Chapter 8

Overview of Using Local and Global Parameters

This chapter provides an overview of using local and global parameters in policies. Topics include:

- Overview of Global and Local Parameters on page 189
- Parameter Types on page 190

Overview of Global and Local Parameters

Policy definitions are templates that the policy engine uses to construct policies that the SAE installs on the router or provisions on the CMTS device. When you configure the policy template, you can assign parameter values. Before it creates a policy for installation on the router, the policy engine substitutes parameter values with specific values. The policy engine uses the parameter value acquisition process to obtain the specific values. For information about parameter value acquisition, see *Generating Policies by Specifying Parameters* on page 395.

Policies can use global or local parameters:

- Global parameters—Are available to use in any policy. With global parameters, you can define parameters once and then reuse them in many policies. Typically, global parameters are not changed often, and if changes are necessary, local parameters are used.
- Local parameters—Are available only for the policy group in which the parameter is defined.

The SRC software provides many predefined built-in parameters and runtime parameters. Runtime parameters are built-in parameters that are filled in with an actual value from the running system when the policy is installed on the router. For example, the interface_speed parameter is filled in with the actual speed of the router interface. You cannot change the values of built-in or runtime parameters.

Parameter Types

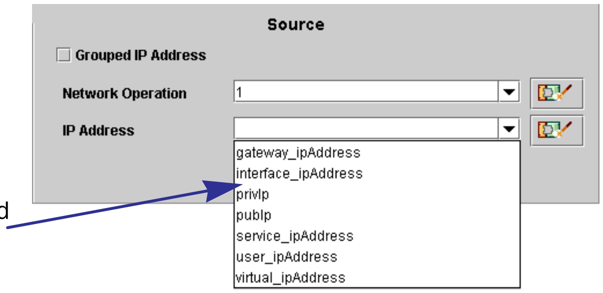
Global and local parameters are assigned a type. (Note that the term *type* is used in the SRC CLI and Policy Editor, and the term *role* is used in SDX Admin. Both terms have the same meaning.) The type indicates in which options in the SRC CLI or which Policy Editor fields you can use the parameter.

For example, address is a type of parameter. In the SRC CLI, whenever there is an option for which you can specify an IP address, you can use the ? to display a list of all local and global parameters of type address. For example:

```
user@host# set source-network network ip-address ?
Possible completions:
  <ip-address>        IP address of the source or destination network or host
  gateway_ipAddress
  interface_ipAddress
  service_ipAddress
  user_ipAddress
  virtual_ipAddress
```

In Policy Editor, wherever there is a field for which you can specify an address, a drop-down list displays all the global parameters of type address as well as local parameters of type address that are defined in the policy group in which you are working. Figure 26 shows a drop-down list of global and local parameters for parameter type address.

Figure 26: Drop-Down List of Local and Global Address Parameters in Policy Editor



Global and local parameters defined for parameter type address

There are a few cases in which a global parameter value appears, but because of the context, the value does not make sense to use. For example, in NAT actions, the global parameter any appears in for the IP network setting. In this context, any is not a valid value.

Table 23 lists the parameter types, the predefined parameters for each type, the policy objects in which you can use the parameter type, and how the type is used.

Table 23: Parameter Types (or Roles)

Type	Predefined Parameters	Used In	Used to Specify
address	gateway_ipAddress interface_ipAddress service_ipAddress user_ipAddress virtual_ipAddress	Classify-traffic condition Next-interface action Next-hop action	IP addresses in dotted decimal notation.
addressMask	interface_ipMask service_ipMask user_ipMask	Classify-traffic condition	IP masks in dotted decimal notation. For JUNOSe policies and JUNOS policies (except for firewall policies), a mask must be equivalent to some prefix length. For example, 255.255.255.0 is allowed, but 255.255.255.1 is not. Policy Editor searches this constraint for default parameter values, but not for any other substitution values until runtime when the policy engine constructs the policy.
allowIpOptions		Classify-traffic condition	
any			The set of all values.
applicationProtocol	bootp, dce_rpc, dce_rpc_portmap, dns, exec, ftp, h323, green, icmp_app, iiop, netbios, netshow, realaudio, rpc, rpc_portmap, rtsp, shell, snmp, sqlnet, tftp, traceroute, winframe, yellow	Classify-traffic condition (Predefined parameters map protocol numbers to synonyms.)	
bandwidthSizeUnit	bps percent	Policer action	
boolean	false true		
burst		Rate-limit action Policer action DOCSIS action	Burst sizes. The range is 2^{14}—2^{32}–1.
color	green red yellow	Classify-traffic condition Color action	Color of action or classifier. The policy engine and Policy Editor validate these values; the substitution engine does not.
dceRpcUuid		Classify-traffic condition	
dropProfileProtocol	any_protocol non_tcp tcp_only	Scheduler action	
dropProfileType	interpolated segmented	Scheduler action	
forwardingClass		Classify-traffic condition QoS condition	

Table 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
fragOffset		Classify-traffic condition	The value of the fragment offset field of IP packets. For JUNOSe routers: ■ eq 0—Equal to 0 ■ eq 1—Equal to 1 ■ gt 1—Greater than 1 ■ any—Any value For JUNOS routing platforms and PCMM policies, integer in the range 0–8191. The policy engine and Policy Editor validate these values; the substitution engine does not.
grantSize		DOCSIS action	
icmpCode icmpType		Classify-traffic condition	8-bit values that represent patterns in the ICMP code and ICMP type fields in IP packets. The policy engine and Policy Editor validate these values; the substitution engine does not.
igmpType		Classify-traffic condition	8-bit values that represent patterns in the IGMP type field in IP packets. The policy engine and Policy Editor validate these values; the substitution engine does not.
interfaceGroup		Classify-traffic condition	
InterfaceSpec	bfwIf gfwIf	Next-interface action	The router interface. For JUNOSe interfaces, the format is: ‘<type of specifier> = <value>’ For example: name=‘fastEthernet3/0’ For JUNOS interfaces, the format is: ‘name= <mediatype>-<slot>/<pic>/<port>.<unit>’ For example: ‘name=AT-0/1/0.0’
interval		DOCSIS action	
ipFlags ipFlagsMask		Classify-traffic condition	3-bit values that represent patterns for the IP flags field in an IP packet. The high bit is reserved, the middle bit is don’t fragment, and the low bit is more fragments.
ipSecSpi		Classify-traffic condition	
IPv4range			
jitter		DOCSIS action	
matchDirection	both input output	Classify-traffic condition	
maxLatency		DOCSIS action	
messageType		Reject action	

Table 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
microSecond			
natTranslationType		NAT action	
network	any	Classify-traffic condition NAT action	IP subnets using two forms: <address>/<mask> <address>/<prefixLength> where <address> and <mask> are in the traditional dotted decimal notation. <prefixLength> is a number in the range 0–32, which specifies how many of the first bits in the address specify the network. In policy conditions, network specifies patterns for the address fields in packets. Networks can be preceded by “not” to indicate that the condition matches every address not in the subnet.
networkOperation		Classify-traffic condition	Whether a network field of a packet should match or not match the value specified in a policy condition. ■ 0—Does not match ■ 1—Matches
packetLength		Classify-traffic condition DOCSIS action FlowSpec action	
packetLossPriority	any_priority high_priority low_priority	Loss priority action	
packetOperation		Rate-limit action Policer action Stateful firewall	Actions taken on packets. For rate-limit actions, valid values are: $’forward’, $’filter’, and $’mark <tosByte> <tosMask>’. For policer actions, valid values are: filter, forwardingClass, lossPriority. For stateful firewalls, valid values are: filter, forward, reject. The policy engine and Policy Editor validate these values; the substitution engine does not.
percent		Scheduler action	
policedUnit		FlowSpec action	
port	service_port	Classify-traffic condition NAT action	16-bit values that represent patterns in the port fields in IP packets.

Table 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
portOperation	eq neq	Classify-traffic condition	Whether a port field should match or not match the value(s) specified in a condition. For JUNOSe policies valid values are: $'eq', $'lt', $'gt', $'neq' and $'range'. For JUNOS the allowed values are: ■ 0—Does not match ■ 1—Matches The policy engine and Policy Editor validate these values; the substitution engine does not.
prPrecedence		Policy rule	
protocol	ah, egp, esp, gre, icmp, igmp, ip, ipip, ospf, pim, rsvp, tcp, udp	Classify-traffic condition (Predefined parameters map protocol numbers to synonyms.)	8-bit values that represent patterns in the protocol field in IP packets. The policy engine and Policy Editor validate these values; the substitution engine does not.
protocolOperation	is not	Classify-traffic condition	Whether a protocol field of a packet should match or not match the value specified in a policy condition. ■ 0—Does not match ■ 1—Matches
qosProfileSpec		QoS-attachment action	Strings in QoS attachment actions that specify QoS profiles. They can be any string that names a QoS profile on the JUNOSe router.
rate	interface_speed	Rate-limit action Policer action DOCSIS action FlowSpec action Traffic-shape action	Rates in the range 0—2^{32}–1.
rateLimitType	one_rate two_rate	Rate-limit action	Rate-limit type. The allowed values are $'one-rate' and $'two-rate'. The policy engine and Policy Editor validate these values; the substitution engine does not.
requestTransmissionPolicy		DOCSIS action	
routingInstance		Routing instance action	
rpcProgramNumber		Classify-traffic condition	
schedulerBufferSize		Scheduler action	
schedulerBufferSizeUnit	buffer_size_percentage buffer_size_remainder temporal	Scheduler action	

Table 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
schedulerPriority	high low medium_high medium_low strict_high	Scheduler action	
schedulerTransmitRate		Scheduler action	
schedulerTransmitRateUnit	rate_in_bps rate_in_percentage rate_in_remainder	Scheduler action	
serviceClassName		Service class name action	
serviceNumber	controlled_load_service guaranteed_service	FlowSpec action	
sessionClassIdPriority		GateSpec action	
slackTerm		FlowSpec action	
snmpCommand	get get_next set trap	Classify-traffic condition	
tcpFlags tcpFlagsMask		Classify-traffic condition	6-bit values that represent patterns for the TCP flags field in IP packets. The bits from high to low mean: urgent, acknowledge, push, reset, synchronize, finish.
timeout		Classify-traffic condition	
tokenBucketSize		FlowSpec action	
tosByte tosByteMask		Classify-traffic condition Rate-limit action Mark action	8-bit values that represent patterns in the ToS byte field in IP packets. When tosByteMask is used in ToS conditions, the allowed values are 0, 224, 252, and 255. The policy engine and Policy Editor validate these values; the substitution engine does not.
traceRouteTtlThreshold		Classify-traffic condition	
trafficClassSpec		Traffic-class action	Strings in traffic-class actions that specify traffic-class profiles. They can be any string that names a traffic class on the JUNOSe router.
trafficPriority		DOCSIS action	

Table 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
trafficProfileType	best_effort unsolicited_grant down_stream unsolicited_grant_with_activity_detection real_time non_real_time	DOCSIS action	Service flow scheduling type
translationType			
userPacketClass		User packet class action	4-bit value. For JUNOSe IPv6 policies, valid values are in the range 0–15. The policy engine and Policy Editor validate these values; the substitution engine does not.

Predefined Global Parameters

Table 24 describes the predefined built-in and runtime global parameters that the SRC software provides. Only three of the predefined parameters can be modified: any, bfwlf, and gfwlf.

Table 24: Predefined Global Parameters

Predefined Parameter	Description	Type	Runtime
ah	Maps protocol 51 to AH	protocol	
any	This network matches any address	network	
any_priority	Sets packet loss priority to “any”	packetLossPriority	
any_protocol	Sets drop profile protocol to “any”	dropProfileProtocol	
best_effort	Sets the service flow scheduling type to best effort	trafficProfileType	
bfwlf	Specifier of the interface that leads to the bronze firewall server	interfaceSpec	Yes
bootp	Specifies the BOOTP protocol	applicationProtocol	
both	Specifies the direction of the policy as input and output	matchdirection	
bps	Specifies that the indicated bandwidth size is in bps	bandwidthSizeUnit	
buffer_size_percentage	Specifies that the indicated buffer size is a percentage	schedulerBufferSizeUnit	
buffer_size_remainder	Specifies that the indicated buffer size is a remainder	schedulerBufferSizeUnit	
controlled_load_service	Specifies that the type of FlowSpec service is controlled-load service	serviceNumber	
dce_rpc	Specifies the DCE RPC protocol	applicationProtocol	
dce_rpc_portmap	Specifies the DCE RPC portmap	applicationProtocol	
dns	Specifies the DNS protocol	applicationProtocol	
down_stream	Sets the service flow scheduling type to downstream	trafficProfileType	
egp	Maps protocol 8 to EGP	protocol	

Table 24: Predefined Global Parameters (continued)

Predefined Parameter	Description	Type	Runtime
eq	Matches packets with a port that is equal to the specified port	portOperation	
esp	Maps protocol 50 to ESP	protocol	
exec	Specifies the Exec protocol	applicationProtocol	
false	Sets Boolean values to false	boolean	
ftp	Specifies the FTP protocol	applicationProtocol	
gateway_ipAddress	IP address of the gateway as specified by the service object	address	Yes
get	Specifies the get SNMP command	snmpCommand	
get_next	Specifies the get-next SNMP command	snmpCommand	
gfwlf	Specifier of the interface that leads to gold firewall server	interfaceSpec	Yes
gre	Maps protocol 47 to GRE	protocol	
green	Specifies the color that indicates a low drop preference	color	Yes
guaranteed	Specifies that the type of FlowSpec service is guaranteed service	serviceNumber	
h323	Specifies the H.323 protocol	applicationProtocol	
high	Sets the scheduler priority to high	schedulerPriority	
high_priority	Sets the packet loss priority (PLP) to high	packetLossPriority	
icmp	Maps protocol 1 to ICMP	protocol	
icmp_app	Specifies the ICMP protocol	applicationProtocol	
igmp	Maps protocol 2 to IGMP	protocol	
iiop	Specifies the Internet Inter-ORB Protocol, a TCP protocol	applicationProtocol	
input	Specifies the direction of the policy as input	matchdirection	
interface_ipAddress	IP address of the interface	address	Yes
interface_ipMask	IP mask of the interface	addressMask	Yes
interface_speed	Speed of the subscriber's IP interface on the router or the speed of the subscriber's DOCSIS interface	rate	
interpolated	Sets the drop profile type to interpolate	dropProfileType	
ip	Maps protocol 0 to IP	protocol	
ipip	Maps protocol 4 to IP-IP	protocol	
is	Matches packets with the protocol that is equal to the specified protocol	protocolOperation	
low	Sets scheduler priority to low	schedulerPriority	
low_priority	Sets packet loss priority to low	packetLossPriority	
medium_high	Sets scheduler priority to medium-high	schedulerPriority	
medium_low	Sets scheduler priority to medium-low	schedulerPriority	
neq	Matches packets with a port that is not equal to the specified port	portOperation	
netbios	Specifies the NetBIOS protocol	applicationProtocol	
netshow	Specifies the NetShow protocol	applicationProtocol	

Table 24: Predefined Global Parameters (continued)

Predefined Parameter	Description	Type	Runtime
non_real_time	Sets the service flow scheduling type to NRTPS	trafficProfileType	
non_tcp	Sets the drop profile protocol to any protocol other than TCP	dropProfileProtocol	
not	Matches packets with the protocol that is not equal to the specified protocol	protocolOperation	
one_rate	Sets the rate-limit type to one rate	rateLimitType	
ospf	Maps protocol 89 to OSPF	protocol	
output	Specifies the direction of the policy as output	matchdirection	
percent	Specifies that the indicated bandwidth size is a percentage of bandwidth	bandwidthSizeUnit	
pim	Maps protocol 103 to PIM	protocol	
rate_in_bps	Specifies that the indicated transmit rate is in bps	schedulerTransmitRateUnit	
rate_in_percentage	Specifies that the indicated transmit rate is a percentage	schedulerTransmitRateUnit	
rate_in_remainder	Specifies that the indicated transmit rate is a remainder	schedulerTransmitRateUnit	
realaudio	Specifies the RealAudio protocol	applicationProtocol	
real_time	Sets the service flow scheduling type to RTPS	trafficProfileType	
red	Specifies the color that indicates a high drop preference	color	Yes
rpc	Specifies the RPC UDP or TCP protocols	applicationProtocol	
rpc_portmap	Specifies the RPC portmap protocol	applicationProtocol	
rsvp	Maps protocol 46 to RSVP	protocol	
rtsp	Specifies the Real-Time Streaming Protocol	applicationProtocol	
sctp	Maps protocol 132 to the Stream Control Transmission Protocol	protocol	
segmented	Sets the drop profile type to segmented	dropProfileType	
service_ipAddress	IP address of the service as specified by the service object	address	Yes
service_ipMask	IP mask of the service as specified by the service object	address	Yes
service_port	Service port as specified by the service object	port	Yes
set	Specifies the set SNMP command	snmpCommand	
shell	Specifies the Shell protocol	applicationProtocol	
snmp	Specifies the SNMP protocol	applicationProtocol	
sqlnet	Specifies the SQLNet protocol	applicationProtocol	
strict_high	Sets scheduler priority to strict-high	schedulerPriority	
tcp	Maps protocol 6 to TCP	protocol	
tcp_only	Sets the drop profile protocol to TCP	dropProfileProtocol	
temporal	Specifies that the indicated buffer size is temporal	schedulerBufferSizeUnit	
tftp	Specifies the Trivial File Transfer Protocol	applicationProtocol	
traceroute	Specifies the Traceroute protocol	applicationProtocol	
trap	Specifies the trap SNMP command	snmpCommand	
true	Sets the Boolean value to true	boolean	

Table 24: Predefined Global Parameters (continued)

Predefined Parameter	Description	Type	Runtime
two_rate	Sets the rate-limit type to two rate	rateLimitType	
udp	Maps protocol 17 to UDP	protocol	
unsolicited_grant	Sets the service flow scheduling type to UGS	trafficProfileType	
unsolicited_grant_with_activity_detection	Sets the service flow scheduling type to UGS-AD	trafficProfileType	
user_ipAddress	IP address of the subscriber	address	Yes
user_ipMask	IP mask of the subscriber	address	Yes
virtual_ipAddress	Virtual portal address of the SSP that is used in redundant SAE installations	address	Yes
winframe	Specifies the WinFrame protocol	applicationProtocol	
yellow	Specifies the color that indicates a medium drop preference	color	Yes

Naming Global Parameters

A global parameter is stored in the directory with the parameter name as its naming attribute. The directory stores the case for the parameter name; however, the directory does not allow you to create another global parameter with a name that differs only by the use of upper and lowercase letters. For example, if there is a parameter named fastspeed, the directory will not allow the creation of a parameter named fastSpeed without first deleting fastspeed.

Also, when you define a substitution for a global parameter, make sure that the case in the substitution matches the case of the global parameter.

When you perform a SaveAs operation to a directory with Policy Editor, the SRC software does not verify the names of local parameters in the policy group with the names of existing global parameters in the directory. After the SaveAs operation is complete, the directory may contain global parameters and local parameters with the same names. You will not receive any messages about duplicate names. If local and global parameters have duplicate names, the policy engine uses the local parameter definitions.