Chapter 6

Using JUNOSe Routers 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 a JUNOSe router so that the router can be used the SRC network. It inlcudes information about how to monitor the interactions between the SAE and the JUNOSe router and how to troubleshoot SRC problems on the router.

You can also use the CLI that runs on Solaris platforms and the C-series platform to configure the SRC software to work with JUNOSe routers. See *Chapter 5, Using JUNOSe Routers in the SRC Network with the SRC CLI*.

Topics in this chapter include:

COPS Connection Between JUNOSe Routers and the SAE

Configuring the SRC client on a JUNOSe router opens a Common Open Policy Service (COPS) protocol layer connection to the SAE. When the SRC client software establishes a TCP/IP connection to the SAE, the SAE starts to manage the JUNOSe router. Subsequently, the SRC client sends configuration changes made on the JUNOSe router to the SAE, and the SAE updates SRC configurations for services and policies accordingly.

The SAE supports two versions of COPS:

- COPS usage for policy provisioning (COPS-PR)
- COPS External Data Representation Stand (COPS XDR)

The version of COPS that you use depends on the version of COPS that your JUNOSe router supports. When you set up your JUNOSe router to work with the SAE, you enable either COPS-PR mode or COPS XDR mode.

Adding JUNOSe Routers and Virtual Routers

The SAE uses router and virtual router objects in the directory to manage interfaces on JUNOSe virtual routers. Each JUNOSe router in the SRC network and its virtual routers (VRs) must appear in the directory. There are two ways to add routers to the directory:

- Use SDX Admin to detect operative routers and configured JUNOSe VRs in the SRC network and add them to the directory.
- 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 or unconfigured JUNOSe VRs.

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 89.

Adding Operative JUNOSe Routers and Virtual Routers

To simultaneously add to the directory routers and JUNOSe VRs that are currently operative and have an operating Simple Network Management Protocol (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**.

   Objects for all routers and JUNOSe VRs that meet the criteria you specified appear under the Networks object in the navigation pane. You can modify the configuration of these objects. For information about configuring these objects, see *Adding Routers Individually* on page 83 and *Adding Virtual Routers Individually* on page 85.

Adding Routers Individually

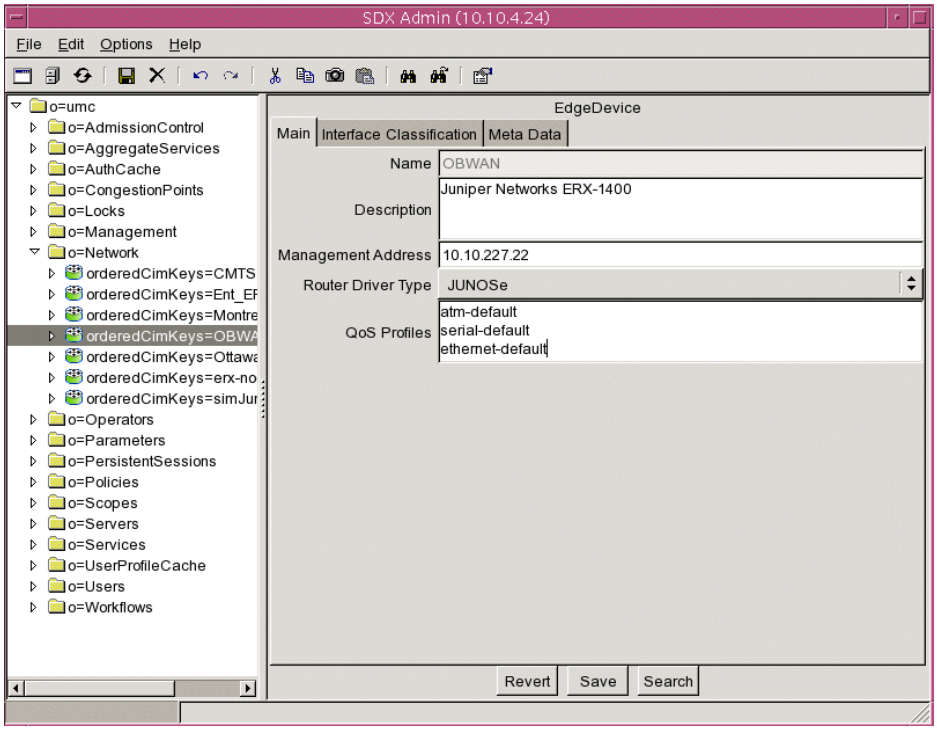
To add a single router to the directory with SDX Admin:

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 on the router, and click **OK**.

   The name of the new device appears in the navigation pane, and information about the router appears in the Main Tab of the EdgeDevice pane.



3. In the content pane, edit or accept the default values for the router fields.

   See *Router Fields* on page 84.

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 find utility uses.
- Value—Text string
- Example—ERX-1400 router located in Ottawa

Management Address

- IP address of the router. 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 router 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. To update this list, see *SRC-PE Solutions Guide, Chapter 1, Managing Tiered and Premium Services with QoS on JUNOSe Routers*.
- 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 to the directory with SDX Admin:

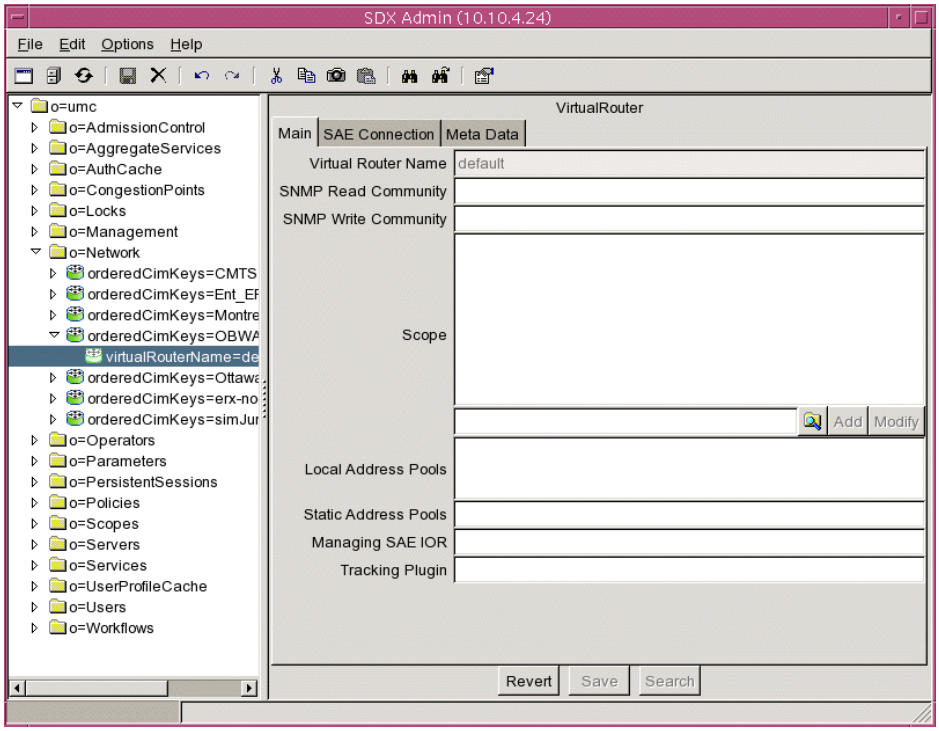
1. In the navigation pane, right-click the device to which you want to add the VR, and select **New > VirtualRouter**.

   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 name of the new VR appears in the navigation pane, and the VirtualRouter pane appears.



3. In the Main tab in the VirtualRouter pane, edit or accept the default values for the fields.

   See *Virtual Router Fields* on page 86.

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 89.

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 = < name of virtual router > 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—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

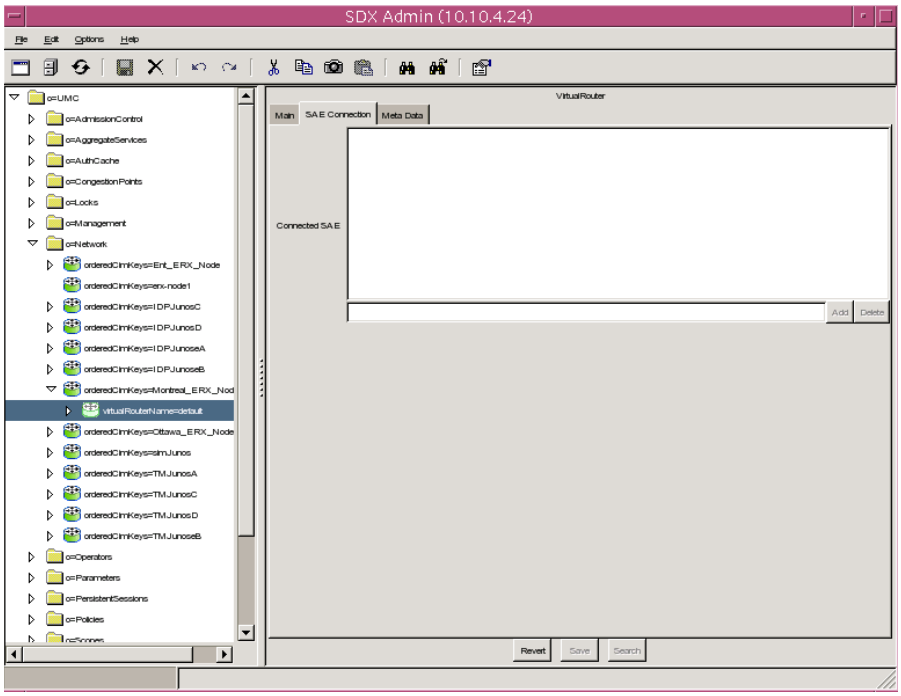
- Common Object Request Broker Architecture (CORBA) reference for the SAE managing this VR.
- Value—One of the following items:
  - The actual CORBA reference for the SAE
  - The absolute path to the interoperable object reference (IOR) file
  - A corbaloc URL in the form corbaloc:: < host > :8801/SAE
    - < 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 use 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

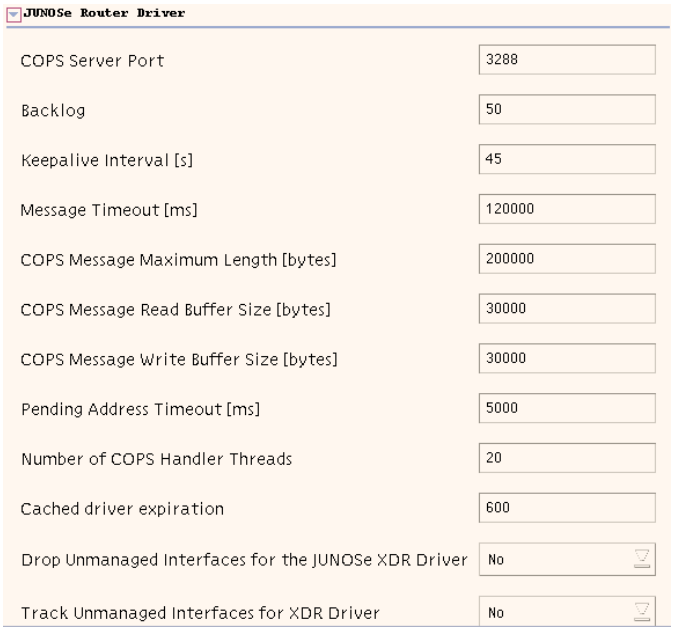
- SAEs that are connected to the router or CMTS device.
- Value—IP addresses
- Default—No value

Configuring the SAE to Manage JUNOSe Routers

To set up the SAE to manage JUNOSe routers, you need to configure a router driver that specifies the COPS connection between the SAE COPS server and the COPS client in the JUNOSe router.

To use SDX Configuration Editor to configure a JUNOSe 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 **JUNOSe Router Driver** section.



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

   See *JUNOSe Router Driver Fields* on page 91.

4. You can also configure a session store for the JUNOSe router driver.

   See *Storing Subscriber and Service Session Data* on page 41.

5. Select **File > Save**.

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

JUNOSe Router Driver Fields

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

COPS Server Port

- Port number of the SAE COPS server.
- Value—Port number that matches the configuration of the SRC client in the JUNOSe router
- Default—3288
- Property name—Router.junose.server_port

Backlog

- Number of connection attempts before connections are dropped.
- Value—Integer
- Default—50
- Property name—Router.junose.backlog_connections

Keepalive Interval [s]

- Interval between keepalive messages sent from the COPS client (the JUNOSe router). The COPS client monitors the COPS connection by sending keepalive messages at random intervals between one-fourth and three-fourths of the specified interval. If the client does not receive the expected keepalive answer within the specified timeout, the client terminates the connection.
- Value—Number of seconds in the range 0–32768. A value of 0 means that timeout is disabled.
- Guidelines—A short interval results in a high load on the COPS interface. A long interval results in a long time before a COPS failure is detected.
- Default—45
- Property name—Router.junose.keepalive

Message Timeout [ms]

- Timeout interval in which the COPS server waits for a response to COPS requests. Under a high load the router may not be able to respond fast enough to COPS requests. Change this value only if a high number of COPS timeout events appear in the error log.
- Value—Number of milliseconds
- Default—60000
- Property name—Router.junose.message_timeout

COPS Message Maximum Length [bytes]

- Maximum length of a COPS message.
- Value—Number of bytes in the range 4 bytes to 2 GB
- Guidelines—We recommend that you use the default setting.
- Default—200000
- Property name—Router.junose.message_max_length

COPS Message Read Buffer Size [bytes]

- Buffer size for receiving COPS messages from the JUNOSe client.
- Value—Number of bytes in the range 4 bytes to 2 GB
- Guidelines—We recommend that you use the default setting unless you are instructed to change it by Juniper Networks engineers.
- Default—30000
- Property name—Router.junose.message_read_buffer_size

COPS Message Write Buffer Size [bytes]

- Buffer size for sending COPS messages to the JUNOSe client.
- Value—Number of bytes in the range 4 bytes to 2 GB
- Guidelines—We recommend that you use the default setting unless you are instructed to change it by Juniper Networks engineers
- Default—30000
- Property name—Router.junose.message_write_buffer_size

Pending Address Timeout [ms]

- Maximum time that an address request remains pending.
- Value—Number of milliseconds
- Guidelines—Realistic values are in the range 1000–15000 (5 seconds to 15 seconds).
- Default—5000
- Property name—Router.junose.pending_address_timeout

Number of COPS Handler Threads

- Size of the thread pool for handling unsolicited messages. These threads are shared among all JUNOSe router drivers.
- Value—Number of threads
- Default—20
- Property name—Router.junose.handler_threads

Cache driver expiration

- Minimum amount of time to keep the state of a router driver after its COPS connection has been closed.
- Value—Number of seconds in the range 0–2147483647
- Default—600
- Property name—Router.junose.cachedDriverExpiration

Drop Unmanaged Interfaces for the JUNOSe XDR Driver

- Specifies whether or not the JUNOSe router driver keeps a record of unmanaged interfaces.
- Value
  - Yes—The router driver does not keep a record of unmanaged interfaces. With this setting, next interface rules may not work properly.
  - No—The router driver keeps a record of unmanaged interfaces.
- Default—No
- Property name—Router.junose.drop_unmanaged_xdr

Using SNMP to Retrieve Information from JUNOSe Routers

Some scripts in the SRC software use SNMP to get information from the router. For example, the **poolPublisher** router initialization script uses SNMP to read the IP pools.

- On the router, you can configure access to the router's SNMP server. See *Configuring the SNMP Server on the JUNOSe Router* on page 93.
- On the SAE, you can configure global default SNMP communities that are used for read and write access to the router. See *Configuring Global SNMP Communities in the SRC Software* on page 94.
- In the directory, you can specify SNMP communities for each virtual router. We recommend that you specify communities for each virtual router instead of global communities. See *Adding Virtual Routers Individually* on page 85.

Configuring the SNMP Server on the JUNOSe Router

Access to the SNMP server on the router by an SNMP client is governed by a proprietary SNMP community table. This table identifies communities that have read-only, read-write, or administrative permission to the SNMP Management Information Base (MIB) stored on a particular server.

When an SNMP server receives a request, the server extracts the client's IP address and the community name. The SNMP server searches the community table for a matching community.

- If a match is found, its access list name is used to validate the IP address.
  - If the access list name is null, the IP address is accepted.
  - If an invalid IP address results, an SNMP authentication error is sent to the SNMP client.
- If a match is not found, an SNMP authentication error results.

To configure the SNMP agent on the JUNOSe router:

1. Switch to the virtual router for which you want to create an SRC client.

   host1#(config)**virtual-router** <vrName>

2. Enable the SNMP agent.

   host1:<vrName>#(config)**snmp-server**

3. Configure at least one authorized SNMP read-write community (SNMPv1/v2c), which provides SNMP client access.

   host1:<vrName>(config)#**snmp-server community boston rw**

4. (Optional) Configure a read-only community.

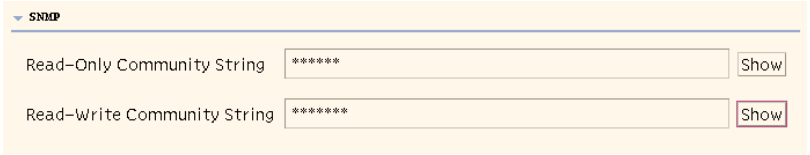
   host1:<vrName>#(config)**snmp-server public ro**

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 configuration file 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 95.

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.

For JUNOSe VRs that supply IP addresses from a local pool, a router initialization script is provided that identifies which VR supplies each IP pool and writes the information to the directory. The SAE runs the script only when a COPS connection is established to the JUNOSe router. Consequently, if you modify information about IP pools on a VR after the COPS connection is established, the SAE will not automatically register the changes, and you must update the directory.

Table 6 describes the router initialization scripts that we provide with the SRC software in the */opt/UMC/sae/lib* directory.

Table 6: Router Initialization Scripts

Script Name	Function	When to Use Script
IorPublisher	Publishes the IOR of the SAE in the directory so that a NIC can associate a router with an SAE.	Use with JUNOSe routers that do not supply IP addresses from local pools, and JUNOS routing platforms.
poolPublisher	Publishes the IOR of the SAE and local IP address pools in the directory so that a NIC can associate a router with an SAE and resolve the IP-to-SAE mapping.	Use with JUNOSe virtual routers that supply IP addresses from local pools.

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 7 describes the fields that the SAE exports.

Table 7: 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

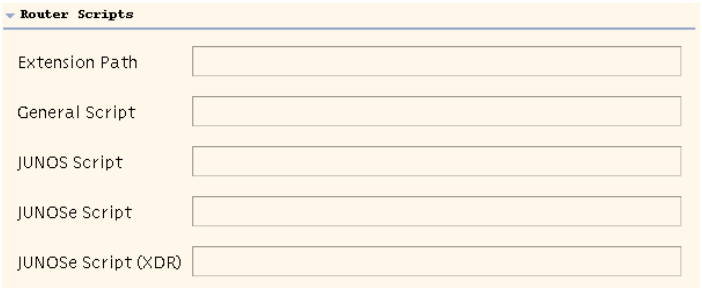
The following script defines a router initialization class named *SillyRouterInit*. The interface class does not implement any useful functionality. The interface class just writes messages to the infoLog when the router connection is created or terminated.

```
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 configuration file 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 fields.

   See *JUNOSe Router Script Fields* on page 98.
4. Select File > Save.
5. Right-click the configuration file, select **SDX System Configuration > Export to LDAP Directory**.

JUNOSe Router Script Fields

In SDX Configuration Editor, you can edit the following fields in 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.*

JUNOSe Script

- Router initialization script for JUNOSe routers when the JUNOSe driver uses COPS-PR mode when connecting to the SAE. 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.junose

JUNOSe Script (XDR)

- Router initialization script for JUNOSe routers when the JUNOSe driver uses XDR mode when connecting to the SAE. 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
- Guidelines—In COPS XDR mode, the router does not send the network access server (NAS) IP address to the SAE. If your configuration requires this value, add the following line to a JUNOSe script:

  import ERXnasip

  When you add the **import ERXnasip** entry, the script obtains the NAS-IP address from the router through SNMP. This mechanism can affect performance, especially when the SAE manages a large number of virtual routers.
- Default—Unspecified
- Examples—iorPublisher, poolPublisher
- Property name—Router.script.junose_xdr

Updating Local IP Address Pools for JUNOSe VRs

When you reconfigure local IP address pools on a JUNOSe VR, you must update in the directory the local IP addresses that the VR provides.

Before you update local IP address pools, make sure that:

- The JUNOSe router and VR appear in the directory.
- The VR has an operating SNMP agent.
- The host that supports SDX Admin or the SAE can communicate with the VR through SNMP.
- You have write permissions for the *o = Network* subtree.

There are two ways to add routers to the directory:

- SDX Admin—Updates on VR at a time.
- The **poolRepublish** command—simultaneously updates any number of VRs in the same directory.

Updating Local IP Address Pools with SDX Admin

To allow updates of IP address pools with SDX Admin, the host that supports SDX Admin must be able to communicate with the VR through SNMP. To update local IP address pools for a VR in the directory with SDX Admin:

1. In the navigation pane, expand **o = Network**.
2. In the navigation pane, expand the object for the router on which the VR is configured.
3. Right-click the object for the VR in the navigation pane.
4. Select **Update IP Pools**.

   The SDX Admin dialog box appears.
5. Enter the IP address for the VR, enter the SNMP community if the default value is incorrect, and click **OK**.

   SDX Admin updates the local IP addresses for the VR in the directory and displays the information in the Local IP Address field of the Main tab in the VirtualRouter pane.

Updating Local IP Address Pools with the poolRepublish Command

You can use the **poolRepublish** command on the SAE host to update local IP address pools. You can specify multiple VRs with the **poolRepublish** command that use the same SNMP read community. For each VR you must specify the name of the VR, the name of the JUNOSe router on which it is configured, the VR's corresponding IP address, and the directory connection.

To update local IP addresses using the **poolRepublish** command:

1. On the SAE host, access the folder */opt/UMC/sae/etc*.

   cd /opt/UMC/sae/etc

2. Run the command.

   ./poolRepublish -v vr1@erx1 -i 192.0.2.1 -v vr2@erx2 -i 192.0.2.3 -h 192.0.2.5 -w admin123 -D cn=umcAdmin,o=umc -b o=Network,o=umc -c public

   The software updates and displays the local IP address pools for each VR you specified.

   vr1@erx1 pools: ([10.227.11.242 10.227.11.250][10.227.11.226 10.227.11.239]{10.227.11.208/255.255.255.240}{10.227.11.240/255.255.255.240}{10.227.11.224/255.255.255.240})
   vr2@erx2: ([10.227.12.242 10.227.12.250][10.227.12.226 10.227.12.239]{10.227.12.208/255.255.255.240}{10.227.12.240/255.255.255.240}{10.227.12.224/255.255.255.240})

Syntax of poolRepublish Command

The syntax for the poolRepublish command is:

poolRepublish { { -v <vrName> @ <routerName> -i <ipAddress> }*
-h <host> -b <baseDn> -D <bindDN> -w <password>
-c <readCommunity>] | -H }

<vrName>

- Name of the VR.
- Value—Text string (value is case sensitive and must match the name in the JUNOSe configuration)
- Guideline—You must enter a value for this property.
- Example—vr-boston

<routerName>

- Name of JUNOSe router on which VR is configured.
- Value—Text string (value is case sensitive and must match the name in the JUNOSe configuration)
- Example—erx1

<ipAddress>

- VR's IP address.
- Value—IP address or text string
- Example—192.0.2.1

<host>

- IP address or name of the host that supports the directory.
- Value—IP address or text string
- Example—192.0.2.2 or ottawa

<baseDn>

- DN of the root of the tree in the directory.
- Value—DN
- Example—*o = Network, o = umc*

<bindDn>

- DN of the username for authentication with the directory server.
- Value—DN
- Example—*cn = umcAdmin, o = umc*

<password>

- Password for authentication with the directory server.
- Value—Text string
- Example—Admin123

<readCommunity>

- Name of the SNMP read community for the VR. If the SNMP read community for a VR is defined in the directory, you do not need to specify this value.
- Value—Text string
- Example—public

-H

- Displays help for this tool.

Troubleshooting the poolRepublish Command

You must specify the correct arguments for the **poolRepublish** command. In addition, the specified router and directory must be available for the command to run successfully.

If no SNMP read community is configured in the directory for the VR and you do not specify this value when you run the **poolRepublish** command, you will see the following error message:

```
Could not perform ip pools update due to No 'snmpReadCommunity' attribute is
provided for virtual router: vr1@bigfoot
```

If you run the **poolRepublish** command again and supply this SNMP read community, the command should run correctly.

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.

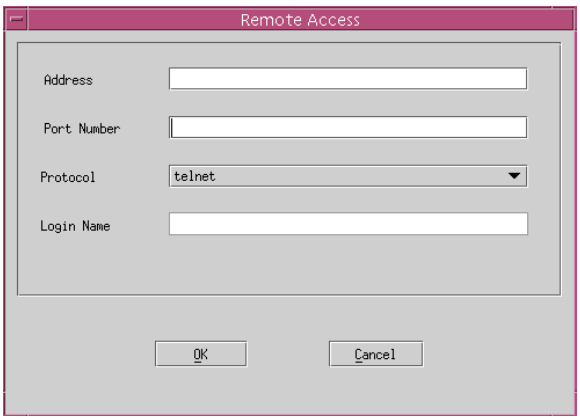
You must have the Telnet or SSH applications installed and available to Policy Editor or SDX Admin. You can open multiple Telnet or SSH sessions.

Using Policy Editor

To access a router from Policy Editor:

1. In the Policy Editor window, click **Tools** in the menu bar; then click **Manage**.

   The Remote Access dialog box appears.



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

   See *Remote Access Fields* on page 104.

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

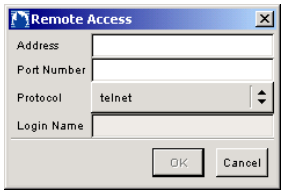
Using SDX Admin

To access a router from SDX Admin:

1. In the navigation pane, expand **o = Network**.
2. Select the router to which you want to connect, and right-click.

3. Select **Manage**.

   The Remote Access dialog box appears.



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

   See *Remote Access Fields* on page 104.

   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 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

Starting the SRC Client on a JUNOSe Router

JUNOSe routers use an SRC client to interact with the SAE. See *JUNOSe Broadband Access Configuration Guide* for complete information about configuring the SRC client on the JUNOSe router.

To start the SRC client:

1. Access the router CLI.
2. Access Global configuration mode.

   host1#**configure terminal**

3. Switch to the virtual router for which you want to create an SRC client.

   host1(config)#**virtual-router** <vrName>

4. Enable the SRC client.

   To enable COPS-PR mode:

   host1:<vrName>(config)#**sscc enable cops-pr**

   To enable COPS-XDR mode:

   host1:<vrName>(config)#**sscc enable**

5. Set the primary address from the configuration directory.

   host1:<vrName>(config)#**sscc primary address** <ipAddress> **port 3288**

Stopping the SRC Client on a JUNOSe Router

JUNOSe routers use an SRC client to interact with the SAE. See *JUNOSe Broadband Access Configuration Guide* for complete information about configuring the SRC client on the JUNOSe router.

To stop the SRC client:

1. Access the router CLI.

   See *Accessing the Router CLI* on page 103.

2. Access Global configuration mode.

   host1#**configure terminal**

3. Switch to the virtual router for which you want to stop an SRC client.

   host1(config)#**virtual-router** <vrName>

4. Disable the SRC client.

   host1:<vrName>(config)#**no sscc enable**

Monitoring Interactions Between the SAE and the JUNOSe Router

To monitor the connection between the router and the SAE:

- Use the **show sscc info** command on the JUNOSe router

To display the version number of the SRC client:

- Use the **show sscc version** command on the JUNOSe router.

See the *JUNOSe Command Reference Guides* for details about these commands.

You can also monitor the interactions between the SRC software and the router in the log files for the SAE and in the log files generated by the JUNOSe router. 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 about configuring logging on JUNOSe routers, see the *JUNOSe System Event Logging Reference Guide*.

Troubleshooting the SRC Client on JUNOSe Routers

To troubleshoot SRC problems on the router:

1. Look at the log files for the SAE and the log files generated by the SRC client on the JUNOSe router.

   - If the log files indicate a problem with specific interfaces on the router, review the configuration of the associated policies in the SRC software, and fix any errors.

   - If the log files indicate a problem with a specific service or its associated policy rules, review the configuration of the service or policies in the SRC software, and fix any errors.

   - If the log files indicate only that the SRC client is not responding, ensure that the values in the SAE configuration match the values in the SRC client configuration on the router.

2. Restart the SRC client on the JUNOSe router.

   When you restart the SRC client, the SRC client removes all policies that were installed by the SRC software and reports all interfaces again.

NOTE: DHCP addresses that were managed are not reported again, so we recommend that you do not restart the SRC client if you are managing DHCP sessions.

   To restart the SRC client in COPS-PR mode, enter the following commands:

```
host1:<vrName>(config)#no sscc enable
host1:<vrName>(config)#sscc enable cops-pr
```

   To restart the SRC client in COPS XDR mode, enter the following commands:

```
host1:<vrName>(config)#no sscc enable
host1:<vrName>(config)#sscc enable
```

If restarting the SRC client does not resolve the problem, rebuild the router configuration and restart the client.