Chapter 7

Using JUNOS Routing Platforms in the SRC Network with the SRC CLI

This chapter describes how to use the SRC CLI to set up the SRC software and how to set up JUNOS routing platforms so that the routing platforms can be used the SRC network. It also shows how to monitor the interactions between the SAE and JUNOS routing platforms and troubleshoot SRC problems on JUNOS routing platforms.

You can also use the following to configure JUNOS routers:

- To use the C-Web interface, see *SRC-PE C-Web Interface Configuration Guide, Chapter 18, Using JUNOS Routing Platforms in the SRC Network with the C-Web Interface*.
- To use the Solaris platform, see *Chapter 8, Using JUNOS Routing Platforms in the SRC Network with a Solaris Platform*.

Topics in this chapter include:

- Monitoring Interactions Between the SAE and JUNOS Routing Platforms on page 108
- Troubleshooting Problems with the SRC Software Process on page 109

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:

- Detect operative routers and configured JUNOS VRs in the SRC network and add them to the configuration.
- Add each router and VR individually.

Adding Operative JUNOS Routing Platforms

To add to the directory routers and JUNOS VRs that are currently operative and have an operating SNMP agent:

- In operational mode, enter the following command:

  ```
  request network discovery network network <community community>
  ```

  where:

  - *network*—Address (with or without mask) of the network to discover
  - *community*—Name of the SNMP community to which the devices belong

If you add a router using the discover network feature, the software adds the IP address of the first SNMP agent on the router to respond to the discover request.

Adding Routers Individually

Use the following configuration statements to add a router device:

```
shared network device name {
    description description;
    management-address management-address;
    device-type (junose| junos| pcmm| proxy);
    qos-profile [qos-profile...];
}
```

To add a router device:

1. From configuration mode, access the configuration statements that configure network devices. This procedure uses junos_boston as the name of the router.

   ```
   user@host# edit shared network device junos_boston
   ```

2. (Optional) Add a description for the router.

   ```
   [edit shared network device junos_boston]
   user@host# set description description
   ```

3. (Optional) Add the IP address of the router.

   ```
   [edit shared network device junos_boston]
   user@host# set management-address management-address
   ```

4. (Optional) Specify the type of device that you are adding.

    ```
    [edit shared network device junos_boston]
    user@host# set device-type junos
    ```

5. (Optional) Verify your configuration.

    ```
    [edit shared network device junos_boston]
    user@host# show
    description "This is a core-facing JUNOS router.";
    management-address 10.117.8.32;
    device-type junos;
    interface-classifier {
      rule rule-0 {
        script #;
      }
    }
    ```

Adding Virtual Routers Individually

Use the following configuration statements to add a virtual router:

```
shared network device name virtual-router name {
    sae-connection [sae-connection...];
    snmp-read-community snmp-read-community;
    snmp-write-community snmp-write-community;
    scope [scope...];
    tracking-plug-in [tracking-plug-in...];
}
```

To add a virtual router:

1. From configuration mode, access the configuration statements for virtual routers. This procedure uses junos_Boston as the name of the router. For JUNOS routing platforms, use the name default for the virtual router.

    ```
    user@host# edit shared network device junos_boston virtual-router default
    ```

2. Specify the addresses of SAEs that can manage this router. This step is required for the SAE to work with the router.

    ```
    [edit shared network device junos_boston virtual-router default]
    user@host# set sae-connection [sae-connection...]
    ```

    To specify the active SAE and the redundant SAE, enter an exclamation point (!) after the hostname or IP address of the connected SAE. For example:

    ```
    [edit shared network device junos_boston virtual-router vr1]
    user@host# set sae-connection [sae1! sae2!]
    ```

3. (Optional) Specify an SNMP community name for SNMP read-only operations for this VR.

    ```
    [edit shared network device junos_boston virtual-router default]
    user@host# set snmp-read-community snmp-read-community
    ```

4. (Optional) Specify an SNMP community name for SNMP write operations for this virtual router.

   ```
   [edit shared network device junos_boston virtual-router default]
   user@host# set snmp-write-community snmp-write-community
   ```

5. (Optional) Specify service scopes assigned to this virtual router. The scopes are available for subscribers connected to this virtual router for selecting customized versions of services.

   ```
   [edit shared network device junos_boston virtual-router default]
   user@host# set scope [scope...]
   ```

6. (Optional) Specify the plug-ins that track interfaces that the SAE manages on this virtual router.

   ```
   [edit shared network device junos_boston virtual-router default]
   user@host# tracking-plug-in [tracking-plug-in...]
   ```

7. (Optional) Verify your configuration.

   ```
   [edit shared network device junos_boston virtual-router default]
   user@host# show
   sae-connection 192.168.80.1;
   snmp-read-community ********;
   snmp-write-community ********;
   scope POP-Cambridge;
   tracking-plug-in flexRadius;
   ```

Related Topics

- For information about service scopes, see *SRC-PE Services and Policies Guide, Chapter 1, Managing Services with the SRC CLI*
- For information about tracking plug-ins, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 11, Configuring Accounting and Authentication Plug-Ins with the SRC CLI*

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.

Use the following configuration statements to configure the JUNOS router driver:

```
shared sae configuration driver junos {
```

```
    beep-server-port beep-server-port;
    tls-beep-server-port tls-beep-server-port;
    connection-attempts connection-attempts;
    keepalive-interval keepalive-interval;
    message-timeout message-timeout;
    batch-size batch-size;
    transaction-batch-time transaction-batch-time;
    sdx-group-name sdx-group-name;
    sdx-session-group-name sdx-session-group-name;
    send-commit-check send-commit-check;
}
```

To configure the JUNOS router driver:

1. From configuration mode, access the configuration statement that configures the JUNOS router driver. In this sample procedure, the JUNOS driver is configured in the west-region group.

   ```
   user@host# edit shared sae group west-region configuration driver junos
   ```

2. Specify the 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.

   If you set this value to zero and the TLS BEEP server port is set, the SAE accepts only TLS connections.

   ```
   [edit shared sae group west-region configuration driver junos]
   user@host# set beep-server-port beep-server-port
   ```

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

   If you set this value to zero, the SAE does not accept TLS connections.

   ```
   [edit shared sae group west-region configuration driver junos]
   user@host# set tls-beep-server-port tls-beep-server-port
   ```

4. Specify the number of outstanding connection attempts before new connection attempts are dropped.

   ```
   [edit shared sae group west-region configuration driver junos]
   user@host# set connection-attempts connection-attempts
   ```

5. Specify the interval between keepalive messages sent from the router.

   ```
   [edit shared sae group west-region configuration driver junos]
   user@host# set keepalive-interval keepalive-interval
   ```

6. Specify the 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.

```
[edit shared sae group west-region configuration driver junos]
user@host# set message-timeout message-timeout
```

7. Specify the minimum number of service configuration transactions that are committed at the same time

   ```
   [edit shared sae group west-region configuration driver junos]
   user@host# set batch-size batch-size
   ```

8. Specify the maximum time to collect configuration transactions in a batch.

   ```
   [edit shared sae group west-region configuration driver junos]
   user@host# set transaction-batch-time transaction-batch-time
   ```

9. Specify the name of a session group on the JUNOS routing platform in which provisioning objects are stored.

   ```
   [edit shared sae group west-region configuration driver junos]
   user@host# set sdx-session-group-name sdx-session-group-name
   ```

10. Enable or disable commit check. If enabled, a more detailed error message is logged if a batch fails, which lets you verify individual transactions in a batch.

    ```
    [edit shared sae group west-region configuration driver junos]
    user@host# set send-commit-check send-commit-check
    ```

11. (Optional) Verify your configuration.

    ```
    [edit shared sae group west-region configuration driver junos]
    user@host# show
    beep-server-port 3333;
    tls-beep-server-port 0;
    connection-attempts 50;
    keepalive-interval 45;
    message-timeout 30000;
    batch-size 10;
    transaction-batch-time 2000;
    sdx-group-name sdx;
    sdx-session-group-name sdx-sessions;
    send-commit-check true;
    ```

Related Topics

- For information about setting up SAE groups, see *SRC-PE Getting Started Guide, Chapter 21, Setting Up an SAE with the SRC CLI.*

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.

You can use SRC CLI commands to manage certificates manually, or through the Simple Certificate Enrollment Protocol (SCEP).

Certificates are in the format defined in the X.509 standard for public key infrastructure. The certificate requests are in the Public Key Cryptology Standard (PKCS) #10 format.

Tasks to set up the SAE and the JUNOS routing platform to use TLS are:

1. Manually Obtaining Digital Certificates on page 96

   Or

2. Obtaining Digital Certificates through SCEP on page 98
3. Installing the Server Certificate on the Routing Platform on page 99
4. Creating a Client Certificate for the Router on page 100
5. Installing the Client Certificate on the Router on page 100
6. Configuring the SAE to Use TLS on page 100
7. Configuring TLS on the SAE on page 100

Manually Obtaining Digital Certificates

You can manually add digital certificates, or you can use SCEP to help manage how you obtain certificates. See *Obtaining Digital Certificates through SCEP* on page 98.

To manually add a signed certificate:

1. Create a certificate signing request.

   user@host> **request security generate-certificate-request subject** *subject* **password** *password*

   where:

   - subject is the distinguished name of the SRC host; for example cn=src1,ou=pop,o=Juniper,l=kanata, st=Ontario,c=Canada.

- password is the password received from the certificate authority for the specified subject.

By default, this request creates the file /tmp/certreq.csr and encodes the file by using Privacy-Enhanced Mail (pem) encoding.

2. Copy the file generated in Step 1 to another system, and submit the certificate signing request file generated in Step 1 to VeriSign, Inc. (http://www.verisign.com) for signing.

   You can transfer the file through FTP by using the file copy command.

   user@host> **file copy** *source_file* **ftp://***username***@***server*[:*port*]**/***destination_file*

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

3. When you receive the signed certificate, copy the file back to the SRC system to the */tmp* directory.

   You can transfer the file through FTP, as shown in Step 2.

4. Add the certificate to the SRC configuration.

   user@host> **request security import-certificate file-name** *file-name* **identifier** *identifier*

   where:

   - file-name is the name of the certificate file in the */tmp* folder. The file must be in one of the following formats, which is indicated by the following extensions:
     - CER—Windows extension
     - PEM—Privacy-Enhanced Mail encoding
     - DER—Binary encoding
     - BER—Binary encoding
   - identifier is the name of the certificate.

   For example, to import the file src.cer that is identified as web:

   user@host> **request security import-certificate file-name src.cer identifier web**

5. Verify that the certificate is part of the SRC configuration.

```
user@host> show security certificate
web subject:CN=host
```

   If there are no certificates on the system, the CLI displays the following message:

```
No entity certificates in key store
```

Obtaining Digital Certificates through SCEP

You can use SCEP to help manage how you obtain digital certificates, or you can manually add certificates. See *Manually Obtaining Digital Certificates* on page 96.

Before you can obtain certificates for your use, you must get the CA's certificate and install it in the local store of trusted certificates.

To add a signed certificate that you obtain through SCEP:

1. Request your CA's certificate through SCEP.

   ```
   user@host> request security get-ca-certificate url url ca_identifier ca_identifier
   ```

   where:

   - url is the URL of the certificate authority (which is the SCEP server).
   - ca-identifier is the identifier that designates the authority.

   For example, to request a certificate from the CA authority SrcCA at a specified URL on the server security_server:

   ```
   user@host> request security get-ca-certificate url
   http://security_server:8080/ejbca/publicweb/apply/scep/pkiclient.exe
   ca-identifier SrcCA

   Version: 3
   Serial Number: 5721058705923989279
   Signature Algorithm: SHA1withRSA
   Issuer: CN=SrcCA
   Valid From: Wed Sep 06 17:00:55 EDT 2006
   Valid Until: Sat Sep 03 17:10:55 EDT 2016
   Subject: CN=SrcCA
   Public key: RSA
   Thumbprint Algorithm: SHA1
   Thumbprint: 3c 57 a9 77 af 83 3 e9 c7 1e ee e2 4a e8 ff f3 89 f4 11 a9
   Do you want to add the above certificate as a trusted CA [yes,no] ? (no) y
   ```

2. Request that the certificate authority automatically sign the certificate request.

   ```
   user@host> request security enroll subject subject password password
   ```

   where:

   - subject is the distinguished name of the SRC host; for example cn=myhost.
   - password is the password received from the certificate authority.

   For example, to request a certificate from the CA authority SrcCA at a specified URL on the server security_server:

   ```
   user@host> request security enroll url
   http://security_server:8080/ejbca/publicweb/apply/scep/pkiclient.exe
   identifier web ca-identifier SrcCA subject cn=myhost password mypassword
   ```

```
Received certificate:
Version: 3
Serial Number: 6822890691617224432
Signature Algorithm: SHA1withRSA
Issuer: CN=SrcCA
Valid From: Tue Sep 19 16:33:11 EDT 2006
Valid Until: Thu Sep 18 16:43:11 EDT 2008
Subject: CN=myhost
Public key: RSA
Do you want to install the above certificate [yes,no] ? (no) y
```

3. Verify that the certificate is part of the SRC configuration.

```
user@host> show security certificate
web subject:CN=myhost
```

If there are no certificates on the system, the CLI displays the following message:

```
No entity certificates in key store
```

Installing the Server Certificate on the Routing Platform

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;
            }
    }
}
```

Configuring the SAE to Use TLS

To configure the SAE to accept TLS connections, enter a port number with the **set beep-server-port** command in the JUNOS router driver configuration.

See *Configuring the SAE to Manage JUNOS Routing Platforms* on page 93.

Configuring TLS on the SAE

Use the following configuration statements to configure TLS on the SAE:

```
shared sae configuration driver junos security {
    need-client-authentication;
    certificate-identifier private-key;
}
```

To configure TLS on the SAE:

1. From configuration mode, access the configuration statement that configures security for the JUNOS TLS connection. In this sample procedure, the JUNOS driver is configured in the west-region group.

```
user@host# edit shared sae group west-region configuration driver junos security
```

2. (Optional) Specify whether or not the SAE requests a client certificate from the router when a connection to the router is established.

   ```
   [edit shared sae group west-region configuration driver junos security]
   user@host# set need-client-authentication
   ```

3. Specify the name of certificate to be used for TLS communications.

   ```
   [edit shared sae group west-region configuration driver junos security]
   user@host# set certificate-identifier private-key
   ```

4. (Optional) Verify your TLS configuration.

   ```
   [edit shared sae group west-region configuration driver junos security]
   user@host# show
   need-client-authentication;
   certificate-identifier privatekey;
   ```

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:

- The SAE takes the state of the session layer on the router to be correct and updates its local state to be consistent with the router. The SAE then sends stop events for all sessions where the corresponding provisioning in the router has been removed.
- The SAE takes its local state to be the correct state and updates the router to be consistent with its local state.
- The SAE does not solve the state discrepancy. It reports disparities through the SAE device driver event trap called routerConfOutOfSynch and through the info log.

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

Setting Up Periodic Configuration Checking

Use the following configuration statements to configure the SAE to periodically check the configuration of the JUNOS routing platform:

```
shared sae configuration driver junos configuration-checking
    configuration-checking-schedule configuration-checking-schedule;
    configuration-checking-action (enforce | synchronize | detect);
}
```

To configure the SAE to periodically check the configuration of the JUNOS routing platform:

1. From configuration mode, access the configuration statement that configures the configuration checking feature.

   ```
   user@host# edit shared sae configuration driver junos configuration-checking
   ```

2. Specify when the SAE checks the router configuration.

   ```
   [edit shared sae configuration driver junos configuration-checking]
   user@host# set configuration-checking-schedule configuration-checking-schedule
   ```

3. Specify the action that the SAE takes when it detects disparities between the configuration of the SAE and the configuration on the router.

   ```
   [edit shared sae configuration driver junos configuration-checking]
   user@host# set configuration-checking-action enforce | synchronize | detect
   ```

4. (Optional) From operational mode, verify your configuration checking configuration.

   ```
   [edit shared sae configuration driver junos configuration-checking]
   user@host# show
   configuration-checking-schedule "0 0 * * * * *";
   configuration-checking-action synchronize;
   ```

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 92.) 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 if the community strings have not been configured for the VR.

Use the following configuration statements to configure global default SNMP communities:

```
shared sae configuration driver snmp {
    read-only-community-string read-only-community-string;
    read-write-community-string read-write-community-string;
}
```

To configure global default SNMP communities:

1. From configuration mode, access the configuration statements that configure default SNMP communities.

   ```
   user@host# edit shared sae configuration driver snmp
   ```

2. Configure the default SNMP community string used for read access to the router.

   ```
   [edit shared sae configuration driver snmp]
   user@host# set read-only-community-string read-only-community-string
   ```

3. Configure the default SNMP community string used for write access to the router.

   ```
   [edit shared sae configuration driver snmp]
   user@host# set read-write-community-string read-write-community-string
   ```

4. (Optional) Verify your configuration.

   ```
   [edit shared sae configuration driver snmp]
   user@host# show
   read-only-community-string ********;
   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 into an internal part of the shared configuration so that a NIC can associate a router with an SAE.

Interface Object Fields

Router initialization scripts are written in the Python programming language (www.python.org) and executed in the Jython environment (www.jython.org).

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 8 describes the fields that the SAE exports.

Table 8: 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, in the 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

Use the following configuration statements to specify router initialization scripts for JUNOS routing platforms:

```
shared sae configuration driver scripts {
    extension-path extension-path;
    general general;
    junos junos;
}
```

To configure router initialization scripts for JUNOS routing platforms:

1. From configuration mode, access the configuration statements that configure router initialization scripts. In this sample procedure, the scripts are configured in the west-region group.

   ```
   user@host# edit shared sae group west-region configuration driver scripts
   ```

2. Specify the router initialization script for JUNOS routing platforms.

   ```
   [edit shared sae group west-region configuration driver scripts]
   user@host# set junos junos
   ```

3. Configure a router initialization script that can be used for all types of routers that the SRC software supports.

   ```
   [edit shared sae group west-region configuration driver scripts]
   user@host# set general general
   ```

4. Configure a path to router initialization scripts that are not in the default location, */opt/UMC/sae/lib*.

   ```
   [edit shared sae group west-region configuration driver scripts]
   user@host# set extension-path extension-path
   ```

5. (Optional) From operational mode, verify your router initialization script configuration.

   ```
   [edit shared sae group west-region configuration driver scripts]
   user@host# show
   extension-path ;
   junos iorPublisher;
   ```

Accessing the Router CLI

You can access the CLIs of Juniper Networks routers through a Telnet or secure shell connection.

- To open a Telnet session to a router, use the telnet operational mode command. For example:

  ```
  user@host> telnet 10.10.10.3
  ```

- To open a secure shell connection, use the ssh operational command. For example:

  ```
  user@host> ssh host 10.10.10.3
  ```

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 platform 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 3, Configuring Logging for SRC Components with the CLI*.
- For information about configuring logging on JUNOS routing platforms, see *JUNOS System Basics Configuration Guide*.

Troubleshooting Problems with the SRC Software Process

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 on the JUNOS routing platform 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 108.

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.

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**

Viewing the State of JUNOS Device Drivers with the SRC CLI

To display the state of JUNOS drivers, use the following operational mode command:

show sae drivers <device-name *device-name*> < (brief) > <maximum-results *maximum-results*>

For example:

```
user@host> show sae drivers device-name default@jrouter
JUNOS Driver
Device name                          default@jrouter
Device type                          junos
Device IP                            /10.10.6.113:1879
Local IP                             10.10.6.113
TransportRouter
Device version                       8.2R1.7
Start time                           Thu Mar 08 21:00:50 UTC 2007
Number of notifications              0
Number of processed added            0
Number of processed changed          0
Number of processed deleted          0
Number of provisioning attempt       0
Number of provisioning attempt failed 0
Device type                          JunosRouterDriver
Job queue size                       0
Number of SAP                        3
Number of PAP                        0
Start time                           Thu Mar 08 21:00:55 UTC 2007
End time                             Thu Mar 08 21:00:55 UTC 2007

  Transaction Manager
  Transaction queue size 0
  Router name            default@troll
```

Viewing Statistics for Specific JUNOS Device Drivers with the SRC CLI

To display statistics for a specific JUNOS device driver, use the following operational mode command:

show sae statistics device <name *name*> < (brief) >

For example:

```
user@host> show sae statistics device name default@jrouter
SNMP Statistics
Add notification handle time      7
Change notification handle time   0
Client ID                         default@troll
Delete notification handle time   0
Failover IP                       0.0.0.0
Failover port                     0
Handle message time               40
Job queue age                     0
Job queue time                    0
```

```
Number message send              3
Number of added jobs             0
Number of add notifications      0
Number of change notifications   0
Number of delele notifications   0
Number of managed interfaces     3
Number of message errors         0
Number of message timeouts       0
Number of removed jobs           0
Number of user session established 0
Number of user session removed   0
Router type                      JUNOS
Up time                          7036120
Using failover server            false
```

Viewing Statistics for All JUNOS Device Drivers with the SRC CLI

To display SNMP statistics for all JUNOS device drivers, use the following operational mode command:

show sae statistics device common junos

For example:

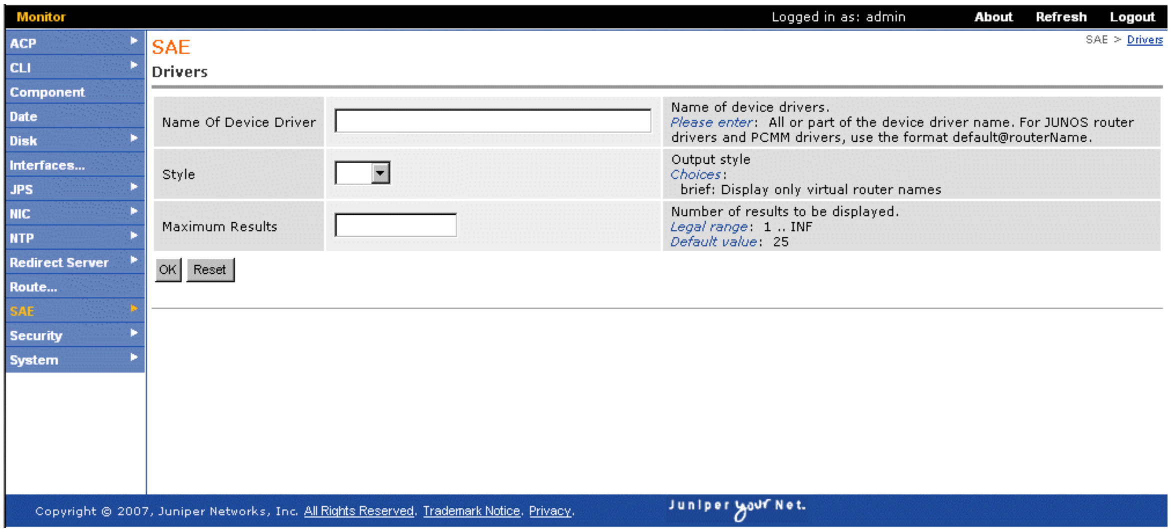
```
user@host> show sae statistics device common junos
SNMP Statistics
Driver type                    JUNOS
Number of close requests       0
Number of connections accepted 0
Number of current connections  0
Number of open requests        0
Server address                 0.0.0.0
Server port                    3288
Time since last redirect       0
```

Viewing the State of JUNOS Device Drivers with the C-Web Interface

If the log files indicate a problem with a specific driver, review the configuration of the associated JUNOS router driver with C-Web.

1. Select **SAE** from the side pane, and click **Drivers**.

   The Drivers pane appears.



SAE

Drivers

Name Of Device Driver		Name of device drivers. *Please enter*: All or part of the device driver name. For JUNOS router drivers and PCMM drivers, use the format default@routerName.
Style		Output style *Choices*: brief: Display only virtual router names
Maximum Results		Number of results to be displayed. *Legal range*: 1 .. INF *Default value*: 25

2. In the Name of Device Driver box, enter a full or partial device driver name for which you want to display information, or leave the box blank to display all devices. Use the format:

   default@<router name>

3. Select an output style from the Style list.

4. In the Maximum Results box, enter the maximum number of results that you want to receive.

5. Click **OK**.

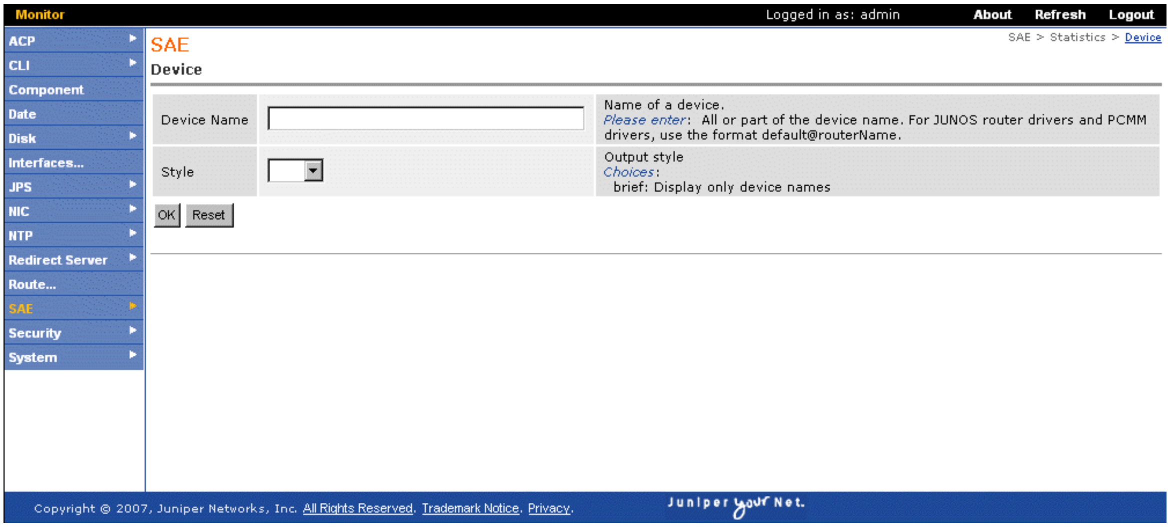
   The Drivers pane displays information about the JUNOS device driver.

Viewing Statistics for Specific JUNOS Device Drivers with the C-Web Interface

To view SNMP statistics about devices:

1. Select **SAE** from the side pane, click **Statistics**, and then click **Device**.

   The Device pane appears.



Name of a device.
Please enter: All or part of the device name. For JUNOS router drivers and PCMM drivers, use the format default@routerName.

Output style
Choices:
brief: Display only device names

2. In the Device Name box, enter a full or partial device name for which you want to display information, or leave the box blank to display all devices.
3. Select an output style from the Style list.
4. Click **OK**.

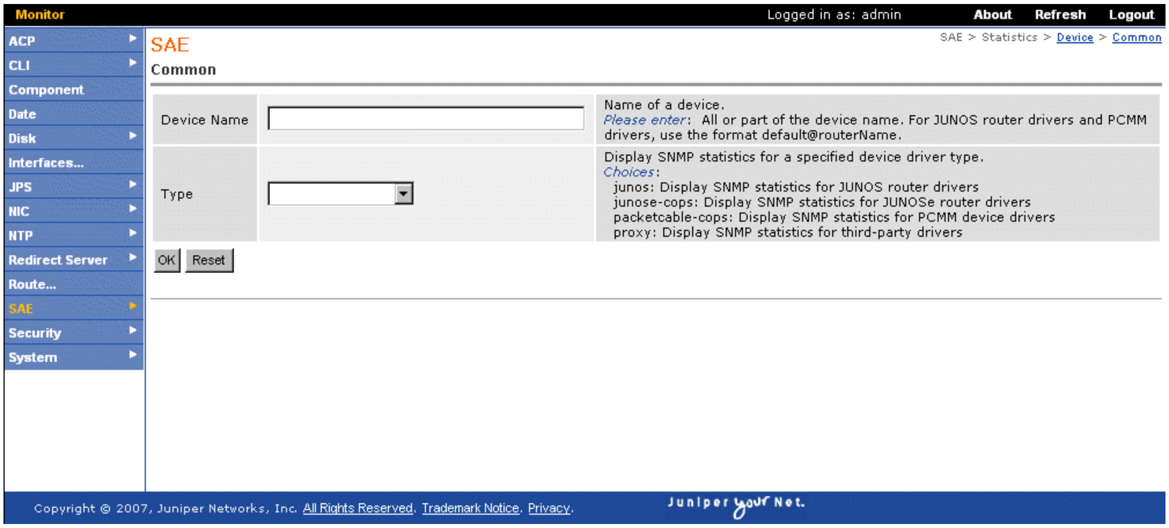
   The Device pane displays statistics for all devices.

Viewing Statistics for All JUNOS Device Drivers with the C-Web Interface

To view SNMP statistics about specific devices:

1. Select **SAE** from the side pane, click **Statistics**, click **Device**, and then click **Common**.

   The Common pane appears.



Device Name	Name of a device. *Please enter*: All or part of the device name. For JUNOS router drivers and PCMM drivers, use the format default@routerName.
Type	Display SNMP statistics for a specified device driver type. *Choices*: junos: Display SNMP statistics for JUNOS router drivers junose-cops: Display SNMP statistics for JUNOSe router drivers packetcable-cops: Display SNMP statistics for PCMM device drivers proxy: Display SNMP statistics for third-party drivers

2. In the Device Name box, enter a full or partial device name for which you want to display information, or leave the box blank to display all devices.

3. Select the junos from the Type list:

4. Click **OK**.

   The Common pane displays statistics for the specified device.