

Classifying Subscribers (SRC CLI)

Changes that you make to subscriber classification scripts do not affect subscriber sessions that are already established. One effect of this behavior is that static IP subscriber sessions are not closed if the classification script is changed in a way that would no longer cause the SAE to load a profile for certain subscribers.

On JUNOSe routers that use the COPS-PR or COPS XDR router drivers, you can create a subscriber session for the router interface to start services such as script services and aggregate services. The SAE creates the router interface, but does not install any policies on it. You can create a subscriber classification rule, but not an interface classification rule for this interface.

Use the following configuration statements to define subscriber classification scripts:

```
shared sae subscriber-classifier rule name {  
    target target ;  
    script script ;  
    include include ;  
}  
shared sae subscriber-classifier rule name condition name ...
```

A classification script can contain either a target and a condition or a script. If you do not define a script, the classifier must have both a target and a condition.

To define subscriber classification scripts:

1. From configuration mode, enter the subscriber classifier configuration. In this sample procedure, the subscriber classifier is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region subscriber-classifier
```

2. Create a rule for the subscriber classifier. You can create multiple rules for the classifier.

```
[edit shared sae group west-region subscriber-classifier]  
user@host# edit rule rule-2
```

3. Configure either a script, a target, or an included script for the rule.

```
[edit shared sae group west-region subscriber-classifier rule rule-2]  
user@host# set target target
```

OR

```
[edit shared sae group west-region subscriber-classifier rule rule-2]  
user@host# set script script
```

OR

```
[edit shared sae group west-region subscriber-classifier rule rule-2]  
user@host# set include include
```

Where *include* identifies a script that has already been created.

If you configure a target, see Subscriber Classification Targets.

4. If you configured a target for the rule, configure a match condition for the rule. You can create multiple conditions for the rule. See “Subscriber Classification Conditions” on page 4.

```
[edit shared sae group west-region subscriber-classifier rule rule-2]
user@host# edit condition name
```

5. (Optional) Change the order of rules.

```
[edit shared sae group west-region subscriber-classifier]
user@host# insert rule rule-5 before rule-4
```

6. (Optional) Rename a rule.

```
[edit shared sae group west-region subscriber-classifier]
user@host# rename rule rule-5 to Retailer
```

7. (Optional) Verify the classifier rule configuration.

```
[edit shared sae group west-region subscriber-classifier rule rule-2]
user@host# show
target <-unauthenticatedUserDn->;
condition {
  "loginType == \"ADDR\"";
  "loginType == \"AUTHADDR\"";
}
```

8. (Optional) Verify the subscriber classifier configuration.

```
[edit shared sae group west-region subscriber-classifier]
user@host# show
rule rule-1 {
  script "# User Classification script
#
# The following attributes MAY be available for comparison.
# Attributes that are not available will have the value \"\" (empty
string).
#
# loginType: one of \"INTF\", \"AUTHINTF\", \"ADDR\", \"AUTHADDR\",
#             \"PORTAL\", \"ASSIGNEDIP\"
# userName: Everything before the \"@\" in the user's login name.
# domainName: Everything after the \"@\" in the user's login name.
# serviceBundle: A RADIUS VSA available if the login event involves
#                 authentication with a properly configured RADIUS server.
# radiusClass: The RADIUS class of user's ERX interface.
# virtualRouterName: The name of the user's virtual router.
# interfaceName: The name of the user's ERX interface (e.g.
#                 \"fastEthernet3/1.0\")
# ifAlias: The alias of the user's ERX interface, as configured on the
ERX.
# ifDesc: The description of the user's ERX interface, as configured on
#          the ERX.
# nasPortId: The user's ERX interface including Layer 2 access
```



```

rule rule-2 {
  target <-unauthenticatedUserDn->;
  condition {
    "loginType == \"ADDR\"";
    "loginType == \"AUTHADDR\"";
  }
}
rule rule-3 {
  target <-retailerDn->??sub?(uniqueID=<-userName->);
  condition {
    "retailerDn != \"\"";
    "& userName != \"\"";
  }
}

```

- Related Topics**
- Classifying Subscribers (C-Web Interface)
 - Sending DHCP Options to the JUNOS Router
 - Example: Subscriber Classification Scripts for Static IP Subscriber
 - Overview of Classification Scripts
 - Overview of Subscribers

Subscriber Classification Conditions

Subscriber classification conditions define match criteria that are used to find the subscriber profile. Use the fields in this section to define subscriber classification conditions.

dhcp

- DHCP options. See Sending DHCP Options to the JUNOS Router .

domainName

- Domain name of the subscriber.
- Value—Valid domain name
- Example—domainName = “ isp99.com”

ifAlias

- Description of the interface.
- Value—Interface description that is configured on the router. For JUNOS routers, it is the description configured with the **interface description** command
- Example—ifAlias = “ dhcp-subscriber12”

ifDesc

- Alternate name for the interface that is used by SNMP. This name is a system-generated name.
- Value

- On a JUNOSe router, the format of the description is
ip<slot>/<port>.<subinterface>
- On the JUNOS routing platform, ifDesc is the same as interfaceName.
- Example—ifDesc = “ IP3/1.1”

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
 - Router for a JUNOSe router instance
- Example—For JUNOSe routers: interfaceName = “ fastEthernet6/0”

For JUNOS routing platforms: interfaceName = “ fe-0/1/0.0”

For forwarding interface: interfaceName = “ FORWARDING_INTERFACE”

loginName

- Name to be used to create a loginName attribute for a subscriber session for JUNOSe interfaces that are not otherwise assigned a loginName when a session starts, such as unauthenticated DHCP addresses, unauthenticated IP interfaces (that are not using PPP connections), or core-facing interfaces.

The loginName can also be used to identify a subscriber session through the SAE CORBA remote API.

- Value—Name in the form subscriber@domain
- Guideline—The format is not defined. A loginName can be of form subscriber, domain\subscriber, subscriber@domain, or as otherwise defined by the login setup of the operator.
- < Login name >
- Example—idp@idp

loginType

- Type of subscriber session to be created.
- Value—One of the following login types:

- ASSIGNEDIP—For assigned IP subscribers. Triggered when an application accesses a subscriber object for an assigned IP subscriber that is not currently loaded into memory. (Supported on JUNOSe routers.)
- AUTHINTF—For authenticated interface login requests. Triggered when a login Name is reported together with the interface, such as authenticated PPP or autoconfigured ATM interface, by means of the **subscriber** command. (Supported on JUNOSe routers.)
- INTF—For unauthenticated interface login requests. Triggered when an interface comes up and the interface classification script determines that the SAE should manage the interface. (Supported on JUNOS routing platforms and JUNOSe routers.)
- ADDR—For unauthenticated address login requests. Triggered when the DHCP server in the JUNOSe router provides an unauthenticated IP address. (Supported on JUNOSe routers.)
- AUTHADDR—For authenticated address login requests. Triggered when the DHCP server in the JUNOSe router provides an authenticated IP address. (Supported on JUNOSe routers.)
- PORTAL—Triggered when the portal API is invoked to log in a subscriber. (Supported on JUNOS routing platforms and JUNOSe routers.)
- Example—loginType = “ AUTHADDR”

macAddress

- String representation of the DHCP subscriber media access control (MAC) address.
- Value—Valid MAC address
- Example—macAddress = “ 00:11:22:33:44:55”

nasPortId

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—nasPortId = “ fastEthernet 3/1” (There is a space between fastEthernet and slot number 3/1 in the nasPortId.)

radiusClass

- RADIUS class used for authorization.
- Value—RADIUS class name
- Example—radiusClass = “ Premium”

retailerDn

- DN of the retailer object. The object is found when the domain name is mapped to a retailer object in LDAP.
- Value—DN of a retailer

serviceBundle

- Content of the vendor-specific RADIUS attribute for the service bundle.
- Value—Name of a service bundle
- Example—serviceBundle = “ goldSubscriber”

unauthenticatedUserDn

- DN of the unauthenticated subscriber profile (usable for target expressions only).
- Value—DN of a subscriber profile

userName

- Name of the subscriber.
- Value—Subscriber name without the domain name
- Example—userName = “ peter”

virtualRouterName

- Name of the virtual router or routing instance.
- Value—For JUNOS routers: name of the virtual router in the format vname@hostname
For JUNOS routing platforms: name of the routing instance
- Example—virtualRouterName = “ default@e_series5”

