

Chapter 22

Interface Configuration Guidelines

For the interfaces on a router to function, you must configure them, specifying properties such as the interface location (that is, which slot the Flexible PIC Concentrator [FPC] is installed in and which location on the FPC the Physical Interface Card [PIC] is installed in), the interface type (such as Synchronous Optical Network [SONET] or Asynchronous Transfer Mode [ATM]), encapsulation, and interface-specific properties. You can configure the interfaces that are currently present in the router, and you can also configure interfaces that are not currently present but that you might add in the future. When a configured interface appears, the JUNOS software detects its presence and applies the appropriate configuration to it. For more information on the general configuration of interfaces, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

You can configure two different sets of properties at the interface level:

- Properties that apply to an entire Adaptive Services (AS) PIC interface on a global level, including default values for system logging and timeout properties.

- Assignment of service sets and filters to a network interface.

To configure default properties for the AS PIC interface, you include the following statements at the [edit interfaces] hierarchy level of the configuration:

```
[edit]
interfaces sp-fpc/pic/port {
  services-options {
    inactivity-timeout seconds;
    open-timeout seconds;
    syslog {
      host hostname {
        services severity-level;
        facility-override facility-name;
        log-prefix prefix-number;
      }
    }
  }
}
```

To apply services on network interfaces, you include the following statements at the [edit interfaces] hierarchy level of the configuration:

```
[edit]
interfaces interface-name {
  unit logical-unit-number {
    clear-dont-fragment-bit;
    encapsulation type;
    family inet {
      address address {
        ...
      }
      mtu bytes;
      service {
        input {
          [ service-set service-set-names <service-filter filter-name> ];
          post-service-filter filter-name;
        }
        output {
          [ service-set service-set-names <service-filter filter-name> ];
        }
      }
      service-domain (inside | outside);
    }
  }
}
```

This chapter contains the following sections:

Naming Services Interfaces on page 264

Configuring Interface Properties on page 266

Applying Filters and Services to an Interface on page 269

Example: Configuring a Services Interface on page 272

Naming Services Interfaces

Each interface has an interface name, which specifies the media type, the slot the FPC is located in, the location on the FPC that the PIC is installed in, and the PIC port. The interface name uniquely identifies an individual network connector in the system. You use the interface name when configuring interfaces and when enabling various functions and properties, such as routing protocols, on individual interfaces. The system uses the interface name when displaying information about the interface, for example, in the `show interfaces` command.

The interface name is represented by a physical part, a logical part, and a channel part in the following format:

physical<:*channel*>.*logical*

The channel part of the name is optional for all interfaces except Channelized DS3, E1, OC12, and STM1 interfaces.

The physical part of an interface name identifies the physical device, which corresponds to a single physical network connector. This part of the interface name has the following format:

type-fpc/pic/port

type is the media type, which identifies the network device. For service interfaces, it can be one of the following:

cp—Flow collector interface.

es—Encryption interface.

gr—Generic Route Encapsulation tunnel interface.

gre—This interface is internally generated and not configurable.

ip—IP-over-IP encapsulation tunnel interface.

ipip—This interface is internally generated and not configurable.

ls—Link services interface.

ml—Multilink interface.

mo—Monitoring services interface. The logical interface *mo-fpc/pic/port.16383* is an internally generated, non-configurable interface for routing platform control traffic.

mt—Multicast tunnel interface.

mtun—This interface is internally generated and not configurable.

sp—Adaptive services interface. The logical interface *sp-fpc/pic/port.16383* is an internally generated, non-configurable interface for routing platform control traffic.

tap—This interface is internally generated and not configurable.

vsp—Voice services interface.

vt—Virtual loopback tunnel interface.

Configuring Interface Properties

This section describes the following tasks for configuring properties specific to Adaptive Services interfaces:

Configuring the Interface Address and Domain on page 266

Configuring Default Timeout Settings on page 266

Configuring Default System Log Properties on page 267

Enabling Fragmentation on GRE Tunnels on page 268

Configuring the Interface Address and Domain

On the AS PIC, you configure a source address for system log messages by including the address statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet]
address address {
...
}
```

Assign an IP address to the interface by configuring the *address* value. The AS PIC supports only Internet Protocol version 4 (IPv4) addresses configured using the family inet statement.

For information on other addressing properties you can configure that are not specific to service interfaces, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

The service-domain statement specifies whether the interface is used within the network or to communicate with remote devices. The software uses this setting to determine which default stateful firewall rules to apply, and to determine the default direction for service rules. To configure, include the service-domain statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
service-domain (inside | outside);
```

If you are configuring the interface in a next-hop service-set definition, the service-domain setting must match the configuration for the inside-service-interface and outside-service-interface statements; for more information, see “Configuring Service Interfaces” on page 246.

Configuring Default Timeout Settings

You can specify global default settings for certain timers that apply for the entire interface. There are two statements of this type:

inactivity-timeout—Sets the inactivity timeout period for established flows, after which they are no longer valid.

open-timeout—Sets the timeout period for TCP session establishment, for use with SYN cookie defenses against network intrusion.

To configure a setting for the inactivity timeout period, include the inactivity-timeout statement at the [edit interfaces *interface-name* services-options] hierarchy level:

```
[edit interfaces interface-name services-options]
inactivity-timeout seconds;
```

The default value is 30 seconds. The range of possible values is from 4 through 86,400 seconds. Any value you configure in the application protocol definition overrides the value specified here; for more information, see “Configuring Application Protocol Properties” on page 44.

To configure a setting for the TCP session establishment timeout period, include the open-timeout statement at the [edit interfaces *interface-name* services-options] hierarchy level:

```
[edit interfaces interface-name services-options]
open-timeout seconds;
```

The default value is 30 seconds. The range of possible values is from 4 through 86,400 seconds. Any value you configure in the intrusion detection services (IDS) definition overrides the value specified here; for more information, see “Configuring Intrusion Detection Properties” on page 95.

Configuring Default System Log Properties

You specify properties that control how system log messages are generated for the interface as a whole. If you configure different values for the same properties at the [edit services service-set *service-set-name*] hierarchy level, the service-set values override the values configured for the interface. For more information on configuring service-set properties, see “Configuring System Log Properties” on page 248.

To configure interface-wide default system logging values, include the syslog statement at the [edit interfaces *interface-name* services-options] hierarchy level:

```
[edit interfaces interface-name services-options]
syslog {
  host hostname {
    services severity-level;
    facility-override facility-name;
    log-prefix prefix-number;
  }
}
```

Configure the host statement with a hostname that specifies the system log target server. The hostname local directs system log messages to the Routing Engine. For external system log servers, the hostname must be included in inet.0. You can specify only one system logging hostname.

Table 11 lists the severity levels that you can specify in configuration statements at the [edit interfaces *interface-name* services-options syslog host *hostname*] hierarchy level. The levels from emergency through info are in order from highest severity (greatest effect on functioning) to lowest.

Table 11: System Log Message Severity Levels

Severity Level	Description
any	Includes all severity levels
emergency	System panic or other condition that causes the routing platform to stop functioning
alert	Conditions that require immediate correction, such as a corrupted system database
critical	Critical conditions, such as hard drive errors
error	Error conditions that generally have less serious consequences than errors in the emergency, alert, and critical levels
warning	Conditions that warrant monitoring
notice	Conditions that are not errors but might warrant special handling
info	Events or nonerror conditions of interest

We recommend setting the system logging severity level to error during normal operation. To monitor PIC resource usage, set the level to warning. To gather information about an intrusion attack when an intrusion detection system error is detected, set the level to notice for a specific interface. To debug a configuration or log NAT functionality, set the level to info.

For more information about system log messages, see the *JUNOS System Log Messages Reference*.

To use one particular facility code for all logging to the specified system log host, include the facility-override statement at the [edit interfaces *interface-name* services-options syslog host *hostname*] hierarchy level:

```
[edit interfaces interface-name services-options syslog host hostname]
  facility-override facility-name;
```

The supported facilities include: authorization, daemon, ftp, kernel, user, and local0 through local7.

To specify an address prefix for all logging to this system log host, include the log-prefix statement at the [edit interfaces *interface-name* services-options syslog host *hostname*] hierarchy level:

```
[edit interfaces interface-name services-options syslog host hostname]
  log-prefix prefix-number;
```

Enabling Fragmentation on GRE Tunnels

To enable fragmentation of IPv4 packets in generic routing encapsulation (GRE) tunnels, include the clear-dont-fragment-bit statement and a maximum transmission unit (MTU) setting for the tunnel as part of an existing GRE configuration, as shown the following example:

```
[edit interfaces]
gr-fpc/pic/port {
  unit logical-unit-number {
    clear-dont-fragment-bit;
    ...
    family inet {
      mtu 1000;
      ...
    }
  }
}
```

This statement clears the Don't Fragment (DF) bit in the packet header, regardless of the packet size. If the packet size exceeds the tunnel MTU value, the packet is fragmented before encapsulation. The maximum MTU size configurable on the AS PIC is 9192 bytes.

Fragmentation is enabled only on IPv4 packets being encapsulated in IPv4-based GRE tunnels.



NOTE: This configuration is supported only on AS PIC GRE tunnels. If you commit gre-fragmentation as the encapsulation type on a standard Tunnel PIC interface, the following console log message appears when the PIC comes online:

```
gr-fpc/pic/port: does not support this encapsulation
```

Applying Filters and Services to an Interface

When you have defined and grouped the service rules by configuring the service-set definition, you can apply services to one or more interfaces installed on the router. To associate a defined service set with an interface, include the service-set statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet service (input | output)] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet service]
input {
  [ service-set service-set-name <service-filter filter-name> ];
  post-service-filter filter-name;
}
output {
  [ service-set service-set-name <service-filter filter-name> ];
}
```



NOTE: When you enable services on an interface, reverse-path forwarding is not supported. You cannot configure services on the management interface (fxp0) or the loopback interface (lo0).

You can configure different service sets on the input and output sides of the interface. However, for service sets with bidirectional service rules, you must include the same service set definition in both the input and output statements. Any service set you include in the service statement must be configured with the interface-service statement at the [edit services service-set *service-set-name*] hierarchy level; for more information, see “Configuring Service Interfaces” on page 246.



NOTE: If you configure an interface with an input firewall filter that includes a reject action and with a service set that includes stateful firewall rules, the router executes the input firewall filter before the stateful firewall rules are run on the packet. As a result, when the Packet Forwarding Engine sends an Internet Control Message Protocol (ICMP) error message out through the interface, the stateful firewall rules might drop the packet because it was not seen in the input direction.

Possible workarounds are to include a forwarding-table filter to perform the reject action, because this type of filter is executed after the stateful firewall in the input direction, or to include an output service filter to prevent the locally generated ICMP packets from going to the stateful firewall service.

Configuring Service Filters

You can optionally include filters associated with each service set to refine the target and additionally process the traffic. If you include the service-set statement without a service-filter definition, the router software assumes that the match condition is true and selects the service set for processing automatically.

To configure service filters, include the firewall statement at the [edit] hierarchy level:

```
[edit]
firewall {
  family inet {
    service-filter filter-name {
      term term-name {
        from {
          match-conditions;
        }
        then {
          action;
          action-modifiers;
        }
      }
    }
  }
}
```



NOTE: You must specify inet as the address family to configure a service filter.

Service filters are configured the same way as firewall filters. Service filters have the same match conditions as firewall filters, but two specific actions:

accept—Accept traffic for service processing.

skip—Omit traffic from service processing.

For more information about configuring firewall filters, see the *JUNOS Policy Framework Configuration Guide*.

You can also include more than one service set definition on each side of the interface. If you include multiple service sets, the router software evaluates them in the order specified in the configuration. It executes the first service set for which it finds a match in the service filter and ignores the subsequent definitions.

An additional statement allows you to specify a filter for processing the traffic after the input service set is executed. To configure this type of filter, include the post-service-filter statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet service input] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet service input]
post-service-filter filter-name;
```



NOTE: The software performs post-service filtering only when it has selected and executed a service set. If the traffic does not meet the match criteria for any of the configured service sets, the post-service filter is ignored.

For an example of applying a service set to an interface, see “Example: Configuring a Services Interface” on page 272.

For more information on applying filters to interfaces, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*. For general information on filters, see the *JUNOS Policy Framework Configuration Guide*.



NOTE: After Network Address Translation (NAT) processing is applied to packets, they are not subject to output service filters. The service filters affect only untranslated traffic.

Example: Configuring a Services Interface

The following example applies my-service-set on an interface-wide basis. All traffic that is accepted by my_input_filter has my-input-service-set applied to it. After the service set is applied, additional filtering is done using my_post_service filters.

```
[edit interfaces fe-0/1/0]
unit 0 {
  family inet {
    filter {
      input my_input_filter;
      output my_output_filter;
    }
    service {
      input {
        service-set my-input-service-set;
        post-service-filter my_post_service_input_filter;
      }
      output {
        service-set my-output-service-set;
      }
    }
  }
}
```