CHAPTER 3

Configuring T1/E1 Interfaces

This chapter provides information about configuring the T1/E1 interface module on the Cisco ME 3600X 24CX Series Switch. It includes the following sections:

- Configuration Tasks, page 3-1
- Configuration Examples, page 3-14

For information about managing your system images and configuration files, refer to the Cisco IOS Configuration Fundamentals Configuration Guide and Cisco IOS Configuration Fundamentals Command Reference publications.

For more information about the commands used in this chapter, refer to the Cisco IOS Command Reference publication for your Cisco IOS software release.

For more information, see the Related Publications, page -2.

Configuration Tasks

This section describes how to configure the T1/E1 interface module for the Cisco ME 3600X 24CX Series Switch and includes information about verifying the configuration.

It includes the following topics:

- Required Configuration Tasks, page 3-2
- Optional Configurations, page 3-4
- Saving the Configuration, page 3-6

Limitations

This section describes the software limitations that apply when configuring the T1/E1 interface module on the Cisco ME 3600X 24CX Series Switch.

The following features are not currently supported on the T1/E1 interface module:

- Serial interfaces—The Cisco ME 3600X 24CX Series Switch does not currently support serial interfaces or features applied to serial interfaces. We recommend that you use a configuration with CEM as a workaround.

- Channel groups—The Cisco ME 3600X 24CX Series Switch does not currently support channel-groups or features applied to channel-groups. We recommend that you use a configuration with CEM as a workaround.
Configuration Tasks

- Supported BERT patterns—Currently only the 2^11, 2^15, 2^20-O153, and 2^20-QRSS patterns are supported.

Required Configuration Tasks

This section lists the required configuration steps to configure the T1/E1 interface module. Some of the required configuration commands implement default values that might be appropriate for your network. If the default value is correct for your network, then you do not need to configure the command.

- Setting the Card Type, page 3-2
- Configuring the Controller, page 3-3
- Verifying Controller Configuration, page 3-4
- Optional Configurations, page 3-4

Setting the Card Type

The interface module is not functional until the card type is set. Information about the interface module is not indicated in the output of any `show` commands until the card type has been set. There is no default card type.

**Note**
Mixing of interface types is not supported. All ports on the interface module must be of the same type.

To set the card type for the T1/E1 interface module, complete these steps:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 2</td>
<td>Router(config)# card type {e1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>Router(config)# exit</td>
</tr>
</tbody>
</table>

**Note**
On doing no card type T1/E1 0 1, the peer box controller will not go down until the device is reloaded as prompted.
**Configuring the Controller**

To create the interfaces for the T1/E1 interface module, complete these steps:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | Router(config)# controller {t1 | e1} card/number | Selects the controller to configure and enters controller configuration mode.  
  * t1—Specifies the T1 controller.  
  * e1—Specifies the E1 controller.  
  * card/number—Specifies the location of the interface.  
  
  **Note**  
  The card numbers supported are 0 or 1. The number range for T1 is 0 to 15. The number range for E1 is 0 to 31. |
| **Step 2** | Router(config-controller)# clock source {internal | line} | Sets the clock source.  
  
  **Note**  
  The clock source is set to internal if the opposite end of the connection is set to line and the clock source is set to line if the opposite end of the connection is set to internal.  
  * internal—Specifies that the internal clock source is used.  
  * line—Specifies that the network clock source is used. This is the default for T1 and E1. |
| **Step 3** | Router(config-controller)# linecode {ami | b8zs | hdb3} | Selects the linecode type.  
  * ami—Specifies Alternate Mark Inversion (AMI) as the linecode type. Valid for T1 and E1 controllers.  
  * b8zs—Specifies binary 8-zero substitution (B8ZS) as the linecode type. Valid for T1 controller only. This is the default for T1 lines.  
  * hdb3—Specifies high-density binary 3 (HDB3) as the linecode type. Valid for E1 controller only. This is the default for E1 lines. |
| **Step 4** | For T1 Controllers:  
  Router(config-controller)# framing {sf | esf}  
  For E1 Controllers:  
  Router(config-controller)# framing {crc4 | no-crc4} | Selects the framing type.  
  * sf—Specifies Super Frame as the T1 frame type.  
  * esf—Specifies Extended Super Frame as the T1 frame type. This is the default for E1.  
  * crc4—Specifies CRC4 as the E1 frame type. This is the default for E1.  
  * no-crc4—Specifies no CRC4 as the E1 frame type. |
Verifying Controller Configuration

Use the `show controllers` command to verify the controller configuration:

```
Router# show controllers e1 0/11
E1 0/11 is up.
Applique type is Channelized E1 - balanced
Cablelength is long gain36 0db
No alarms detected.
alarm-trigger is not set
Soaking time: 3, Clearance time: 10
AIS State:Clear  LOS State:Clear  LOF State:Clear
Framing is ESF, Line Code is B8ZS, Clock Source is Internal.
Data in current interval (230 seconds elapsed):
  0 Line Code Violations, 0 Path Code Violations
  0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
  0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  0 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
Total Data (last 24 hours)
  136 Line Code Violations, 63 Path Code Violations,
  0 Slip Secs, 6 Fr Loss Secs, 4 Line Err Secs, 0 Degraded Mins,
  7 Errored Secs, 1 Bursty Err Secs, 6 Severely Err Secs, 458 Unavail Secs
  2 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
```

Optional Configurations

There are several standard, but optional, configurations that might be necessary to complete the configuration of your Ethernet interface module.

- Configuring Framing, page 3-5
- Saving the Configuration, page 3-6

---

### Configuration Tasks

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td>`cablelength {long</td>
<td>short}`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-controller)# cablelength long</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Exits configuration mode and returns to the EXEC command interpreter prompt.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Framing

Framing is used to synchronize data transmission on the line. Framing allows the hardware to determine when each packet starts and ends. To configure framing, use the following commands.

### Command | Purpose
--- | ---
Router# configure terminal | Enters global configuration mode.

Router(config)# controller \{t1 | e1\} card/number | Selects the controller to configure.
- \( t1 \) — Specifies the T1 controller.
- \( e1 \) — Specifies the E1 controller.
- \( card/number \) — Specifies the location of the controller.

**Note** The card is always 0.

**For T1 controllers**

Router(config-controller)# framing \{sf | esf\} | Set the framing on the interface.
- \( sf \) — Specifies Super Frame as the T1 frame type.
- \( esf \) — Specifies Extended Super Frame as the T1 frame type. This is the default for T1.

**For E1 controllers**

Router(config-controller)# framing \{crc4 | no-crc4\} | Specifies CRC4 frame as the E1 frame type. This is the default for E1.
- \( no-crc4 \) — Specifies no CRC4 as the E1 frame type.

### Verifying Framing Configuration

Use the `show controllers` command to verify the framing configuration:

Router# show controllers t1 0/11
T1 0/11 is up.
Applique type is Channelized T1 - balanced
Cablelength is long gain36 0db
No alarms detected.
alarm-trigger is not set
Soaking time: 3, Clearance time: 10
AIS State:Clear LOS State:Clear LOF State:Clear
Framing is ESF, Line Code is B8ZS, Clock Source is Line.
Data in current interval (740 seconds elapsed):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
0 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs

Total Data (last 24 hours):
0 Line Code Violations, 0 Path Code Violations,
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
0 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), use the following command in privileged EXEC configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router# copy running-config startup-config</code></td>
<td>Writes the new configuration to NVRAM.</td>
</tr>
</tbody>
</table>

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and *Cisco IOS Configuration Fundamentals Command Reference* publications.

Troubleshooting E1 and T1 Controllers

You can use the following methods to troubleshoot the E1 and T1 controllers using Cisco IOS software:

- Setting Loopbacks
- Run Bit Error Rate Test

Setting Loopbacks

The following sections describe how to set loopbacks:

- Setting a Loopback on the E1 Controller, page 3-6
- Setting a Loopback on the T1 Controller, page 3-7

Setting a Loopback on the E1 Controller

To set a loopback on the E1 controller, perform the first task followed by any of the following tasks beginning in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the E1 controller and enter controller configuration mode.</td>
<td><code>controller e1 card/number</code></td>
</tr>
<tr>
<td>Note The card is always 0.</td>
<td></td>
</tr>
<tr>
<td>Set a diagnostic loopback on the E1 line.</td>
<td><code>loopback diag</code></td>
</tr>
<tr>
<td>Set a network payload loopback on the E1 line.</td>
<td>`loopback network {line</td>
</tr>
<tr>
<td>Exit configuration mode when you have finished configuring the controller.</td>
<td><code>end</code></td>
</tr>
</tbody>
</table>
Setting a Loopback on the T1 Controller

To set a loopback on the T1 controller, perform the first task followed by any of the following tasks beginning in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the T1 controller and enter controller configuration mode.</td>
<td>controller t1 card/number</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>The card is always 0.</td>
</tr>
<tr>
<td>Set a diagnostic loopback on the T1 line.</td>
<td>loopback diag</td>
</tr>
<tr>
<td>Set a local loopback on the T1 line.</td>
<td>loopback local {line</td>
</tr>
<tr>
<td></td>
<td>Set a remote loopback on the T1 line. This</td>
</tr>
<tr>
<td></td>
<td>loopback setting will loopback the far end</td>
</tr>
<tr>
<td></td>
<td>at line or payload, using IBOC (in band</td>
</tr>
<tr>
<td></td>
<td>bit-orientated code) or the ESF loopback</td>
</tr>
<tr>
<td></td>
<td>codes to communicate the request to the far</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
<tr>
<td>Exit configuration mode when you have</td>
<td>end</td>
</tr>
<tr>
<td>finished configuring the controller.</td>
<td></td>
</tr>
</tbody>
</table>

Note: To remove a loopback, use the no loopback command.

Table 3-1 Loopback Descriptions

<table>
<thead>
<tr>
<th>Loopback</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>loopback diag</td>
<td>Loops the outgoing transmit signal back to the receive signal. This is done using the diagnostic loopback feature in the interface module’s PMC framer. The interface module transmits AIS in this mode. Set the clock source command to internal for this loopback mode.</td>
</tr>
<tr>
<td>loopback local</td>
<td>Loops the incoming receive signal back out the transmitter. You can specify whether to use the line or payload.</td>
</tr>
<tr>
<td>local line</td>
<td>The incoming signal is looped back in the interface module using the framer’s line loopback mode. The framer does not re-clock or re-frame the incoming data. All incoming data is receive by the interface module’s driver.</td>
</tr>
<tr>
<td>local payload</td>
<td>The incoming signal is looped back in the interface module using the framer’s payload loopback mode. The framer re-clocks and re-frames the incoming data before sending it back out to the network. When in payload loopback, an all 1s data pattern is received by the local HDLC receiver, and the clock source is automatically set to line (overriding the clock source command). When the payload loopback is ended, the clock source returns to the last setting selected by the clock source command.</td>
</tr>
</tbody>
</table>
Chapter 3 Configuring T1/E1 Interfaces

Configuration Tasks

### Run Bit Error Rate Test

Bit error rate testing (BERT) is supported on each of the E1 or T1 links. The BERT testing is done only over a framed E1 or T1 signal and can be run only on one port at a time.

The interface modules contain onboard BERT circuitry. With this, the interface module software can send and detect a programmable pattern that is compliant with CCITT/ITU O.151, O.152, and O.153 pseudo-random and repetitive test patterns. BERTs allow you to test cables and signal problems in the field.

When running a BER test, your system expects to receive the same pattern that it is transmitting. To help ensure this, two common options are available:

- Use a loopback somewhere in the link or network
- Configure remote testing equipment to transmit the same BER test pattern at the same time

To run a BERT on an E1 or T1 controller, perform the following optional tasks beginning in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the E1 or T1 controller and enter controller configuration mode.</td>
<td>`controller {e1</td>
</tr>
<tr>
<td>Note</td>
<td>The card is always 0.</td>
</tr>
<tr>
<td>Specify the BERT pattern for the E1 or T1 line and the duration of the test in minutes (1 to 1440 minutes).</td>
<td>`bert pattern {0s</td>
</tr>
<tr>
<td>Note</td>
<td>2^23 and 2^20-O153 patterns are not supported.</td>
</tr>
<tr>
<td>Exit configuration mode when you have finished configuring the controller.</td>
<td><code>end</code></td>
</tr>
<tr>
<td>View the BERT results.</td>
<td>`show controllers {e1</td>
</tr>
</tbody>
</table>

---

### Table 3-1 Loopback Descriptions

<table>
<thead>
<tr>
<th>Loopback</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>loopback remote <code>iboc</code></td>
<td>Attempts to set the far-end T1 interface into line loopback. This command sends an in-band bit-oriented code to the far-end to cause it to go into line loopback. This command is available when using ESF or SF framing mode.</td>
</tr>
<tr>
<td>network line</td>
<td>The incoming signal is looped back in the interface module using the framer’s line loopback mode. The framer does not re-clock or re-frame the incoming data. All incoming data is received by the interface module's driver.</td>
</tr>
<tr>
<td>network payload</td>
<td>The incoming signal is looped back in the interface module using the framer’s payload loopback mode. The framer re-clocks and re-frames the incoming data before sending it back out to the network. When in payload loopback, an all 1s data pattern is received by the local HDLC receiver, and the clock source is automatically set to line (overriding the <code>clock source</code> command). When the payload loopback is ended, the clock source returns to the last setting selected by the <code>clock source</code> command.</td>
</tr>
</tbody>
</table>
The following keywords list different BERT keywords and their descriptions.

⚠️ **Caution**
Currently 2^23, and 2^20-O153 patterns are not supported.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0s</td>
<td>Repeating pattern of zeros (...000...).</td>
</tr>
<tr>
<td>1s</td>
<td>Repeating pattern of ones (...111...).</td>
</tr>
<tr>
<td>2^11</td>
<td>Pseudo-random test pattern that is 2,048 bits in length.</td>
</tr>
<tr>
<td>2^15</td>
<td>Pseudo-random O.151 test pattern that is 32,768 bits in length.</td>
</tr>
<tr>
<td>2^20-O153</td>
<td>Pseudo-random O.153 test pattern that is 1,048,575 bits in length.</td>
</tr>
<tr>
<td>2^20-QRSS</td>
<td>Pseudo-random QRSS O.151 test pattern that is 1,048,575 bits in length.</td>
</tr>
<tr>
<td>2^23</td>
<td>Pseudo-random O.151 test pattern that is 8,388,607 bits in length.</td>
</tr>
<tr>
<td>alt-0-1</td>
<td>Repeating alternating pattern of zeros and ones (...01010...).</td>
</tr>
</tbody>
</table>

Both the total number of error bits received and the total number of bits received are available for analysis. You can select the testing period from 1 minute to 24 hours, and you can also retrieve the error statistics anytime during the BER test.

📝 **Note**
To terminate a BER test during the specified test period, use the `no bert` command.

You can view the results of a BER test at the following times:

- After you terminate the test using the `no bert` command
- After the test runs completely
- Anytime during the test (in real time)

---

**Monitor and Maintain the T1/E1 Interface Module**

After configuring the new interface, you can monitor the status and maintain the interface module by using `show` commands. To display the status of any interface, complete any of the following tasks in EXEC mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display the status of the E1 or T1 controller.</td>
<td>`show controllers {e1</td>
</tr>
</tbody>
</table>
Configuring CEM

This section provides information about how to configure CEM. CEM provides a bridge between a time-division multiplexing (TDM) network and a packet network, such as Multiprotocol Label Switching (MPLS). The router encapsulates the TDM data in the MPLS packets and sends the data over a CEM pseudowire to the remote provider edge (PE) router. Thus, function as a physical communication link across the packet network.

The following sections describe how to configure CEM:
- Configuring a CEM Group, page 3-10
- Using CEM Classes, page 3-11
- Configuring CEM Parameters, page 3-12

Note: CEM is used as an element in configuring pseudowires including Structure-Agnostic TDM over Packet (SAToP) and Circuit Emulation Service over Packet-Switched Network (CESoPSN). For more information about configuring pseudowires, see Chapter 6, “Configuring Pseudowire.”

Configuring a CEM Group

The following section describes how to configure a CEM group on the Cisco ME 3600X 24CX Series Switch.

SUMMARY STEPS

1. enable
2. configure terminal
3. controller {t1 | e1} card/number
4. cem-group group-number {unframed | timeslots timeslot}
5. end

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 3 controller {t1</td>
<td>e1} card/number</td>
</tr>
<tr>
<td></td>
<td>• Use the card, number arguments to specify the card number and bay number to be configured.</td>
</tr>
<tr>
<td></td>
<td>Note: The card number is always 0.</td>
</tr>
</tbody>
</table>
Using CEM Classes

A CEM class allows you to create a single configuration template for multiple CEM pseudowires. Follow these steps to configure a CEM class:

**Note** The CEM parameters at the local and remote ends of a CEM circuit must match; otherwise, the pseudowire between the local and remote PE routers will not come up.

**Note** You cannot apply a CEM class to other pseudowire types such as ATM over MPLS.

### SUMMARY STEPS

1. enable
2. configure terminal
3. class cem classname
4. payload-size size
5. dejitter-buffer size
6. idle-pattern \{pattern \ length pattern1 \ pattern2\}\
7. exit
8. interface cem card/number
9. no ip address
10. cem card/number
11. cem group-number
12. `xconnect peer-ip-address vc-id [encapsulation {l2tpv3 [manual] | mpls [manual]} | pw-class pw-class-name] [pw-class pw-class-name] [sequencing {transmit | receive | both}]`

13. `exit`

14. `exit`

---

**Command** | **Purpose**
---|---
Step 1
| enable |
| Enables privileged EXEC mode.  
  • Enter your password if prompted. |
Step 2
| configure terminal |
| Enters global configuration mode. |
Step 3
| Router(config)# class cem mycemclass |
| Creates a new CEM class |
Step 4
| payload-size size  
dejitter-buffer size  
idle-pattern {pattern | length pattern1 [pattern2]} |
| Enter the configuration commands common to the CEM class. This example specifies a sample rate, payload size, dejitter buffer, and idle pattern. |
Step 5
| Router(config-cem-class)# exit |
| Returns to the config prompt. |
Step 6
| Router(config)# interface cem 0/1  
Router(config-if)# no ip address  
Router(config-if)# cem 0  
Router(config-if-cem)# xconnect 10.10.10.10 200 encapsulation mpls |
| Configure the CEM interface that you want to use for the new CEM class. | Note: The use of the `xconnect` command can vary depending on the type of pseudowire you are configuring. |
Step 7
| Router(config-if-cem)# exit  
Router(config-if)# |
| Exits the CEM interface. |
Step 8
| exit |
| Exits configuration mode. |

### Configuring CEM Parameters

The following sections describe the parameters you can configure for CEM circuits.

- Configuring Payload Size (Optional), page 3-12
- Setting the Dejitter Buffer Size, page 3-13
- Setting an Idle Pattern (Optional), page 3-13
- Enabling Dummy Mode, page 3-13
- Setting a Dummy Pattern, page 3-13
- Shutting Down a CEM Channel, page 3-13

**Note**

The CEM parameters at the local and remote ends of a CEM circuit must match; otherwise, the pseudowire between the local and remote PE routers will not come up.

### Configuring Payload Size (Optional)

To specify the number of bytes encapsulated into a single IP packet, use the `payload-size` command. The `size` argument specifies the number of bytes in the payload of each packet. The range is from 32 to 1312 bytes.
Default payload sizes for an unstructured CEM channel are as follows:

- \( E1 = 256 \) bytes
- \( T1 = 192 \) bytes
- \( DS0 = 32 \) bytes

Default payload sizes for a structured CEM channel depend on the number of time slots that constitute the channel. Payload size (L in bytes), number of time slots (N), and packetization delay (D in milliseconds) have the following relationship: \( L = 8*N*D \). The default payload size is selected in such a way that the packetization delay is always 1 millisecond. For example, a structured CEM channel of 16xDS0 has a default payload size of 128 bytes.

The payload size must be an integer of the multiple of the number of time slots for structured CEM channels.

**Setting the Dejitter Buffer Size**

To specify the size of the dejitter buffer used to compensate for the network filter, use the `dejitter-buffer size` command. The configured dejitter buffer size is converted from milliseconds to packets and rounded up to the next integral number of packets. Use the size argument to specify the size of the buffer, in milliseconds. The range is from 1 to 500 ms; the default is 5 ms.

**Setting an Idle Pattern (Optional)**

To specify an idle pattern, use the `[no] idle-pattern pattern1` command. The payload of each lost CESoPSN data packet must be replaced with the equivalent amount of the replacement data. The range for pattern is from 0x0 to 0xFF; the default idle pattern is 0xFF.

**Enabling Dummy Mode**

Dummy mode enables a bit pattern for filling in for lost or corrupted frames. To enable dummy mode, use the `dummy-mode [last-frame | user-defined]` command. The default is last-frame. The following is an example:

```
Router(config-cem)# dummy-mode last-frame
```

**Setting a Dummy Pattern**

If dummy mode is set to user-defined, you can use the `dummy-pattern pattern` command to configure the dummy pattern. The range for pattern is from 0x0 to 0xFF. The default dummy pattern is 0xFF. The following is an example:

```
Router(config-cem)# dummy-pattern 0x55
```

**Shutting Down a CEM Channel**

To shut down a CEM channel, use the `shutdown` command in CEM configuration mode. The `shutdown` command is supported only under CEM mode and not under the CEM class.
Configuration Examples

This section includes the following configuration examples:

- Framing and Encapsulation Configuration Example, page 3-14
- CRC Configuration Example, page 3-14
- Facility Data Link Configuration Example, page 3-15
- Invert Data on the T1/E1 Interface Example, page 3-15

Framing and Encapsulation Configuration Example

The following example sets the framing and encapsulation for the controller and interface:

! Specify the controller and enter controller configuration mode
! Router(config)# controller t1 0/1
! Specify the framing method
! Router(config-controller)# framing esf
! Exit controller configuration mode and return to global configuration mode
! Router(config-controller)# exit
! Specify the interface and enter interface configuration mode
! Router(config)# interface serial 0/1
! Specify the encapsulation protocol
! Router(config-if)# encapsulation ppp
! Exit interface configuration mode
! Router(config-if)# exit
! Exit global configuration mode
! Router(config)# exit

CRC Configuration Example

The following example sets the CRC size for the interface:

! Specify the interface and enter interface configuration mode
! Router(config)# interface serial 0/1
! Specify the CRC size
! Router(config-if)# crc 32
! Exit interface configuration mode and return to global configuration mode
! Router(config-if)# exit
! Exit global configuration mode
! Router(config)# exit

Facility Data Link Configuration Example

The following example configures Facility Data Link:

! Specify the controller and enter controller configuration mode
! Router(config)# controller t1 0/1
! ! Specify the FDL specification
! Router(config-controller)# fdl ansi
! ! Exit controller configuration mode and return to global configuration mode
! Router(config-controller)# exit
! ! Exit global configuration mode
! Router(config)# exit

Invert Data on the T1/E1 Interface Example

The following example inverts the data on the serial interface:

! Enter global configuration mode
! Router# configure terminal
! ! Specify the serial interface and enter interface configuration mode
! Router(config)# interface serial 0/1
! ! Configure invert data
! Router(config-if)# invert data
! ! Exit interface configuration mode and return to global configuration mode
! Router(config-if)# exit
! ! Exit global configuration mode
! Router(config)# exit

CEM Configuration Example

The following example shows how to add a T1 interface to a CEM group as a part of a SAToP pseudowire configuration. For more information about how to configure pseudowires, see Chapter 6, “Configuring Pseudowire.”
This section displays a partial configuration intended to demonstrate a specific feature.

controller T1 0/1
framing unframed
clock source internal
linecode b8zs
cablelength short 110
cem-group 0 unframed

interface CEM0/0
no ip address
cem 0
  xconnect 18.1.1.1 1000 encapsulation mpls