Chapter 10

Managing SRC Configurations

This chapter provides basic information about managing configurations. Topics include:

- How the SRC Configuration Is Stored on page 87
- Updating the SRC Configuration on page 88
- About SRC Configuration Files in XML Format on page 90
- About SRC Configuration Files in Text Format on page 93
- Preparing a File to Be Loaded into the Current SRC Configuration on page 94
- Loading an SRC Configuration on page 96
- Reverting to a Previous SRC Configuration on page 102
- Cutting and Pasting Configuration Information at the SRC CLI on page 102

How the SRC Configuration Is Stored

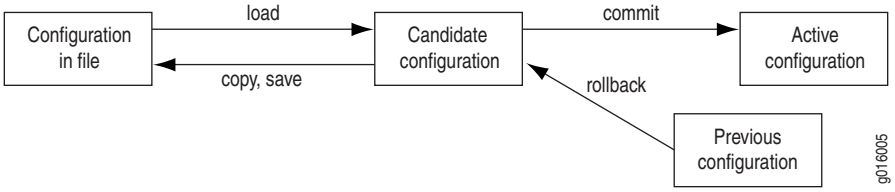
When you edit a configuration, you work in a copy of the current configuration to create a candidate configuration. Changes you make to the candidate configuration are visible in the CLI immediately. If multiple users edit the configuration at the same time, all users can see all changes.

To have a candidate configuration take effect, you *commit* the changes. At this point, the software verifies the candidate configuration for proper syntax. If multiple users are editing the configuration, when you commit the candidate configuration, all changes made by all the users take effect.

Slot (local) configuration is stored in files, and the remainder of the configuration is stored in the Juniper Networks Database or another directory that you have configured to store SRC configuration data.

Figure 10 illustrates the various configuration states and the configuration mode commands that you use to load, commit, copy, and save the configuration.

Figure 10: Commands for Storing and Modifying the Configuration



Updating the SRC Configuration

You can update the SRC configuration to include configuration changes from a file or to revert to the configuration supplied with the SRC product. You can also retain the active configuration and discard changes not yet committed. After you load one of these configurations, you can commit it to activate the configuration on the C-series Controller, or you can edit the configuration interactively using the CLI and commit it at a later time.

Before You Load a Configuration

Before you load a configuration, make a copy of the current configuration. This configuration contains the active configuration plus any configuration changes that have been made at the CLI. For information about the syntax in the files, see:

- Text format—*SRC-PE CLI Command Reference, Volume 1* and *SRC-PE CLI Command Reference, Volume 2*
- XML format—*SRC-PE XML API Configuration Reference*

You can save the configuration to text or XML format. By default, the configuration is saved to a file in XML format.

To make a backup copy of the configuration:

- From the [edit] hierarchy level of configuration mode, save the configuration to XML format. For example:

  ```
  [edit]
  user@host> save backupcfg.xml format xml
  ```

- From the [edit] hierarchy level of configuration mode, save the configuration to text format. For example:

  ```
  [edit]
  user@host> save backupcfg.txt format text
  ```

Commands to Load a Configuration

You can use the following commands in configuration mode to make configuration changes:

- load factory-default—Replace the existing configuration with the configuration supplied with the SRC software.
- load merge *filename* <relative>—Combine the configuration that is currently shown in the CLI and the configuration in the specified file.
- load override *filename* —Discard the entire configuration that is currently shown in the CLI, and load the entire configuration in the specified file.
- load replace *filename* <relative>— Look for replace attributes in the specified file, delete the existing statements of the same name, and replace them with the configuration in the specified file.
- load set *filename* <relative>—Execute configuration mode commands such as set, edit, exit, and top from a text file.

The relative option for the load merge, load replace, and load set commands lets you load the configuration at a specified hierarchy level.

The load merge, load override, and load replace commands let you update configuration statements in the SRC configuration from a text file or an XML file. The structure of this file must conform to the structure for an SRC configuration file. For this reason, we recommend that you copy the file based on the file format you plan to use:

- XML format—Save a configuration to an XML file and modify that file.
- Text format—Save a configuration to a text file, or copy the output from a show command to a file, and modify that file.

For a merge or replace operation, you can save a copy of the configuration at any level in the configuration hierarchy, then load the updated configuration at the same level.

Use the editor of your choice to modify a saved configuration file. When you edit a file that is to be loaded into the SRC configuration, you can add specified attributes to specify actions to be taken.

Attributes in SRC Configuration Files

You can add the following attributes to text files or to XML tags in a configuration file to be loaded through the load merge, load override, and load replace commands. If you do not add any attributes, the software merges all changes.

- operation="create"—Create the specified configuration.

  If you try to create a configuration object that already exists, the software does not create the new objects and generates an error message to that effect.

- operation="delete"—Delete the specified configuration.

- operation="merge"—Merge the specified configuration.

- operation="replace"—Replace a specified configuration with another defined configuration.

  If the replace attribute is in the file whose contents are merged, the command disregards the replace attribute.

About SRC Configuration Files in XML Format

The XML structure follows the same hierarchy as the CLI. For example, in configuration mode the following statements are available at the [edit system] hierarchy level:

```
[edit system]
user@host# set ?
Possible completions:
+ authentication-order Order in which authentication methods are invoked
+ domain-search        List of domain names to search
  host-name            Hostname for C-series Controller
> ldap                 LDAP properties
> login                Login properties
+ name-server          DNS name servers
> ntp                  Configure NTP
> radius-server        RADIUS server configuration
> services             System services configuration
> syslog               System log configuration
> tacplus-server       TACACS+ server configuration
  time-zone            Time zone definition name
```

In an XML file, the tags within the <system> tags are the same as the statements in the [edit system] hierarchy. The tags under <system> can appear in any order.

```
<configuration>
    <system>
      <authentication-order> </authentication-order>
      <domain-search> </domain-search>
      <host-name> </host-name>
      <ldap> </ldap>
      <login> </login>
      <name-server> </name-server>
      <ntp> </ntp>
      <radius-server> </radius-server>
      <services> </services>
```

```
        <syslog> </syslog>
        <tacplus-server> </tacplus-server>
        <time-zone> </time-zone>
      </system>
</configuration>
```

The following example shows parts of a configuration file for statements in the [edit system] hierarchy:

```
<?xml version="1.0"?>
<configuration>
    <system>
      <time-zone>Canada/Eastern</time-zone>
      <services>
        <telnet/>
        <ssh>
          <root-login>allow</root-login>
        </ssh>
      </services>
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.20</name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>juniper.net</domain-search>
      <ntp>
        <server> <address>192.2.2.100</address>
        </server>
        <boot-server>192.2.2.100</boot-server>
      </ntp>
      <ldap>
        <server>
          <address>10.227.2.100</address>
        </server>
        <boot-server>10.227.2.100</boot-server>
      </ldap>
      <ldap>
        <server>
          <community>
            <primary-neighbors>neighbor1</primary-neighbors>
            <role>primary</role>
          </community>
        </server>
      </ldap>
      <ldap>
        <client>
          <connection-manager-id>CLI_DATA_MANAGER
          </connection-manager-id>
        . . .
        </client>
      </ldap>
      <login>
        <class>
          <name>class-cfg</name>
          <allow-configuration>s.*m$|s.*m l.*n</allow-configuration>
          <permissions>configure</permissions>
          <permissions>interface</permissions>
        </class>
```

```
            <user>
              <user-name>admin</user-name>
              <class>super-user</class>
              <full-name>admin</full-name>
              <uid>500</uid>
              <gid>100</gid>
              <authentication>
                . . .
              </authentication>
              <level>normal</level>
              <complete-on-space>on</complete-on-space>
            </user>
          </login>
          <syslog>
            . . .
          </syslog>
        </system>
    </configuration>
```

Example: Using Attributes When Editing an XML Configuration File

You can modify a single value by inserting an attribute into one tag. For example, to delete the name server that has the IP address 192.2.2.20:

```
<configuration>
    <system>
      <name-server operation="delete">192.2.2.20</name-server>
    </system>
</configuration>
```

You can also modify a number of values within a hierarchy by adding an attribute at a higher level in the hierarchy. For example, to replace permissions for the class named class-cfg in the following configuration:

```
<configuration>
    <system>
        <class>
          <name>class-cfg</name>
          <allow-configuration>s.*m$|s.*m l.*n</allow-configuration>
          <permissions>configure</permissions>
          <permissions>interface</permissions>
        </class>
    </system>
</configuration>
```

Enter the replace attribute for the class:

```
<configuration>
    <system>
      <login>
        <class operation="replace">
          <name>class-cfg</name>
          <allow-configuration>s.*m$|s.*m l.*n</allow-configuration>
          <permissions>control</permissions>
          <permissions>maintenance</permissions>
        </class>
      </login>
    </system>
</configuration>
```

About SRC Configuration Files in Text Format

You can create a configuration file in text format by saving the configuration to a file in text format or by running the show command at a specified hierarchy level, and then copying the output into a text file. The hierarchical format you see when you run a show command shows the statement hierarchy as it appears in a text file.

You can also create a text file that includes configuration mode commands to be executed and then load this file though the load set command. Use the editor of your choice to create the text file.

For example, to add a name server that has the IP address 192.2.2.30 and to delete the name server that has the IP address 192.2.2.20 add the following lines to a text file:

```
edit system
set name-server 192.2.2.30
delete name-server 192.2.2.20
```

Example: Using Attributes When Editing a Text Configuration File

You can modify a single value by inserting an attribute. For example, to delete the name server that has the IP address 192.2.2.20:

```
configuration{
    system{
      delete:
      name-server 192.2.2.20;
    {
{
```

You can also modify a number of values within a hierarchy by adding an attribute at a higher level in the hierarchy. For example, to replace permissions for the class named class-cfg in the following configuration:

```
configuration{
    system{
        class{
            name class-cfg;
            allow-configuration s.*m$|s.*m l.*n;
            permissions configure;
            permissions interface;
        }
    }
}
```

Enter the **replace** attribute for the class:

```
configuration{
    system{
        login{
            replace:
            class{
                name class-cfg;
                allow-configuration s.*m$|s.*m l.*n;
                permissions control;
                permissions maintenance;
            }
        }
    }
}
```

Preparing a File to Be Loaded into the Current SRC Configuration

When you save your current configuration to a file, the file contains the configuration in its current form, including any uncommitted changes. If more than one user is modifying the configuration, all changes made by all users are saved.

When you save a configuration to a file, the contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it. When you save a file to XML format, the software inserts a line in the saved file to indicate the level at which the file was saved. For example:

```
<configuration>
    <system>
        <services sdx:current="true">
            <ssh>
                <root-login>deny</root-login>
            </ssh>
            <editor>
                <password-encryption>sha</password-encryption>
            </editor>
        </services>
    </system>
</configuration>
```

The file is saved in the current working directory. When you load a file that was saved at a specific hierarchy level, use the relative option for a load command. If you do not use the relative option, the command disregards the sdx:current="true" text in XML files.

If you plan to copy a configuration file from the C-series Controller to another system and back, make sure that you have SSH or Telnet enabled on the C-series Controller.

The examples in the following procedure show how to prepare a file in XML format; the procedure is the same for files in text format.

To prepare a configuration file for loading into the SRC configuration:

1. In configuration mode, navigate to the level at or below which you want to save the configuration.

2. Run the save command.

   For example:

   ```
   [edit system]
   user@host# save systemcfg.xml
   172 lines written to systemcfg.xml
   [edit system]
   ```

3. Edit the file.

   On a C-series Controller:

   a. Copy the file to a remote system, and then edit it. For example:

```
user@host> file copy /root/systemcfg.xml ftp://user@myserver/systemcfg.xml
Password:

user@host>
```

      For information about specifying the filename, see *Chapter 5, Using the SRC CLI Operational Commands to Monitor the SRC Software*.

   b. Edit the file in the editor of your choice.

c. Copy the edited file back to the C-series Controller. For example:

```
user@host> file copy ftp://user@myserver/systemcfg.xml /root/systemcfg2.xml
Password:

user@host>
```

On a Solaris platform:

- Edit the file in the text editor of your choice.

Related Topics

- For information about enabling SSH and Telnet on the C-series Controller, see *SRC-PE Getting Started Guide, Chapter 10, Managing SRC Configurations*.

Loading an SRC Configuration

You can use the load commands to perform the following tasks:

- *Replacing the Current Configuration with the Default SRC Configuration* on page 96
- *Merging the Active Configuration with Another Configuration* on page 96
- *Replacing the Configuration* on page 98
- *Replacing Parts of the Configuration* on page 99
- *Adding a Configuration Through Configuration Mode Commands* on page 101
- *Loading a Configuration at a Specified Hierarchy Level* on page 101

Replacing the Current Configuration with the Default SRC Configuration

To restore the full default SRC configuration:

- In configuration mode, enter the load factory-default command.

```
[edit]
user@host# load factory-default
```

This command removes the active configuration and replaces it with the basic, default SRC configuration.

Merging the Active Configuration with Another Configuration

A merge operation is useful when you are adding a new section to an existing configuration. If the existing configuration and the incoming configuration contain conflicting statements, the statements in the incoming configuration override those in the existing configuration.

You can merge a configuration from files in XML or text format. The examples in this section use files in XML format.

You can merge all of the configuration, or the configuration at a specified hierarchy level. For information about loading a configuration at a specified hierarchy level, see *Loading a Configuration at a Specified Hierarchy Level* on page 101.

To combine the active configuration and the configuration in a specified file:

- In configuration mode, specify the load merge command. For example:

```
[edit]
user@host# load merge newcfg.xml format xml
```

The following example shows part of an existing configuration, the configuration in the file to be loaded, and the resulting configuration. In the resulting configuration, bold text indicates the configuration that changed.

Existing configuration:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.20</name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>juniper.net</domain -search>
      . . .
    </system>
    . . .
</configuration>
```

Configuration in the file to be loaded:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.30</name-server>
      <domain-search>newdomain.juniper.net
      </domain-search>
      . . .
      </system>
    . . .
</configuration>
```

Resulting configuration:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.20</name-server>
      <name-server>192.2.2.30</name-server>
```

```
        <domain-search>mydomain.juniper.net</domain-search>
        <domain-search>juniper.net</domain-search>
        <domain-search>newdomain.juniper.net</domain-search>
        . . .
      </system>
      . . .
</configuration>
```

Replacing the Configuration

You can replace a configuration from files in XML or text format. The examples in this section use files in XML format.

To replace all of the active configuration with a full configuration in a specified file:

- In configuration mode, specify the load override command. For example:

  ```
  [edit]
  user@host# load override complete-newcfg.xml format xml
  ```

  When you use the load override command and commit the configuration, all system processes reparse the configuration.

The following example shows part of an existing configuration, the configuration in the file to be loaded, and the resulting configuration. In the resulting configuration, bold text indicates the configuration that changed.

Existing configuration:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.20</name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>juniper.net</domain -search>
      . . .
</system>
    . . .
</configuration>
```

Configuration in the file to be loaded:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.30</name-server>
      <name-server>192.2.2.40</name-server>
      <domain-search>newdomain.juniper.net
      </domain-search>
      . . .
    </system>
    . . .
</configuration>
```

Resulting configuration:

```
<configuration>
    . . .
    <system>
    . . .
    <host-name>myhost</host-name>
      <name-server>192.2.2.30</name-server>
      <name-server>192.2.2.40</name-server>
      <domain-search>newdomain.juniper.net</domain-search>
    . . .
      </system>
    . . .
</configuration>
```

Replacing Parts of the Configuration

A replace operation searches for replace attributes in the specified file, deletes the existing statements of the same name, if any, and replaces them with the incoming configuration. If there is no existing statement of the same name, the replace operation adds to the configuration the statements marked with the replace attribute. You can also use create, delete, and merge attributes in the file.

If you are performing a replace operation and the file you specify does not contain any replace attributes, the replace operation is effectively equivalent to a merge operation. This type of operation might be useful if you are running automated scripts and cannot know in advance whether the scripts need to perform a replace or a merge operation. The scripts can use the replace operation to cover either case.

You can merge a configuration from files in XML or text format. The examples in this section use files in XML format.

You can replace all of the configuration, or the configuration at a specified hierarchy level. For information about loading a configuration at a specified hierarchy level, see *Loading a Configuration at a Specified Hierarchy Level* on page 101.

To replace portions of a configuration:

1. Make sure that the incoming configuration file has replace attributes in place for each part of the configuration to be replaced.

   See *Example: Using Attributes When Editing an XML Configuration File* on page 92.

2. In configuration mode, specify the load replace command. For example:

   user@host# **load replace newcfg.xml format xml**

The following example shows part of the existing configuration, the configuration in the file to be loaded, and the resulting configuration. In the resulting configuration, bold text indicates the configuration that changed.

For an example of a file snippet that shows how to replace a number a values within a hierarchy, see *Example: Using Attributes When Editing an XML Configuration File* on page 92.

Existing configuration:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.20</name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>juniper.net</domain -search>
      . . .
</system>
    . . .
</configuration>
```

Configuration in the file to be loaded:

```
<configuration>
    . . .
    <system>
      . . .
      <host-name>myhost</host-name>
      < name-server operation=”replace”>192.2.2.10</name-server>
      <name-server>192.2.2.30
      </name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>mydomain.juniper.net
      </domain-search>
      <domain-search>juniper.net</domain -search>
      . . .
    </system>
    . . .
</configuration>
```

Resulting configuration:

```
<configuration>
    . . .
    <system>
    . . .
      <host-name>myhost</host-name>
      <name-server>192.2.2.10</name-server>
      <name-server>192.2.2.30</name-server>
      <domain-search>mydomain.juniper.net</domain-search>
      <domain-search>juniper.net</domain -search>
      . . .
    </system>
    . . .
</configuration>
```

Adding a Configuration Through Configuration Mode Commands

When you use the load set command, it executes the configuration instructions line by line as they are stored in a file. The instructions can contain any configuration mode command, such as set, edit, exit, and top.

To load a configuration that contains configuration mode commands:

1. Create a text file that includes set and other configuration mode commands. For example:

   ```
   edit system login class name newclass permissions system
   delete system login class name newclass permissions interface
   set system login class name newclass permissions configure
   ```

2. In configuration mode, use the load set command.

   ```
   user@host# load set newcfg2.txt
   ```

Loading a Configuration at a Specified Hierarchy Level

The load merge, load replace, and load set commands let you load the configuration at a specified hierarchy level by using the relative option.

To load a configuration at a hierarchy level:

1. In configuration mode, move to the hierarchy level at which you want to load a configuration.

2. At the hierarchy level, enter a load merge, load replace, or load set command. For example:

   ```
   [edit system login class name newclass]
   user@host# load merge mynewcfg.xml relative format xml
   ```

Reverting to a Previous SRC Configuration

You can revert to the active configuration and discard configuration changes not yet committed.

To revert to the full committed configuration:

- In configuration mode, at the [edit] hierarchy level enter the rollback command.

  user@host> **rollback**

Cutting and Pasting Configuration Information at the SRC CLI

You can also create a configuration by cutting and pasting existing portions of the configuration. You can copy configuration text from another source or from another part of the configuration to a new location. Use the cut and paste functions for your windowing system, such as X Windows.