

## **Before You Configure SRC Policies**

---

Building policies is a top-down operation. For example, before you can add a subordinate to the policy group, the policy group itself must exist.

### ***Creating a Worksheet***

Before you configure policies, you must determine what information you want to enter and where it will go. It is best to create a worksheet where you can record such things as names, priorities, addresses, and so on.

To create a worksheet:

1. Determine the policy requirements for your system.
2. Consider information that contains (at a minimum) names and parameters for:
  - Policy group
  - Policy list
  - Policy rules
  - Conditions
  - Actions
  - Record the policy information about the worksheet.

### ***Naming Objects***

Object names must be unique and must conform to LDAP distinguished name (DN) constraints.

### ***Using the `apply-groups` Statement***

When you use the `apply-groups` statement on the JUNOS routing platform to apply a configuration group to a hierarchy level in a configuration, you need to make sure that the SAE configuration group (default name is `sdx`) is in the first position in the `apply-groups` statement.

### ***Using Expressions***

Many of the policy options can take expressions in addition to literal values. If you can enter an expression for an option, the expression type is noted in the documentation for the command. For information about using and formatting expressions, see Expressions in Parameters.

### ***Policy Values***

As you are planning your policy configuration, you need to understand how invalid values in policies are handled on JUNOS routing platforms and JUNOSe routers.

## SAE to JUNOS Routing Platforms

When the SAE sends policies to JUNOS routing platforms, it uses JUNOScript on the Blocks Extensible Exchange Protocol (BEEP), which is an XML-based protocol. Many of the configuration values in JUNOScript are strings in which the value is a number. If you enter an integer value that is too large, the policy software flags the entry as invalid, but the value is still sent to the router because JUNOScript on BEEP allows for its transmission. The router is the authority that decides whether values are valid for the particular version of the JUNOS software and the routing platform. If the value is too large, the router sends an error message to the SAE.

For example, the valid range for the burst size limit in a policer action is 1,500 to 100,000,000. If you specify a value greater than 100,000,000, it is flagged as invalid. However, as usual, the SRC software attempts to activate the service, but the activation will fail because the burst size is an invalid value on the router.

## SAE to JUNOSe Routers

When the SAE sends policies to JUNOSe routers, it uses the Common Open Policy Service (COPS) protocol with specific standard Policy Information Bases (PIBs) and private PIBs. Many of the configuration values in the PIBs are not strings in which the value is a number. Sometimes the numeric range in the PIB is larger than the valid range of values on the router. For integer values in policies, the eventual policy on the router has only the portion of a value that can be converted to an integer in the usable range.

The example below for ToS byte is such a case. From the JUNOSe-IP-PIB:

```
JunoselpPolicyClacIRuleEntry ::= SEQUENCE {  
    junoselpPolicyClacIRuleTosByte Integer32,  
    junoselpPolicyClacIRuleTosMask Integer32,  
}
```

If a policy has a literal ToS byte value of 300, the high bits are ignored (or a mask of 255 is used) so that the value used for the ToS byte is 44; that is, 300 minus 256.