The completion time is calculated from the creation of a batch. Note that the batch time is a function of the total configuration size and not of the number of commands in the configuration transactions.
- Default—2000
- Property name—Router.junos.batch_time

SDX Group Name

- Name of group on the JUNOS routing platform in which provisioning objects are stored.
- Value—Name configured on the JUNOS routing platform
- Default—sdx
- Property name—Router.junos.group.config

SDX Session Group Name

- Name of group on the JUNOS routing platform in which session objects are stored.
- Value—Name configured on the JUNOS routing platform
- Default—sdx-sessions
- Property name—Router.junos.group.session

Send Commit Check

- Enables or disables commit check. If enabled, a more detailed error message is logged if a batch fails, which lets you verify individual transactions in a batch.
- Value—True or false
- Guidelines—To maximize service activation performance, commit check should be disabled.
- Default—True
- Property name—Router.junos.send_commit_check

Configuring Secure Connections Between the SAE and JUNOS Routing Platforms

You can use TLS to protect communication between the SAE and JUNOS routing platforms.

To complete the handshaking protocol for the TLS connection, the client (JUNOS routing platform) and the server (SAE) must exchange and verify certificates. You need to create a client certificate and a server certificate. Both certificates must be signed by a certificate authority (CA). JUNOS software supports VeriSign, Inc. (http://www.verisign.com). You must then install both certificates on the SAE and on the JUNOS routing platform.

To set up the SAE and the JUNOS routing platform to use TLS, perform the following tasks:

1. Creating a Server Certificate for the SAE on page 148
2. Installing the Server Certificate on the SAE on page 149
3. Installing the Server Certificate on the Router on page 150
4. Creating a Client Certificate for the Router on page 150
5. Installing the Client Certificate on the Router on page 150
6. Installing the Client Certificate on the SAE on page 151
7. Configuring the SAE to Use TLS on page 151
8. Configuring the Keystore for TLS Certificates and Keys on page 151

Creating a Server Certificate for the SAE

The SRC software provides a sample security certificate that you must replace with a real one. You can obtain a signed certificate from a CA. The SAE stores certificates in a keystore, which is a database of keys and certificates from trusted entities.

To remove the sample certificate and create a site certificate:

1. Access the SAE installation directory.

   cd /opt/UMC/sae

2. Remove the sample certificate.

   rm -f lib/jetty/saeKeystore

3. Generate a self-signed certificate using the **keytool** command; for example:

   /opt/UMC/jre/bin/keytool -genkey -keyalg RSA -keystore keystore/keystore.jks -keypass router -storepass router -alias sae -dname <DN> -validity 365

The values specified for the **-keystore**, **-keypass**, **-storepass**, and **-alias** arguments must match the following values that you configure for the keystore on the SAE:

- The value of the **-keystore** argument must match the value of the Keystore Location field.
- The value of the **-keypass** and **-storepass** arguments must both match the value of the Keystore Password field.

See *Configuring the Keystore for TLS Certificates and Keys* on page 151.

Replace < DN > with the distinguished name that identifies your HTTPS server. For example, if XYM Corp in Canada has an HTTPS server with a hostname of ssp1.domain.org, then the DN might be:

"cn=ssp1.domain.org, o=XYM Corp, c=CA"

Be sure to include the quotation marks. Do not use the "#" character in DNs.

For complete documentation of the Java **keytool**, see:

http://java.sun.com/j2se/1.4.1/docs/tooldocs/solaris/keytool.html

4. Create a certificate signing request (CSR).

   /opt/UMC/jre/bin/keytool -certreq -alias sae -file server.csr -keypass router -keystore keystore/keystore.jks -storepass router

   The command creates a CSR and places it in the *server.csr* file.

5. Send the CSR from the file */opt/UMC/sae/server.csr* for signing to VeriSign, Inc. (http://www.verisign.com).

   VeriSign authenticates you and returns a certificate, signed by them, that authenticates your public key.

Installing the Server Certificate on the SAE

To install the server certificate on the SAE, import the server certificate into the SAE keystore using the **keytool** command:

/opt/UMC/jre/bin/keytool -import -alias sae -file server.crt -keypass router -noprompt -trustcacerts -keystore keystore/keystore.jks -storepass router

Installing the Server Certificate on the Router

The TLS client (JUNOS routing platform) needs a copy of the certificate that was used to sign the SAE certificate so that it can verify the SAE certificate. To install the SAE certificate on the JUNOS routing platform:

1. Include the following statements at the [edit security certificates certificate-authority] hierarchy level.

```
[edit security certificates certificate-authority]
security{
    certificates{
        certificate-authority SAECert{
            File /var/db/certs/cert.pem
        }
    }
}
```

2. Include the following statements at the [system services service-deployment] hierarchy level.

```
system{
    services{
      service-Deployment{
          servers {
            server-address port port-number{
              Security-options {
                tls;
              }
            }
          }
      }
    }
}
```

Creating a Client Certificate for the Router

For information about how to obtain a certificate for the router from a certificate authority, see *Obtaining a Certificate from a Certificate Authority* in the *JUNOS System Basics Configuration Guide*.

Installing the Client Certificate on the Router

To install the client (router) certificate on the JUNOS routing platform:

1. Include the following statements at the [edit security certificates certificate-authority] hierarchy level.

```
[edit security certificates certificate-authority]
security{
    certificates{
          local clientCERT { .... } ;
    }
}
```

2. Include the following statements at the [system services service-deployment] hierarchy level.

```
system{
    services{
        service-Deployment{
            local-certificate clientCert;
        }
    }
}
```

Installing the Client Certificate on the SAE

To install the client certificate on the SAE, you must import the client (router) certificate to the SAE keystore using the **keytool** command. For example:

/opt/UMC/jre/bin/keytool -import -alias router -file client.crt -keypass router -noprompt -trustcacerts -keystore keystore/keystore.jks -storepass router

Configuring the SAE to Use TLS

To configure the SAE to accept TLS connections, enter a port number in the TLS BEEP Server Port field in the JUNOS router driver configuration.

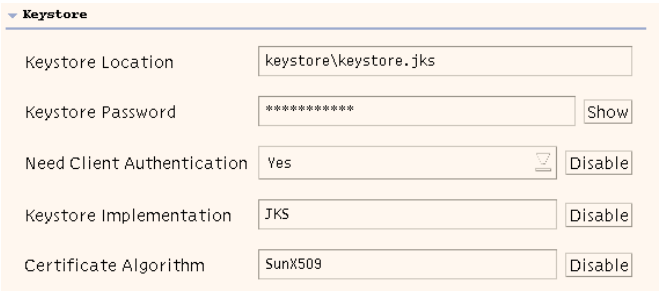
See *Configuring the SAE to Manage JUNOS Routing Platforms* on page 144.

Configuring the Keystore for TLS Certificates and Keys

A keystore is a database of keys and certificates from trusted entities. To use SDX Configuration Editor to configure the TLS keystore on the SAE:

1. In the navigation pane, select a configuration file for the SAE that you want to configure.

2. Select the **Router** tab, expand the **JUNOS Router Driver** section, and then expand the **Keystore** section.



3. Edit or accept the default values in the fields.

See *Keystore Fields for the JUNOS Router Driver* on page 152.

4. Select **File > Save**.

5. Right-click the configuration file, select **SDX System Configuration > Export to LDAP Directory**.

Keystore Fields for the JUNOS Router Driver

In SDX Configuration Editor, you can edit the Keystore fields in the JUNOS Router Driver section in the Router pane in an SAE configuration file.

Keystore Location

- Location of the keystore that contains the key/certificate pair that the SAE sends to the router. If the SAE requires client authentication, it also specifies the location of the CA certificate that was used to sign the certificate that the router sends to the SAE.
- Value—Path and name of the keystore
- Guidelines—The value of this field must match the value of the **-keystore** argument that you entered with the **keytool** command when you created the server certificate for the SAE.

  See *Creating a Server Certificate for the SAE* on page 148.
- Default—keystore\keystore.jks
- Property name—Router.junos.keystore.location

Keystore Password

- Password required for the keystore.
- Value—Password; must be at least six characters
- Guidelines—The value of this field must match the value of the **-keypass** and **-storepass** arguments that you entered with the **keytool** command when you created the server certificate for the SAE.

  See *Creating a Server Certificate for the SAE* on page 148.
- Default—No value
- Example—{BASE64}c2FlS2V5c3RvcmU=
- Property name—Router.junos.keystore.storePass

Need Client Authentication

- Specifies whether or not the SAE requests a client certificate from the router.
- Value
  - Yes—The SAE asks the router for a client certificate when a connection to the router is established.
  - No—The SAE does not ask the router for a client certificate when a connection to the router is established.
- Default—Yes
- Property name—Router.junos.keystore.needClientAuth

Keystore Implementation

- Implementation type of the keystore.
- Value
  - JKS (JKS is the standard Java keystore implementation)
  - PKCS12 (Public Key Cryptography Standard #12)
- Default—JKS
- Property name—Router.junos.keystore.type

Certificate Algorithm

- Implementation type of the certificates contained in the keystore.
- Value—SUN509
- Default—SUNX509, which is the type defined in X.509 the ITU-T standard for Public Key Infrastructure (PKI).
- Property name—Router.junos.keystore.certType

Checking Changes to the JUNOS Configuration

The SAE can check the configuration of a JUNOS routing platform under its control to detect whether the configuration has changed by a means other than through the SAE. If the SAE finds a disparity between the router and the SAE configurations, it can take several actions. The SAE checks the configuration installed on the router against the state of the SAE session layer (subscriber, service, and interface sessions). While the check is occurring, the SAE does not handle jobs from the router, and all provisioning activity is blocked, including event notifications.

The SAE can take the following actions if it finds a disparity between the router and SAE configurations:

- Remove the disparate sessions from the router. When the SAE removes a session, it generates Stop events for the session and removes the session from the session store and the SAE.

- Re-create the sessions that have been removed. Subscribers whose sessions have been removed need to log back in before they can activate services. During session re-creation, the SAE responds to event notifications and provisioning operations.

  If the state of the router configuration is lost because of a failover or a restart, it is not possible to re-create the sessions.

- Report disparities to the operator without making any changes to the router configuration.

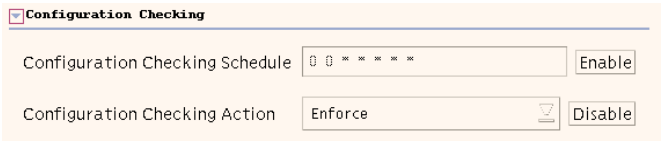
  The disparities are reported through the SAE router driver event trap called routerConfOutOfSynch and through the info log.

Note that it is not possible to check the consistency of individual provisioning objects. Therefore, modifications to a provisioning object while the SAE is disconnected from the router cannot be detected.

Setting Up Periodic Configuration Checking

To use SDX Configuration Editor to configure the SAE to periodically check the configuration of the JUNOS routing platform:

1. In the navigation pane, select a configuration file for the SAE that you want to configure.
2. Select the **Router** tab, expand the **JUNOS Router Driver** section, and then expand the **Configuration Checking** section.



3. Edit or accept the default values in the fields.

   See *Configuration Checking Fields for the JUNOS Router Driver* on page 154.

4. Select **File > Save**.
5. Right-click the configuration file, select **SDX System Configuration > Export to LDAP Directory**.

Configuration Checking Fields for the JUNOS Router Driver

In SDX Configuration Editor, you can modify the Configuration Checking fields in the JUNOS Router Driver section of the Router pane in an SAE configuration file.

Configuration Checking Schedule

- Specifies when the SAE checks the router configuration.
- Value—The schedule format is modeled on the UNIX crontab Entry Format (see UNIX crontab man pages). It consists of seven fields separated by space or tabs. The fields specify:
  - Minute (0-59)
  - Hour (0-23)
  - Day of month (1-31, or the first three letters of the day of month)
  - Month of the year (1-12)
  - Day of the week (0-6 with 0 = Sunday, or the first three letters of the name of the day)
  - Year (4 digits indicating the year)
  - Time Zone ID: An * indicates the SAE local time zone. For custom time zones, specify the format:
    - zone = “GMT” (“+” | “-“) (hour : minute | hour minute | hour)
    - hour = digit digit

    - minute = digit digit
    - digit = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
- Guidelines
    - An asterisk (*) is interpreted as 0 for minutes and hours and as the SAE local time zone for time zone. For all other fields, it stands for "first-last."
    - Ranges of numbers and names are allowed. Ranges are two values separated with a hyphen. The specified range is inclusive. For example, 1-5 for the hour field specifies checking at hours 1, 2, 3, 4, and 5.
    - Lists are allowed. A list is a set of numbers (or ranges) separated by commas. Examples: "1,2,5,9", "0-4,8-12".
    - Step values can be used with ranges. Following a range with "/<number>" specifies skips in the number's value through the range. For example, "0-23/2" in the hours field specifies event execution every other hour. Steps are also permitted after an asterisk, so "*/2" to specifies every 2 hours.
    - When determining the next event time based on a specific time pattern, the following rules apply:
        - Seconds and milliseconds are ignored (that is, rounded up to the closest minute).
        - If you set both a day of the month and a day of the week, only the day of month is used.
- Default—No value
- Property name—Router.junos.configcheck_schedule

Configuration Checking Action

- Action that the SAE takes when it detects disparities between the configuration of the SAE and the configuration on the router.
- Value
    - Enforce—Enforces the state of the session layer on the router. The SAE removes all sessions that have disparities and creates new sessions with the same activation parameters as the original ones.
    - Synchronize—Synchronizes the state of the session layer on the router. The SAE removes all sessions that have disparities.
    - Check only—Reports disparities through the SAE router driver event trap called routerConfOutOfSynch and through the info log. The SAE does not make any changes on the router.
- Default—Enforce
- Property name—Router.junos.configcheck_action

Using SNMP to Retrieve Information from JUNOS Routing Platforms

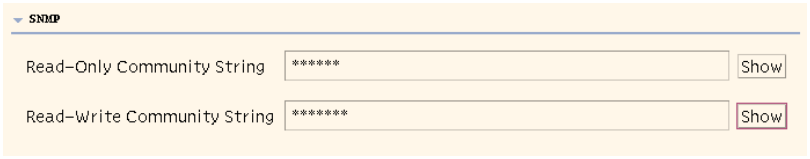
You can use SNMP to retrieve information from the router. For example, if you create a router initialization script that uses SNMP, you need to specify the SNMP communities that are on the router.

We recommend that you specify SNMP communities for each virtual router. (See *Adding Virtual Routers Individually* on page 139.) You can also configure global default SNMP communities.

Configuring Global SNMP Communities in the SRC Software

You can configure global default SNMP communities that are used if a VR does not exist on the router or the community strings have not been configured for the VR. To use SDX Configuration Editor to configure global default SNMP communities:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the **Router** tab, and expand the **SNMP** section.



3. Edit or accept the default values in the fields.

   See *Global SNMP Community Fields* on page 156.
4. Select **File > Save**.
5. Right-click the configuration file, select **SDX System Configuration > Export to LDAP Directory**.

Global SNMP Community Fields

In SDX Configuration Editor, you can edit the following fields in the Router pane in an SAE configuration file.

Read-Only Community String

- Default SNMP community string used for read access to the router.
- Value—SNMP community string that matches a read-only community string configured on the router
- Default—Public
- Property name—Router.read-only.community.string

Read-Write Community String

- Default SNMP community string used for write access to the router.
- Value—SNMP community string that matches a read-write community string configured on the router
- Default—Private
- Property name—Router.read-write.community.string

Developing Router Initialization Scripts

When the SAE establishes a connection with a router, it can run a router initialization script to customize the setup of the connection. Router initialization scripts are run when the connection between a router and the SAE is established and again when the connection is dropped.

We provide the IorPublisher script in the */opt/UMC/sae/lib* folder. The IorPublisher script publishes the IOR of the SAE in the directory so that a NIC can associate a router with an SAE.

Interface Object Fields

Router initialization scripts interact with the SAE through an interface object called Ssp. The SAE exports a number of fields through the interface object to the script and expects the script to provide the entry point to the SAE.

Table 9 describes the fields that the SAE exports.

Table 9: Exported Fields

Ssp Attribute	Description
Ssp.properties	System properties object (class: java.util.Properties)—The properties should be treated as read-only by the script.
Ssp.errorLog	Error logger—Use the SsperrorLog.printIn (message) to send error messages to the log.
Ssp.infoLog	Info logger—Use the Ssp.infoLog.printIn (message) to send informational messages to the log.
Ssp.debugLog	Debug logger—Use the Ssp.debugLog.printIn (message) to send debug messages to the log.

The router initialization script must set the field Ssp.routerInit to a factory function that instantiates a router initialization object:

- < VRName > —Name of the virtual router in which the COPS client has been configured, format: virtualRouterName@RouterName
- < virtualIp > —Virtual IP address of the SAE (string, dotted decimal; for example: 192.168.254.1)
- < realIp > —Real IP address of the SAE (string, dotted decimal; for example, 192.168.1.20)

- < VRIp > —IP address of the virtual router (string, dotted decimal)
- < transportVR > —Name of the virtual router used for routing the COPS connection, or None, if the COPS client is directly connected

The factory function must implement the following interface:

```
Ssp.routerInit(VRName,
virtualIp,
realIp,
VRIp,
transportVR)
```

The factory function returns an interface object that is used to set up and tear down a connection for a given COPS server. A common case of a factory function is the constructor of a class.

The factory function is called directly after a COPS server connection is established. In case of problems, an exception should be raised that leads to the termination of the COPS connection.

Required Methods

Instances of the interface object must implement the following methods:

- *setup()*—Is called when the COPS server connection is established and is operational. In case of problems, an exception should be raised that leads to the termination of the COPS connection.
- *shutdown()*—Is called when the COPS server connection is terminated to the virtual router. This method should not raise any exceptions in case of problems.

Example: Router Initialization Script

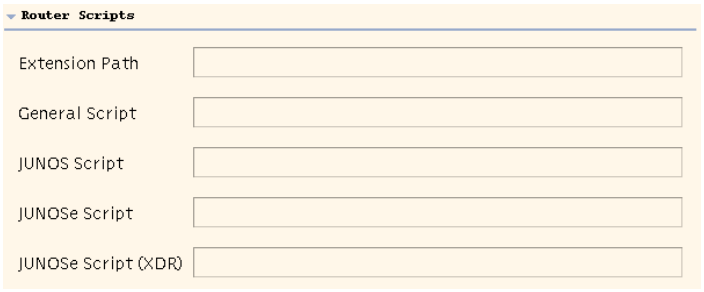
The following script defines a router initialization class named *SillyRouterInit*. The interface class does not implement any useful functionality; it just writes messages to the infoLog when the router connection is created or terminated.

```
class SillyRouterInit:
    def __init__(self, vrName, virtualIp, realIp, vrIp, transportVr):
        """ initialize router initialization object """
        self.vrName = vrName
        Ssp.infoLog.printin("SillyRouterInit created")

    def setup(self):
        """ initialize connection to router """
        Ssp.infoLog.printin("Setup connection to VR %(vrName)s" %
                    vars(self))

    def shutdown(self):
        """ shutdown connection to router """
        Ssp.infoLog.printin("Shutdown connection to VR %(vrName)s" %
                    vars(self))

#
# publish interface object to Ssp core
#
Ssp.routerInit = SillyRouterInit
```

Specifying Router Initialization Scripts on the SAE

To use SDX Configuration Editor to specify router initialization scripts:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the **Router** tab, and expand the **Router Scripts** section.



3. Edit or accept the default values in the appropriate fields.

   See *JUNOS Router Script Fields* on page 159.

4. Select **File > Save**.
5. Right-click the configuration file, select **SDX System Configuration > Export to LDAP Directory**.

JUNOS Router Script Fields

In SDX Configuration Editor, you can edit the following fields in the Router Scripts section of the Router pane in an SAE configuration file.

Extension Path

- Path to router initialization scripts that are not in the default location, */opt/UMC/sae/lib*.
- Value—List of paths separated by semicolons (;)
- Default—No value
- Property name—Extension.path

General Script

- Router initialization script that can be used for all types of routers that the SRC software supports. The script is run when the connection between a router and the SAE is established and again when the connection is dropped.
- Value—Name of a script
- Default—No value
- Property name—Router.script.*

JUNOS Script

- Router initialization script for JUNOS routing platforms. The script is run when the connection between a router and the SAE is established and again when the connection is dropped.
- Value—Name of a script
- Default—iorPublisher
- Property name—Router.script.junos

Accessing the Router CLI

You can access the CLIs of Juniper Networks routers from Policy Editor and from SDX Admin through a Telnet or SSH connection. This access allows you to display and change the configuration of the router.

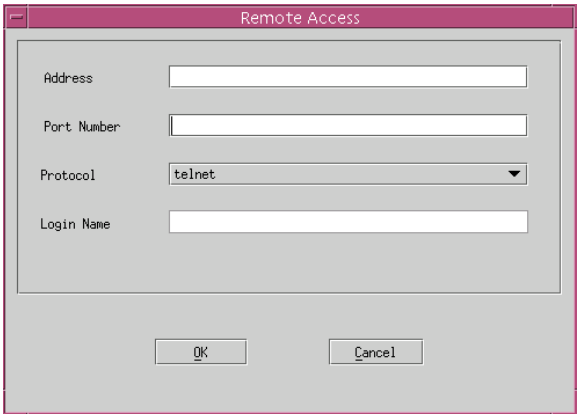
You must have the Telnet or SSH applications installed and available to Policy Editor or SDX Admin. You can open multiple Telnet or SSH sessions.

Using Policy Editor

To access a router from Policy Editor:

1. In the Policy Editor menu, select **Tools > Manage**.

   The Remote Access dialog box appears.



2. Fill in the Remote Access fields, and click **OK**.

   See *Remote Access Fields* on page 161.

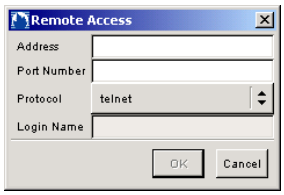
   A Telnet or an SSH window with a CLI prompt appears.

Using SDX Admin

To access a router from SDX Admin:

1. In the navigation pane, expand **o = Network**.
2. Right-click on the router to which you want to connect, and select **Manage**.

   The Remote Access dialog box appears.



3. Fill in the Remote Access fields, and click **OK**.

   See *Remote Access Fields* on page 161.

   A Telnet or an SSH window with a CLI prompt appears.

Remote Access Fields

In Policy Editor, you can edit the following fields in the Remote Access dialog box, in the select Tools > Manage menu.

In SDX Admin, you can edit the following fields in the Remote Access dialog box by right-clicking on the router object, and selecting Manage.

Address

- IP address or hostname of the router.
- Value—IP address
- Default—No value
- Example—192.0.2.1

Port Number

- TCP port over which you want to connect to the router.
- Value—TCP port
- Default—No value
- Example—22

Protocol

- Type of connection
- Value—telnet | ssh

- Default—telnet
- Example—ssh

Login Name

- Login name for SSH connections.
- Value—Text string
- Default—No value
- Guideline—You must enter a value for this property.
- Example—admin

Configuring JUNOS Routing Platforms to Interact with the SAE

To configure the JUNOS routing platform to interact with the SAE:

1. Include the following statements at the [edit system services service-deployment] hierarchy level.

   ```
   [edit system services service-deployment]
   servers server-address {
   port port-number;
   }
   source-address source-address;
   ```

2. Use the following guidelines for the variables in these statements.

server-address

- Specifies the IP address of the host on which you install the SAE.
- Value—IP address
- Guidelines—Be sure this setting matches the corresponding value in the SAE configuration.
- Default—None
- Example—192.0.2.2

port-number

- Specifies the port number for the SAE.
- Value—TCP port number
- Guidelines—Be sure this setting matches the corresponding value in the SAE configuration.
- Default—3333
- Example—3333

source-address

- Specifies the IP address of the source that sends traffic to the SAE.
- Value—IP address
- Guidelines—This setting is optional.
- Default—None
- Example—192.0.2.2

Configuring the JUNOS Routing Platform to Apply Changes It Receives from the SAE

To configure the JUNOS routing platform to receive configuration statements from the SAE and apply those statements to the configuration:

1. Create a configuration group called sdx that contains the configuration statements that the SAE sends to the JUNOS routing platform. To do so, include the groups statement at the [edit] level, and specify the name sdx.

   ```
   [edit]
   groups {
       sdx;
   }
   ```

2. Configure the JUNOS routing platforms to apply these statements to the configuration. To do so, include the apply-groups statement at the [edit] level.

   ```
   [edit]
   set apply-groups sdx;
   ```

Disabling Interactions Between the SAE and JUNOS Routing Platforms

To disable the SRC software process, enter the following command.

```
root@ui1#set system processes service-deployment disable
root@ui1#commit
```

When you disable the SRC software process, it is still available on the JUNOS routing platform.

To reenable the SRC software process, enter the following command.

```
root@ui1#delete system processes service-deployment disable
root@ui1#commit
```

The SRC software process attempts to reconnect the JUNOS routing platform to the SAE.

Monitoring Interactions Between the SAE and JUNOS Routing Platforms

Use the following command on JUNOS routing platforms to monitor the connection between the JUNOS routing platform and the SAE.

```
root@ui1> show system services service-deployment
Connected to 172.17.20.151 port 3333 since 2004-02-06 14:50:31 PST
Keepalive settings: Interval 15 seconds
Keepalives sent: 100, Last sent: 6 seconds ago
Notifications sent: 0
Last update from peer: 00:00:06 ago
```

You can also monitor the interactions between the SRC software and JUNOS routing platforms in the log files for the SAE and in the log files generated by the SRC software process on the JUNOS routing platform. For information about configuring logging for the SAE, see *SRC-PE Monitoring and Troubleshooting Guide, Chapter 4, Configuring Logging for SRC Components on a Solaris Platform*. For information configuring logging on JUNOS routing platforms, see *JUNOS System Basics Configuration Guide*.

Troubleshooting SRC Problems on JUNOS Routing Platforms

To troubleshoot SRC problems on the JUNOS routing platform, review the log files for the SAE and the log files generated by the SRC software process on the router.

- If the log files indicate that the SRC software process is not responding, see *Troubleshooting Problems with the SRC Software Process* on page 164.
- If the log files indicate a problem with a specific interface, see *Troubleshooting Problems with Interfaces* on page 165.
- If the log files indicate a problem with a specific service or its associated firewall rules, see *Troubleshooting Problems with Services* on page 168.

Troubleshooting Problems with the SRC Software Process

If the log files indicate that the SRC software process is not responding:

1. Look at the status of the process on the JUNOS routing platform.

   ```
   root@ui1>show system services service-deployment
   Connected to 172.17.20.151 port 3333 since 2004-02-06 14:50:31 PST
   Keepalive settings: Interval 15 seconds
   Keepalives sent: 100, Last sent: 6 seconds ago
   Notifications sent: 0
   Last update from peer: 00:00:06 ago
   ```

2. If you see the message “error: the service-deployment subsystem is not running,” reenable the SRC software process (see *Disabling Interactions Between the SAE and JUNOS Routing Platforms* on page 163).

3. If the process is already enabled, review the configurations of the router and the SAE in the directory, and fix any problems.

4. Restart the SRC software process on the router.

   root@ui1>**restart service-deployment**

   The SAE synchronizes with the SRC software process and deletes unnecessary data from the router.

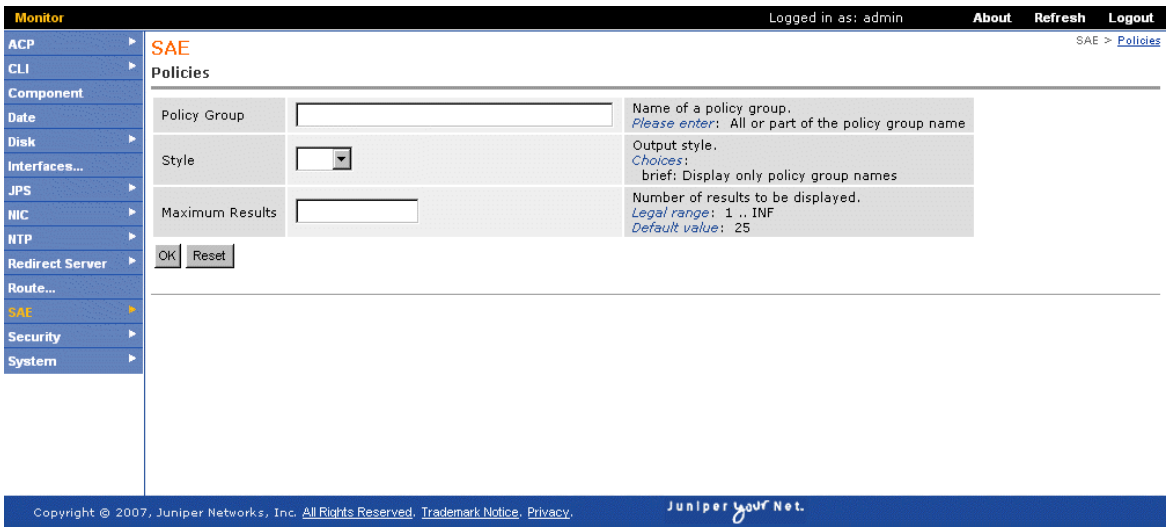
Troubleshooting Problems with Interfaces

If the log files indicate a problem with a specific interface or its associated firewall rules:

1. Review the configuration of the policies associated with the interfaces with the C-Web interface.

   a. Select **SAE** from the side pane, and click **Policies**.

      The Policies pane appears.



      For more information on these fields, see *SRC-PE Monitoring and Troubleshooting Guide, Chapter 14, Monitoring SAE Data with the C-Web Interface*.

   b. Click **Execute**.

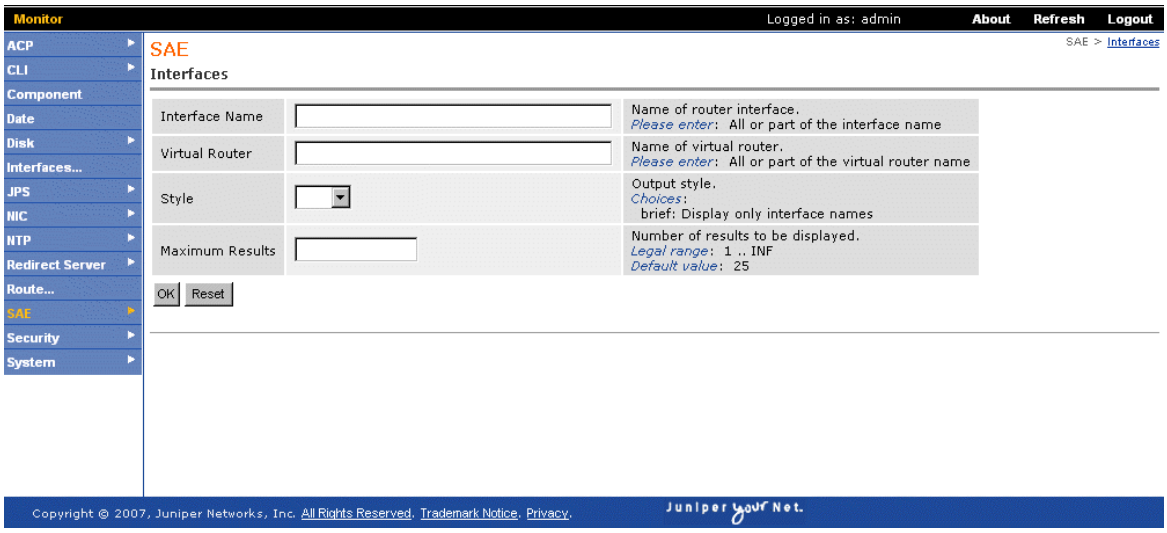
      The Policies pane displays the interfaces available on the router.

   c. If you find any errors, fix the configuration in the directory, and proceed to Step 5. Otherwise, proceed to Step 2.

2. Review the configuration of interfaces on the JUNOS routing platform with the C-Web interface.

   a. Select **SAE** from the side pane, and click **Interfaces**.

      The Information About Router Interfaces pane appears.



   b. Click **Execute**.

      The Information About Router Interfaces pane displays the interfaces available on the router.

   c. If you find any errors, fix the configuration in the directory, and proceed to Step 5. Otherwise, proceed to Step 2.

3. Display the corresponding interfaces on the JUNOS routing platform.

```
root@olive1# show groups sdx interfaces
<fe-0/0/0> {
    unit <0> {
        family inet {
            filter {
                input SDX_PRIVATE_ID00000000000001092282;
                output SDX_PRIVATE_ID00000000000001223352;
            }
        }
    }
}
```

   If you find any errors, fix the configuration in the directory, and proceed to Step 5. Otherwise, proceed to Step 4.

4. Remove the configuration for this interface from the JUNOS routing platform.

   a. Disable the SRC software process.

      root@ui1#s**et system processes service-deployment disable**
      root@ui1#**commit**

   b. Delete the interfaces from the router.

      delete groups sdx interfaces <interfaceName> <interfaceIdentifier>
      root@ui1#commit

      For example, to delete the interface with identifier fe-0/0/0 unit 0, enter:

      root@ui1#**delete groups sdx interfaces <fe-0/0/0> unit <0>**
      root@ui1#**commit**

   c. Reenable the SRC software process.

      root@ui1#**delete system processes service-deployment disable**
      root@ui1#**commit**

5. Restart the SRC software process on the router.

   root@ui1>r**estart service-deployment**

   The SAE reconfigures the interface that you deleted.

6. Review the log files again.

   If the action you took did not fix the problem, return to the last step you performed, and proceed with this troubleshooting procedure. If you have performed all the tasks in the troubleshooting procedure and the problem persists, delete all SRC data on the JUNOS routing platform (see *Deleting All SRC Data on JUNOS Routing Platforms* on page 171*).*

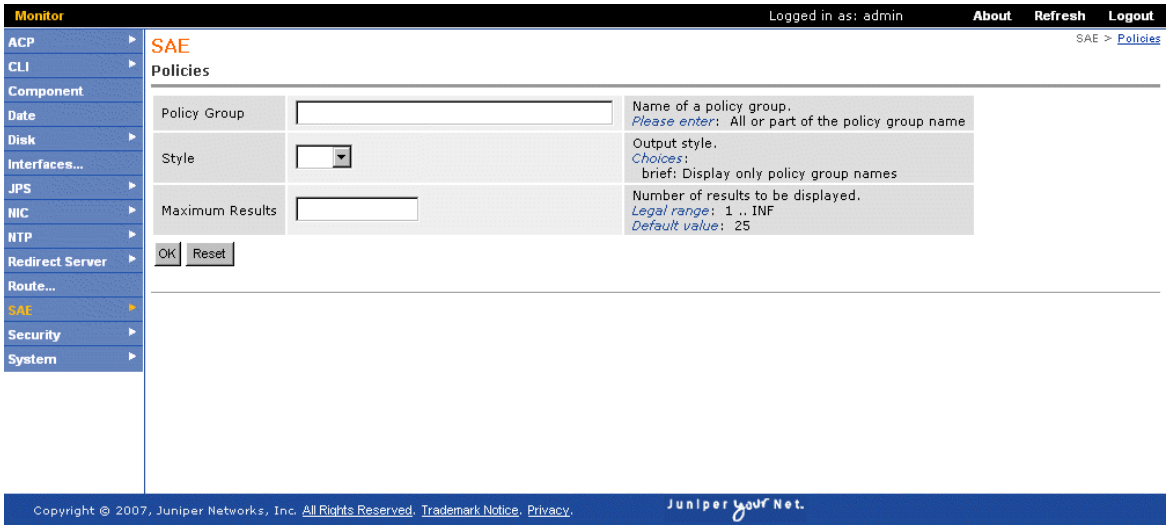
Troubleshooting Problems with Services

If the log files indicate a problem with a specific service or its associated firewall rules:

1. Review the configuration of the policies associated with the interfaces with C-Web.

   a. Select **SAE** from the side pane, and click **Policies**.

      The Policies pane appears.



      For more information on these fields, see *SRC-PE Monitoring and Troubleshooting Guide, Chapter 14, Monitoring SAE Data with the C-Web Interface*.

   b. Click **Execute**.

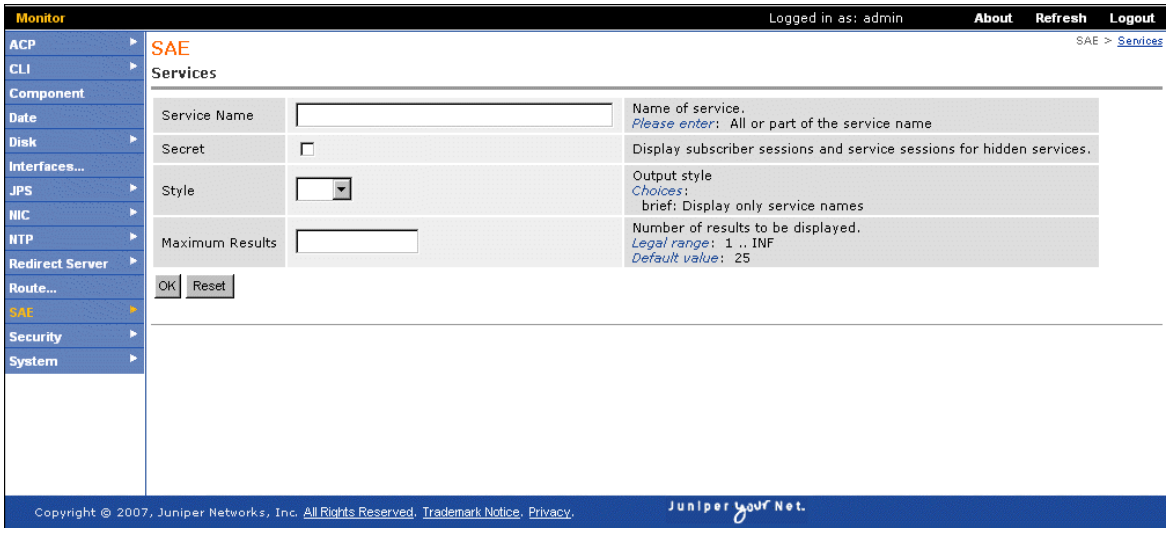
      The Policies pane displays the interfaces available on the router.

   c. If you find any errors, fix the configuration in the directory, and proceed to Step 5. Otherwise, proceed to Step 2.

2. Review the configuration of the service on the JUNOS routing platform with C-Web.

   a. Select **SAE** from the side pane, and click **Services**.

      The State of Services Running on the SAE pane appears.



For more information on these fields, see *SRC-PE Monitoring and Troubleshooting Guide, Chapter 14, Monitoring SAE Data with the C-Web Interface*.

b. Click **Execute**.

   The State of Services Running on the SAE pane displays the interfaces available on the router.

c. Locate an active service session for this service, and observe the ProvisioningSet field of that session.

d. Locate an identifier that is associated with the service that is causing the problem.

   For example, in the above display, the identifier `SDX_PRIVATE_ID0000000000002075317` is associated with a Network Address Translation (NAT) rule.

3. Review the corresponding configuration on the JUNOS routing platform.

```
root@olive1# show groups sdx services nat rule SDX_PRIVATE_ID0000000
000002075317
    match-direction input;
    term SDX_PRIVATE_TERM {
        from {
            source-address {
                0.0.0.0/0;
            }
            destination-address {
                0.0.0.0/0;
            }
        }
        then {
            translated {
                source-pool SDX_PRIVATE_ID0000000000002009780;
                translation-type source dynamic;
            }
        }
    }
```

If you find any errors, fix the configuration in the directory and proceed to Step 5. Otherwise, proceed to Step 4.

4. Remove the configuration for this service from the JUNOS routing platform.

   a. Disable the SRC software process.

   ```
   root@ui1#set system processes service-deployment disable
   root@ui1#commit
   ```

   b. Delete the service on the JUNOS routing platform.

   ```
   delete groups sdx services <serviceName> <filterID>
   root@ui1#commit
   ```

   For example, to delete a firewall filter of the service called firewall with filterID SDX_PRIVATE_ID0000000000001223352, enter:

   ```
   delete groups sdx services firewall filter
   SDX_PRIVATE_ID0000000000001223352
   root@ui1#commit
   ```

   c. Reenable the SRC software process.

   ```
   root@ui1#delete system processes service-deployment disable
   root@ui1#commit
   ```

5. Restart the SRC software process on the JUNOS routing platform.

   root@ui1>**restart service-deployment**

   The SAE reconfigures the service that you deleted on the JUNOS routing platform.

6. Review the log files again.

   If the action you took did not fix the problem, return to the last step you performed, and proceed with this troubleshooting procedure. If you have performed all the tasks in the troubleshooting procedure and the problem persists, delete all SRC data on the JUNOS routing platform (see *Deleting All SRC Data on JUNOS Routing Platforms* on page 171*).*

Deleting All SRC Data on JUNOS Routing Platforms

If deleting parts of the SRC data on a JUNOS routing platform fails to solve problems, delete all the SRC data and restart the SRC software process. To do so:

1. Delete all SRC interfaces and services.

   delete groups sdx
   root@ui1#**commit**

2. If you are running SDX software releases 5.0 through 6.1, you should also delete interface sessions. (After release 6.2, session data is no longer stored on the router, it is stored on the SAE host using the session store feature.)

   delete groups sdx-sessions
   root@ui1#**commit**

3. Restart the SRC software process on the router.

   root@ui1>**restart service-deployment**

   The SAE reconfigures all the interfaces and services that you deleted from the router.