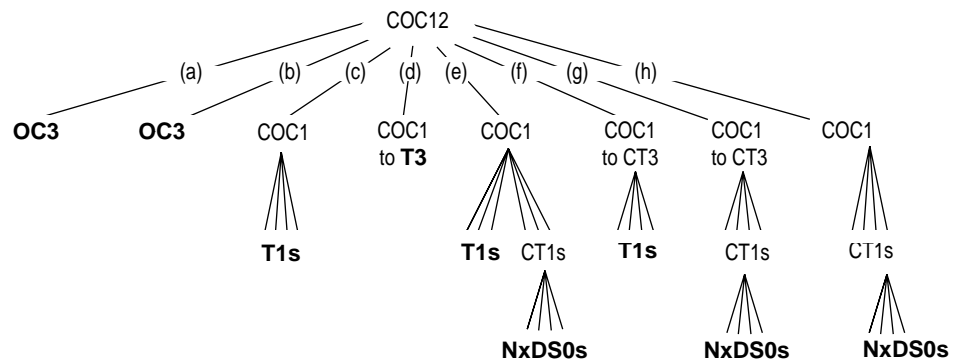


Chapter 15

Configuring Channelized OC12 Interfaces

Channelized intelligent queuing (IQ) interfaces allow arbitrary and dynamic channelization of serial links, allowing greater flexibility than the channelized interfaces. Figure 19 and Figure 20 on page 274 illustrate the difference in flexibility between a Channelized OC12 IQ Physical Interface Card (PIC) and a channelized OC12 PIC.

Figure 19: Sample Channelization of OC12 IQ PIC



Bold entries correspond to actual packet channels.

1991

In Figure 19, a Channelized OC12 IQ PIC is partitioned into the following OC slices:

- An OC3 interface
- Another OC3 interface
- A channelized OC1 partitioned into T1 interfaces
- A channelized OC1 converted into a T3 interface
- A channelized OC1 partitioned into T1 interfaces and channelized T1s, which are partitioned into *NxDS0* interfaces
- A channelized OC1 converted into a channelized T3, which is partitioned into T1 interfaces

- g. A channelized OC1 converted into a channelized T3, which is partitioned into T1 interfaces and a channelized T1, which is partitioned into $N \times$ DS0 interfaces
- h. A channelized OC1 partitioned into channelized T1s, which are partitioned into $N \times$ DS0 interfaces

This is one of thousands of ways to configure a Channelized OC12 IQ PIC. To configure the interfaces shown in Figure 19 on page 273, see “Example: Configuring Channelized OC12 IQ Interfaces” on page 282.

Figure 20: Sample Channelization of OC12 PIC

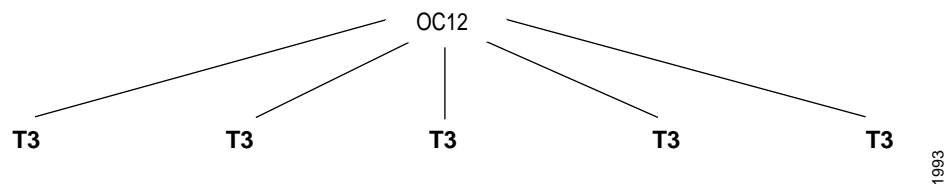


Figure 20 shows five T3 channels configured on the Channelized OC12 PIC. You can configure seven additional T3 channels. For more information about configuring Channelized OC12 PICs, see “Configuring Channelized OC12 Interfaces” on page 281.

This chapter is organized as follows:

Configuring Channelized OC12 IQ Interfaces on page 275

“Configuring Channelized OC12 Interfaces” on page 281

For examples of channelized OC12 interface configuration, see the following sections:

Example: Configuring Channelized OC12 IQ Interfaces on page 282

Example: Configuring Channelized OC12 Interfaces on page 286

For a full configuration example, see the *JUNOS Feature Guide*.

Configuring Channelized OC12 IQ Interfaces

This section describes how to configure channelized OC12 IQ interfaces, discussing the following topics:

Configuring an OC12 IQ Interface on page 275

Configuring T3 IQ Interfaces on page 275

Configuring OC3 IQ Interfaces on page 276

Configuring T1 IQ Interfaces on page 277

Configuring NxDS0 IQ Interfaces on page 279

Configuring Fractional T1 IQ Interfaces on page 281

Configuring an OC12 IQ Interface

On a 1-port Channelized OC12 IQ PIC, you can configure one OC12 interface. To configure an OC12 interface, include the `no-partition` and `interface-type` statements at the `[edit interfaces coc12-fpc/pic/port]` hierarchy level:

```
[edit interfaces coc12-fpc/pic/port]
no-partition interface-type so;
```

This configuration creates interface `so-fpc/pic/port`.

Configuring T3 IQ Interfaces

To configure a T3 interface on an OC12 PIC, include the `partition`, `oc-slice`, and `interface-type` statements at the `[edit interfaces coc12-fpc/pic/port]` hierarchy level, specifying the `coc1` interface type:

```
[edit interfaces coc12-fpc/pic/port]
partition partition-number oc-slice oc-slice-range interface-type coc1;
```

This configuration creates interface `coc1-fpc/pic/port:channel`.

Then, include the `no-partition interface-type` statement at the `[edit interfaces coc1-fpc/pic/port:channel]` hierarchy level, specifying the `t3` interface type:

```
[edit interfaces coc1-fpc/pic/port:channel]
no-partition interface-type t3;
```

This configuration creates interface `t3-fpc/pic/port:channel`.

The partition number is the sublevel interface partition index and is correlated with the channel number. For channelized OC1 interfaces, the partition number can be from 1 through 12.



NOTE: For channelized OC12 interfaces, channel numbering begins with 0 (:0). For channelized OC12 IQ interfaces, channel numbering begins with 1 (:1).

The OC-slice range is the range of SONET/SDH slices. For SONET/SDH interfaces, the OC-slice range specifies the bandwidth size required for the interface type you are configuring. For channelized OC1 interfaces, the OC slice can be from 1 through 12. You can configure only one OC slice per channelized OC1 interface.

The interface type is the channelized interface type or clear channel you are creating. For channelized OC12 interfaces, *type* can be *so* or *coc1*.

Example: Configuring T3 IQ Interfaces

Configure a T3 interface using partition 3 and OC slice 3. This configuration creates interface *t3-1/1/0:3*.

```
[edit interfaces coc12-1/1/0]
partition 3 oc-slice 3 interface-type coc1;
```

```
[edit interfaces coc1-1/1/0:3]
no-partition interface-type t3;
```

For a full configuration example, see the *JUNOS Feature Guide*.

Configuring OC3 IQ Interfaces

To configure an OC3 IQ interface, include the partition, oc-slice, and interface-type statements at the `[edit interfaces coc12-fpc/pic/port]` hierarchy level, specifying the *so* interface type:

```
[edit interfaces coc12-fpc/pic/port]
partition partition-number oc-slice oc-slice-range interface-type so;
```

The partition number is the sublevel interface partition index. For SONET/SDH interfaces, the partition number does not correlate with bandwidth size. For OC3 interfaces, the partition number can be 1 through 4.



NOTE: For channelized OC12 interfaces, channel numbering begins with 0 (:0). For channelized OC12 IQ interfaces, channel numbering begins with 1 (:1).

The OC-slice range is the range of SONET/SDH slices. For SONET/SDH interfaces, the OC-slice range specifies the bandwidth size required for the interface type you are configuring. OC3 IQ interfaces must occupy three consecutive OC slices per interface, in one of the following forms:

1-3

4-6

7-9

10-12

By contrast, the T3 and OC1 IQ interfaces each occupy one OC slice per interface.

The interface type is the channelized interface type or data channel you are creating. For channelized OC12 interfaces, the interface type can be *coc1* or *so*.

Example: Configuring OC3 IQ Interfaces

Configure an OC3 interface, using partition 1 and OC slices 4 through 6. This configuration creates interface so-1/1/0:1.

```
[edit interfaces coc12-1/1/0]
partition 1 oc-slice 4-6 interface-type so;
```

For a full configuration example, see the *JUNOS Feature Guide*.

Configuring T1 IQ Interfaces

To configure T1 interfaces on a Channelized OC12 IQ PIC, you perform the following tasks:

1. Partition the channelized OC12 interface into channelized OC1 interfaces by including the partition, oc-slice, and interface-type statements at the [edit interfaces coc12-fpc/pic/port] hierarchy level, specifying the coc1 interface type:

```
[edit interfaces coc12-fpc/pic/port]
partition partition-number oc-slice oc-slice-range interface-type coc1;
```

2. If your network equipment uses VT mapping, partition the channelized OC1 interface into T1 interfaces by including the partition and interface-type statements at the [edit interfaces coc1-fpc/pic/port:channel] hierarchy level, specifying the t1 interface type:

```
[edit interfaces coc1-fpc/pic/port:channel]
partition partition-number interface-type t1;
```

If your network equipment uses M13 or C-bit parity, convert the channelized OC1 interface into a channelized T3 interface by including the no-partition and interface-type statements at the [edit interfaces coc1-fpc/pic/port:channel] hierarchy level, specifying the ct3 interface type:

```
[edit interfaces coc1-fpc/pic/port:channel]
no-partition partition-number interface-type ct3;
```

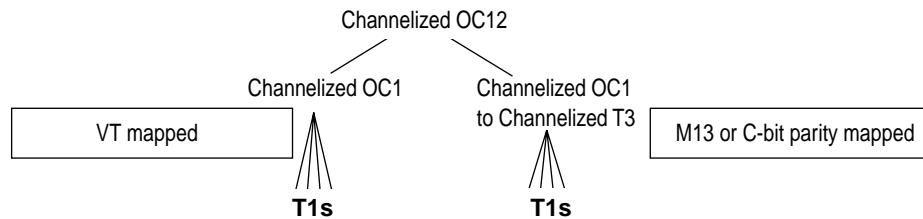
Note that because the no-partition statement is included, this configuration does not create another level of channelization, as denoted by the number of colons in the resulting interface.

You then partition the channelized T3 interface into T1 interfaces by including the partition and interface-type statements at the [edit interfaces ct3-fpc/pic/port:channel] hierarchy level, specifying the t1 interface type:

```
[edit interfaces ct3-fpc/pic/port:channel]
partition partition-number interface-type t1;
```

Figure 21 shows VT-mapped and M13 or C-bit parity-mapped configurations of T1 IQ interfaces.

Figure 21: T1 Interfaces on a Channelized OC12 PIC



Bold entries correspond to actual packet channels.

>

g003013

Example: Configuring T1 IQ Interfaces

Configure the following T1 interfaces:

```

t1-0/0/0:1:1
t1-0/0/0:1:2
t1-0/0/0:1:3
t1-0/0/0:1:4
t1-0/0/0:1:5
  
```

VT-Mapped Configuration

```

[edit interfaces coc12-0/0/0]
partition 1 oc-slice 1 interface-type coc1;

[edit interfaces coc1-0/0/0:1]
partition 1-5 interface-type t1;
  
```

M13 or C-bit Parity-Mapped Configuration

```

[edit interfaces coc12-0/0/0]
partition 1 oc-slice 1 interface-type coc1;

[edit interfaces coc1-0/0/0:1]
no-partition interface-type ct3;

[edit interfaces ct3-0/0/0:1]
partition 1-5 interface-type t1;
  
```

For a full configuration example, see the *JUNOS Feature Guide*.

Configuring NxDS0 IQ Interfaces

To configure NxDS0 interfaces on a Channelized OC12 IQ PIC, you perform the following tasks:

1. Partition the channelized OC12 interface into channelized OC1 interfaces by including the partition, oc-slice, and interface-type statements at the [edit interfaces coc12-fpc/pic/port] hierarchy level, specifying the coc1 interface type:

```
[edit interfaces coc12-fpc/pic/port]
partition partition-number oc-slice oc-slice-range interface-type coc1;
```

2. If your network equipment uses VT mapping, partition the channelized OC1 interface into channelized T1 interfaces by including the partition and interface-type statements at the [edit interfaces coc1-fpc/pic/port:channel] hierarchy level, specifying the ct1 interface type:

```
[edit interfaces coc1-fpc/pic/port:channel]
partition partition-number interface-type ct1;
```

If your network equipment uses M13 or C-bit parity, convert the channelized OC1 interface into a channelized T3 interface by including the no-partition and interface-type statements at the [edit interfaces coc1-fpc/pic/port:channel] hierarchy level, specifying the ct3 interface type:

```
[edit interfaces coc1-fpc/pic/port:channel]
no-partition partition-number interface-type ct3;
```

Note that because the no-partition statement is included, this configuration task does not create another level of channelization, as denoted by the number of colons in the resulting interface.

You then partition the channelized T3 interface into channelized T1 interfaces by including the partition and interface-type statements at the [edit interfaces ct3-fpc/pic/port:channel] hierarchy level, specifying the ct1 interface type:

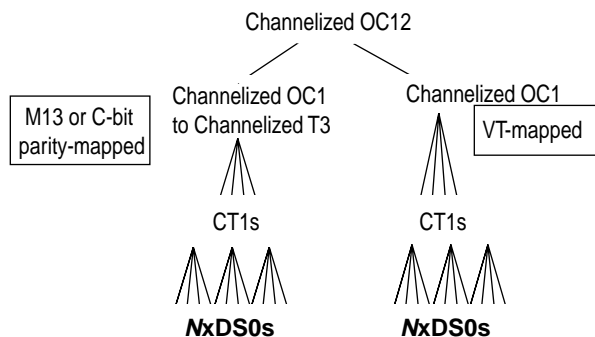
```
[edit interfaces ct3-fpc/pic/port:channel]
partition partition-number interface-type ct1;
```

3. Configure channelized NxDS0 IQ interfaces on the channelized T1 IQ interface by including the partition, timeslots, and interface-type statements at the [edit interfaces ct1-fpc/pic/port<:channel>] hierarchy level, specifying the ds interface type:

```
[edit interfaces ct1-fpc/pic/port:channel:channel]
partition partition-number timeslots time-slot-range interface-type ds;
```

Figure 22 shows VT-mapped and M13 or C-bit parity-mapped configurations of NxDS0 IQ interfaces.

Figure 22: Sample Channelization of OC12 IQ PIC



Bold entries correspond to actual packet channels.

9003014

Example: Configuring NxDS0 IQ Interfaces

Configure the following two NxDS0 interfaces with 10 time slots and 4 time slots, respectively:

```

ds-0/0/0:1:2:1
ds-0/0/0:1:2:2

VT-Mapped Configuration
[edit interfaces coc12-0/0/0]
partition 1 oc-slice 1 interface-type coc1;

[edit interfaces coc1-0/0/0:1]
partition 2 interface-type ct1;

[edit interfaces ct1-0/0/0:1:2]
partition 1 timeslots 1-10 interface-type ds;
partition 2 timeslots 12-16 interface-type ds;

M13 or C-bit Parity-Mapped Configuration
[edit interfaces coc12-0/0/0]
partition 1 oc-slice 1 interface-type coc1;

[edit interfaces coc1-0/0/0:1]
no-partition interface-type ct3;

[edit interfaces ct3-0/0/0:1]
partition 2 interface-type ct1;

[edit interfaces ct1-0/0/0:1:2]
partition 1 timeslots 1-10 interface-type ds;
partition 2 timeslots 12-16 interface-type ds;
    
```

For a full configuration example, see the *JUNOS Feature Guide*.

Configuring Fractional T1 IQ Interfaces

By default, all the time slots on a channelized T1 interface are used. To configure a fractional T1 interface on a Channelized OC12 IQ PIC, you must perform the following tasks:

1. Configure a T1 IQ interface. For more information, see “Configuring T1 IQ Interfaces” on page 277.
2. Configure the number of time slots allocated to the T1 IQ interface by including the `timeslots` statement at the `[edit interfaces t1-fpc/pic/port<:channel> t1-options]` hierarchy level:

```
[edit interfaces t1-fpc/pic/port<:channel> t1-options]
timeslots time-slot-range;
```

For channelized T1 IQ interfaces, the time-slot range is from 1 through 24. You can designate any combination of time slots. To configure ranges, use hyphens. To configure discontinuous time slots, use commas. Do not include spaces. For more information, see “Configuring Fractional T1 Time Slots” on page 553.

Examples: Configuring Fractional T1 IQ Interfaces

Configure a fractional T1 interface that uses time slots 1 through 5 and 10:

```
[edit interfaces coc12-0/0/0]
partition 1 oc-slice 1 interface-type coc1;
```

```
[edit interfaces coc1-0/0/0:1]
partition 1 interface-type t1;
```

```
[edit interfaces t1-0/0/0:1:1 t1-options]
timeslots 1-5,10;
```

For a full configuration example, see the *JUNOS Feature Guide*.

Configuring Channelized OC12 Interfaces

On Channelized OC12 PICs, you can configure 12 T3 channels per port. To configure channelized OC12 interface properties, you can include the `sonet-options` and `t3-options` statements at the `[edit interfaces interface-name]` hierarchy level. Some SONET/SDH options are ignored, and some can only be configured for channel 0, though they apply equally to all channels. The `long-buildout` statement under `t3-options` is also ignored.

For T3 channels on a channelized OC12 interface, the clocking statement is supported only for channel 0; it is ignored if included in the configuration of channels 1 through 11. The clock source configured for channel 0 applies to all channels on the channelized OC12 interface. The individual T3 channels use a gapped 45-MHz clock as the transmit clock. When you configure the clock source for a channelized interface—`ds-x/y/z:0`, for example—you must also include the `channel-group` statement at the `[edit chassis]` hierarchy level and specify channel group 0. For more information, see “Clock Sources on Channelized Interfaces” on page 250.

For more information, see “Configuring SONET/SDH Interfaces” on page 509 and “Configuring T3 Interfaces” on page 555. For a configuration example, see “Configuring Aggregated SONET/SDH Interfaces” on page 538.

Table 26 summarizes the OC12-to-DS3 numbering scheme.

Table 26: OC12-to-DS3 Numbering Scheme

Two-Level STS-1 Number (STS-3,STS-1)	One-Level STS Number	OC12-to-DS3 PIC DS3 Number
1,1	1	0
1,2	2	1
1,3	3	2
2,1	4	3
2,2	5	4
2,3	6	5
3,1	7	6
3,2	8	7
3,3	9	8
4,1	10	9
4,2	11	10
4,3	12	11

Example: Configuring Channelized OC12 IQ Interfaces

Configuring a SONET/SDH Interface

Configure a channelized OC12 interface as an unpartitioned, clear channel:

```
[edit interfaces]
coc12-5/0/0 {
    no-partition interface-type so; # so-5/0/0
}
```

Configuring Multiple Interface Types

Configure the following interfaces on a Channelized OC12 IQ PIC:

- a. An OC3 interface
- b. Another OC3 interface
- c. A channelized OC1 partitioned into T1 interfaces
- d. A channelized OC1 converted into a T3 interface
- e. A channelized OC1 partitioned into T1 interfaces and channelized T1s, which are partitioned into NxDS0 interfaces
- f. A channelized OC1 converted into a channelized T3, which is partitioned into T1 interfaces

- g. A channelized OC1 converted into a channelized T3, which is partitioned into T1 interfaces and a channelized T1, which is partitioned into $N \times$ DS0 interfaces
- h. A channelized OC1 partitioned into channelized T1s, which are partitioned into $N \times$ DS0 interfaces

Configuring the Interface Partitions

```
[edit interfaces]
coc12-1/1/0 {
  sonet-options {
    sonet-options-statements;
  }
  partition 1 oc-slice 1-3 interface-type so; # (a) so-1/1/0:1
  partition 2 oc-slice 4-6 interface-type so; # (b) so-1/1/0:2
  partition 3 oc-slice 7 interface-type coc1; # (c) coc1-1/1/0:3
  partition 4 oc-slice 8 interface-type coc1; # (d) coc1-1/1/0:4
  partition 5 oc-slice 9 interface-type coc1; # (e) coc1-1/1/0:5
  partition 6 oc-slice 10 interface-type coc1; # (f) coc1-1/1/0:6
  partition 7 oc-slice 11 interface-type coc1; # (g) coc1-1/1/0:7
  partition 8 oc-slice 12 interface-type coc1; # (h) coc1-1/1/0:8
}

(a) so-1/1/0:1 {
  description "(a) OC-slice 1-3 of coc12-1/1/0. COC12 > OC3.";
  sonet-options {
    sonet-options-statements;
  }
}

(b) so-1/1/0:2 {
  description "(b) OC-slice 4-6 of coc12-1/1/0. COC12 > OC3.";
  sonet-options {
    sonet-options-statements;
  }
}

(c) coc1-1/1/0:3 {
  description "(c) OC-slice 7 of coc12-1/1/0. COC12 to COC1 VT-mapped to
  T1s.";
  sonet-options {
    sonet-options-statements;
  }
  partition 1 - 10 interface-type t1; # t1-1/1/0:[1-10]
}
t1-1/1/0:3:1 {
  description "(c) OC-slice 7 of coc12-1/1/0. T1 interface configuration.";
  t1-options {
    t1-options-statements;
  }
}
...

```

```

(d) coc1-1/1/0:4 {
    description "(d) OC-slice 8 of coc12-1/1/0. COC12 to COC1 converted to a
                T3.";
    sonet-options {
        sonet-options-statements;
    }
    no-partition interface-type t3; # t3-1/1/0:4
}
t3-1/1/0:4 {
    description "(d) OC-slice 8 of coc12-1/1/0. T3 interface configuration.";
}

(e) coc1-1/1/0:5 {
    description "(e) OC-slice 9 of coc12-1/1/0. COC12 to COC1 VT-mapped to
                T1s.";
    sonet-options {
        sonet-options-statements;
    }
    partition 1 - 3 interface-type t1; # t1-1/1/0:5:[1-3]
    partition 4 interface-type ct1; # ct1-1/1/0:5:4
}
t1-1/1/0:5:1 {
    description "(e) OC-slice 9 of coc12-1/1/0. T1 interface configuration.";
    t1-options {
        t1-options-statements;
    }
}
...
ct1-1/1/0:5:4 {
    description "(e) OC-slice 9 of coc12-1/1/0. CT1 to NxDSOs.";
    t1-options {
        t1-options-statements;
    }
    partition 1 timeslots 0 - 10 interface-type ds0; # ds-1/1/0:5:4:1
    partition 2 timeslots 11- 23 interface-type ds0; # ds-1/1/0:5:4:2
    ...
}

(f) coc1-1/1/0:6 {
    description "(f) OC-slice 10 of coc12-1/1/0. COC12 to COC1 converted to a
                CT3 to T1s.";
    sonet-options {
        sonet-options-statements;
    }
    no-partition interface-type ct3; # ct3-1/1/0:6
}
ct3-1/1/0:6 {
    description "(f) COC12 to CT3 M-13 and C-bit parity-mapped to T1s.";
    sonet-options {
        sonet-options-statements;
    }
    partition 1 - 10 interface-type t1; # t1-1/1/0:6:[1-10]
}

```

```

t1-1/1/0:6:1 {
    description "(f) T1 interface configuration.";
    t1-options {
        t1-options-statements;
    }
}
...
(g) coc1-1/1/0:7 {
    description "(g) OC-slice 11 of coc12-1/1/0. COC12 to COC1 converted to a
        CT3 to T1s and CT1 to NxDSOs.";
    sonet-options {
        sonet-options-statements;
    }
    no-partition interface-type ct3; # ct3-1/1/0:7
}
ct3-1/1/0:7 {
    description "(g) COC12 to CT3 M-13 and C-bit parity-mapped to T1s and
        CT1.";
    sonet-options {
        sonet-options-statements;
    }
    partition 1 - 10 interface-type t1; # t1-1/1/0:7:[1-10]
    partition 2 interface-type ct1; # ct1-1/1/0:7:11
}
t1-1/1/0:7:1 {
    description "(g) T1 interface configuration.";
    t1-options {
        t1-options-statements;
    }
}
...
ct1-1/1/0:7:11 {
    description "(g) CT1 to NxDSOs.";
    t1-options {
        t1-options-statements;
    }
    partition 1 timeslots 0 - 10 interface-type ds0; # ds-1/1/0:7:11:1
    partition 2 timeslots 11- 23 interface-type ds0; # ds-1/1/0:7:11:2
    ...
}

```

```
(h) coc1-1/1/0:8 {
    description "(h) OC-slice 12 of coc12-1/1/0. COC12 to COC1 VT-mapped to
        CT1 to NxDSOs.";
    sonet-options {
        sonet-options-statements;
    }
    partition 1 interface-type t1; # ct1-1/1/0:8:1
}
ct1-1/1/0:8:1 {
    description "(h) CT1 to NxDSOs.";
    t1-options {
        t1-options-statements;
    }
    partition 1 timeslots 0 - 10 interface-type ds0; # ds-1/1/0:8:1:1
    partition 2 timeslots 11- 23 interface-type ds0; # ds-1/1/0:8:1:2
    ...
}
```

For a full configuration example, see the *JUNOS Feature Guide*.

Example: Configuring Channelized OC12 Interfaces

The following configuration is sufficient to get the channelized OC12 interface up and running. The OC12 interface can be divided into 12 channels. DS3 channels can use the following encapsulation types:

PPP, PPP CCC, and PPP TCC

Frame Relay, Frame Relay CCC, and Frame Relay TCC

Cisco HDLC, Cisco HDLC CCC, and Cisco HDLC TCC

The channels can also have logical interfaces.

```
[edit interfaces]
t3-fpc/pic/port:0 {
    encapsulation cisco-hdlc;
    t3-options {
        compatibility-mode larscom;
        payload-scrambler;
    }
    unit 0 {
        family inet {
            address 10.11.30.1/30;
        }
        family iso;
    }
}
```

```

t3-fpc/pic/port:1 {
  encapsulation ppp;
  t3-options {
    compatibility-mode larscom;
    payload-scrambler;
  }
  unit 0 {
    family inet {
      address 10.11.30.5/30;
    }
    family iso;
  }
}
t3-fpc/pic/port:2 {
  encapsulation frame-relay;
  t3-options {
    compatibility-mode larscom;
    payload-scrambler;
  }
  unit 0 {
    dlci 100;
    family inet {
      address 10.11.30.9/30;
    }
    family iso;
  }
  unit 1 {
    dlci 101;
    family inet {
      address 10.11.31.9/30;
    }
    family iso;
  }
}
t3-1fpc/pic/port:3 {
  encapsulation cisco-hdlc-ccc;
  t3-options {
    compatibility-mode larscom;
    payload-scrambler;
  }
  unit 0;
}
t3-fpc/pic/port:4 {
  encapsulation ppp-ccc;
  t3-options {
    compatibility-mode larscom;
    payload-scrambler;
  }
  unit 0;
}

```

```
t3-fpc/pic/port:5 {  
  dce;  
  encapsulation frame-relay-ccc;  
  t3-options {  
    compatibility-mode larscom;  
    payload-scrambler;  
  }  
  unit 0 {  
    encapsulation frame-relay-ccc;  
    dlcI 1000;  
  }  
  unit 1 {  
    encapsulation frame-relay-ccc;  
    dlcI 1001;  
  }  
}
```