Security Products

ScreenOS Wide Area Network Interfaces and Protocols Reference

ScreenOS Release 5.1.0
# Table of Contents

**About This Document**

- Documentation Conventions
- CLI Conventions
- Naming Conventions and Character Types
- WebUI Conventions
- Juniper Networks Documentation

## Chapter 1 Wide Area Network Overview

- Wide Area Network Interfaces
- Wide Area Network Protocols
- Wide Area Network Interface Names
- Before You Begin
- Basic WAN Configuration Steps

## Chapter 2 Configuring Serial Interfaces

- Overview of Serial Interfaces
- Configuring Serial Interface Options with the WebUI
- Configuring Serial Interface Options with the CLI
  - Hold Time
  - Clocking Mode
  - Transmit Clock Inversion
  - Internal Clock Rate
  - Signal Handling
  - Loopback Capability
  - Line Encoding
- Example Configurations

## Chapter 3 Configuring T1 Interfaces

- Overview of T1 Interfaces
- Configuring T1 Interface Options with the WebUI
- Configuring T1 Interface Options with the CLI
  - Hold Time
  - Clocking Source
  - Line Buildout
  - Line Encoding
  - Byte Encoding
  - Frame Checksum
  - Framing Mode
  - Idle Cycle Flag
  - Start/End Flags
  - Data Inversion
  - Remote Loopback Response
Table of Contents

<table>
<thead>
<tr>
<th>Chapter 4 Configuring E1 Interfaces</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of E1 Interfaces</td>
<td>19</td>
</tr>
<tr>
<td>Configuring E1 Interface Options with the WebUI</td>
<td>20</td>
</tr>
<tr>
<td>Configuring E1 Interface Options with the CLI</td>
<td>21</td>
</tr>
<tr>
<td>Hold Time</td>
<td>21</td>
</tr>
<tr>
<td>Clocking Source</td>
<td>21</td>
</tr>
<tr>
<td>Frame Checksum</td>
<td>21</td>
</tr>
<tr>
<td>Framing Mode</td>
<td>22</td>
</tr>
<tr>
<td>Idle Cycle Flag</td>
<td>22</td>
</tr>
<tr>
<td>Start/End Flags</td>
<td>22</td>
</tr>
<tr>
<td>Data Inversion</td>
<td>22</td>
</tr>
<tr>
<td>Fractional E1 Time Slots</td>
<td>22</td>
</tr>
<tr>
<td>Loopback Mode</td>
<td>23</td>
</tr>
<tr>
<td>Bit Error Rate Testing Options</td>
<td>23</td>
</tr>
<tr>
<td>Configuring BERT</td>
<td>24</td>
</tr>
<tr>
<td>Starting and Stopping a BERT</td>
<td>24</td>
</tr>
<tr>
<td>Example Configurations</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5 Configuring T3 Interfaces</th>
<th>27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of T3 Interfaces</td>
<td>27</td>
</tr>
<tr>
<td>Configuring T3 Interface Options with the WebUI</td>
<td>28</td>
</tr>
<tr>
<td>Configuring T3 Interface Options with the CLI</td>
<td>29</td>
</tr>
<tr>
<td>Hold Time</td>
<td>29</td>
</tr>
<tr>
<td>Clocking Source</td>
<td>30</td>
</tr>
<tr>
<td>Frame Checksum</td>
<td>30</td>
</tr>
<tr>
<td>Idle Cycle Flag</td>
<td>30</td>
</tr>
<tr>
<td>Start/End Flags</td>
<td>30</td>
</tr>
<tr>
<td>C-Bit Parity Mode</td>
<td>31</td>
</tr>
<tr>
<td>Line Buildout</td>
<td>31</td>
</tr>
<tr>
<td>FEAC Response</td>
<td>31</td>
</tr>
<tr>
<td>HDLC Payload Scrambling</td>
<td>31</td>
</tr>
<tr>
<td>CSU Compatibility Mode</td>
<td>32</td>
</tr>
<tr>
<td>Loopback Mode</td>
<td>33</td>
</tr>
<tr>
<td>Bit Error Rate Testing Options</td>
<td>33</td>
</tr>
<tr>
<td>Configuring BERT</td>
<td>34</td>
</tr>
<tr>
<td>Starting and Stopping a BERT</td>
<td>34</td>
</tr>
<tr>
<td>Example Configurations</td>
<td>35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 6 Configuring Frame Relay and Multilink Frame Relay</th>
<th>37</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of the Frame Relay Network</td>
<td>38</td>
</tr>
<tr>
<td>Configuring Frame Relay Encapsulation on a Single Physical Link</td>
<td>38</td>
</tr>
<tr>
<td>Basic Configuration Steps</td>
<td>38</td>
</tr>
<tr>
<td>Configuring Frame Relay Encapsulation on a Physical Link</td>
<td>39</td>
</tr>
<tr>
<td>Assigning the Interface to a Zone</td>
<td>39</td>
</tr>
<tr>
<td>Setting the Protocol MTU</td>
<td>39</td>
</tr>
<tr>
<td>Configuring Frame Relay Options for the Link</td>
<td>40</td>
</tr>
</tbody>
</table>
Frame Relay Keepalives .................................................................40
Frame Relay LMI Type .................................................................41
Keepalives for Frame Relay LMI .....................................................41
Creating and Configuring PVCs ....................................................42
Inverse Address Resolution Protocol ...........................................43
Unnumbered Interfaces .................................................................43
Configuring Multilink Frame Relay .................................................44
Basic Configuration Steps ............................................................45
Creating a Bundle ........................................................................45
  Configuring a Bundle Identifier .................................................46
  Assigning the Interface to a Zone ...............................................46
  Setting the Protocol MTU ..........................................................46
Configuring Frame Relay Options for the Bundle .......................47
  MLFR LMI Keepalives ............................................................47
  Frame Relay LMI Type ............................................................48
  Drop Timeout ...........................................................................48
  Minimum Links .........................................................................49
Assigning Bundle Links to the Bundle .........................................49
Configuring MLFR Options for Bundle Links ...............................50
  Acknowledge Retries ...............................................................50
  Acknowledge Timer ..................................................................51
  Fragment Threshold ...............................................................51
  Hello Timer ............................................................................51
Creating and Configuring PVCs Within the Bundle .......................52
Inverse Frame Relay Address Resolution Protocol .......................53
Example Configurations ...............................................................53
  Frame Relay Configuration Examples .......................................53
  SSG Device ............................................................................53
  J2300 and SSG 520 Peer Devices ..............................................53
Unnumbered Interface Configuration Examples ............................54
  Multilink Frame Relay Configuration Example ...........................55

Chapter 7 Configuring Point-to-Point Protocol and Multilink
Point-to-Point Protocol 57
Overview of Point-to-Point Protocol ...........................................57
Configuring PPP Encapsulation on a Single Physical Link .............58
Basic Configuration Steps ............................................................58
Configuring PPP Encapsulation on a Physical Link .......................58
  Assigning the Interface to a Zone .............................................59
  Configuring a Static IP Address for the Interface .......................59
Configuring PPP Options for the Link ...........................................59
  PPP Keepalives ......................................................................60
  Keepalive Interval ..................................................................60
  Keepalive Down Count ..........................................................60
Configuring a PPP Access Profile ...............................................61
  Authentication Method ..........................................................62
  Static IP Address ....................................................................62
  Netmask ................................................................................63
  Unnumbered Interfaces ..........................................................63
  Passive Mode .........................................................................63
  CHAP Local Name ..................................................................64
  Password ..............................................................................64
Configuring WAN Users in the Local Database ............................64
Configuring Multilink Point-to-Point Protocol ...............................65
Table of Contents

Product Rev# Book Title

Basic Configuration Steps ................................................................. 65
Creating a Bundle ......................................................................... 65
Assigning the Interface to a Zone .................................................. 66
Assigning an IP Address to the Interface ....................................... 66
Setting the Protocol MTU .............................................................. 67
Configuring MLPPP Options for the Bundle ..................................... 67
  Drop Timeout ............................................................................. 67
  Minimum Links .......................................................................... 68
  Maximum Received Reconstructed Unit ....................................... 68
  Fragment Threshold .................................................................... 69
  Sequence Header Format .......................................................... 69
Configuring and Binding PPP Access Profiles .................................. 69
Configuring WAN Users in the Local Database ............................... 69
Assigning Bundle Links to the Bundle .......................................... 70
Configuring PPP Options for Bundle Links ..................................... 70
Example Configurations ............................................................... 71
  PPP Configuration Examples ..................................................... 71
    SSG Peer Devices ................................................................... 71
    J2300 and SSG 520 Peer Devices ........................................... 71
Unnumbered Interfaces Configuration Examples ............................ 72
MLPPP Configuration Example ..................................................... 72

Chapter 8 Configuring Cisco High-Level Data Link Control ............. 75
  Overview of Cisco High-Level Data Link Control .......................... 75
  Configuring Cisco HDLC Encapsulation ....................................... 76
    Basic Configuration Steps ....................................................... 76
    Configuring Cisco HDLC Encapsulation on a Physical Link ........... 76
      Assigning the Interface to a Zone ........................................... 76
      Configuring a Static IP Address for the Interface .............. 77
    Configuring HDLC Options for the Link .................................... 77
      Keepalives ........................................................................ 77
      Keepalive Interval ............................................................. 78
      Keepalive Down and Up Counts ......................................... 78
    Unnumbered Interfaces .......................................................... 78
Example Configurations ............................................................... 79

Chapter 9 New and Modified CLI Commands ................................. 81
  counter ................................................................................... 81
  Syntax .................................................................................... 81
  get....................................................................................... 81
  Keywords and Variables ........................................................... 81
    Variable Parameter .............................................................. 81
    statistics ............................................................................ 82
  interface ................................................................................ 82
  Syntax ................................................................................... 82
  exec (T1/E1/T3 Interfaces) ........................................................ 82
  get (E1 Interfaces) ................................................................. 82
  get (Serial Interfaces) .............................................................. 82
  get (T1 Interfaces) ................................................................. 82
  get (T3 Interfaces) ................................................................. 82
  get (Cisco HDLC) ................................................................ 82
  get (Frame Relay or Multilink Frame Relay) ............................... 82
  get (PPP or Multilink PPP) ...................................................... 82
set/unset (Cisco HDLC) ................................................................. 82
set/unset (E1 Interfaces) ................................................................. 83
set/unset (Frame Relay) ................................................................. 83
set/unset (Multilink Frame Relay) ................................................... 84
set/unset (PPP) ............................................................................. 84
set/unset (Multilink PPP) ................................................................. 85
set/unset (Serial Interfaces) ............................................................. 85
set/unset (T1 Interfaces) ................................................................. 86
set/unset (T3 Interfaces) ................................................................. 86
Keywords and Variables ................................................................. 87
  Variable Parameter ................................................................. 87
  bert-test ...................................................................................... 87
  bundle ....................................................................................... 87
  bundle-ID .................................................................................. 87
  cisco-hdlc ................................................................................ 88
  clocking .................................................................................... 88
  disable ...................................................................................... 88
  drop-timeout ............................................................................ 88
  e1-options ................................................................................ 88
  encapsulation .......................................................................... 90
  fragment-threshold .................................................................. 90
  frame-relay .............................................................................. 90
  hold-time ................................................................................ 91
  ip unnumbered interface ......................................................... 91
  keepalives ............................................................................... 92
  minimum-links ......................................................................... 92
  mfr-uni-nni ............................................................................. 92
  mrru ........................................................................................ 92
  ppp .......................................................................................... 93
  ppp profile ............................................................................... 93
  serial-options ......................................................................... 93
  short-sequence ....................................................................... 95
  t1-options ............................................................................... 95
  t3-options ............................................................................... 97
  zone ...................................................................................... 100

ppp .......................................................................................... 101
  Syntax ...................................................................................... 101
  get .......................................................................................... 101
  set/unset ............................................................................... 101
Keywords and Variables ................................................................. 101
  profile .................................................................................... 101

user .......................................................................................... 102
  Syntax ...................................................................................... 102
  set .......................................................................................... 102
Keywords and Variables ................................................................. 102
  Variable Parameter ................................................................. 102
    password ............................................................................. 102
    type .................................................................................... 102

Chapter 10 Message Descriptions ............................................... 103
  Interface .................................................................................. 104
  Notification (00009) ................................................................. 104
  Cisco HDLC ............................................................................... 104
  Alert (000087) ......................................................................... 104
<table>
<thead>
<tr>
<th>Appendix A</th>
<th>Glossary</th>
<th>A-I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td></td>
<td>IX-I</td>
</tr>
</tbody>
</table>

| Notification (00062) | ................................................................. | 104  |
| Notification (00561) | ................................................................. | 105  |
| Frame Relay         | ...................................................................... | 106  |
| Alert (00086)       | ...................................................................... | 106  |
| Notification (00060)  | ..................................................................... | 106  |
| Notification (00559)  | ..................................................................... | 107  |
| Multilink Frame Relay | ..................................................................... | 107  |
| Alert (00085)       | ...................................................................... | 107  |
| Notification (00061)  | ..................................................................... | 107  |
| Notification (00560)  | ..................................................................... | 110  |
| PPP                 | ...................................................................... | 111  |
| Notification (00017)  | ..................................................................... | 111  |
| Alert (00088)       | ...................................................................... | 111  |
| Notification (00063)  | ..................................................................... | 111  |
| Notification (00562)  | ..................................................................... | 113  |
| WAN                 | ...................................................................... | 115  |
| Critical (00091)    | ...................................................................... | 115  |
List of Figures

Figure 1: WebUI Navigation ......................................................... xvi
Figure 2: Navigational Path and Configuration Settings ................... xvii
Figure 3: Serial Interface Clocking Mode ...................................... 6
Figure 4: Serial Interface LIU Loopback ...................................... 8
Figure 5: Serial Interface Local Loopback .................................... 9
Figure 6: Remote and Local T1 Loopback ................................... 16
Figure 7: Remote and Local E1 Loopback ................................... 23
Figure 8: Remote and Local T3 Loopback ................................... 33
Figure 9: Devices in the Frame Relay Network ............................. 38
Figure 10: Point-to-Point Frame Relay Subinterfaces ...................... 42
Figure 11: Multilink Frame Relay (MLFR) Bundle ......................... 44
List of Tables

Table 1: Default and Alternate Values for Serial Interface Options ............... 4
Table 2: Signal Handling by Serial-Interface Type .................................. 7
Table 3: Default and Alternate Values for T1 Interface Options ................. 12
Table 4: Default and Alternate Values for E1 Interface Options ............... 20
Table 5: Default and Alternate Values for T3 Interface Options ............... 29
Table 6: Default and Alternate Values for Frame Relay Options ............... 40
Table 7: Default and Alternate Values for Multilink Frame Relay Interfaces 47
Table 8: Default and Alternate Values for Multilink Frame Relay Bundle Links 50
Table 9: Default and Alternate Values for Point-to-Point Protocol Options ... 59
Table 10: Default and Alternate Values for Point-to-Point Protocol Access Profile Options 62
Table 11: Default and Alternate Values for Multilink Point-to-Point Protocol Interfaces 67
Table 12: Default and Alternate Values for Cisco High-Level Data Link Control Options 77
About This Document

This document describes how to configure wide area network (WAN) data links on the Juniper Networks Secure Services Gateway (SSG) device. This document contains the following information:

- Brief overviews of the WAN interface types and encapsulation protocols supported on SSG devices
- Instructions for configuring each WAN interface type and protocol
- Syntax and descriptions for new or modified command line interface (CLI) commands that support the WAN interfaces and protocols
- Descriptions of messages related to WAN interfaces and protocols
- Glossary of acronyms and terms related to WAN interfaces and protocols

This document is intended to be a supplement to the ScreenOS 5.1.0 documentation set. For more information about ScreenOS features, CLI commands, and messages, refer to the following ScreenOS 5.1.0 documents:

- Concepts & Examples ScreenOS Reference Guide
- ScreenOS CLI Reference Guide IPv4 Command Descriptions
- Message Log Reference

For information about installing the SSG device and performing basic configuration, see the Secure Services Gateway (SSG) 500 Series Hardware Installation and Configuration Guide.

Documentation Conventions

This document uses several types of conventions, which are introduced in the following sections:

- “CLI Conventions” on page xiv
- “Naming Conventions and Character Types” on page xv
- “WebUI Conventions” on page xvi
**CLI Conventions**

The following conventions are used to present the syntax of CLI commands in examples and in text.

In examples:

- Anything inside square brackets `[ ]` is optional.
- Anything inside braces `{ }` is required.
- If there is more than one choice, each choice is separated by a pipe ( | ). For example:

  ```
  set interface { ethernet1 | ethernet2 | ethernet3 } manage
  ```

  means “set the management options for the ethernet1, ethernet2, or ethernet3 interface.”

- Variables are in *italic* type:

  ```
  set admin user *name1* password *xyz*
  ```

In text:

- Commands are in **boldface** type.
- Variables are in *italic* type.

**NOTE:** When typing a keyword, you only have to enter enough letters to identify the word uniquely. For example, entering `set adm u kath j12fmt54` is enough to enter the command `set admin user kathleen j12fmt54`. Although you can use this shortcut when entering commands, all the commands documented here are presented in their entirety.
Naming Conventions and Character Types

ScreenOS employs the following conventions regarding the names of objects—such as addresses, admin users, auth servers, IKE gateways, virtual systems, VPN tunnels, and zones—defined in ScreenOS configurations:

- If a name string includes one or more spaces, the entire string must be enclosed within double quotes (""); for example:

  ```bash
  set address trust "local LAN" 10.1.1.0/24
  ```

- ScreenOS trims any spaces leading or trailing text within a set of double quotes; for example, “local LAN” becomes “local LAN”.

- ScreenOS treats multiple consecutive spaces as a single space.

- Name strings are case sensitive, although many CLI key words are case insensitive. For example, “local LAN” is different from “local lan”.

ScreenOS supports the following character types:

- Single-byte character sets (SBCS) and multiple-byte character sets (MBCS). Examples of SBCS are ASCII, European, and Hebrew. Examples of MBCS—also referred to as double-byte character sets (DBCS)—are Chinese, Korean, and Japanese.

- ASCII characters from 32 (0x20 in hexadecimals) to 255 (0xff), except double quotes (""), which have special significance as an indicator of the beginning or end of a name string that includes spaces.

**NOTE:** A console connection only supports SBCS. The WebUI supports both SBCS and MBCS, depending on the character sets that your web browser supports.
WebUI Conventions

A chevron ( > ) shows the navigational sequence through the WebUI, which you follow by clicking menu options and links. The following figure shows the following path to the address configuration dialog box—Objects > Addresses > List > New:

Figure 1: WebUI Navigation

To perform a task with the WebUI, you first navigate to the appropriate dialog box, where you then define objects and set parameters. The set of instructions for each task is divided into navigational path and configuration settings:

The next figure lists the path to the address configuration dialog box with the following sample configuration settings:

Objects > Addresses > List > New: Enter the following, and then click OK:

Address Name: addr_1
IP Address/Domain Name: 10.2.2.5/32
Zone: Untrust
Figure 2: Navigational Path and Configuration Settings

Juniper Networks Documentation

To obtain technical documentation for any Juniper Networks product, visit www.juniper.net/techpubs/.

For technical support, open a support case using the Case Manager link at http://www.juniper.net/support/ or call 1-888-314-JTAC (within the United States) or 1-408-745-9500 (outside the United States).

If you find any errors or omissions in this document, please contact us at the email address below:

techpubs-comments@juniper.net
Chapter 1
Wide Area Network Overview

The Juniper Networks Secure Services Gateway (SSG) device uses wide area network (WAN) data links to transmit and receive traffic across geographically dispersed networks. These networks can be privately owned but more typically include public or shared networks. For WAN links to operate, you must configure properties such as the clocking and signal-handling options for the physical line and the encapsulation method to be used to transfer data across the WAN.

In ScreenOS, you define the properties of the data link by configuring the WAN interface that corresponds to a port on an SSG Physical Interface Module (PIM).

Wide Area Network Interfaces

SSG devices support PIMs for the following WAN interface types:

- Serial
- T1
- E1
- T3 (also known as DS3)

Wide Area Network Protocols

SSG devices support the following encapsulation protocols on WAN interfaces:

- Frame Relay
- Multilink Frame Relay (MLFR)
- Point-to-Point Protocol (PPP)
- Multilink PPP (MLPPP)
- Cisco High-Level Data Link Control (HDLC)
Wide Area Network Interface Names

Each WAN interface on an SSG device is named serialn1/n2, where n1 is the slot number in the SSG chassis that is occupied by the PIM, and n2 is the port on the PIM. For example, serial1/0 refers to the WAN interface in slot 1, port 0. In this document, the variable interface is used to represent a WAN interface name.

Refer to the SSG 500 Series Hardware Installation and Configuration Guide for information about slot numbers on the SSG chassis and the ports available for each PIM.

Before You Begin

Before you configure WAN interfaces on an SSG device, you need to perform the following tasks:

1. Install and power on the SSG device.
2. Connect network cables to the SSG Console or management ports.
3. Establish connectivity for configuring and administering the SSG device.

For more information on the above tasks, see the SSG 500 Series Hardware Installation and Configuration Guide.

Basic WAN Configuration Steps

To configure a WAN interface on an SSG device:

1. Connect network cables between the WAN port and the peer device.
2. (Optional) Configure the properties of the serial, T1, E1, or T3 physical link. This step is required only if you need to change the default options for the physical link.
3. Configure the encapsulation protocol for the data link.
Chapter 2
Configuring Serial Interfaces

Serial Physical Interface Modules (PIMs) for Secure Services Gateway (SSG) devices have two serial ports per PIM, which support full-duplex, synchronous data transmission. These ports can transmit packets at speeds up to 8 Megabits per second (Mbps). You cannot use these serial ports to connect a console or modem.

This chapter describes how to configure interfaces on a serial PIM in an SSG device. It contains the following sections:

- “Overview of Serial Interfaces” on page 3
- “Configuring Serial Interface Options with the WebUI” on page 4
- “Configuring Serial Interface Options with the CLI” on page 5
- “Example Configurations” on page 9

Overview of Serial Interfaces

Devices that communicate over a serial interface are divided into two classes: data terminal equipment (DTE) and data circuit-terminating equipment (DCE). The SSG Serial PIM supports DTE mode only. On the Serial PIM, you can configure the following types of serial interfaces:

- **TIA/EIA 530**—The Telecommunications Industry Association/Electronics Industries Alliance (TIA/EIA) Standard 530, *High-Speed 25-Position Interface for Data Terminal Equipment and Data Circuit-Terminating Equipment*, describes the interconnection of DTE and DCE using serial binary data interchange with control information exchanged on separate control circuits.

- **V.35**—The Telecommunication Standardization Sector of the International Telecommunications Union (ITU-T) Recommendation V.35, *Data Transmission at 48 kbit/s Using 60-108 kHz Group Band Circuits*, describes a synchronous, Physical Layer protocol used for communications between a network access device and a packet network. V.35 is most commonly used in the United States and in Europe. Note that the Juniper Networks Serial PIM supports V.35 interfaces with speeds higher than 48 kilobits per second (kbps).

- **X.21**—The ITU-T Recommendation X.21, *Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Synchronous Operation on Public Data Networks*, describes serial communications over synchronous digital lines. The X.21 protocol is used primarily in Europe and Japan.
- RS-232—TIA/EIA-232-F (the current revision), *Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*, describes the physical interface and protocol for communication with modems and other serial devices.

- RS-449—The EIA standard *EIA-449 General Purpose 37-Position and 9-Position Interface for Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*, specifies the interface between data terminal equipment and data communications equipment.

### Configuring Serial Interface Options with the WebUI

To use the ScreenOS Web User Interface (WebUI) to configure serial interface options, navigate to the WAN dialog box for the serial interface:

Network > Interfaces > Edit *(interface)* > WAN: Enter or select the applicable option value, and then click **Apply**.

Table 1 shows the default settings for serial interface options along with the alternate values you can enter or select. For more information about each option, see “Configuring Serial Interface Options with the CLI.”

#### Table 1: Default and Alternate Values for Serial Interface Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold time:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Up</td>
<td>0 milliseconds</td>
<td>1-65534 milliseconds</td>
</tr>
<tr>
<td>Down</td>
<td>0 milliseconds</td>
<td>1-65534 milliseconds</td>
</tr>
<tr>
<td>Clock mode</td>
<td>Loop</td>
<td>DCE or Internal</td>
</tr>
<tr>
<td>Transmit clock invert</td>
<td>Not set</td>
<td>Enable</td>
</tr>
<tr>
<td>Clock rate</td>
<td>8.0 megahertz</td>
<td>Specific kilohertz or megahertz rates, starting from 1.2 kilohertz to 4.0 megahertz</td>
</tr>
<tr>
<td>DTE signal handling options:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clear-to-send (CTS) signal</td>
<td>Normal</td>
<td>Ignore or Require</td>
</tr>
<tr>
<td>Data carrier-detect (DCD) signal</td>
<td>Normal</td>
<td>Ignore or Require</td>
</tr>
<tr>
<td>Data-set-ready (DSR) signal</td>
<td>Normal</td>
<td>Ignore or Require</td>
</tr>
<tr>
<td>Data-transfer-ready (DTR) signal</td>
<td>Normal</td>
<td>Assert, Auto-synchronize, or Deassert</td>
</tr>
<tr>
<td>Request-to-send (RTS) signal</td>
<td>Normal</td>
<td>Assert or Deassert</td>
</tr>
<tr>
<td>Test mode (TM) signal</td>
<td>Normal (for TIA/EIA 530 interfaces only; not applicable for V.35 or X.21 interfaces)</td>
<td>Ignore or Require</td>
</tr>
<tr>
<td>Ignore all control leads</td>
<td>Not set</td>
<td>Enable</td>
</tr>
<tr>
<td>Line encoding</td>
<td>Non-return to zero (NRZ)</td>
<td>Non-return to zero inverted (NRZI)</td>
</tr>
<tr>
<td>Loopback mode</td>
<td>None</td>
<td>DCE Local, Local, or Remote/line interface unit (LIU)</td>
</tr>
</tbody>
</table>
Configuring Serial Interface Options with the CLI

You can use the command line interface (CLI) to configure the serial interface options shown in Table 1. This section describes the serial interface options and shows the CLI command you enter to configure each option.

**Hold Time**

By default, when an interface changes from up to down or from down to up, this transition is advertised immediately to the hardware and the ScreenOS software. In some situations—for example, when an interface is connected to an add-drop multiplexer (ADM) or wavelength-division multiplexer (WDM), or to protect against SONET/SDH framer holes—you might want to damp interface transitions. This means not advertising the interface's transition until a certain period of time, called the *hold time*, has passed. When you have damped interface transitions and the interface goes from up to down, the interface is not advertised to the rest of the system as being down until it has remained down for the hold-time period. Similarly, when an interface goes from down to up, it is not advertised as being up until it has remained up for the hold-time period.

To damp interface transitions:

```
set interface interface hold-time { down milliseconds | up milliseconds }
```

The time can be a value from 0 through 65,534 milliseconds. The default value is 0, which means that interface transitions are not damped. The ScreenOS software advertises the transition within 100 milliseconds of the time value you specify.

**Clocking Mode**

By default, a serial interface uses loop clocking mode. For TIA/EIA 530, V.35, RS0232, and RS-449 interfaces, you can configure each port on the PIM independently to use loop, DCE, or internal clocking mode. For X.21 interfaces, only loop clocking mode is supported.

The three clocking modes work as follows:

- **Loop clocking mode**—Uses the DCE’s Receive (RX) clock to clock data from the DCE to the DTE.

- **DCE clocking mode**—Uses the Transmit (TX) clock, which is generated by the DCE specifically to be used by the DTE as the DTE’s transmit clock.

- **Internal clocking mode**—Also known as *line timing*, uses an internally generated clock. You can configure the speed of this clock by including the *clock-rate* option in the `set interface` command. For more information about the clock rate, see “Internal Clock Rate” on page 6.

Note that DCE-clocking mode and loop-clocking mode use external clocks generated by the DCE.

Figure 3 shows the clock sources of loop, DCE, and internal clocking modes.
To configure the clocking mode of a serial interface:

```
set interface interface serial-options clocking-mode { dce | internal | loop }
```

### Transmit Clock Inversion

When an externally timed clocking mode (DCE or loop) is used, long cables might introduce a phase shift of the DTE-transmitted clock and data. At high speeds, this phase shift might cause errors. Inverting the transmit clock corrects the phase shift, thereby reducing error rates.

By default, the transmit clock is not inverted. To invert the transmit clock:

```
set interface interface serial-options transmit-clock invert
```

### Internal Clock Rate

By default, the serial interface has a clock rate of 8.0 MHz. For TIA/EIA 530, V.35, RS-232, and RS-449 interfaces with internal clocking mode configured, you can configure the clock rate. For more information about internal clocking mode, see “Clocking Mode” on page 5.

**NOTE:** For RS-232 interfaces with internal clocking mode configured, the clock rate must be less than 20khz.

To configure the clock rate:

```
set interface interface serial-options clock-rate rate
```

You can configure the following interface rates:

- 1.2 Khz
- 2.4 Khz
- 9.6 Khz
- 19.2 Khz
- 38.4 Khz
- 56.0 Khz
- 64.0 Khz
- 72.0 Khz
- 125.0 Khz
- 148.0 Khz
- 250.0 Khz
- 500.0 Khz
- 800.0 Khz
- 1.0 Mhz
- 1.3 Mhz
- 2.0 Mhz
- 4.0 Mhz
- 8.0 Mhz
- 1.0 Mhz
- 2.0 Mhz
- 4.0 Mhz
- 8.0 Mhz
- 1.0 Mhz
- 2.0 Mhz
- 4.0 Mhz
- 8.0 Mhz
Although the serial interface is intended for use at the default rate of 8.0 MHz, you might need to use a slower rate under any of the following conditions:

- The interconnecting cable is too long for effective operation.
- The interconnecting cable is exposed to an extraneous noise source that might cause an unwanted voltage in excess of +1 volt measured differentially between the signal conductor and circuit common at the load end of the cable, with a 50-ohm resistor substituted for the generator.
- You need to minimize interference with other signals.
- You need to invert signals.

For detailed information about the relationship between signaling rate and interface cable distance, see the following standards:

- EIA 422-A, *Electrical Characteristics of Balanced Voltage Digital Interface Circuits*
- EIA 423-A, *Electrical Characteristics of Unbalanced Voltage Digital Interface Circuits*

### Signal Handling

By default, normal signal handling is enabled for all signals. For each signal, normal signal handling is defined by the following standards:

- TIA/EIA Standard 530
- ITU-T Recommendation V.35
- ITU-T Recommendation X.21

Table 2 shows the serial-interface modes that support each signal type.

**Table 2: Signal Handling by Serial-Interface Type**

<table>
<thead>
<tr>
<th>Signal</th>
<th>Serial Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>From-DCE signals:</td>
<td></td>
</tr>
<tr>
<td>Clear-to-send (CTS)</td>
<td>TIA/EIA 530, V.35, RS-232, RS-449</td>
</tr>
<tr>
<td>Data-carrier-detect (DCD)</td>
<td>TIA/EIA 530, V.35, RS-232, RS-449</td>
</tr>
<tr>
<td>Data-set-ready (DSR)</td>
<td>TIA/EIA 530, V.35, RS-232, RS-449</td>
</tr>
<tr>
<td>Test-mode (TM)</td>
<td>TIA/EIA 530 only</td>
</tr>
<tr>
<td>To-DCE signals:</td>
<td></td>
</tr>
<tr>
<td>Data-transfer-ready (DTR)</td>
<td>TIA/EIA 530, V.35, RS-232, RS-449</td>
</tr>
<tr>
<td>Request-to-send (RTS)</td>
<td>TIA/EIA 530, V.35, RS-232, RS-449</td>
</tr>
</tbody>
</table>

You configure serial-interface signal characteristics by including the `serial-options dte-options` keyword in the `set interface` command.
For TIA/EIA 530, V.35, RS-232, and RS-449 interfaces, configure from-DCE signals by including the cts, cdc, and dsr options, and specify ignore, normal, or require as the value. For TIA/EIA 530 interfaces only, you can configure from-DCE test-mode (TM) signaling by including the tm option and specifying ignore, normal, or require as the value.

For all interfaces, configure to-DCE signals by including the dtr and rts options. For dtr, specify assert, de-assert, or auto-synchronize as the value. For rts, specify assert, de-assert, or normal as the value. Assertion is when the positive side of a given signal is at potential high-level output voltage (Voh), while the negative side of the same signal is at potential low-level output voltage (Vol). Deassertion is when the positive side of a given signal is at potential Vol, while the negative side of the same signal is at potential Voh.

You can configure the serial interface to ignore all signal-handling options by including the ignore-all option. If you specify ignore-all, you cannot configure any other signal-handling option.

**Loopback Capability**

From the SSG device, remote line interface unit (LIU) loopback loops the TX (transmit) data and TX clock back to the device as RX (receive) data and RX clock. From the line, LIU loopback loops the RX data and RX clock back out the line as TX data and TX clock, as shown in Figure 4.

**Figure 4: Serial Interface LIU Loopback**

DCE local and DCE remote control the TIA/EIA 530 interface-specific signals for enabling local and remote loopback on the link-partner DCE. Figure 5 shows local loopback.
By default, no loopback is enabled on the SSG device. For TIA/EIA 530, RS-232, and RS-449 interfaces, you can configure DCE local, local, and remote options for loopback capability. For V.35 and X.21 interfaces, you can configure remote LIU and local options for loopback capability; the DCE local loopback option is not supported on V.35 and X.21 interfaces.

To configure loopback capability on a serial interface:

```
set interface interface serial-options loopback { dce-local | local | remote }
```

**Line Encoding**

By default, serial interfaces use non-return to zero (NRZ) line encoding. You can configure non-return to zero inverted (NRZI) line encoding if necessary.

To have the interface use NRZI line encoding, specify the `nrzi` option:

```
set interface interface serial-options encoding nrzi
```

When setting the line-encoding parameter, you can set different encoding parameters for ports on the same PIM.

**Example Configurations**

The following shows the CLI configurations on two SSG devices connected as peers through serial ports.

Peer A:

```
set int s6/0 serial-options encoding nrz
set int s6/0 serial-options clocking-mode internal
set int s6/0 serial-options clock-rate 8.0
```

Peer B:

```
set int s6/0 serial-options encoding nrz
set int s6/0 serial-options clocking-mode dce
set int s6/0 serial-options clock-rate 8.0
```
Overview of T1 Interfaces

T1 is the basic Physical Layer protocol used by the Digital Signal level 1 (DS1) multiplexing method in North America. A T1 interface operates at a bit rate of 1.544 megabits per second (Mbps) and can support 24 digital signal 0 (DS0) channels. Supported DS1 standards include:

- American National Standards Institute (ANSI) T1.107, *Digital Hierarchy - Formats Specifications*, describes digital-hierarchy formats and is used in conjunction with T1.102, *Digital Hierarchy - Electrical Interfaces*.


- International Telecommunications Union (ITU-T) Recommendations G.751 and G.703 describe physical and electrical characteristics of hierarchical digital interfaces.
Configuring T1 Interface Options with the WebUI

To use the Web User Interface (WebUI) to configure T1 interface options, navigate to the WAN dialog box for the T1 interface:

Network > Interfaces > Edit (interface) > WAN: Enter or select the applicable option value, and then click Apply.

Table 3 shows the default settings for T1 interface options along with the alternate values you can enter or select. For more information about each option, see “Configuring T1 Interface Options with the CLI” on page 13.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold time:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Up</td>
<td>0 milliseconds</td>
<td>0-65534 milliseconds</td>
</tr>
<tr>
<td>Down</td>
<td>0 milliseconds</td>
<td>0-65534 milliseconds</td>
</tr>
<tr>
<td>Clocking source</td>
<td>Internal</td>
<td>External</td>
</tr>
<tr>
<td>T1 options:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line buildout</td>
<td>0-132 feet (0-40 meters)</td>
<td>133-265 feet (40-81 meters) 266-398 feet (81-121 meters) 399-531 feet ((121-162 meters) 532-655 feet (162-200 meters)</td>
</tr>
<tr>
<td>Line encoding</td>
<td>Binary 8 zero substitution (B8ZS)</td>
<td>Alternate mark inversion (AMI)</td>
</tr>
<tr>
<td>Byte encoding</td>
<td>7 bits per byte (nx56)</td>
<td>8 bits per byte (nx64)</td>
</tr>
<tr>
<td>Frame checksum</td>
<td>16 bits</td>
<td>32 bits</td>
</tr>
<tr>
<td>Framing mode</td>
<td>Extended superframe</td>
<td>Superframe</td>
</tr>
<tr>
<td>Transmitting flag in idle cycles</td>
<td>Flags (0x7E)</td>
<td>All ones (0xFF)</td>
</tr>
<tr>
<td>Start/end flags on transmission</td>
<td>Filler</td>
<td>Shared</td>
</tr>
<tr>
<td>Invert data</td>
<td>Not set</td>
<td>Enabled</td>
</tr>
<tr>
<td>Remote loopback respond</td>
<td>Not set</td>
<td>Enabled</td>
</tr>
<tr>
<td>Time slots</td>
<td>All active</td>
<td>Specify active time slots</td>
</tr>
<tr>
<td>Loopback mode</td>
<td>None</td>
<td>Local, Payload, or Remote</td>
</tr>
<tr>
<td>BERT options:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT algorithm</td>
<td>pseudo-2e15-o151 (2^15-1)</td>
<td>Select from drop-down list</td>
</tr>
<tr>
<td>BERT error rate</td>
<td>0 (10^-9)</td>
<td>0 (10^-9) - 7 (10^-7)</td>
</tr>
<tr>
<td>BERT length</td>
<td>10 seconds</td>
<td>1-240 seconds</td>
</tr>
</tbody>
</table>
Configuring T1 Interface Options with the CLI

You can use the command line interface (CLI) to configure the T1 options shown in Table 3. This section describes the T1 interface options and shows the CLI command you enter to configure each option.

**Hold Time**

By default, when an interface changes from up to down or from down to up, this transition is advertised immediately to the hardware and the ScreenOS software. In some situations—for example, when an interface is connected to an add-drop multiplexer (ADM) or wavelength-division multiplexer (WDM), or to protect against SONET/SDH framer holes—you might want to damp interface transitions. This means not advertising the interface’s transition until a certain period of time, called the *hold time*, has passed. When you have damped interface transitions and the interface goes from up to down, the interface is not advertised to the rest of the system as being down until it has remained down for the hold-time period. Similarly, when an interface goes from down to up, it is not advertised as being up until it has remained up for the hold-time period.

To damp interface transitions:

```
set interface interface hold-time { down milliseconds | up milliseconds }
```

The time can be a value from 0 through 65,534 milliseconds. The default value is 0, which means that interface transitions are not damped. The ScreenOS software advertises the transition within 100 milliseconds of the time value you specify.

**Clocking Source**

You can configure the transmit clock on each interface. The transmit clock aligns each outgoing packet transmitted over the SSG device’s interfaces. The clock source can be the device’s internal stratum 3 clock, which resides on the control board, or an external clock that is received from the interface you are configuring. By default, the interface clock source is internal, which means that each interface uses the device’s internal stratum 3 clock. For interfaces that can use different clock sources, the source can be internal (also called line timing or normal timing) or external (also called loop timing).

To set the clock source of an interface to use an external clock:

```
set interface interface clocking external
```
**Line Buildout**

A T1 interface has five possible setting ranges for the line buildout:

- 0-132 feet (0-40 meters)
- 133-265 feet (40-81 meters)
- 266-398 feet (81-121 meters)
- 399-531 feet (121-162 meters)
- 532-655 feet (162-200 meters)

By default, the T1 interface uses the shortest setting (0-132 feet, or 0-40 meters).

To set the interface to drive a line at one of the longer distance ranges, specify the range value:

```
set interface interface t1-options buildout 266-398
```

**Line Encoding**

By default, a T1 interface uses 8 bits zero suppression (B8ZS) line encoding. You can configure Automatic Mark Inversion (AMI) line encoding if necessary. When setting the line-encoding parameter, you must set the same value for paired ports. Ports 0 and 1 must share the same value.

To configure the interface to use AMI line encoding:

```
set interface interface t1-options line-encoding ami
```

**Byte Encoding**

By default, a T1 interface uses a byte encoding of 8 bits per byte (nx64). You can configure an alternative byte encoding of 7 bits per byte (nx56).

To configure the interface to use 7 bits per byte encoding:

```
set interface interface t1-options byte-encoding nx56
```

**Frame Checksum**

By default, a T1 interface uses a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment might not support 32-bit checksums.

To configure the interface to use a 32-bit checksum:

```
set interface interface t1-options fcs 32
```
**Framing Mode**

By default, a T1 interface uses extended superframe (ESF) framing format. You can configure superframe (SF) as an alternative.

To set the interface to use SF framing format:

```
set interface interface t1-options framing sf
```

**Idle Cycle Flag**

By default, a T1 interface transmits the value 0x7E in the idle cycles. To have the interface transmit the value 0xFF (all ones) instead:

```
set interface interface t1-options idle-cycle-flag ones
```

**Start/End Flags**

By default, a T1 interface waits two idle cycles between sending start and end flags. To configure the interface to share the transmission of start and end flags:

```
set interface interface t1-options start-end-flag shared
```

**Data Inversion**

When you enable data inversion, all data bits in the data stream are transmitted inverted; that is, zeroes are transmitted as ones and ones as zeroes. Data inversion is normally used only in AMI mode to provide ones density in the transmitted stream. By default, data inversion is disabled.

To enable data inversion:

```
set interface interface t1-options invert-data
```

**Remote Loopback Response**

The T1 facilities data-link loop request signal is used to communicate various network information in the form of in-service monitoring and diagnostics. ESF, through the facilities data link (FDL), supports nonintrusive signaling and control, thereby offering clear-channel communication. Remote loopback requests can be over the FDL or inband. By default, the SSG device does not respond to remote-loopback requests.

To configure the interface to respond to remote-loopback requests:

```
set interface interface t1-options remote-loopback-respond
```
**Fractional T1 Time Slots**

By default, all 24 time slots on a T1 interface are used. You can designate any combination of time slots. For a T1 interface, the time-slot range is from 1 through 24.

Use hyphens to configure ranges of time slots. Use commas to configure discontinuous time slots. Do not include spaces.

To allocate a specific set of time slots to a fractional T1 interface:

```
set interface interface t1-options timeslots 1-5,10,24
```

**Loopback Mode**

You can configure loopback capability between the local T1 interface and the remote channel service unit (CSU), as shown in Figure 6 on page 16. You can configure the loopback to be local or remote. With local loopback, the T1 interface can transmit packets to the CSU but receives its own transmission back again and ignores data from the CSU. With remote loopback, packets sent from the CSU are received by the T1 interface, forwarded if there is a valid route, and immediately retransmitted to the CSU. Local and remote loopback transmissions loop back both data and clocking information.

**Figure 6: Remote and Local T1 Loopback**

For T1 interfaces, you can specify the loopback payload option to loop back data only without clocking information on the remote router’s PIM. Overhead is recalculated.

To configure loopback capability on a T1 interface:

```
set interface interface t1-options loopback { local | payload | remote }
```

**Bit Error Rate Testing Options**

Bit error rate testing (BERT) allows you to troubleshoot problems by checking the quality of links. You can configure any of the WAN interfaces on an SSG device to execute a BERT when the interface receives a request to run this test.

A BERT requires a line loop to be in place on either the transmission device or the far-end router. The local router generates a known bit pattern and sends it out the transmit path. The received pattern is then verified against the sent pattern. The higher the bit error rate (BER) of the received pattern, the worse the noise is on the physical circuit. As you move the position of the line loop increasingly downstream toward the far-end router, you can isolate the troubled portion of the link.
Configuring BERT

You must specify the following parameters for the BERT:

- Duration of the test. By default, the BERT period is 10 seconds. You can configure the BERT period to last from 1 through 240 seconds on SSG PIMs.

- Bit pattern or algorithm to send on the transmit path. The default is pseudo-2e15-o151. To see a list of supported algorithms in the CLI, enter a `?` after the `bert-algorithm` option.

- Error rate to monitor when receiving the inbound pattern. You specify this rate in the form of an integer from 0 (the default) through 7, which corresponds to a BER from $10^{-0}$ (1 error per bit) to $10^{-7}$ (1 error per 10 million bits).

To configure BERT parameters:

```
set interface interface t1-options [ bert-algorithm string ] [ bert-error-rate number ] [ bert-period seconds ]
```

Starting and Stopping a BERT

Before you can start the BERT, you must disable the interface. To do this, issue the `set interface interface disable` command.

After you configure the BERT properties and commit the configuration, begin the test by issuing the `exec interface interface bert-test start` command.

The test runs for the duration you specify with the `bert-period` option. If you wish to terminate the test sooner, issue the `exec interface interface bert-test stop` command.

To view the results of the BERT:

```
get counter statistics interface interface extensive
```

BERT counts:
  bit_count 0, error_bit_count 0
BERT stats:
  started 0, completed 1, los 0, elapsed 0, los_trans 0
  los_errorseconds 0, bit_count_oflow 0, ebit_count_oflow 0

NOTE: To exchange BERT patterns between a local routing platform and a remote routing platform, include the `loopback remote` option in the interface configuration at the remote end of the link. From the local routing platform, issue the `exec interface interface bert-test start` command.
Example Configurations

The following shows the CLI configurations on two SSG devices connected as peers through T1 ports.

Peer A:

set int s3/0 t1-options buildout 0-132
set int s3/0 t1-options fcs 16
set int s3/0 t1-options framing esf
set int s3/0 t1-options idle-cycle-flag flags
set int s3/0 t1-options start-end-flag filler
set int s3/0 clocking external

Peer B:

set int s3/0 t1-options buildout 0-132
set int s3/0 t1-options fcs 16
set int s3/0 t1-options framing esf
set int s3/0 t1-options idle-cycle-flag flags
set int s3/0 t1-options start-end-flag filler
set int s3/0 clocking internal
Chapter 4

Configuring E1 Interfaces

E1 Physical Interface Modules (PIMs) on Secure Services Gateway (SSG) devices have two E1 ports with integrated Channel Service Unit/Data Service Unit (CSU/DSU). These ports provide physical connections to E1 or fractional E1 network media types.

This chapter describes how to configure interfaces on an E1 PIM in an SSG device. It contains the following sections:

- “Overview of E1 Interfaces” on page 19
- “Configuring E1 Interface Options with the WebUI” on page 20
- “Configuring E1 Interface Options with the CLI” on page 21

Overview of E1 Interfaces

E1 is a standard wide area network (WAN) digital communications format designed to operate over copper facilities at a rate of 2.048 megabits per second (Mbps). Widely used outside North America, it is a basic time-division multiplexing scheme used to carry digital circuits. The following standards apply to E1 interfaces:

- ITU-T Recommendation G.775, Loss of Signal (LOS) and Alarm Indication Signal (AIS) Defect Detection and Clearance Criteria, describes alarm reporting methods.

NOTE: The Juniper Networks E1 Physical Interface Module (PIM) does not support Channel Associated Signaling (CAS).
Configuring E1 Interface Options with the WebUI

To use the Web User Interface (WebUI) to configure E1 interface options, navigate to the WAN dialog box for the E1 interface:

```
Network > Interfaces > Edit (interface) > WAN: Enter or select the applicable option value, and then click Apply.
```

Table 4 shows the default settings for E1 interface options along with the alternate values you can enter or select. For more information about each option, see “Configuring E1 Interface Options with the CLI” on page 21.

**Table 4: Default and Alternate Values for E1 Interface Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold time:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Up</td>
<td>0 milliseconds</td>
<td>0-65534 milliseconds</td>
</tr>
<tr>
<td>Down</td>
<td>0 milliseconds</td>
<td>0-65534 milliseconds</td>
</tr>
<tr>
<td>Clocking source</td>
<td>Internal</td>
<td>External</td>
</tr>
<tr>
<td>E1 options:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frame checksum</td>
<td>16 bits</td>
<td>32 bits</td>
</tr>
<tr>
<td>Framing mode</td>
<td>G704 with CRC4</td>
<td>G704 without CRC4 or unframed</td>
</tr>
<tr>
<td>Transmitting flag in idle cycles</td>
<td>Flags (0x7E)</td>
<td>All ones (0xFF)</td>
</tr>
<tr>
<td>Start/end flags on transmission</td>
<td>Filler</td>
<td>Shared</td>
</tr>
<tr>
<td>Invert data</td>
<td>Not set</td>
<td>Enabled</td>
</tr>
<tr>
<td>Time slots</td>
<td>All active</td>
<td>Specify active timeslots</td>
</tr>
<tr>
<td>Loopback mode</td>
<td>None</td>
<td>Local or Remote</td>
</tr>
<tr>
<td>BERT options:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT algorithm</td>
<td>pseudo-2e15-o151 (2^15-1)</td>
<td>Select from drop-down list</td>
</tr>
<tr>
<td>BERT error rate</td>
<td>0 (10^-5)</td>
<td>0 (10^-5) - 7 (10^-7)</td>
</tr>
<tr>
<td>BERT length</td>
<td>10 seconds</td>
<td>1-240 seconds</td>
</tr>
</tbody>
</table>
You can use the command line interface (CLI) to configure the T1 options shown in Table 4. This section describes the T1 interface options and shows the CLI command you enter to configure each option.

**Hold Time**

By default, when an interface changes from up to down or from down to up, this transition is advertised immediately to the hardware and the ScreenOS software. In some situations—for example, when an interface is connected to an add-drop multiplexer (ADM) or wavelength-division multiplexer (WDM), or to protect against SONET/SDH framer holes—you might want to damp interface transitions. This means not advertising the interface's transition until a certain period of time, called the *hold time*, has passed. When you have damped interface transitions and the interface goes from up to down, the interface is not advertised to the rest of the system as being down until it has remained down for the hold-time period. Similarly, when an interface goes from down to up, it is not advertised as being up until it has remained up for the hold-time period.

To damp interface transitions:

```
set interface interface hold-time { down milliseconds | up milliseconds }
```

The time can be a value from 0 through 65,534 milliseconds. The default value is 0, which means that interface transitions are not damped. The ScreenOS software advertises the transition within 100 milliseconds of the time value you specify.

**Clocking Source**

You can configure the *transmit clock* on each interface. The transmit clock aligns each outgoing packet transmitted over the SSG device’s interfaces. The clock source can be the device’s internal stratum 3 clock, which resides on the control board, or an external clock that is received from the interface you are configuring. By default, the interface clock source is internal, which means that each interface uses the device’s internal stratum 3 clock. For interfaces that can use different clock sources, the source can be internal (also called *line timing* or *normal timing*) or external (also called *loop timing*).

To set the clock source of an interface to use an external clock:

```
set interface interface clocking external
```

**Frame Checksum**

By default, an E1 interface supports a 16-bit checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment might not support 32-bit checksums.

To configure the interface to use a 32-bit checksum:

```
set interface interface e1-options fcs 32
```
**Framing Mode**

By default, an E1 interface uses the G704 framing mode with cyclic redundancy check 4 (CRC 4). If needed, you can configure the unframed mode or the G704 mode without CRC 4.

To set the interface to use unframed mode:

```
set interface interface e1-options framing unframed
```

To explicitly configure an interface’s G704 framing to not use CRC4:

```
set interface interface e1-options framing g704-no-crc4
```

**Idle Cycle Flag**

By default, an E1 interface transmits the value 0x7E in the idle cycles. To have the interface transmit the value 0xFF (all ones) instead:

```
set interface interface e1-options idle-cycle-flag ones
```

**Start/End Flags**

By default, an E1 interface waits two idle cycles between sending start and end flags. To configure the interface to share the transmission of start and end flags:

```
set interface interface e1-options start-end-flag shared
```

**Data Inversion**

When you enable data inversion, all data bits in the data stream are transmitted inverted; that is, zeroes are transmitted as ones and ones as zeroes. Data inversion is normally used only in AMI mode to provide ones density in the transmitted stream. By default, data inversion is disabled.

To enable data inversion:

```
set interface interface e1-options invert-data
```

**Fractional E1 Time Slots**

By default, all available time slots on an E1 interface are used. You can designate any combination of time slots. Time slots 0 and 1 are reserved for framing and cannot be used to configure a fractional E1 interface. For an E1 interface, the time-slot range is from 2 through 32.

Use hyphens to configure ranges of time slots. Use commas to configure discontinuous time slots. Do not include spaces.

To allocate a specific set of time slots to a fractional E1 interface:

```
set interface interface e1-options timeslots 4-6,11,25
```
**Loopback Mode**

You can configure loopback capability between the local E1 interface and the remote channel service unit (CSU), as shown in Figure 7. You can configure the loopback to be local or remote. With local loopback, the E1 interface can transmit packets to the CSU but receives its own transmission back again and ignores data from the CSU. With remote loopback, packets sent from the CSU are received by the E1 interface, forwarded if there is a valid route, and immediately retransmitted to the CSU. Local and remote loopback transmissions loop back both data and clocking information.

**Figure 7: Remote and Local E1 Loopback**

To configure loopback capability on an E1 interface:

```
set interface interface e1-options loopback { local | remote }
```

Packets can be looped on either the local routing platform or the remote CSU.

**Bit Error Rate Testing Options**

Bit error rate testing (BERT) allows you to troubleshoot problems by checking the quality of links. You can configure any of the WAN interfaces on an SSG device to execute a BERT when the interface receives a request to run this test.

A BERT requires a line loop to be in place on either the transmission devices or the far-end router. The local router generates a known bit pattern and sends it out the transmit path. The received pattern is then verified against the sent pattern. The higher the bit error rate (BER) of the received pattern, the worse the noise is on the physical circuit. As you move the position of the line loop increasingly downstream toward the far-end router, you can isolate the troubled portion of the link.
Configuring BERT

You must configure the following parameters for the BERT:

- Duration of the test. By default, the BERT period is 10 seconds. You can configure the BERT period to last from 1 through 240 seconds on SSG PIMs.

- Bit pattern or algorithm to send on the transmit path. The default is pseudo-2e15-o151. To see a list of supported algorithms in the CLI, enter a ? after the bert-algorithm option.

- Error rate to monitor when receiving the inbound pattern. You specify this rate in the form of an integer from 0 (the default) through 7, which corresponds to a BER from $10^{-0}$ (1 error per bit) to $10^{-7}$ (1 error per 10 million bits).

To configure BERT parameters:

```
set interface interface e1-options [ bert-algorithm string ] [ bert-error-rate number ] [ bert-period seconds ]
```

Starting and Stopping a BERT

Before you can start the BERT, you must disable the interface. To do this, issue the `set interface interface disable` command.

After you configure the BERT properties and commit the configuration, begin the test by issuing the `exec interface interface bert-test start` command.

The test runs for the duration you specify with the bert-period option. If you wish to terminate the test sooner, issue the `exec interface interface bert-test stop` command.

To view the results of the BERT:

```
get counter statistics interface interface extensive
```

BERT counts:
- bit_count 0, error_bit_count 0
BERT stats:
- started 0, completed 1, los 0, elapsed 0, los_trans 0
- los_errorseconds 0, bit_count_oflow 0, ebit_count_oflow 0

**NOTE:** To exchange BERT patterns between a local routing platform and a remote routing platform, include the loopback remote option in the interface configuration at the remote end of the link. From the local routing platform, issue the `exec interface interface bert-test start` command.
Example Configurations

The following shows the CLI configurations on two SSG devices connected as peers through E1 ports.

Peer A:

set int s6/1 e1-options fcs 16
set int s6/1 e1-options framing g704
set int s6/1 e1-options idle-cycle-flag flags
set int s6/1 e1-options start-end-flag filler
set int s6/1 e1-options timeslots 2-32
set int s6/1 clocking external

Peer B:

set int s6/1 e1-options fcs 32
set int s6/1 e1-options framing g704
set int s6/1 e1-options idle-cycle-flag flags
set int s6/1 e1-options start-end-flag filler
set int s6/1 e1-options timeslots 2-32
set int s6/1 clocking internal
Chapter 5

Configuring T3 Interfaces

Digital Signal level 3 (DS3) Physical Interface Modules (PIMs) on Secure Services Gateway (SSG) devices contain one physical DS3 port with integrated Data Service Unit (DSU). This port provides physical connection to T3 network media types at a bit rate of 44.736 megabits per second (Mbps).

This chapter describes how to configure an interface on a T3 PIM in an SSG device. It contains the following sections:

- “Overview of T3 Interfaces” on page 27
- “Configuring T3 Interface Options with the WebUI” on page 28
- “Configuring T3 Interface Options with the CLI” on page 29
- “Example Configurations” on page 35

Overview of T3 Interfaces

T3 is the Physical Layer protocol used by the DS3 multiplexing method in North America. Supported DS3 standards include:

- American National Standards Institute (ANSI) T1.107, *Digital Hierarchy - Formats Specifications*, describes digital-hierarchy formats and is used in conjunction with T1.102, *Digital Hierarchy - Electrical Interfaces*.


- Telcordia TR-TSY-000009, *Asynchronous Digital Multiplexes, Requirements and Objectives*, describes generic technical requirements and objectives for asynchronous multiplexes that operate at DS1C (3.152 Mbps), DS2 (6.312 Mbps), and/or DS3 (44.736 Mbps) digital rates.

ITU G.751, *Digital multiplex equipment operating at the third order bit rate of 34 368 kbit/s and the fourth order bit rate of 139 264 kbit/s and using positive justification*, G.703, *Physical/electrical characteristics of hierarchical digital interfaces*, and G.823, *The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy*, describe transmission systems and media, digital systems, and networks.

### Configuring T3 Interface Options with the WebUI

To use the Web User Interface (WebUI) to configure T3 interface options, navigate to the WAN dialog box for the T3 interface:

```
Network > Interfaces > Edit (interface) > WAN: Enter or select the applicable option value, and then click Apply.
```

Table 5 shows the default settings for T3 interface options along with the alternate values you can enter or select. For more information about each option, see “Configuring T3 Interface Options with the CLI” on page 29.
Table 5: Default and Alternate Values for T3 Interface Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold time:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Up</td>
<td>0 milliseconds</td>
<td>0-65534 milliseconds</td>
</tr>
<tr>
<td>Down</td>
<td>0 milliseconds</td>
<td>0-65534 milliseconds</td>
</tr>
<tr>
<td>Clocking source</td>
<td>Internal</td>
<td>External</td>
</tr>
<tr>
<td>T3 options:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frame checksum</td>
<td>16 bits</td>
<td>32 bits</td>
</tr>
<tr>
<td>Transmitting flag in idle cycles</td>
<td>Flags (0x7E)</td>
<td>All ones (0xFF)</td>
</tr>
<tr>
<td>Start/end flags on transmission</td>
<td>Filler</td>
<td>Shared</td>
</tr>
<tr>
<td>C-bit parity mode</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Long buildout</td>
<td>Not set</td>
<td>Enabled</td>
</tr>
<tr>
<td>Respond to FEAC loopback requests</td>
<td>Not set</td>
<td>Enabled</td>
</tr>
<tr>
<td>Payload scrambling</td>
<td>Not set</td>
<td>Enabled</td>
</tr>
<tr>
<td>CSU compatibility mode</td>
<td>Not set</td>
<td>Adtran, Kentrox, Larscom, Verilink, or Digital Link subrates</td>
</tr>
<tr>
<td>Loopback mode</td>
<td>None</td>
<td>Local, Payload, or Remote</td>
</tr>
<tr>
<td>BERT options:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT algorithm</td>
<td>pseudo-2e15-o151</td>
<td>Select from drop-down list</td>
</tr>
<tr>
<td>BERT error rate</td>
<td>0 (10^-5)</td>
<td>0 (10^-5) - 7 (10^-7)</td>
</tr>
<tr>
<td>BERT length</td>
<td>10 seconds</td>
<td>1-240 seconds</td>
</tr>
</tbody>
</table>

Configuring T3 Interface Options with the CLI

You can use the command line interface (CLI) to configure the T3 options shown in Table 5. This section describes the T3 interface options and shows the CLI command you enter to configure each option.

**Hold Time**

By default, when an interface changes from up to down or from down to up, this transition is advertised immediately to the hardware and the ScreenOS software. In some situations—for example, when an interface is connected to an add-drop multiplexer (ADM) or wavelength-division multiplexer (WDM), or to protect against SONET/SDH framer holes—you might want to damp interface transitions. This means not advertising the interface’s transition until a certain period of time, called the hold time, has passed. When you have damped interface transitions and the interface goes from up to down, the interface is not advertised to the rest of the system as being down until it has remained down for the hold-time period. Similarly, when an interface goes from down to up, it is not advertised as being up until it has remained up for the hold-time period.
To damp interface transitions:

```
set interface interface hold-time { down milliseconds | up milliseconds }
```

The time can be a value from 0 through 65,534 milliseconds. The default value is 0, which means that interface transitions are not damped. The ScreenOS software advertises the transition within 100 milliseconds of the time value you specify.

**Clocking Source**

You can configure the *transmit clock* on each interface. The transmit clock aligns each outgoing packet transmitted over the SSG device’s interfaces. The clock source can be the device’s internal stratum 3 clock, which resides on the control board, or an external clock that is received from the interface you are configuring. By default, the interface clock source is internal, which means that each interface uses the device’s internal stratum 3 clock. For interfaces that can use different clock sources, the source can be internal (also called line timing or normal timing) or external (also called loop timing).

To set the clock source of an interface to use an external clock:

```
set interface interface clocking external
```

**Frame Checksum**

By default, a T3 interface uses a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment might not support 32-bit checksums.

To configure a 32-bit checksum:

```
set interface interface t3-options fcs 32
```

**Idle Cycle Flag**

By default, a T3 interface transmits the value 0x7E in the idle cycles. To have the interface transmit the value 0xFF (all ones) instead, specify the `idle-cycle-flag` option with the `ones` value:

```
set interface interface t3-options idle-cycle-flag ones
```

**Start/End Flags**

By default, a T3 interface waits two idle cycles between sending start and end flags. To configure the interface to share the transmission of start and end flags:

```
set interface interface t3-options start-end-flag
```
**C-Bit Parity Mode**

C-bit parity mode controls the type of framing that is present on the transmitted T3 signal. When C-bit parity mode is enabled, the C-bit positions are used for the FEBE, FEAC, terminal data link, path parity, and mode indicator bits, as defined in ANSI T1.107a-1989. When C-bit parity mode is disabled, the basic T3 framing mode (M13) is used.

By default, C-bit parity mode is enabled. To disable C-bit parity mode and use M13 framing for your T3 link, unset the `cbit-parity` option for the interface:

```
unset interface interface t3-options cbit-parity
```

**Line Buildout**

A T3 interface has two settings for the T3 line buildout: a short setting, which is less than 255 feet (about 68 meters), and a long setting, which is greater than 255 feet and less than 450 feet (about 137 meters). By default, the interface uses the short setting.

To set the interface to drive a line that is longer than 255 feet and shorter than 450 feet, specify the `long-buildout` option:

```
set interface interface t3-options long-buildout
```

**FEAC Response**

The T3 far-end alarm and control (FEAC) signal is used to send alarm or status information from the far-end terminal back to the near-end terminal and to initiate T3 loopbacks at the far-end terminal from the near-end terminal.

By default, the routing platform does not respond to FEAC requests. To allow the remote Channel Service Unit (CSU) to place the local routing platform into loopback, you must configure the routing platform to respond to the CSU’s FEAC request by specifying the `feac-loop-respond` option:

```
set interface interface t3-options feac-loop-respond
```

If you configure remote or local loopback with the T3 `loopback` option, the routing platform does not respond to FEAC requests from the CSU even if you include the `feac-loop-respond` option in the configuration. For the routing platform to respond, you must delete the `loopback` option from the configuration.

**HDLC Payload Scrambling**

T3 HDLC payload scrambling, which is disabled by default, provides better link stability. Both sides of a connection must either use or not use scrambling.

To configure scrambling on the DS3 channels on the interface:

```
set interface interface t3-options payload-scrambler
```
**CSU Compatibility Mode**

Subrating a T3 interface reduces the maximum allowable peak rate by limiting the HDLC-encapsulated payload. Subrate modes configure the PIM to connect with channel service units (CSUs) that use proprietary methods of multiplexing.

You can configure a T3 interface to be compatible with a Digital Link, Kentrox, Adtran, Verilink, or Larscom CSU.

To configure a T3 interface so that it is compatible with the CSU at the remote end of the line, specify the `compatibility-mode` option and the `subrate` value. The subrate of a T3 interface must exactly match that of the remote CSU.

```
set interface interface t3-options compatibility-mode { adtran | digital-link | kentrox | larscom | verilink } subrate number
```

- For an Adtran CSU, specify the subrate as a number from 1 through 588 that exactly matches the value configured on the CSU. A subrate value of 588 corresponds to 44.2 Mbps, or 100 percent of the HDLC-encapsulated payload. A subrate value of 1 corresponds to 44.2 / 588, which is 75.17 Kbps, or 0.17 percent of the HDLC-encapsulated payload.

- For a Digital Link CSU, specify the subrate as the data rate you configured on the CSU in the format xKb or x.xMb. For Digital Link CSUs, you can specify the subrate value to match the data rate configured on the CSU in the format xKb or x.xMb. For a list of supported values, enter a ? after the `compatibility-mode digital-link subrate` option.

- For a Kentrox CSU, specify the subrate as a number from 1 through 69 that exactly matches the value configured on the CSU. A subrate value of 69 corresponds to 34.995097 Mbps, or 79.17 percent of the HDLC-encapsulated payload (44.2 Mbps). A subrate value of 1 corresponds to 999.958 Kbps, which is 2.26 percent of the HDLC-encapsulated payload. Each increment of the subrate value corresponds to a rate increment of about 0.5 Mbps.

- For a Larscom CSU, specify the subrate as a number from 1 through 14 that exactly matches the value configured on the CSU. A subrate value of 14 corresponds to 44.2 Mbps, or 100 percent of the HDLC-encapsulated payload. A subrate value of 1 corresponds to 44.2 / 14, which is 3.16 Mbps, 7.15 percent of the HDLC-encapsulated payload.

- For a Verilink CSU, specify the subrate as a number from 1 through 28 that exactly matches the value configured on the CSU. To calculate the maximum allowable peak rate, multiply the configured subrate by 1.578 Mbps. For example, a subrate value of 28 corresponds to 28 x 1.578 Mbps, which is 44.2 Mbps, 100 percent of the HDLC-encapsulated payload. A subrate value of 1 corresponds to 1.578 Mbps, 3.57 percent of the HDLC-encapsulated payload. A subrate value of 20 corresponds to 20 x 1.578 Mbps, which is 31.56 Mbps, 71.42 percent of the HDLC-encapsulated payload.
Loopback Mode

You can configure loopback capability between the local T3 interface and the remote CSU, as shown in Figure 8. You can configure the loopback to be local or remote. With local loopback, the T3 interface can transmit packets to the CSU but receives its own transmission back again and ignores data from the CSU. With remote loopback, packets sent from the CSU are received by the T3 interface, forwarded if there is a valid route, and immediately retransmitted to the CSU.

Figure 8: Remote and Local T3 Loopback

To configure loopback capability on a T3 interface:

```
set interface interface t3-options { local | payload | remote }
```

Packets can be looped on either the local routing platform or the remote CSU. Local and remote loopback transmissions loop back both data and clocking information.

For channelized T3 interfaces, you can specify the `loopback payload` option to loop back data only (without clocking information) on the remote router’s PIM. Overhead is recalculated.

Bit Error Rate Testing Options

Bit error rate testing (BERT) allows you to troubleshoot problems by checking the quality of links. You can configure any of the WAN interfaces on an SSG device to execute a BERT when the interface receives a request to run this test.

A BERT requires a line loop to be in place on either the transmission device or the far-end router. The local router generates a known bit pattern and sends it out the transmit path. The received pattern is then verified against the sent pattern. The higher the bit error rate (BER) of the received pattern, the worse the noise is on the physical circuit. As you move the position of the line loop increasingly downstream toward the far-end router, you can isolate the troubled portion of the link.
Configuring BERT

You must configure the following parameters for the BERT:

- Duration of the test. By default, the BERT period is 10 seconds. You can configure the BERT period to last from 1 through 240 seconds on SSG PIMs.

- Bit pattern or algorithm to send on the transmit path. The default is pseudo-2e15-o151. To see a list of supported algorithms in the CLI, enter a ? after the `bert-algorithm` option.

- Error rate to monitor when receiving the inbound pattern. You specify this rate in the form of an integer from 0 (the default) through 7, which corresponds to a BER from $10^{-0}$ (1 error per bit) to $10^{-7}$ (1 error per 10 million bits).

To configure BERT parameters:

```
set interface interface t3-options [ bert-algorithm string ] [ bert-error-rate number ]
    [ bert-period seconds ]
```

Starting and Stopping a BERT

Before you can start the BERT, you must disable the interface. To do this, issue the `set interface interface disable` command.

After you configure the BERT properties and commit the configuration, begin the test by issuing the `exec interface interface bert-test start` command.

The test runs for the duration you specify with the `bert-period` option. If you wish to terminate the test sooner, issue the `exec interface interface bert-test stop` command.

To view the results of the BERT:

```
get counter statistics interface interface extensive
```

BERT counts:
- bit_count 0, error_bit_count 0
BERT stats:
- started 0, completed 1, los 0, elapsed 0, los_trans 0
- los_errorseconds 0, bit_count_oflow 0, ebit_count_oflow 0

**NOTE:** To exchange BERT patterns between a local routing platform and a remote routing platform, include the `loopback remote` option in the interface configuration at the remote end of the link. From the local routing platform, issue the `exec interface interface bert-test start` command.
**Example Configurations**

The following shows the CLI configurations on two SSG devices connected as peers through T3 ports.

**Peer A:**

set int s4/0 t3-options fcs 16  
set int s4/0 t3-options idle-cycle-flag flags  
set int s4/0 t3-options start-end-flag filler  
set int s4/0 clocking external

**Peer B:**

set int s4/0 t3-options fcs 16  
set int s4/0 t3-options idle-cycle-flag flags  
set int s4/0 t3-options start-end-flag filler  
set int s4/0 clocking internal
Chapter 6

Configuring Frame Relay and Multilink Frame Relay

Wide area network (WAN) interfaces on Secure Services Gateway (SSG) devices support Frame Relay, a WAN protocol that operates at the Data Link Layer of the OSI Reference Model. Frame Relay encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*, and the Frame Relay Forum Implementation Agreement FRF3.1/3.2. The Frame Relay protocol allows you to reduce costs by using shared data-transmission facilities that are managed by a Frame Relay service provider. You pay fixed charges for the local connections from each site in the Frame Relay network to the first point of presence (POP) in which the provider maintains a Frame Relay switch. The portion of the network between Frame Relay switches is shared by all customers of the service provider.

WAN interfaces on SSG devices also support Multilink Frame Relay (MLFR) for User-to-Network Interface (UNI), based on the Frame Relay Forum FRF16, *Multilink Frame Relay UNI/Network-to-Network Interface (NNI) Implementation Agreement*. MLFR enables you to bundle multiple physical links into a single logical data link across the Frame Relay network. You use MLFR to increase bandwidth in smaller, more cost-effective increments. Bundling multiple physical links can add a level of fault-tolerance to WAN data links because you can implement bundling across multiple PIMs, thus protecting against the failure of any single PIM.

This chapter describes how to configure Frame Relay and MLFR on an SSG device. It contains the following sections:

- “Overview of the Frame Relay Network” on page 38
- “Configuring Frame Relay Encapsulation on a Single Physical Link” on page 38
- “Configuring Multilink Frame Relay” on page 44
- “Example Configurations” on page 53
Overview of the Frame Relay Network

Figure 9 depicts the devices in a Frame Relay network.

Figure 9: Devices in the Frame Relay Network

There are two categories of devices that can be attached to a Frame Relay Network:

- Data terminal equipment (DTE) devices are generally the terminating equipment for a specific network and are typically located on the customer premises. SSG devices and other routers are examples of DTEs.

- Data circuit-terminating equipment (DCE) devices are generally carrier-owned internetworking devices that provide switching services in a network. DCEs are typically packet switches.

A Frame Relay permanent virtual circuit (PVC) provides a logical connection between two DTE devices across a Frame Relay network. A number of PVCs can be multiplexed into a single physical circuit for transmission across the network. Each PVC is assigned a unique data-link connection identifier (DLCI) to ensure that each customer receives only their own traffic.

Configuring Frame Relay Encapsulation on a Single Physical Link

This section describes how to configure Frame Relay on a physical link that is not part of an MLFR bundle. To configure MLFR, see “Configuring Multilink Frame Relay” on page 44.

Basic Configuration Steps

To configure Frame Relay on a single physical link on an SSG device:

1. Configure Frame Relay encapsulation on the physical link, and assign the link to a security zone. You can also set other options, such as the maximum transfer unit (MTU), for the link.

2. (Optional) Configure Frame Relay options for the physical link. This step is required only if you need to change the default Frame Relay options for the link.

3. Create one or more PVCs for the physical link, and assign each PVC a Frame Relay DLCI and IP address.
Configuring Frame Relay Encapsulation on a Physical Link

On the SSG device, you configure a physical link by configuring the interface that represents the link. There is no default encapsulation type for WAN links on the SSG device; therefore, you must explicitly configure the Frame Relay encapsulation type for these interfaces.

To configure Frame Relay encapsulation on a physical interface:

**WebUI**

Network > Interfaces > Edit (interface): Select **Frame Relay** for the WAN Encapsulation, and then click **Apply**.

**CLI**

```
set interface interface encapsulation frame-relay
save
```

Assigning the Interface to a Zone

Security zones are logical entities to which one or more interfaces are bound. Through policies that you define, you can permit traffic between security zones. To permit traffic to flow from zone to zone, you bind an interface to the zone. (For more information about creating and configuring security zones, see the Fundamentals volume in the Concepts & Examples ScreenOS Reference Guide.)

**WebUI**

Network > Interfaces > Edit (interface): Select the zone from the **Zone Name** drop-down list, and then click **Apply**.

**CLI**

```
set interface interface zone zone
save
```

Setting the Protocol MTU

For ScreenOS, you can configure the protocol MTU on each physical interface. The default protocol MTU is 1500 bytes for serial, T1, E1, and multilink interfaces, and 4470 bytes for T3 interfaces. You can specify a value between 800 and 8192 bytes.

To configure the protocol MTU on a physical interface:

**WebUI**

Network > Interfaces > Edit (interface): Enter a value between 800 and 8192 in the **Maximum Transfer Unit (MTU)** field, and then click **OK**.

**CLI**

```
set interface interface mtu number
save
```
Note that in ScreenOS, the media MTU is derived from the protocol MTU. If you increase the size of the protocol MTU, you must ensure that the size of the media MTU is equal to or greater than the sum of the protocol MTU and the encapsulation overhead.

**Configuring Frame Relay Options for the Link**

Table 6 shows the default Frame Relay settings for a WAN interface configured for Frame Relay encapsulation. The sections that follow the table describe how to change option settings.

Table 6: Default and Alternate Values for Frame Relay Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Management Interface (LMI):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keepalive</td>
<td>Enabled</td>
<td>Disable</td>
</tr>
<tr>
<td>Type</td>
<td>ANSI</td>
<td>ITU</td>
</tr>
<tr>
<td>DTE full station polling interval count</td>
<td>6</td>
<td>1-5, 7-255</td>
</tr>
<tr>
<td>DTE error threshold count</td>
<td>3</td>
<td>1-2, 4-10</td>
</tr>
<tr>
<td>DTE monitored event count</td>
<td>4</td>
<td>1-3, 5-10</td>
</tr>
<tr>
<td>DTE keepalive timer</td>
<td>10 seconds</td>
<td>5-30 seconds</td>
</tr>
</tbody>
</table>

**Frame Relay Keepalives**

Frame Relay keepalives are implemented by the sending of Local Management Interface (LMI) packets. The sending of LMI keepalives is enabled by default on a Frame Relay interface.

For back-to-back Frame Relay connections, either disable the sending of keepalives on both sides of the connection, or configure one side of the connection as a data terminal equipment (DTE) (the default ScreenOS configuration) and the other as data circuit-terminating equipment (DCE).

If keepalives are enabled, the number of possible DLCI configurations on a multipoint or multicast connection is limited by the MTU size selected for the interface. To calculate the available DLCIs, use the formula \((MTU - 12) / 5\). To increase the number of possible DLCIs, disable keepalives.

To disable the sending of keepalives on a physical interface:

**WebUI**

Network > Interfaces > Edit (interface) > FR: Select No-Keepalive, and then click Apply.

**CLI**

```
set interface interface frame-relay lmi no-keepalive
save
```
Frame Relay LMI Type

By default, ScreenOS sends LMIs specified by ANSI T1.617 Annex D. To send ITU Q933 Annex A LMIs:

**WebUI**

Network > Interfaces > Edit (interface) > FR: Select ITU, and then click **Apply**.

**CLI**

set interface interface frame-relay lmi type itu
save

Keepalives for Frame Relay LMI

If the sending of keepalives is enabled (the default operation), you can tune keepalive settings. You can configure the following Frame Relay LMI keepalive options:

- **DTE full status polling interval** (denoted by the **n391-dte** keyword in the CLI). The DTE sends a status inquiry to the DCE at the interval specified by the DTE polling timer. The polling interval specifies the frequency at which these inquiries expect a full status report; for example, a value of 10 would specify a full status report in response to every tenth inquiry. The intermediate inquiries ask for a keepalive exchange only. The range is from 1 through 255, with a default value of 6.

- **DTE error threshold** (denoted by the **n392-dte** keyword in the CLI). The number of errors required to bring down the link, within the event-count specified by the DTE monitored event-count. The range is from 1 through 10, with a default value of 3.

- **DTE monitored event-count** (denoted by the **n393-dte** keyword in the CLI). The range is from 1 through 10, with a default value of 4.

- **DTE keepalive timer** (denoted by the **t391-dte** keyword in the CLI). The period at which the DTE sends out a keepalive response request to the DCE and updates status depending on the DTE error-threshold value. The range is from 5 through 30 seconds, with a default value of 10 seconds.

To configure Frame Relay keepalive options:

**WebUI**

Network > Interfaces > Edit (interface) > FR: Enter appropriate values for the LMI options, and then click **Apply**.

**CLI**

set interface interface frame-relay lmi option
save
Creating and Configuring PVCs

Frame Relay allows creation of multiple point-to-point virtual interfaces, or subinterfaces, within a single physical interface. Each subinterface maps to a permanent virtual circuit (PVC), which is identified by a data-link connection identifier (DLCI). A number of PVCs can be multiplexed onto a single physical link for transmission across a Frame Relay packet-switched network.

Figure 10 illustrates two point-to-point PVCs configured for the physical interface serial1. Note that each PVC can be associated with a different security zone; the security zone for each PVC can be different from the security zone assigned to the physical interface.

Figure 10: Point-to-Point Frame Relay Subinterfaces

To configure a point-to-point Frame Relay subinterface, create the subinterface and assign it to a security zone. Then assign a Frame Relay DLCI and IP address to the subinterface.

NOTE: You can assign a subinterface to a different security zone than that assigned to the physical interface.

WebUI

Network > Interface > New > WAN Sub-IF: Enter the following, and then click OK:

- Interface Name: interface (select)
- Zone Name: (select)
- Frame Relay DLCI: (enter id_num)
- IP Address/Netmask: (enter ip_addr)

CLI

```
set interface subinterface zone zone
set interface subinterface frame-relay dlci id_num
set interface subinterface ip ip_addr
save
```

The subinterface name consists of the physical interface name and a subinterface number. For example, if the physical interface name is serial1/1, its subinterfaces can be serial1/1.1 and serial1/1.2.

NOTE: In the WebUI, the subinterface number is automatically added when you select the interface name. In the CLI, you must enter both the interface name and the subinterface number.
You can specify only one DLCI for each subinterface. The DLCI is a value from 16 through 1022. (Numbers 1 through 15 are reserved.)

You can configure the subinterface to use inverse Frame Relay ARP; this is described in the next section.

You can also configure the subinterface for management functions such as a manage IP address, service options, and other features. (For more information about configuring manage IP and service options on an interface, see the Fundamentals volume in the Concepts & Examples ScreenOS Reference Guide.)

**Inverse Address Resolution Protocol**

Frame Relay subinterfaces can support inverse Address Resolution Protocol (ARP), as described in RFC 2390, *Inverse Address Resolution Protocol*. When inverse Frame Relay ARP is enabled, the routing platform responds to received inverse Frame Relay ARP requests by providing IP address information to the requesting routing platform on the other end of the Frame Relay PVC.

The routing platform does not initiate inverse Frame Relay ARP requests.

By default, inverse Frame Relay ARP is disabled. To configure a routing platform to respond to inverse Frame Relay ARP requests:

**WebUI**

Network > Interface > Edit (subinterface): Select Frame Relay Inverse ARP, and then click **Apply**.

**CLI**

```
set interface subinterface frame-relay inverse-arp
save
```

**Unnumbered Interfaces**

An unnumbered interface is not assigned an IP address but borrows an IP address from other interfaces. In this way, address space is conserved. If an unnumbered interface is pointing to an interface that is not functioning (Interface status UP or Protocol UP is not displayed) then the unnumbered interface does not work. We recommend that unnumbered interfaces point to a loopback interface since loopback interfaces do not fail.

To configure IP unnumbered interface:

**WebUI**

Network > Interface > Edit (WAN interface): Select the **Unnumbered option**, select the source interface, and then click **Apply**.

**CLI**

```
set interface interface ip unnumbered interface src interface
save
```
Configuring Multilink Frame Relay

Multilink Frame Relay (MLFR) provides a cost-effective way to increase bandwidth for applications by enabling multiple physical links to be aggregated into a bundle. Each physical link in the bundle is referred to as a bundle link. For example, if an application requires more bandwidth than is available on a T1 line, one option is to invest in a T3 line. However, T3 lines can be expensive and are not always available. MLFR allows multiple T1 bundle links to be aggregated into a single bundle.

**Figure 11: Multilink Frame Relay (MLFR) Bundle**

MLFR can also provide fault-tolerance. When a single bundle link in the bundle fails, the bundle continues to support Frame Relay service by transmitting across the remaining bundle links. MLFR also provides load balancing across the bundle links within a bundle. If a bundle link chosen for transmission is busy transmitting a long packet, another link can be used.

On the SSG device, you configure and access a bundle through a virtual interface called a multilink interface. The multilink interface emulates a physical interface for the transport of frames. The Frame Relay data link runs on the bundle, and PVCs are built upon the data link.

You configure Frame Relay functions on the multilink interface and not on each bundle link. (Although bundle links are visible to peer DTE and DCE devices, they are invisible to the Frame Relay Data Link Layer.) The local router and peer devices exchange Link Integrity Protocol (LIP) control messages to determine which bundle links are operational and to synchronize which bundle links are associated with each bundle.

For link management, each end of the bundle link follows the MLFR LIP and exchanges link control messages with its peer at the other end of the bundle link. To bring up a bundle link, both ends of the link must complete an exchange of ADD_LINK and ADD_LINK_ACK messages. To maintain the link, both ends periodically exchange HELLO and HELLO_ACK messages. The exchange of hello messages and acknowledgements serves as a keepalive mechanism for the link. If a router sends a hello message but does not receive an acknowledgement, it resends the hello message up to a configured maximum number of retries. If the router exhausts the maximum number of retries, the bundle link is considered down.

The bundle link’s status is considered to be up when the peer device acknowledges that it will use the link for the bundle. The link remains up when the peer device acknowledges the hello messages from the local router. When Local Management Interface (LMI) is enabled, the bundle link’s status is considered up when the Frame Relay Data Link Layer on the local router and on the peer device synchronize using LMI. The bundle link remains up as long as the LMI keepalives are successful.
The multilink interface’s status is up when the status of at least one bundle link is up. The multilink interface’s status is down when the last bundle link is no longer up. This behavior complies with the class A bandwidth requirement defined in FRF.16.

**Basic Configuration Steps**

To configure MLFR on an SSG device:

1. Create a bundle, and configure it for MLFR encapsulation. Assign the bundle to a security zone. You can also set other options, such as a bundle identification or the MTU, for the bundle.

2. (Optional) Configure Frame Relay options for the bundle. This step is required only if you need to change the default Frame Relay options for the bundle.

3. Assign bundle links to the bundle.

4. (Optional) Configure MLFR options for each bundle link in the bundle. This step is required only if you need to change the default MLFR options for the link.

5. Create one or more PVCs for the bundle, and assign each PVC a Frame Relay DLCI and IP address.

**Creating a Bundle**

On an SSG device, a bundle is accessed by a multilink interface, which you create. The name of the multilink interface must be `mlid_num`. For example, multilink interface names can be `ml1`, `ml2`, and so on.

**NOTE:** In the WebUI, `id_num` is automatically added when you create a new multilink interface. In the CLI, you must specify an `id_num` value.

By default, the encapsulation type for new multilink interfaces is MLPPP. For multilink interfaces that support Frame Relay, you must explicitly configure MLFR encapsulation on the interface.

To create a multilink interface and configure it for MLFR encapsulation:

**WebUI**

Network > Interfaces > New > Multilink IF: Select **Multi-Link Frame Relay** for the WAN Encapsulation, and then click **Apply**.

**CLI**

```
set interface bundle encapsulation mlfr-uni-nni
save
```
Configuring a Bundle Identifier

The bundle ID, as specified by FRF.16, associates a local and a remote endpoint with a specific bundle. All bundle links in the MLFR bundle must use the same bundle ID, which can be up to 80 bytes. If you are configuring more than one bundle between two devices, each bundle ID should be unique. For example, you can use network node identifiers, system serial numbers, or network addresses for bundle IDs.

If you do not configure a specific bundle ID for the multilink interface, the multilink interface name (for example, ml1 or ml2) is used.

**WebUI**

Network > Interfaces > Edit (bundle) > Basic: Enter the Multi-Link Frame Relay for the WAN Encapsulation, and then click **Apply**.

**CLI**

```
set interface bundle bundle-id name_str
save
```

Assigning the Interface to a Zone

Security zones are logical entities to which one or more interfaces are bound. Through policies that you define, you can permit traffic between security zones. To permit traffic to flow from zone to zone, you bind an interface to the zone. (For more information about creating and configuring security zones, see the Fundamentals volume in the Concepts & Examples ScreenOS Reference Guide.)

**WebUI**

Network > Interfaces > Edit (bundle): Select the zone from the Zone Name drop-down list, and then click **Apply**.

**CLI**

```
set interface bundle zone zone
save
```

Setting the Protocol MTU

You can configure the protocol MTU on each multilink interface. The default protocol MTU is 1500 bytes for serial, T1, E1, and multilink interfaces, and 4470 bytes for T3 interfaces. You can specify a value between 800 and 8192 bytes.

To configure the protocol MTU on a multilink interface:

**WebUI**

Network > Interfaces > Edit (bundle): Enter a value between 800 and 8192 in the Maximum Transfer Unit (MTU) field, and then click **OK**.

**CLI**

```
set interface bundle mtu number
save
```
Chapter 6: Configuring Frame Relay and Multilink Frame Relay

Note that in ScreenOS, the media MTU is derived from the protocol MTU. If you increase the size of the protocol MTU, you must ensure that the size of the media MTU is equal to or greater than the sum of the protocol MTU and the encapsulation overhead.

Configuring Frame Relay Options for the Bundle

Table 7 shows the default Frame Relay settings for a multilink interface configured for MLFR encapsulation. The sections that follow the table describe how to change option settings.

Table 7: Default and Alternate Values for Multilink Frame Relay Interfaces

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keepalive</td>
<td>No keepalive</td>
<td>Enable or disable</td>
</tr>
<tr>
<td>Type</td>
<td>ANSI</td>
<td>ITU</td>
</tr>
<tr>
<td>DTE full station polling interval count</td>
<td>6</td>
<td>1-255</td>
</tr>
<tr>
<td>DTE error threshold count</td>
<td>3</td>
<td>1-10</td>
</tr>
<tr>
<td>DTE monitored event count</td>
<td>4</td>
<td>1-10</td>
</tr>
<tr>
<td>DTE keepalive timer</td>
<td>10 seconds</td>
<td>5-30 seconds</td>
</tr>
<tr>
<td>Drop timeout</td>
<td>0 (disabled)</td>
<td>0-127 milliseconds</td>
</tr>
<tr>
<td>Minimum links</td>
<td>1</td>
<td>1-8</td>
</tr>
</tbody>
</table>

MLFR LMI Keepalives

On multilink interfaces configured with MLFR encapsulation, you can tune the keepalive settings.

WebUI

Network > Interfaces > Edit (bundle) > FR: Enter appropriate values for the LMI options, and then click Apply.

CLI

`set interface bundle frame-relay lmi option`

You can configure the following Frame Relay LMI keepalive options:

- DTE full status polling interval (denoted by the `n391-dte` keyword in the CLI). The DTE sends a status inquiry to the DCE at the interval specified by the DTE polling timer. The polling interval specifies the frequency at which these inquiries expect a full status report; for example, a value of 10 would specify a full status report in response to every tenth inquiry. The intermediate inquiries ask for a keepalive exchange only. The range is from 1 through 255, with a default value of 6.

- DTE error threshold (denoted by the `n392-dte` keyword in the CLI). The number of errors required to bring down the link, within the event-count specified by the DTE monitored event-count. The range is from 1 through 10, with a default value of 3.
- DTE monitored event-count (denoted by the `n393-dte` keyword in the CLI). The range is from 1 through 10, with a default value of 4.

- DTE keepalive timer (denoted by the `t391-dte` keyword in the CLI). The period at which the DTE sends out a keepalive response request to the DCE and updates status depending on the DTE error-threshold value. The range is from 5 through 30 seconds, with a default value of 10 seconds.

### Frame Relay LMI Type

By default, ScreenOS sends LMIs specified by ANSI T1.617 Annex D. To send ITU Q933 Annex A LMIs:

**WebUI**

Network > Interfaces > Edit (bundle) > FR: Select ITU, and then click Apply.

**CLI**

```
set interface bundle frame-relay lmi type itu
save
```

### Drop Timeout

By default, the drop-timeout parameter is disabled. You can configure a drop-timeout value to provide a recovery mechanism if individual links in the multilink bundle drop one or more packets. Drop timeout is not a differential delay-tolerance setting and does not limit the overall latency. We recommend setting a drop-timeout value significantly larger than the expected differential delay across the links; this way, the timeout period elapses when there is actual packet loss, not under normal jitter conditions.

To configure the drop-timeout value:

**WebUI**

Network > Interfaces > Edit (bundle) > MLFR: Enter a new value in Drop Timeout, and then click Apply.

**CLI**

```
set interface bundle drop-timeout milliseconds
```

The drop timeout has a duration from 0 through 127 milliseconds. Values less than 5 milliseconds are not recommended; a value of 0 disables the timeout.

---

**NOTE:** For multilink interfaces, if a packet or fragment encounters an error condition and is destined for a disabled bundle or link, it does not contribute to the dropped packet and frame counts in the per-bundle statistics. The packet is counted under the global error statistics and is not included in the global output-bytes and output-packet counts. This unusual accounting happens only if the error conditions are generated inside the multilink interface, not if the packet encounters errors on the wire or elsewhere in the network.
Minimum Links
You can set the minimum number of links that must be up in order for the bundle as a whole to be considered up. By default, only one link must be up for the bundle to be considered up.

To set the minimum number of links in the bundle:

WebUI
Network > Interfaces > Edit (bundle) > MLFR: Enter a new value in Minimum Links, and then click Apply.

CLI
set interface bundle minimum-links number

The number can be from 1 through 8. The maximum number of links supported in a bundle is 8. When 8 is specified, all configured links of a bundle must be up in order for the bundle to be considered up.

Assigning Bundle Links to the Bundle
Each MLFR bundle comprises aggregated physical links. Each bundle link must use MLFR encapsulation, and you must assign the physical interfaces to each multilink interface.

NOTE: You must assign bundle links to the same zone as the multilink interface.

To assign a physical interface to a multilink interface:

WebUI
Network > Interfaces > Edit (interface) > Basic: Enter the following, and then click Apply:

Member Link: (select)
Multilink Interface: (select from scrolling list)

NOTE: In the WebUI, you do not need to explicitly configure MLFR encapsulation for the physical interface. The physical interface uses the encapsulation that is configured for the multilink interface.

CLI
set interface interface encapsulation mlfr-uni-nni
set interface interface bundle bundle
save
Configuring MLFR Options for Bundle Links

For physical links configured for MLFR encapsulation, each link endpoint in a bundle initiates a request for bundle operation with its peer by transmitting an add-link message. A hello message notifies the peer endpoint that the local endpoint is up. Both ends of a link generate a hello message periodically or as configured with the hello timer. A remove-link message notifies the peer that the local end management is removing the link from bundle operation. Endpoints respond to add-link, remove-link, and hello messages by sending acknowledgement messages.

Table 8 shows the default settings for a bundle link configured for MLFR encapsulation. The sections that follow the table describe how to change option settings.

Table 8: Default and Alternate Values for Multilink Frame Relay Bundle Links

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLFR User-to-Network Interface (UNI):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acknowledge retries count</td>
<td>2</td>
<td>1-5</td>
</tr>
<tr>
<td>Acknowledge timer</td>
<td>4 milliseconds</td>
<td>1-10 milliseconds</td>
</tr>
<tr>
<td>Fragment threshold</td>
<td>MTU of physical interface</td>
<td>Multiple of 64 bytes</td>
</tr>
<tr>
<td>Hello timer interval</td>
<td>10 milliseconds</td>
<td>1-180 milliseconds</td>
</tr>
</tbody>
</table>

Acknowledge Retries

For bundle links, you can configure the number of retransmission attempts to be made for consecutive hello or remove-link messages after the expiration of the acknowledgement timer:

**WebUI**

Network > Interfaces > Edit (interface) > MLFR: Enter a value for Line Interface Protocol (LIP) Retransmission Count before Link-Down, then click Apply.

**CLI**

```
set interface interface mlfr-uni-nni acknowledge-retries number
save
```

The retransmission count can be from 1 through 5. The default is 2.
**Acknowledge Timer**

You can configure the maximum period to wait for an add-link acknowledgement, a hello acknowledgement, or a remove-link acknowledgement:

**WebUI**

Network > Interfaces > Edit (interface) > MLFR: Enter a value for Maximum Period to Wait for an Acknowledgement, and then click **Apply**.

**CLI**

```
set interface interface mlfr-uni-nni acknowledge-timer seconds
save
```

The acknowledgement timer can be from 1 through 10 milliseconds. The default is 4 milliseconds.

**Fragment Threshold**

You can configure a fragmentation threshold to set a maximum size for packet payloads transmitted across the individual links within the multilink circuit. The software splits any incoming packet that exceeds the fragmentation threshold into smaller units suitable for the circuit size; it reassembles the fragments at the other end but does not affect the output traffic stream. The threshold value affects the payload only.

**WebUI**

Network > Interfaces > Edit (interface) > MLFR: Enter a value for Bundle-link Fragmentation Threshold, and then click **Apply**.

**CLI**

```
set interface interface mlfr-uni-nni fragment-threshold number
save
```

By default, the fragment threshold is the MTU of the physical interface. (For serial, T1, and E1 bundle links, the default MTU size is 1500 bytes; for T3 bundle links, the default MTU size is 4470 bytes.) The maximum fragment size can be from 128 through 16,320 bytes. Any value you set must be a multiple of 64 bytes (N\times64).

**Hello Timer**

You can configure the rate at which hello messages are sent:

**WebUI**

Network > Interfaces > Edit (interface) > MLFR: Enter a value for LIP Hello Keepalive Interval, and then click **Apply**.

**CLI**

```
set interface interface mlfr-uni-nni hello-timer seconds
save
```

A hello message is transmitted after the specified period (in milliseconds) has elapsed. The hello timer can be from 1 through 180 milliseconds; the default is 10 milliseconds. When the hello timer expires, a link endpoint generates an add-link message.
Creating and Configuring PVCs Within the Bundle

MLFR allows creation of one or more permanent virtual circuits (PVCs) within the bundle. You create a PVC by configuring a subinterface to the multilink interface. Each subinterface maps to a PVC, which is identified by a data-link connection identifier (DLCI). Note that each PVC can be associated with a separate security zone; the security zone for each PVC can be different from the security zone assigned to the multilink interface.

To configure an MLFR PVC, first create a subinterface for the multilink interface and assign it to a security zone. Then assign a Frame Relay DLCI and IP address to the subinterface.

**WebUI**

Network > Interface > New > WAN Sub-IF: Enter the following, and then click OK:

- Interface Name: bundle (select)
- Zone Name: (select)
- Frame Relay DLCI: (enter id_num)
- IP Address/Netmask: (enter ip_addr)

**CLI**

set interface subinterface zone zone
set interface subinterface frame-relay dlci id_num
set interface subinterface ip ip_addr
save

The subinterface name consists of the multilink interface name and a subinterface number. For example, if the multilink interface name is ml1, its subinterfaces can be ml1.1 and ml1.2.

**NOTE:** In the WebUI, the subinterface number is automatically added when you select the multilink interface name. In the CLI, you must enter both the multilink interface name and the subinterface number.

You can only specify one DLCI for each subinterface. The DLCI identifier is a value from 16 through 1022. Numbers 1 through 15 are reserved. A point-to-point interface can have one DLCI.

You can configure the subinterface to use inverse Frame Relay ARP; this is described in the next section.

You can also configure the subinterface for management functions such as a manage IP address, service options, and other features. (For more information about configuring manage IP and service options on an interface, see the Fundamentals volume in the Concepts & Examples ScreenOS Reference Guide.)
Inverse Frame Relay Address Resolution Protocol

MLFR subinterfaces support inverse Frame Relay Address Resolution Protocol (ARP), as described in RFC 2390, Inverse Address Resolution Protocol. When inverse Frame Relay ARP is enabled, the routing platform responds to received inverse Frame Relay ARP requests by providing IP address information to the requesting routing platform on the other end of the Frame Relay PVC.

The routing platform does not initiate inverse Frame Relay ARP requests.

By default, inverse Frame Relay ARP is disabled. To configure a routing platform to respond to inverse Frame Relay ARP requests:

**WebUI**

Network > Interface > Edit > WAN Sub-IF: Select Frame Relay Inverse ARP, and then click **Apply**.

**CLI**

```bash
set interface subinterface frame-relay inverse-arp
save
```

Example Configurations

This section contains examples of configuring Frame Relay and MLFR on SSG devices.

**Frame Relay Configuration Examples**

This section contains examples of configuring Frame Relay on SSG devices.

**SSG Device**

The following shows the CLI commands to configure Frame Relay on an SSG device. This example does not show the configuration of the physical WAN interface to connect the device to a Frame Relay network; see the serial, T1, E1, or T3 configuration chapters for examples on how to configure the physical WAN interface.

```bash
set interface serial1/1 zone Untrust
set interface serial1/1 encap frame-relay
set interface serial1/1 frame-relay lmi type itu
set interface serial1/1.1 frame-relay dli 1002
set interface serial1/1.1 ip 1.1.1.1/24
```

**J2300 and SSG 520 Peer Devices**

The following shows the Frame Relay configuration for a Juniper J2300 Services Router (configured as a DCE) and an SSG 520 device.

J2300:

```bash
t1-0/0/2 {
  dce;
  mtu 1390;
  clocking internal;
  encapsulation frame-relay;
```
Example Configurations

### Unnumbered Interface Configuration Examples

The following shows the unnumbered interface configurations for single link and multiple link frame relay.

This example shows the single link frame relay configuration:

```plaintext
ten options {
timeslots 1;
buidout 0-132;
byte-encoding nx64;
line-encoding b8zs;
framing esf;
fcs 16;
}    
unit 0 {
    description Ascend-P130;
dlci 100;
    family inet {
        address 206.204.42.125/24;
    }
}

SSG 520:

set interface serial4/1 zone Untrust
set interface serial4/1.1 zone Untrust
set interface serial4/1 encap frame-relay
set interface serial4/1.1 ip 206.204.42.124/24
set interface serial4/1.1 route
set interface serial4/1.1 ip manageable
set interface serial4/1.1 frame-relay dcli 100
set interface serial4/1 clocking external
set interface serial4/1 t1-options timeslots 1
```

To configure multilink frame relay serial interface to be unnumbered and to borrow the ethernet0/0 interface:

```plaintext
set interface ml2.1 zone Untrust
set interface ml2.1 zone Untrust
set interface ml2 encap mlfr-uni-nmi
set interface serial5/0 bundle ml2
set interface ml2.1 frame-relay dcli 200
set interface ml2.1 unnumbered interface ethernet0/0
```
**Multilink Frame Relay Configuration Example**

The following shows the CLI commands to configure MLFR on two peer SSG devices. This example does not show the configuration of the physical WAN interfaces that connect the two devices; see the serial, T1, E1, and T3 configuration chapters for examples on how to configure physical WAN interfaces.

Peer A:

```plaintext
set interface "ml65535" zone "Trust"
set interface "ml65535.1" zone "Trust"
set interface "ml65535" encap mlfr-uni-nni
set interface ml65535.1 ip 22.22.22.2/24
set interface serial1/1 bundle ml65535
set interface "ml65535.1" frame-relay dlci 221
```

Peer B:

```plaintext
set interface "ml65535" zone "Trust"
set interface "ml65535.1" zone "Trust"
set interface "ml65535" encap mlfr-uni-nni
set interface ml65535.1 ip 22.22.22.1/24
set interface serial1/1 bundle ml65535
set interface "ml65535.1" frame-relay dlci 221
```

Create a multilink interface and assign it to a zone.  
Create a PVC (subinterface) and assign it to the same zone.  
Configure the multilink interface for MLFR encapsulation.  
Configure an IP address for the subinterface.  
Add the physical interface to the multilink interface.  
Assign a DLCI to the subinterface.
Overview of Point-to-Point Protocol

Point-to-Point Protocol (PPP) encapsulation allows different Network Layer protocols to be multiplexed simultaneously over commonly used physical links. PPP uses High-Level Data Link Control (HDLC) for packet encapsulation.

To establish a PPP connection, each end of a PPP link must first configure the link by exchanging Link Control Protocol (LCP) packets. LCP is used to establish, configure, and test data-link options. These options include encapsulation format options, authentication of the peer on the link, handling of varying limits on sizes of packets, detecting a looped-back link and other common misconfiguration errors; determining when a link is functioning properly or failing; and terminating the link.
PPP allows for authentication during link establishment to permit or deny connection to a device. This authentication can be performed using either Password Authentication Protocol (PAP) or Challenge-Handshake Authentication Protocol (CHAP), as documented in RFC 1334, *PPP Authentication Protocols*. These authentication protocols are intended for use primarily by hosts and routers that connect to a network server via switched circuits or dial-up lines but can also be used with dedicated lines.

### Configuring PPP Encapsulation on a Single Physical Link

This section describes how to configure PPP on a physical link that is not part of an MLPPP bundle. To configure multilink PPP (MLPPP), see “Configuring Multilink Point-to-Point Protocol” on page 65.

#### Basic Configuration Steps

To configure PPP on a single physical link on the SSG device:

1. Configure PPP encapsulation on the physical link, and assign the link to a security zone. Configure the IP address on the physical link.

2. (Optional) Configure PPP options for the physical link. This step is required only if you need to change the default PPP options for the link.

3. Configure a PPP access profile, and bind it to the interface. This step is required even if no authentication is used on the PPP data link.

4. (Optional) If CHAP or PAP authentication is used, configure the peer’s username and password in the local database of the SSG device.

### Configuring PPP Encapsulation on a Physical Link

On the SSG device, you configure a physical link by configuring the interface that represents the link. There is no default encapsulation type for WAN links on the SSG device; therefore, you must explicitly configure the PPP encapsulation type for these interfaces.

To configure PPP encapsulation on a WAN interface:

**WebUI**

Network > Interfaces > Edit *(interface)*: select **PPP** for the WAN Encapsulation, and then click **Apply**.

**NOTE:** Make sure that the Main Link option is selected in the Basic properties page for the interface.

**CLI**

```
set interface *interface* encapsulation ppp
save
```
Assigning the Interface to a Zone

Security zones are logical entities to which one or more interfaces are bound. Through policies that you define, you can permit traffic between security zones. To permit traffic to flow from zone to zone, you bind an interface to the zone. (For more information about creating and configuring security zones, see the Fundamentals volume in the Concepts & Examples ScreenOS Reference Guide.)

**WebUI**

Network > Interfaces > Edit (interface): Select the zone from the Zone Name drop-down list, and then click Apply.

**CLI**

```
set interface interface zone zone
save
```

You can also configure the interface for management functions, such as a manage IP address, service options, and other features. (For more information about configuring manage IP and service options on an interface, see the Fundamentals volume in the Concepts & Examples ScreenOS Reference Guide.)

Configuring a Static IP Address for the Interface

By default, the interface uses an IP address dynamically assigned by a server at the other end of the PPP data link. Alternatively, you can explicitly assign a static IP address to the interface.

**NOTE:** If you configure a static IP address for the interface, you must specify the static IP option in the access profile that you bind to the interface. See “Configuring a PPP Access Profile” on page 61.

To assign an IP address to the interface:

**WebUI**

Network > Interfaces > Edit (interface): Enter an IP address and netmask in the IP Address/Netmask fields, and then click Apply.

**CLI**

```
set interface interface ip ip_addr/mask
save
```

Configuring PPP Options for the Link

Table 9 shows the default PPP settings for a WAN interface configured for PPP encapsulation. The sections that follow the table describe how to change option settings.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keepalive</td>
<td>Enabled</td>
<td>Disable</td>
</tr>
<tr>
<td>Keepalive interval</td>
<td>10 seconds</td>
<td>1-52767 seconds</td>
</tr>
<tr>
<td>Keepalive down count</td>
<td>3</td>
<td>1-225</td>
</tr>
</tbody>
</table>
PPP Keepalives

By default, physical interfaces configured with PPP encapsulation send keepalive packets at 10-second intervals.

To disable the sending of keepalives on a physical interface:

**WebUI**

Network > Interfaces > Edit (*interface*) > PPP: Deselect the **Keepalive** checkbox, and then click **Apply**.

**CLI**

unset interface *interface* keepalives
save

Keepalive Interval

The default interval at which the interface sends keepalive packets on a PPP link is 10 seconds. To configure the interface to send keepalive packets at a different interval:

**WebUI**

Network > Interfaces > Edit (*interface*) > PPP: Enter the number of seconds in **Keepalive Interval**, and then click **Apply**.

**CLI**

set interface *interface* keepalives seconds
save

Keepalive Down Count

The receipt of keepalive packets by a destination determines whether the link is considered down or up. By default, if a destination fails to receive three successive keepalive packets, the link is considered down. A down link is subsequently considered up whenever a keepalive packet is received.

To change the number of counts by which the destination determines a link to be down:

**WebUI**

Network > Interfaces > Edit (*interface*) > PPP: Enter the number of counts for **Down Counter**, and then click **Apply**.

**CLI**

set interface *interface* keepalives down-count *number*
save
**Configuring a PPP Access Profile**

For an interface with PPP encapsulation, you must configure a PPP access profile and bind it to the interface. You create an access profile with a user-defined name that is unique on the SSG device. You can bind the same access profile to more than one interface, but only one profile can be assigned to an interface.

A PPP access profile includes the following information:

- Whether authentication is used to permit or deny connection to devices during Link Control Protocol (LCP) link setup. If authentication is specified, you can configure options for the selected authentication method.

- Whether the interface uses a static IP address that you have already configured. If the interface uses an IP address dynamically assigned by a server, you can specify the netmask for the IP address.

During LCP link setup, authentication can be used to permit or deny connection to devices; if authentication fails, the PPP link is terminated. By default, authentication is disabled on interfaces that are configured for PPP encryption. If you do not explicitly enable authentication on the interface, the interface makes no authentication requests and denies all incoming authentication challenges.

You can configure interfaces to support one or both of the following authentication protocols:

- **Password Authentication Protocol (PAP)**, as defined in RFC 1334, *PPP Authentication Protocols*

- **Challenge Handshake Authentication Protocol (CHAP)**, as defined in RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

To create a PPP access profile and bind it to an interface:

**WebUI**

1. Network > PPP > PPP Profile > New: Enter `profile_name` in the **PPP Profile** field and enter other options, and then click **OK**.

2. Network > Interfaces > Edit (interface): Select `profile_name` in the **Binding a PPP Profile** scrolling list, and then click **Apply**.

**CLI**

```
set ppp profile profile_name ...
set interface interface ppp profile profile_name
save
```

Table 10 shows the default settings for PPP access profile options. The sections that follow the table describe how to change option settings.
Table 10: Default and Alternate Values for Point-to-Point Protocol Access Profile Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication method</td>
<td>None</td>
<td>Use CHAP only&lt;br&gt;Use PAP only&lt;br&gt;Use CHAP first; but, if negotiations with the peer are unsuccessful, use PAP</td>
</tr>
<tr>
<td>Static IP</td>
<td>Disabled (IP address obtained from server)</td>
<td>Enable</td>
</tr>
<tr>
<td>Netmask</td>
<td>255.255.255.255</td>
<td>Valid subnet mask</td>
</tr>
<tr>
<td>Passive mode</td>
<td>Disabled (always challenge peer and respond to challenges from peer)</td>
<td>Enable (do not challenge peer and only respond when challenged)</td>
</tr>
<tr>
<td>CHAP local name</td>
<td>Device's hostname</td>
<td>Valid hostname</td>
</tr>
<tr>
<td>Password</td>
<td>None</td>
<td>Valid password string</td>
</tr>
</tbody>
</table>

**Authentication Method**

During LCP link setup, authentication can be used to permit or deny connection to devices; if authentication fails, the PPP link is terminated.

**WebUI**

Network > PPP > PPP Profile > Edit (profile_name): Select Any, CHAP, PAP, or none, and then click OK.

**CLI**

```
set ppp profile profile_name { chap | pap | any | none }
save
```

**Static IP Address**

By default, the interface uses an IP address dynamically assigned by a server at the other end of the PPP data link. You can choose to have the interface use an IP address that you have manually configured for the interface:

**NOTE:** If you specify use of a static IP address in an access profile, you can only bind the profile to an interface that has an explicitly configured IP address. Conversely, if an interface has a static IP address, the access profile you bind to the interface must specify the static IP option.

**WebUI**

Network > PPP > PPP Profile > Edit (profile_name): Select Static IP, and then click OK.

**CLI**

```
set ppp profile profile_name static-ip
save
```
Netmask

By default, if the IP address for the interface is dynamically assigned by a server, the netmask for the interface is /32 (255.255.255.255). You can specify a different netmask value for the interface:

**WebUI**

Network > PPP > PPP Profile > Edit (profile_name): Enter a new mask value in the Netmask field, and then click OK.

**CLI**

```
set ppp profile profile_name netmask mask
save
```

Unnumbered Interfaces

An unnumbered interface is not assigned an IP address but borrows an IP address from other interfaces. In this way, address space is conserved. If an unnumbered interface is pointing to an interface that is not functioning (Interface status UP or Protocol UP is not displayed) then the unnumbered interface does not work. We recommend that unnumbered interfaces point to a loopback interface since loopback interfaces do not fail.

To configure IP unnumbered interface:

**WebUI**

Network > Interface > Edit (WAN interface): Select Unnumbered option, select the source interface, and then click Apply.

**CLI**

```
set interface interface ip unnumbered interface src interface
save
```

Passive Mode

By default, when PPP authentication is enabled on the interface, the interface always challenges its peer and responds to challenges from its peer. You can configure the interface not to challenge its peer and to respond only when challenged (this behavior is called passive mode).

To enable passive mode:

**WebUI**

Network > PPP > PPP Profile > Edit (profile_name): Select Passive, and then click OK.

**CLI**

```
set ppp profile profile_name passive
save
```
CHAP Local Name

By default, when Challenge Handshake Authentication Protocol (CHAP) authentication is enabled on an interface, the interface uses the SSG device’s system hostname as the name sent in challenge and response packets. You can configure a different name for the interface to use in challenge and response packets.

WebUI

Network > PPP > PPP Profile > Edit (profile_name): Enter a name in the Local Name field, and then click OK.

CLI

set ppp profile profile_name chap-local-name name
save

Password

The password is used to authenticate the PPP client on the interface with its peer.

WebUI

Network > PPP > PPP Profile > Edit (profile_name): Enter a string in Password, and then click OK.

CLI

set ppp profile profile_name password string
save

Configuring WAN Users in the Local Database

If CHAP or PAP is used on the PPP link, the peer device sends its username and password to the SSG device for authentication. The SSG device compares the received username and password with WAN user-type entries configured in its local database. Only a peer whose username and password match an entry in the local database is allowed to connect to the SSG device to send or receive data.

To configure a WAN user:

WebUI

Objects > Users > Local > New: Enter the following, and then click OK:

WAN User: (select)
User Name: name_str
User Password: pswd_str
Confirm Password: pswd_str

CLI

set user name_str password pswd_str
set user name_str type wan
save
Configuring Multilink Point-to-Point Protocol

Multilink Point-to-Point Protocol (MLPPP) provides a cost-effective way to increase bandwidth for applications by enabling multiple physical links to be aggregated into a bundle. For example, if an application requires more bandwidth than is available on a single T1 line, one option is to invest in a T3 line. However, T3 lines can be expensive and are not always available. MLPPP allows you to bundle multiple T1 lines to provide bandwidth equivalent to a T3 line.

MLPPP can also provide fault-tolerance. When a single bundle link in the bundle fails, the bundle continues to support PPP service by transmitting across the remaining bundle links. MLPPP also provides load balancing across the bundle links within a bundle. If a bundle link chosen for transmission is busy transmitting a long packet, another link can be used.

On the SSG device, you configure and access a bundle through a virtual interface called a multilink interface. Each physical link in the bundle is referred to as a bundle link, which is configured and access through the physical interface.

Basic Configuration Steps

To configure MLPPP on an SSG device:

1. Create a bundle, and configure it for MLPPP encapsulation. Assign the bundle to a security zone. You can also set other options, such as the IP address or the MTU, for the bundle.

2. (Optional) Configure MLPPP options for the bundle. This step is required only if you need to change the default MLPPP options for the bundle.

3. Configure a PPP access profile, and bind it to the interface. This step is required even if no authentication is used on the PPP data link.

4. (Optional) If CHAP or PAP authentication is used, configure the peer’s username and password in the local database of the SSG device.

5. Assign bundle links to the bundle.

6. (Optional) Configure PPP options for each bundle link in the bundle. This step is required only if you need to change the default PPP options for the link.

Creating a Bundle

On an SSG device, a bundle is accessed by a multilink interface, which you create. The name of the multilink interface must be mlid_num. For example, multilink interface names can be ml1, ml2, and so on.

NOTE: In the WebUI, id_num is automatically added when you create a new multilink interface. In the CLI, you must specify an id_num value.

By default, the encapsulation type for new multilink interfaces is MLPPP. (For multilink interfaces that support Frame Relay, you must explicitly configure MLFR encapsulation on the interface.)
To create a multilink interface and configure it for MLPPP encapsulation:

**WebUI**
Network > Interfaces > New > Multilink IF: Make sure that Multi-Link PPP is selected for the WAN Encapsulation, and then click Apply.

**CLI**
```
set interface bundle encapsulation mlppp
save
```

**Assigning the Interface to a Zone**
Security zones are logical entities to which one or more interfaces are bound. Through policies that you define, you can permit traffic between security zones. With the routes that you define, you specify the interfaces that traffic from one zone to another must use. To permit traffic to flow from zone to zone, you bind an interface to the zone. (For more information about creating and configuring security zones, see the Fundamentals volume in the Concepts & Examples ScreenOS Reference Guide.)

**WebUI**
Network > Interfaces > Edit (bundle): Select the zone from the Zone Name drop-down list, and then click Apply.

**CLI**
```
set interface bundle zone zone
save
```

**Assigning an IP Address to the Interface**
By default, the interface uses an IP address dynamically assigned by a server at the other end of the PPP data link. Alternatively, you can explicitly assign a static IP address to the interface.

**NOTE:** If you configure a static IP address for the interface, you must specify the static IP option in the access profile that you bind to the interface. See “Configuring a PPP Access Profile” on page 61.

To assign an IP address to the interface:

**WebUI**
Network > Interfaces > Edit (interface): Enter an IP address and a netmask in the IP Address/Netmask fields, and then click Apply.

**CLI**
```
set interface interface ip ip_addr/mask
save
```
Setting the Protocol MTU

You can configure the protocol MTU on each multilink interface. The default protocol MTU is 1500 bytes for serial, T1, E1, and multilink interfaces, and 4470 bytes for T3 interfaces. You can specify a value between 800 and 8192 bytes.

To configure the protocol MTU on a multilink interface:

**WebUI**

Network > Interfaces > Edit (bundle): Enter the number of bytes in the Maximum Transfer Unit (MTU) field, and then click OK.

**CLI**

```
set interface bundle mtu bytes
save
```

Note that in ScreenOS, the media MTU is derived from the protocol MTU. If you increase the size of the protocol MTU, you must ensure that the size of the media MTU is equal to or greater than the sum of the protocol MTU and the encapsulation overhead.

Configuring MLPPP Options for the Bundle

Table 11 shows the default settings for a multilink interface configured for MLPPP encapsulation. The sections that follow the table describe how to change option settings.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop timeout</td>
<td>0 (disabled)</td>
<td>0-127 milliseconds</td>
</tr>
<tr>
<td>Minimum links</td>
<td>1</td>
<td>1-8</td>
</tr>
<tr>
<td>Maximum received reconstructed unit</td>
<td>1500 bytes</td>
<td>1500-4500 bytes</td>
</tr>
<tr>
<td>Fragmentation threshold</td>
<td>0 (disabled)</td>
<td>128-16320 bytes</td>
</tr>
<tr>
<td>Sequence header format</td>
<td>Disabled (24 bits)</td>
<td>Enabled (12 bits)</td>
</tr>
</tbody>
</table>

Drop Timeout

By default, the drop-timeout parameter is disabled. You can configure a drop-timeout value to provide a recovery mechanism if individual links in the multilink bundle drop one or more packets. Drop timeout is not a differential delay tolerance setting and does not limit the overall latency. We recommend setting a drop-timeout value significantly larger than the expected differential delay across the links; this way, the timeout period elapses when there is actual packet loss and not under normal jitter conditions.

To configure the drop-timeout value:

**WebUI**

Network > Interfaces > Edit (bundle) > MLPPP: Enter the number of milliseconds in Drop Timeout, and then click Apply.
**CLI**

```bash
set interface bundle drop-timeout milliseconds
save
```

The drop timeout has a duration from 0 through 127 milliseconds. Values less than 5 milliseconds are not recommended; a value of 0 disables the timeout.

---

**NOTE:** For multilink interfaces, if a packet or fragment encounters an error condition and is destined for a disabled bundle or link, it does not contribute to the dropped packet and frame counts in the per-bundle statistics. The packet is counted under the global error statistics and is not included in the global output bytes and output packet counts. This unusual accounting happens only if the error conditions are generated inside the multilink interface and not if the packet encounters errors elsewhere in the network.

---

**Minimum Links**

You can set the minimum number of links that must be up for the bundle in order for it to be considered up. By default, only one link must be up for the bundle to be considered up.

To set the minimum number of links in the bundle:

**WebUI**

Network > Interfaces > Edit (bundle) > MLPPP: Enter a new number in Minimum Links, and then click Apply.

**CLI**

```bash
set interface bundle minimum-links number
save
```

The number for minimum-links can be from 1 through 8. The maximum number of links supported in a bundle is 8. If you specify 8, all configured links of a bundle must be up for the bundle to be considered up.

---

**Maximum Received Reconstructed Unit**

The maximum received reconstructed unit (MRRU) is similar to a maximum transmission unit (MTU) but applies only to multilink bundles; it is the maximum packet size that the multilink interface can process. By default, the MRRU is set to 1500 bytes; you can configure a different MRRU value if the peer equipment allows it. The MRRU includes the original payload plus the 2-byte PPP header, but it does not include the additional MLPPP header applied while the individual multilink packets are traversing separate links in the bundle.

To configure a different MRRU:

**WebUI**

Network > Interfaces > Edit (bundle) > MLPPP: Enter the number of bytes in Maximum Received Reconstructed Unit, and then click Apply.

**CLI**

```bash
set interface bundle mrru bytes
save
```
Fragment Threshold

By default, the fragment-threshold parameter is disabled. For interfaces with MLPPP encapsulation, you can configure a fragment threshold to set a maximum size for packet payloads transmitted across the individual links within the multilink circuit. The software splits any incoming packet that exceeds the fragment threshold into smaller units suitable for the circuit size; it reassembles the fragments at the other end but does not affect the output traffic stream. The threshold value affects the payload only; it does not affect the MLPPP header.

**NOTE:** To ensure proper load balancing for MLPPP interfaces, do not set both fragment-threshold and short-sequence options in the configuration.

For MLPPP interfaces, if the MTU of links in a bundle is less than the bundle MTU plus encapsulation overhead, then fragmentation is automatically enabled. You should avoid this situation for MLPPP interfaces on which short-sequencing is enabled.

To configure a fragmentation threshold value:

**WebUI**

Network > Interfaces > Edit (bundle) > MLPPP: Enter the number of bytes in Fragment Threshold, and then click Apply.

**CLI**

```
set interface bundle fragment-threshold bytes
save
```

Sequence Header Format

For MLPPP, the sequence header format is set to 24 bits by default. You can configure an alternative value of 12 bits, but 24 bits is considered the more robust value for most networks.

To configure a 12-bit sequence header format:

**WebUI**

Network > Interfaces > Edit (bundle) > MLPPP: Select Short-Sequence MLPPP Number, and then click Apply.

**CLI**

```
set interface bundle short-sequence
save
```

Configuring and Binding PPP Access Profiles

See “Configuring a PPP Access Profile” on page 61.

Configuring WAN Users in the Local Database

See “Configuring WAN Users in the Local Database” on page 64.
Assigning Bundle Links to the Bundle

An MLPPP bundle comprises aggregated physical links. Each bundle link must use MLPPP encapsulation, and you must add the physical interface that represents a bundle link to a multilink interface.

Before you assign physical interfaces to the multilink interface, ensure the following:

- The multilink interface is created and configured for MLPPP encapsulation.
- A PPP access profile is configured and bound to the multilink interface.

To assign a physical interface to a multilink interface:

**WebUI**

Network > Interfaces > Edit (interface) > Basic: Enter the following, and then click Apply:

- Member Link: (select)
- Multilink Interface: (select from scrolling list)

**NOTE:** In the WebUI, you do not need to explicitly configure MLPPP encapsulation for the physical interface. The physical interface uses the encapsulation that is configured for the specified multilink interface.

**CLI**

```
set interface interface encapsulation mllppp
set interface interface bundle bundle
save
```

Configuring PPP Options for Bundle Links

See “Configuring PPP Options for the Link” on page 59.

**NOTE:** You must assign bundle links to the same zone as the multilink interface.
Example Configurations

This section contains examples of configuring PPP and MLPPP on SSG devices.

PPP Configuration Examples

This section contains examples of configuring Frame Relay on peer devices.

SSG Peer Devices

The following shows the CLI commands to configure PPP and MLPPP on two peer SSG devices. This example does not show the configuration of the physical WAN interfaces that connect the two devices; see the serial, T1, E1, and T3 configuration chapters for examples on how to configure physical WAN interfaces.

Peer A:

```
set ppp profile 1
set ppp profile 1 static-ip
set interface serial1/1 zone Untrust
set int s1/1 encap ppp
set int s1/1 ip 1.1.1.1/24
set int s1/1 ppp profile 1
```

Peer B:

```
set ppp profile 1
set ppp profile 1 static-ip
set interface serial1/1 zone Untrust
set int s1/1 encap ppp
set int s1/1 ip 1.1.1.2/24
set int s1/1 ppp profile 1
```

J2300 and SSG 520 Peer Devices

The following shows the PPP configuration for a Juniper J2300 Services Router and an SSG 520 device.

J2300:

```
t1-0/0/3 {
  clocking internal;
  encapsulation ppp;
  t1-options {
    timeslots 1-24;
    buildout 0-132;
    byte-encoding nx64;
    line-encoding b8zs;
    framing esf;
    fcs 16;
  }
  unit 0 {
    family inet {
      address 10.10.10.1/24;
    }
  }
}
```

SSG 520:

```
set ppp profile "J2300-hub"
```
Example Configurations

Unnumbered Interfaces Configuration Examples

To configure ppp serial interface to be unnumbered and to borrow the loopback.1 interface:

set ppp profile test
set ppp profiles test static-ip
set interface serial5/0 encap ppp
set interface serial5/0 ip unnumbered interface loopback.1
set interface serial5/0 ppp profile test

To configure multilink ppp serial interface to be unnumbered and to borrow the loopback.1 interface:

set ppp profile test
set ppp profile test static-ip
set interface ml2 encap mlppp
set interface serial5/0 bundle ml2
set interface ml2 unnumbered interface loopback.1
set interface ml2 ppp profile test

MLPPP Configuration Example

Peer A:

set int s6/0 serial-options clocking-mode internal
set int s6/0 serial-options clock-rate 8.0
set int s6/1 serial-options clocking-mode internal
set int s6/1 serial-options clock-rate 8.0
set int ML100 zone untrust
set int ML100 ip 192.168.1.1/24
set ppp profile test
set ppp profile static-ip
set int ML100 encapsulation MLPPP
set int ML100 PPP profile test
set int ser6/0 bundle ML100
set int ser6/1 bundle ML100

Configure the physical WAN interfaces.

Create a multilink interface and assign it an IP address.

Configure a PPP access profile.

Configure the multilink interface for MLPPP encapsulation.

Add the physical interface to the multilink interface.

Peer B:

set int s6/0 serial-options clocking-mode DCE
set int s6/0 serial-options clock-rate DCE
set in ML100 zone untrust

Configure the physical WAN interfaces.

Create a multilink interface and assign it an IP address.
set in ML100 ip 192.168.1.2/24
set ppp profile test
set ppp profile static-ip
set in ML100 encapsulation MLPPP
set in ML100 PPP profile test
set in ser6/0 bundle ML100
set in ser6/1 bundle ML100

Configure a PPP access profile.

Configure the multilink interface for MLPPP encapsulation.

Add the physical interface to the multilink interface.
Chapter 8
Configuring Cisco High-Level Data Link Control

Wide area network (WAN) interfaces on Secure Services Gateway (SSG) devices support Cisco High-Level Data Link Control (Cisco HDLC), the default protocol for serial interfaces on Cisco routers and bridges. Cisco HDLC is used to encapsulate LAN protocol packets for transfer over WAN links.

This chapter describes how to configure Cisco HDLC on an SSG device. It contains the following sections:

- “Overview of Cisco High-Level Data Link Control” on page 75
- “Configuring Cisco HDLC Encapsulation” on page 76
- “Example Configurations” on page 79

Overview of Cisco High-Level Data Link Control

Cisco High-Level Data Link Control (Cisco HDLC) is an extension to the standard HDLC developed by the International Organization for Standardization (ISO). HDLC is a bit-oriented, synchronous, Data Link Layer protocol that specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

Cisco HDLC monitors line status on a serial interface by exchanging keepalive messages with peer network devices. A keepalive message is a signal from one endpoint to the other that the first endpoint is still active. Keepalives are used to identify inactive or failed connections.

Keepalives can also allow routers to discover IP addresses of neighbors by exchanging Serial Line Address Resolution Protocol (SLARP) address-request and address-response messages with peer network devices. An SSG device does not support SLARP exchanges with peers.
Configuring Cisco HDLC Encapsulation

This section describes how to configure Cisco HDLC on a WAN interface.

Basic Configuration Steps

To configure Cisco HDLC on a WAN link on an SSG device:

1. Configure Cisco HDLC encapsulation on the WAN link, and assign the link to a security zone. If necessary, configure the IP address on the physical link.

2. (Optional) Configure Cisco HDLC options for the physical link. This step is required only if you need to change the default HDLC options for the link.

Configuring Cisco HDLC Encapsulation on a Physical Link

On an SSG device, you configure a physical link by configuring the interface that represents the link. There is no default encapsulation type for WAN links on the SSG device; therefore, you must explicitly configure the Cisco HDLC encapsulation type for these interfaces.

To configure Cisco HDLC encapsulation on a WAN interface:

WebUI

Network > Interfaces > Edit (interface): Select Cisco HDLC for the WAN Encapsulation, and then click Apply.

CLI

set interface interface encapsulation cisco-hdlc
save

Assigning the Interface to a Zone

Security zones are logical entities to which one or more interfaces are bound. Through policies that you define, you can permit traffic between security zones. To permit traffic to flow from zone to zone, you bind an interface to the zone. (For more information about creating and configuring security zones, see the Fundamentals volume in the Concepts & Examples ScreenOS Reference Guide.)

WebUI

Network > Interfaces > Edit (interface): Select the zone from the Zone Name drop-down list, and then click Apply.

CLI

set interface interface zone zone
save

You can also configure the interface for management functions, such as a manage IP address, service options, and other features. (For more information about configuring manage IP and service options on an interface, see the Fundamentals volume in the Concepts & Examples ScreenOS Reference Guide.)
Chapter 8: Configuring Cisco High-Level Data Link Control

Configuring a Static IP Address for the Interface

Layer 3 interfaces must have an IP address and a netmask, which can be either dynamically assigned by a server at the other end of the data link or explicitly configured by an administrator as a static address.

To assign a static IP address to the interface:

**WebUI**

Network > Interfaces > Edit (interface): Enter an IP address and a netmask in the IP Address/Netmask fields, and then click Apply.

**CLI**

```
set interface interface ip ip_addr/mask
save
```

Configuring HDLC Options for the Link

Table 12 shows the default HDLC settings for a WAN interface configured for Cisco HDLC encapsulation. The sections that follow the table describe how to change option settings.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Alternate Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keepalive</td>
<td>Enabled</td>
<td>Disable</td>
</tr>
<tr>
<td>Keepalive interval</td>
<td>10 seconds</td>
<td>1-32767 seconds</td>
</tr>
<tr>
<td>Down count</td>
<td>3</td>
<td>1-225</td>
</tr>
<tr>
<td>Up count</td>
<td>0 (disabled)</td>
<td>1-255</td>
</tr>
</tbody>
</table>

**Keepalives**

By default, physical interfaces configured with Cisco HDLC encapsulation send keepalive packets at 10-second intervals.

To disable the sending of keepalives on a physical interface:

**WebUI**

Network > Interfaces > Edit (interface) > Cisco HDLC: Deselect the Keepalive checkbox, and then click Apply.

**CLI**

```
unset interface interface keepalives
save
```
Keepalive Interval
The default interval at which the interface sends keepalive packets on a Cisco HDLC link is 10 seconds. To configure the interface to send keepalive packets at a different interval:

**WebUI**
Network > Interfaces > Edit (interface) > Cisco HDLC: Enter the number of seconds in Keepalive Interval, and then click Apply.

**CLI**
```
set interface interface keepalives seconds
save
```

Keepalive Down and Up Counts
The receipt of keepalive packets by a destination determines whether the link is considered down or up. By default, if a destination fails to receive three successive keepalive packets, the link is considered down. A down link is subsequently considered up when a single keepalive packet is received.

To change the counts by which the destination determines a link to be down or up:

**WebUI**
Network > Interfaces > Edit (interface) > Cisco HDLC: Enter the number of counts for Down Counter or Up Counter, and then click Apply.

**CLI**
```
set interface interface keepalives down-count number
set interface interface keepalives up-count number
save
```

Unnumbered Interfaces
An unnumbered interface is not assigned an IP address but borrows an IP address from other interfaces. In this way, address space is conserved. If an unnumbered interface is pointing to an interface that is not functioning (Interface status UP or Protocol UP is not displayed) then the unnumbered interface does not work. We recommend that unnumbered interfaces point to a loopback interface since loopback interfaces do not fail.

To configure IP unnumbered interface:

**WebUI**
Network > Interface > Edit (WAN interface): Select Unnumbered option, select the source interface, and then click Apply.

**CLI**
```
set interface interface ip unnumbered interface src interface
save
```
Example Configurations

The following shows the CLI commands to configure Cisco HDLC on two peer SSG devices.

Peer A:

set interface serial1/1 zone Untrust
set interface serial1/1 encap cisco-hdlc
set interface serial1/1 ip 2.1.1.1/24

Peer B:

set interface serial1/1 zone Untrust
set interface serial1/1 encap cisco-hdlc
set interface serial1/1 ip 2.1.1.2/24

To configure CHCLC serial interface to be unnumbered and to borrow the loopback.1 interface:

set interface serial5/0 zone Untrust
set interface serial5/0 encap cisco-hdlc
set interface serial5/0 ip unnumbered interface loopback.1
Chapter 9
New and Modified CLI Commands

This chapter introduces the following new commands:

- “ppp” on page 101

In addition, it describes changes to the following commands:

- “counter” on page 81
- “interface” on page 82
- “user” on page 102

The following command descriptions focus only on the new elements added in this release. For more information about other command elements, refer to the CLI Reference Guide for ScreenOS 5.1.0.

**counter**

Traffic counters provide processing information, which you can use to monitor traffic flow. For WAN interfaces, hardware counters monitor hardware performance and track the number of packets containing errors.

**Syntax**

```
get
get counter statistics interface interface extensive
```

**Keywords and Variables**

**Variable Parameter**

```
ge t
get counter statistics interface interface extensive
```

`interface` The name of the interface. All WAN interfaces on an SSG device, including serial, T1/E1, and T3, are named serialn1/n2, where n1 is the slot number in the SSG chassis that is occupied by the Physical Interface Module (PIM), and n2 is the physical port on the PIM.
**interface**

Interfaces are physical or logical connections that handle network, virtual private network (VPN), High Availability (HA), and administrative traffic. Use the `interface` commands to define or display interface settings for the WAN interfaces on the SSG device.

All WAN interfaces on SSG devices, including serial, T1/E1, and T3 interfaces, are named `serialn1/n2`, where `n1` is the slot number in the SSG chassis that is occupied by the PIM, and `n2` is the physical port on the PIM. For example, `serial1/0` refers to the WAN interface in slot 1, port 0.

**Syntax**

**exec (T1/E1/T3 Interfaces)**

exec `interface` `interface` `bert-test` [ `start` | `stop` ]

**get (E1 Interfaces)**

set `interface` `interface` [ `clocking` | `e1-options` | `hold-time` ]

**get (Serial Interfaces)**

get `interface` `interface` [ `hold-time` | `serial-options` ]

**get (T1 Interfaces)**

get `interface` `interface` [ `clocking` | `hold-time` | `t1-options` ]

**get (T3 Interfaces)**

get `interface` `interface` [ `clocking` | `hold-time` | `t3-options` ]

**get (Cisco HDLC)**

get `interface` `interface` `cisco-hdlc`

**get (Frame Relay or Multilink Frame Relay)**

get `interface` `interface` `frame-relay`

**get (PPP or Multilink PPP)**

get `interface` `interface` `ppp`

**set/unset (Cisco HDLC)**

set `interface` `interface`

```
{  
    encapsulation`cisco-hdlc` |  
    ip unnumbered`interface` `src` `interface` |  
}
```
keepalives
{
  interval seconds |
  down-count number |
  up-count number
}

set/unset (E1 Interfaces)
set interface interface
{
  clocking { external | internal } |
  e1-options
  {
    bert-algorithm name_str |
    bert-error-rate rate |
    bert-period seconds |
    fcs { 16 | 32 } |
    framing { g704 | g704-no-crc4 | unframed } |
    idle-cycle-flag { flags | ones } |
    invert-data |
    loopback { local | remote } |
    start-end-flag { filler | shared } |
    timeslots timeslots
  } |
  hold-time { down milliseconds | up milliseconds }
}

set/unset (Frame Relay)
set interface interface
{
  encapsulation frame-relay |
  ip unnumbered interface src interface |
  frame-relay
  {
    dce |
    lmi
    {
      n391-dte number | n392-dce number | n392-dte number | n393-dce number |
      n393-dte number | t391-dte seconds |
      no-keepalive |
      type { ansi | itu }
    }
  } |
}
set interface subinterface
{
  frame-relay
  {
    dlci id_num |
    inverse-arp |
  } |
  zone zone
}
set/unset (Multilink Frame Relay)

```plaintext
set interface bundle
{
    bundle-ID string |
    drop-timeout milliseconds |
    encapsulation mlfr-uni-nni |
    frame-relay
    {
        n391-dte number | n392-dce number | n392-dte number | n393-dce number |
        n393-dte number | t391-dte seconds |
        no-keepalive |
        type { ansi | itu }
    }
    | minimum-links number |
    zone zone
}
set interface bundle_subinterface
{
    frame-relay
    {
        dlcid id_num |
        inverse-arp |
    }
    | zone zone
}
set interface interface
{
    encapsulation mlfr-uni-nni |
    ip unnumbered interface src interface |
    bundle bundle |
    mlfr-uni-nni
    {
        acknowledge-retries number |
        acknowledge-timer milliseconds |
        fragment-threshold bytes |
        hello-time milliseconds |
    }
}
```

set/unset (PPP)

```plaintext
set interface interface
{
    encapsulation ppp |
    ip unnumbered interface src interface |
    keepalives
    {
        interval seconds |
        down-count number |
    }
}
```
set/unset (Multilink PPP)

set interface bundle
{
  drop-timeout milliseconds |
  encapsulation mlppp |
  fragment-threshold bytes |
  minimum-links number |
  mrru bytes |
  short-sequence |
  zone zone
}

set interface interface
{
  bundle bundle |
  encapsulation ppp |
  ip unnumbered interface src interface |
  keepalives
  {
    interval seconds |
    down-count number |
  } |
}

set/unset (Serial Interfaces)

set interface interface
{
  disable |
  hold-time { down milliseconds | up milliseconds } |
  encapsulation frame-relay |
  serial-options
  {
    clock-rate rate |
    clocking-mode { dce | internal | loop } |
    dte-options
    {
      cts { ignore | normal | require } |
      dcd { ignore | normal | require } |
      dsr { ignore | normal | require } |
      dtr { ignore | normal | require } |
      ignore-all |
      rts { assert | de-assert | normal } |
      tm { ignore | normal | require }
    } |
    encoding { nrz | nrzi } |
    loopback { dce-local | local | remote } |
    transmit-clock [ invert ]
  }
}
set/unset (T1 Interfaces)

set interface interface
{
    disable |
    clocking { external | internal } |
    hold-time { down milliseconds | up milliseconds } |
    t1-options
    {
        bert-algorithm name_str |
        bert-error-rate rate |
        bert-period seconds |
        buildout { 0-132 | 133-265 | 266-398 | 399-531 | 532-655 } |
        byte-encoding { nx56 | nx64 } |
        fcs { 16 | 32 } |
        framing { esf | sf } |
        idle-cycle-flag { flags | ones } |
        invert-data |
        line-encoding { ami | b8zs } |
        loopback { local | payload | remote } |
        remote-loopback-respond |
        start-end-flag { filler | shared } |
        timeslots timeslots |
    }
}

set/unset (T3 Interfaces)

set interface interface
{
    disable |
    clocking { external | internal } |
    hold-time { down milliseconds | up milliseconds } |
    t3-options
    {
        bert-algorithm name_str |
        bert-error-rate rate |
        bert-period seconds |
        cbit-parity |
        compatibility-mode
        {
            adtran subrate rate |
            digital-link subrate rate |
            kentrox subrate rate |
            larscom subrate rate |
            verilink subrate rate |
        } |
        fcs { 16 | 32 } |
        feac-loop-respond |
        idle-cycle-flag { flags | ones } |
        long-buildout |
        loopback { local | payload | remote } |
        payload-scrambler |
        start-end-flag { filler | shared } |
    }
}
Keywords and Variables

**Variable Parameter**

get interface interface | subinterface | bundle | bundle_subinterface ...
set interface interface | subinterface | bundle | bundle_subinterface ...

**interface**
The name of the interface. All WAN interfaces on SSG devices, including serial, T1/E1, and T3, are named serial\textit{n1/n2}, where \textit{n1} is the slot number in the SSG chassis that is occupied by the Physical Interface Module (PIM), and \textit{n2} is the physical port on the PIM. For MLFR and MLPPP, you configure and add physical interfaces to the bundle interface.

**src interface**
The name of the source interface to which an unnumbered interface is assigned an IP address. You can configure an unnumbered interface to use a source interface when the unnumbered interface does not work.

**subinterface**
(\textit{.Frame Relay only}) The name of a virtual interface that is associated with a physical interface. You can create multiple subinterfaces on a physical interface. Subinterface names consist of the physical interface name, followed by a subinterface identification number, for example, serial1/1\textsubscript{1} or serial1/1\textsubscript{2}.

**bundle**
(MLFR and MLPPP only) The name of the bundle interface. Bundle interface names consists of ml, followed by an identification number. For example, bundle interface names can be ml\textsubscript{1}, ml\textsubscript{2}, and so on.

**bundle_subinterface**
(MLFR only) The name of a virtual interface that is associated with a bundle interface. You can create multiple subinterfaces on a bundle interface. Subinterface names consist of the bundle interface name, followed by a subinterface identification number, for example, ml\textsubscript{1.1} or ml\textsubscript{1.2}.

**Example:** The following command specifies the IP address of a remote gateway peer (1.1.1.25) for the serial interface in port 0 of the PIM in slot 1:

```bash
set interface serial1/0 gateway 1.1.1.25
```

**bert-test**

exec interface interface bert-test [ start | stop ]

**bert-test**
Starts or stops bit error rate testing on the specified interface.

**bundle**

set interface interface bundle bundle

**bundle**
(For multilink interfaces only) Adds the physical link \textit{interface} to the multilink interface \textit{bundle}.

**bundle-ID**

set interface bundle bundle-ID string

**bundle-ID**
Specifies an identifier for the bundle interface. If you do not specify a bundle ID, the bundle interface name is used.
**cisco-hdlc**
get interface interface cisco-hdlc

cisco-hdlc Shows the statistics and configuration information for an interface configured for Cisco High-Level Data Link Control protocol.

**clocking**
set interface interface clocking external | internal

clocking Specifies the clocking source for T1/E1 or T3 lines. You can specify one of the following options:
- **external** Specifies that clocking is provided by the DCE (loop timing).
- **internal** Specifies that clocking is provided by the SSG device’s own system clock. This is the default.

**disable**
set interface interface disable

disable Disables the interface. WAN interfaces are enabled by default.

**drop-timeout**
set interface bundle drop-timeout milliseconds

drop-timeout (For multiline bundle interfaces only) Specifies the drop timeout in milliseconds. The drop timeout provides a recovery mechanism if individual links in the multilink bundle drop one or more packets. The default is 0, which means that drop timeout is disabled. Specify a value between 0-127 milliseconds.

**e1-options**
set interface interface e1-options ...

e1-options Specifies options for an E1 interface. You can specify the following:
- **bert-algorithm** Sets the bit error rate testing (BERT) algorithm for the interface. The algorithm is the pattern to send in the bitstream. You can specify one of the following options:
  - **all-ones-repeating** Repeating one bits.
  - **all-zeros-repeating** Repeating zero bits.
  - **alternating-double-ones-zeros** Alternating pairs of ones and zeroes.
  - **alternating-ones-zeros** Alternating ones and zeroes.
  - **pseudo-2e10** Pattern is $2^{10}$-1.
  - **pseudo-2e11-o152** Pattern is $2^{11}$-1 (per O.152 standard).
  - **pseudo-2e15-o151** Pattern is $2^{15}$-1 (per O.152 standard). This is the default.
- **pseudo-2e17** Pattern is $2^{17}-1$.
- **pseudo-2e18** Pattern is $2^{18}-1$.
- **pseudo-2e20-o151** Pattern is $2^{20}-1$ (per O.151 standard).
- **pseudo-2e20-o153** Pattern is $2^{20}-1$ (per O.153 standard).
- **pseudo-2e21** Pattern is $2^{21}-1$.
- **pseudo-2e22** Pattern is $2^{22}-1$.
- **pseudo-2e23-o151** Pattern is $2^{23}$ (per O.151 standard).
- **pseudo-2e25** Pattern is $2^{25}-1$.
- **pseudo-2e28** Pattern is $2^{28}-1$.
- **pseudo-2e29** Pattern is $2^{29}-1$.
- **pseudo-2e3** Pattern is $2^{3}-1$.
- **pseudo-2e31** Pattern is $2^{31}-1$.
- **pseudo-2e32** Pattern is $2^{32}-1$.
- **pseudo-2e4** Pattern is $2^{4}-1$.
- **pseudo-2e5** Pattern is $2^{5}-1$.
- **pseudo-2e6** Pattern is $2^{6}-1$.
- **pseudo-2e7** Pattern is $2^{7}-1$.
- **pseudo-2e9-o153** Pattern is $2^{9}-1$ (per O.153 standard).
- **repeating-1-in-4** One bit in 4 is set.
- **repeating-1-in-8** One bit in 8 is set.
- **repeating-3-in-24** Three bits in 24 are set.
- **bert-error-rate** Sets the bit error rate (BER) to use in BERT. This can be an integer from 0 to 7, which corresponds to a BER from $10^{-0}$ (1 error per bit) to $10^{-7}$. The default is 0.
- **bert-period** Sets the length of the BERT, in seconds. The default is 10. Specify a value between 1 and 240 seconds.
- **fcs** Specifies the number of bits in the frame checksum. You can specify one of the following:
  - **16** 16 bits. This is the default.
  - **32** 32 bits.
- **framing** Sets the framing mode for the E1 line. You can specify the following:
  - **g704** G704 mode with cyclic redundancy check 4 (CRC 4). This is the default.
  - **g704-no-crc4** G704 mode without CRC4.
  - **unframed** Unframed transmission format.
- **idle-cycle-flag** Sets the value to transmit in idle cycles. You can specify one of the following:
  - **flags** Transmit 0x7E in idle cycles. This is the default.
  - **ones** Transmit 0xFF (all ones) in idle cycles.
- **invert-data** Specifies data inversion. Data inversion is normally used only in alternate mark inversion (AMI) mode. By default, this is not set.
- **loopback** Specifies loopback mode. By default, no loopback mode is set. You can specify one of the following:
  - **local** Local loopback.
  - **remote** Remote loopback.

- **start-end-flag** Sets the start and end flags on transmission. You can specify one of the following:
  - **filler** Send two idle cycles between start/end flags. This is the default.
  - **shared** Share start/end flags on transmit.

- **timeslots** Specifies the number of time slots allocated to a fractional E1 interface. By default, all time slots are active. Specify values from 2 to 32. Use hyphens to specify a range. Use commas (with no spaces before or after) to separate individual time slots or ranges. For example, you can specify the following: 3-5,9,22-24,28.

### encapsulation

**set interface** *interface* encapsulation { cisco-hdlc | frame-relay | mlfr-uni-nni | mlppp | ppp }

**set interface** *bundle* encapsulation { mlfr-uni-nni | mlppp }

- **cisco-hdlc** Sets Cisco High-Level Data Link Control (Cisco HDLC) encapsulation on the specified interface.
- **frame-relay** Sets Frame Relay encapsulation on the specified interface.
- **mlfr-uni-nni** (For Frame Relay multilink bundle interfaces and bundle links only) Sets Multilink Frame Relay User-to-Network Interface (UNI) encapsulation, based on Frame Relay Forum Multilink Implementation Agreement FRF.16, on the specified interface.
- **mlppp** (For MLPPP bundle interfaces only) Sets Multilink Point-to-Point Protocol on the specified interface.
- **ppp** Sets Point-to-Point Protocol (PPP) encapsulation on the specified interface.

### fragment-threshold

**set interface** *bundle* fragment-threshold *bytes*

**fragment-threshold** (For MLPPP bundle interfaces only) Specifies the maximum size, in bytes, for packet payloads transmitted across the individual links within the multilink circuit. The threshold value affects the payload only; it does not affect the MLPPP header. The default value is 0 bytes (disabled). Specify a value between 128-16320 bytes.

### frame-relay

**get interface** *interface* frame-relay

**set interface** *interface* frame-relay ...

For the **get** command, shows the statistics and configuration information for an interface configured for Frame Relay or Multilink Frame Relay. The interface can be a WAN interface, a WAN subinterface, a bundle interface, or bundle subinterface.
Chapter 9: New and Modified CLI Commands

**dlci id_num** (For Frame Relay subinterfaces only) Configures the data link connection identifier (DLCI) for a permanent virtual circuit (PVC) for Frame Relay and Multilink Frame Relay user-to-network interface (UNI) encapsulations. Specify a value between 16 and 1022.

**inverse-arp** (For Frame Relay subinterfaces only) Configures the router to respond to inverse Frame Relay Address Resolution Protocol (ARP) requests by providing IP address information to the requesting router at the other end of the Frame Relay PVC.

**lmi** Sets the type of Local Management Interface (LMI) packets used for keepalives and keepalive settings. You can specify the following:

- **n391-dte number** Specifies the data terminal equipment (DTE) full status polling interval. The DTE sends a status inquiry to the data circuit-terminating equipment (DCE) at the interval specified by `t391-dte`. `n391-dte` specifies the frequency at which these inquiries expect a full status report; for example, a `n391-dte` value of 10 would specify a full status report in response to every tenth inquiry. The intermediate inquiries ask for a keepalive exchange only. The range is from 1 through 255, with a default value of 6.

- **t391-dte seconds** Specifies the DTE keepalive timer, which is the period at which the DTE sends out a keepalive response request to the DCE and updates status depending on the DTE error-threshold value. The range is from 5 through 30 seconds, with a default value of 10 seconds.

- **n392-dte number** Specifies the DTE error threshold, which is the number of errors required to bring down the link, within the event-count specified by `n393-dte`. The range is from 1 through 10, with a default value of 3.

- **n393-dte number** Specifies the DTE monitored event-count. The range is from 1 through 10, with a default value of 4.

- **no-keepalive** Disables the sending of keepalives on the interface.

- **type** Specifies the type of LMI packets for keepalives. You can specify one of the following:
  - **ansi** Specifies ANSI T1.617 Annex D LMIs.
  - **itu** Specifies ITU Q933 Annex A LMIs.

**hold-time**

set interface `interface` hold-time { down milliseconds | up milliseconds }

**hold-time** The period of time after an interface has transitioned from up to down (or down to up), in milliseconds, before the interface is advertised to the system as being down (or up). The default is 0 milliseconds. Specify a value between 0 and 65534.

**ip unnumbered interface**

set interface `interface` ip unnumbered interface `src interface`

**ip unnumbered interface** Enables the local address to be derived from a source interface (`src interface`) which has been configured with an IP address.
keepalives

set interface interface keepalives { interval seconds | down-count number | up-count number }

keepalives  By default, physical interfaces configured with Cisco-HDLC or PPP encapsulation send keepalive packets at 10-second intervals. You can configure the following keepalive parameters:

- **interval** Specifies the interval at which the interface sends keepalive packets on a link. The default is 10 seconds. Specify a value between 1-32767 seconds.
- **down-count** Specifies the number of successive times that a destination fails to receive keepalive packets before it considers the link to be down. The default is 3 times. Specify a value between 1-255.
- **up-count** (Cisco HDLC encapsulation only) Specifies the number of times that a destination must receive a keepalive packet before it considers the link to be up. The default is 0 (disabled). Specify a value between 1-255.

minimum-links

set interface bundle minimum-links number

minimum-links  (For multilink bundle interfaces only) Sets the minimum number of bundle links that must be up for the bundle to be considered up. The default is 1. You can specify a value between 1-8.

mlfr-uni-nni

set interface interface mlfr-uni-nni ...

mlfr-uni-nni  (For multilink bundle links only) Configures options for Multilink Frame Relay FRF.16 operations. You can configure the following:

- **acknowledge-retries number** Specifies the number of retransmission attempts to be made for consecutive hello or remove-link messages after the expiration of the acknowledgement timer. The default is 2. Specify a value between 1-5.
- **acknowledge-timer milliseconds** Specifies the maximum period, in milliseconds, to wait for an add-link, hello, or remove-link acknowledgement. The default is 4 milliseconds. Specify a value between 1-10.
- **fragment-threshold bytes** Specifies the maximum size for packet payloads transmitted across bundle links within a multilink circuit. The default is the maximum transmission unit (MTU) of the physical link. Specify a multiple of 64 bytes.
- **hello-timer milliseconds** Specifies the rate, in milliseconds, at which hello messages are sent. The default is 10 milliseconds. Specify a value between 1-180.

mrru

set interface bundle mrru bytes

mrru  (For MLPPP bundle interfaces only) Specifies the maximum packet size, in bytes, that the multilink interface can process. The default is 1500 bytes. Specify a value between 1500-4500.
**ppp**

*get interface interface ppp*

**ppp**  Shows the statistics and configuration information for an interface configured for Point-to-Point Protocol (PPP) or Multilink PPP. The interface can be a WAN interface, a bundle interface, or a WAN interface that is a member of a MLPPP bundle interface.

**ppp profile**

*set interface interface ppp profile profile*

**ppp profile**  Binds the Point-to-Point Protocol (PPP) profile to the specified interface. You configure PPP profiles with the `set ppp profile` command.

**serial-options**

*set interface interface serial-options *

**serial-options**  Specifies options for a serial interface. You can specify the following:

- **clock-rate**  Sets the clock rate for the interface, in Kilohertz (KHz) or Megahertz (MHz), for EIA-530 and V.35 interfaces (for X.21 interfaces, you must specify `loop` for the `clocking-mode` option). The default is 8.0 MHz. You can specify one of the following options:
  - 1.2khz
  - 2.4khz
  - 9.6khz
  - 19.2khz
  - 38.4khz
  - 56.0khz
  - 64.0khz
  - 72.0khz
  - 125.0khz
  - 148.0khz
  - 250.0khz
  - 500.0khz
  - 800.0khz
  - 1.0mhz
  - 8.0mhz
  - 1.3mhz
  - 2.0mhz
  - 4.0mhz
  - 56.0khz
  - 64.0khz
  - 72.0khz
  - 125.0khz
  - 148.0khz
  - 250.0khz
  - 500.0khz
  - 800.0khz
  - 1.0mhz
  - 8.0mhz

- **clocking-mode**  Specifies the clock source to determine the timing on serial interfaces. You can specify one of the following:
  - **dce**  Uses a transmit clock generated by the data circuit-terminating equipment (DCE) for the SSG device’s DTE. When the device is functioning as a DTE, you must use this clocking mode for all interfaces except X.21 serial interfaces.
  - **internal**  Uses the SSG device’s internal clock. When the device is functioning as a DCE, we recommend that you use this clocking mode for all interfaces. You can configure the speed of the clock with the `clock-rate` option.
  - **loop**  Uses the DCE’s or DTE’s receive clock. For X.21 serial interfaces, you must use this clocking mode. This is the default.
**dte-options** Sets data terminal equipment (DTE) options/control leads. You can specify the following:

- **cts** Specifies the from-DCE clear to send (CTS) signal handling for EIA-530 and V.35 interfaces. You can specify one of the following:
  - **ignore** Ignores CTS signal.
  - **normal** Normal CTS signal, as defined by TIA/EIA Standard 530. This is the default.
  - **require** The from-DCE CTS signal must be asserted.

- **dcd** Specifies the from-DCE data carrier detect (DCD) signal handling for EIA-530 and V.35 interfaces. You can specify one of the following:
  - **ignore** Ignores DCD signal.
  - **normal** Normal DCD signal, as defined by TIA/EIA Standard 530. This is the default.
  - **require** The from-DCE DCD signal must be asserted.

- **dsr** Specifies the from-DCE data set ready (DSR) signal handling for EIA-530 and V.35 interfaces. You can specify one of the following:
  - **ignore** Ignores DSR signal.
  - **normal** Normal DSR signal, as defined by TIA/EIA Standard 530. This is the default.
  - **require** The from-DCE DSR signal must be asserted.

- **dtr** Specifies data transmit ready (DTR) signal handling for EIA-530 and V.35 interfaces. You can specify one of the following:
  - **assert** Asserts the DTR signal.
  - **auto-synchronize** Normal DTR signal, with automatic resynchronization.
  - **de-assert** Deasserts the DTR signal.
  - **normal** Normal DTR signal, as defined by TIA/EIA Standard 530. This is the default.

- **ignore-all** Specifies that all control leads are ignored. By default, this is not set.

- **rts** Specifies the to-DCE request to send (RTS) signal handling for EIA-530 and V.35 interfaces. You can specify one of the following:
  - **assert** Asserts RTS signal.
  - **de-assert** Deasserts RTS signal.
  - **normal** Normal RTS signal, as defined by TIA/EIA Standard 530. This is the default.

- **tm** Specifies the test mode (TM) signal for EIA-530 interfaces. You can specify one of the following:
  - **ignore** Ignores TM signal.
  - **normal** Normal TM signal. This is the default.
  - **require** The from-DCE TM signal must be asserted.

**encoding** Sets line encoding. You can specify one of the following:

- **nrz** Non-return-to-zero. This is the default.
- **nrzi** Non-return-to-zero-inverted.
Chapter 9: New and Modified CLI Commands

- **loopback** Sets loopback mode. By default, no loopback mode is specified. You can specify one of the following:
  - **dce-local** DCE local loopback (DTE mode only).
  - **local** Local loopback.
  - **remote** Remote/line interface unit (LIU) loopback.

- **transmit-clock** Sets the transmit-clock phase. By default, this is not set. You can specify the following:
  - **invert** Shift clock phase 180 degrees.

**short-sequence**

```
set interface bundle short-sequence
```

**short-sequence** (For MLPPP bundle interfaces only) Specifies a sequence-header format of 12 bits. The default is 24 bits.

**t1-options**

```
set interface interface t1-options ...
```

**t1-options** Specifies options for a T1 interface. You can specify the following:

- **bert-algorithm** Sets the bit error rate testing (BERT) algorithm for the interface. The algorithm is the pattern to send in the bitstream. You can specify one of the following options:
  - **all-ones-repeating** Repeating one bits.
  - **all-zeros-repeating** Repeating zero bits.
  - **alternating-double-ones-zeros** Alternating pairs of ones and zeroes.
  - **alternating-ones-zeros** Alternating ones and zeroes.
  - **pseudo-2e10** Pattern is \(2^{10}-1\).
  - **pseudo-2e11-o152** Pattern is \(2^{11}-1\) (per O.152 standard).
  - **pseudo-2e15-o151** Pattern is \(2^{15}-1\) (per O.152 standard). This is the default.
  - **pseudo-2e17** Pattern is \(2^{17}-1\).
  - **pseudo-2e18** Pattern is \(2^{18}-1\).
  - **pseudo-2e20-o151** Pattern is \(2^{20}-1\) (per O.151 standard).
  - **pseudo-2e20-o153** Pattern is \(2^{20}-1\) (per O.153 standard).
  - **pseudo-2e21** Pattern is \(2^{21}-1\).
  - **pseudo-2e22** Pattern is \(2^{22}-1\).
  - **pseudo-2e23-o151** Pattern is \(2^{23}\) (per O.151 standard).
  - **pseudo-2e25** Pattern is \(2^{25}-1\).
  - **pseudo-2e28** Pattern is \(2^{28}-1\).
  - **pseudo-2e29** Pattern is \(2^{29}-1\).
  - **pseudo-2e3** Pattern is \(2^{3}-1\).
  - **pseudo-2e31** Pattern is \(2^{31}-1\).
  - **pseudo-2e32** Pattern is \(2^{32}-1\).
  - **pseudo-2e4** Pattern is \(2^{4}-1\).
- **pseudo-2e5** Pattern is $2^5-1$.
- **pseudo-2e6** Pattern is $2^6-1$.
- **pseudo-2e7** Pattern is $2^7-1$.
- **pseudo-2e9-o153** Pattern is $2^9-1$ (per O.153 standard).
- **repeating-1-in-4** One bit in 4 is set.
- **repeating-1-in-8** One bit in 8 is set.
- **repeating-3-in-24** Three bits in 24 are set.

- **bert-error-rate** Sets the bit error rate (BER) to use in BERT. This can be an integer from 0 to 7, which corresponds to a BER from $10^{-0}$ (1 error per bit) to $10^{-7}$. The default is 0.

- **bert-period** Sets the length of the BERT, in seconds. The default is 10. Specify a value between 1 and 240 seconds.

- **buildout** Sets the T1 cable length in feet. You can specify the following:
  - **0-132** 0-40 meters. This is the default.
  - **133-265** 40-81 meters.
  - **266-398** 81-121 meters.
  - **399-531** 121-162 meters.
  - **532-655** 162-200 meters.

- **byte-encoding** Sets the byte-encoding method. You can specify one of the following:
  - **nx56** Seven bits per byte.
  - **nx64** Eight bits per byte. This is the default.

- **fcs** Specifies the number of bits in the frame checksum. You can specify one of the following:
  - **16** 16 bits. This is the default.
  - **32** 32 bits.

- **framing** Sets the framing mode for the T1 line. You can specify the following:
  - **esf** Extended superframe. This is the default.
  - **sf** Superframe.

- **idle-cycle-flag** Sets the value to transmit in idle cycles. You can specify one of the following:
  - **flags** Transmit 0x7E in idle cycles. This is the default.
  - **ones** Transmit 0xFF (all ones) in idle cycles.

- **invert-data** Specifies data inversion. Data inversion is normally used only in alternate mark inversion (AMI) mode. By default, this is not set.

- **line-encoding** Specifies the line-encoding method. You can specify one of the following:
  - **ami** Alternate mark inversion.
  - **b8zs** Binary 8 zero substitution. This is the default.

- **loopback** Specifies loopback mode. By default, no loopback mode is set. You can specify one of the following:
  - **local** Local loopback.
  - **payload** Payload loopback.
  - **remote** Remote loopback.
**remote-loopback-respond** Specifies that the interface responds to loop requests from the remote end. By default, this is not set.

**start-end-flag** Sets the start and end flags on transmission. You can specify one of the following:
- **filler** Sends two idle cycles between start/end flags. This is the default.
- **shared** Shares start/end flags on transmit.

**timeslots** Specifies the number of time slots allocated to a fractional T1 interface. By default, all time slots are active. Specify values from 1 to 24. Use hyphens to specify a range. Use commas (with no spaces before or after) to separate individual time slots or ranges. For example, you can specify the following: 1-3,4,9,22-24.

**t3-options**

```plaintext
set interface interface t3-options ...
```

**t3-options** Specifies options for a T3 interface. You can specify the following:

- **bert-algorithm** Sets the bit error rate testing (BERT) algorithm for the interface. The algorithm is the pattern to send in the bitstream. You can specify one of the following options:
  - **all-ones-repeating** Repeating one bits.
  - **all-zeros-repeating** Repeating zero bits.
  - **alternating-double-ones-zeros** Alternating pairs of ones and zeroes.
  - **alternating-ones-zeros** Alternating ones and zeroes.
  - **pseudo-2e10** Pattern is $2^{10}$-1.
  - **pseudo-2e11-o152** Pattern is $2^{11}$-1 (per O.152 standard).
  - **pseudo-2e15-o151** Pattern is $2^{15}$-1 (per O.152 standard). This is the default.
  - **pseudo-2e17** Pattern is $2^{17}$-1.
  - **pseudo-2e18** Pattern is $2^{18}$-1.
  - **pseudo-2e20-o151** Pattern is $2^{20}$-1 (per O.151 standard).
  - **pseudo-2e20-o153** Pattern is $2^{20}$-1 (per O.153 standard).
  - **pseudo-2e21** Pattern is $2^{21}$-1.
  - **pseudo-2e22** Pattern is $2^{22}$-1.
  - **pseudo-2e23-o151** Pattern is $2^{23}$ (per O.151 standard).
  - **pseudo-2e25** Pattern is $2^{25}$-1.
  - **pseudo-2e28** Pattern is $2^{28}$-1.
  - **pseudo-2e29** Pattern is $2^{29}$-1.
  - **pseudo-2e3** Pattern is $2^{3}$-1.
  - **pseudo-2e31** Pattern is $2^{31}$-1.
  - **pseudo-2e32** Pattern is $2^{32}$-1.
  - **pseudo-2e4** Pattern is $2^{4}$-1.
- **pseudo-2e5** Pattern is $2^{5}-1$.
- **pseudo-2e6** Pattern is $2^{6}-1$.
- **pseudo-2e7** Pattern is $2^{7}-1$.
- **pseudo-2e9-o153** Pattern is $2^{9}-1$ (per O.153 standard).
- **repeating-1-in-4** One bit in 4 is set.
- **repeating-1-in-8** One bit in 8 is set.
- **repeating-3-in-24** Three bits in 24 are set.
- **bert-error-rate** Sets the bit error rate (BER) to use in BERT. This can be an integer from 0 to 7, which corresponds to a BER from $10^{-0}$ (1 error per bit) to $10^{-7}$. The default is 0.
- **bert-period** Sets the length of the BERT in seconds. The default is 10. Specify a value between 1 and 240 seconds.
- **cbit-parity** Disables or enables C-bit parity mode, which controls the type of framing that is present on the transmitted T3 signal. By default, C-bit parity mode is enabled. When C-bit parity mode is enabled, the C-bit positions are used for the FEBE, FEAC, terminal-data-link, path-parity, and mode-indicator bits, as defined in ANSI T1.107a-1989. When C-bit parity mode is disabled, the basic T3 framing mode (M13) is used.
- **compatibility-mode** Sets the T3 interface to be compatible with the channel service unit (CSU) at the remote end of the line. By default, no compatibility mode is set. You can specify one of the following:
  - **adtran subrate** For intelligent-queue (IQ) channels only. Sets the interface to be compatible with Adtran Channel Service Units (CSUs). Specify a value between 1 and 588.
**digital-link subrate** Sets the interface to be compatible with Digital Link CSUs. Specify one of the following bits-per-second values:

- 301Kb 11.4Mb 22.6Mb 53.7Mb
- 601Kb 11.7Mb 22.9Mb 54.0Mb
- 902Kb 12.0Mb 23.2Mb 54.3Mb
- 1.2Mb 12.3Mb 23.5Mb 54.6Mb
- 1.5Mb 12.6Mb 23.8Mb 54.9Mb
- 1.8Mb 12.9Mb 24.1Mb 55.2Mb
- 2.1Mb 13.2Mb 24.4Mb 55.5Mb
- 2.4Mb 13.5Mb 24.7Mb 55.8Mb
- 2.7Mb 13.8Mb 25.0Mb 56.1Mb
- 3.0Mb 14.1Mb 25.3Mb 56.4Mb
- 3.3Mb 14.4Mb 25.6Mb 56.7Mb
- 3.6Mb 14.7Mb 25.9Mb 57.0Mb
- 3.9Mb 15.0Mb 26.2Mb 57.3Mb
- 4.2Mb 15.3Mb 26.5Mb 57.6Mb
- 4.5Mb 15.6Mb 26.8Mb 57.9Mb
- 4.8Mb 15.9Mb 27.1Mb 58.2Mb
- 5.1Mb 16.2Mb 27.4Mb 58.5Mb
- 5.4Mb 16.5Mb 27.7Mb 58.8Mb
- 5.7Mb 16.8Mb 28.0Mb 59.1Mb
- 6.0Mb 17.1Mb 28.3Mb 59.4Mb
- 6.3Mb 17.4Mb 28.6Mb 59.7Mb
- 6.6Mb 17.7Mb 28.9Mb 60.0Mb
- 6.9Mb 18.0Mb 29.2Mb 60.3Mb
- 7.2Mb 18.3Mb 29.5Mb 60.6Mb
- 7.5Mb 18.6Mb 29.8Mb 60.9Mb
- 7.8Mb 18.9Mb 30.1Mb 61.2Mb
- 8.1Mb 19.2Mb 30.4Mb 61.5Mb
- 8.4Mb 19.5Mb 30.7Mb 61.8Mb
- 8.7Mb 19.8Mb 31.0Mb 62.1Mb
- 9.0Mb 20.1Mb 31.3Mb 62.4Mb
- 9.3Mb 20.5Mb 31.6Mb 62.7Mb
- 9.6Mb 20.8Mb 31.9Mb 63.0Mb
- 9.9Mb 21.1Mb 32.2Mb 63.3Mb
- 10.2Mb 21.4Mb 32.5Mb 63.6Mb
- 10.5Mb 21.7Mb 32.8Mb 63.9Mb
- 10.8Mb 22.0Mb 33.1Mb 64.2Mb
- 11.1Mb 22.3Mb 33.4Mb

**kentrox subrate** For IQ channels only. Sets the interface to be compatible with Kentrox CSUs. Specify a value between 1 and 69.

**larscom subrate** For IQ channels only. Sets the interface to be compatible with Larscom CSUs. Specify a value between 1 and 14.
- **verilink subrate** For IQ channels only. Sets the interface to be compatible with Verilink CSUs. Specify a value between 1 and 28.

- **fcs** Specifies the number of bits in the frame checksum. The checksum must be the same on both ends of the link. You can specify one of the following:
  - 16 16 bits. This is the default.
  - 32 32 bits.

- **feac-loop-respond** Sets the interface to respond to far-end alarm and control (FEAC) loop requests. By default, this is not set.

- **idle-cycle-flag** Sets the value to transmit in idle cycles. You can specify one of the following:
  - flags Transmit 0x7E in idle cycles. This is the default.
  - ones Transmit 0xFF (all ones) in idle cycles.

- **long-buildout** Specifies a long cable length (longer than 225 feet or 68.6 meters) for copper-cable-based T3 interfaces. By default, this is not set.

- **loopback** Specifies loopback mode. By default, no loopback mode is set. You can specify one of the following:
  - local Local loopback.
  - payload Payload loopback.
  - remote Remote loopback.

- **payload-scrambler** Enables High-Level Data Link Control (HDLC) payload scrambling on the interface. This type of scrambling provides better link stability, but both sides of the connection must either use or not use scrambling. By default, this is not set.

- **start-end-flag** Sets the start and end flags on transmission. You can specify one of the following:
  - filler Sends two idle cycles between start/end flags. This is the default.
  - shared Shares start/end flags on transmit.

### zone

**set interface** interface zone zone

**set interface** bundle zone zone

**zone zone** Assigns the interface to a security zone. For Multilink Frame Relay (MLFR) or Multilink PPP (MLPPP), assigns the multilink bundle interface or subinterface to a zone. You must assign a bundle link interface or subinterface to the same zone as the bundle interface to which it belongs.
PPP provides a standard method for encapsulating Network Layer protocol information over point-to-point links. PPP encapsulation is defined in RFC 1661, The Point-to-Point Protocol (PPP).

### Syntax

**get**

get ppp profile { all | profile }

**set/unset**

set ppp profile profile

{ auth
 { chap | pap | any | none } | chap-local-name name_str
 } | netmask mask | passive | password pswd_str | static-ip ip_addr

### Keywords and Variables

**profile**

set ppp profile profile ...

get ppp profile { all | profile }

**profile**

- Creates and configures a PPP access profile, which specifies authentication parameters for the PPP link. You bind the PPP access profile to an interface with the **set interface** command. You can configure the following parameters in the PPP access profile:
  - **auth** Specifies the authentication method to be used when establishing the link, or the hostname to be used in Challenge Handshake Authentication Protocol (CHAP) requests and responses. Specify one of the following values:
    - **chap** Specifies CHAP, as defined in RFC 1994, PPP Challenge Handshake Authentication Protocol (CHAP).
    - **pap** Specifies Password Authentication Protocol (PAP), as defined in RFC 1334, PPP Authentication Protocols.
    - **any** Specifies that CHAP is used first, but if negotiations with the peer are unsuccessful, PAP is used.
    - **none** Specifies that no authentication requests are made and that all incoming authentication challenges are denied.
    - **chap-local-name** Specifies the hostname used in CHAP challenge and response packets. By default, the system hostname is used.
  - **netmask** Specifies the netmask for the interface. The default is 255.255.255.255.
Use the `user` commands to create, remove, or display WAN user entries in the internal user authentication database.

**Syntax**

```sh
class set
set user name_str
  { password pswd_str | type wan }
```

**Keywords and Variables**

**Variable Parameter**

- `user` Defines the WAN user’s name (`name_str`).

- `password` Defines the password for the WAN user.

- `type` (For PPP and MLPPP encapsulated data links only) Defines the user as a WAN user. If CHAP or PAP authentication is configured for the PPP data link, the username and password for the peer device must be configured as a WAN user type.
Chapter 10

Message Descriptions

This chapter introduces new ScreenOS messages for this release. The description of each message includes an explanation of its meaning and, when appropriate, a recommended action. Messages are grouped by type and then within that type by severity level, from the most severe to the least severe.

“Interface” on page 104  “Multilink Frame Relay” on page 107
“Cisco HDLC” on page 104  “PPP” on page 111
“Frame Relay” on page 106  “WAN” on page 115

All messages reporting an administrative action include the location from which that action has been made: either from the console, from an administrator’s host IP address via SCS, Telnet, or the Web. When devices are used in a redundant cluster for high availability, the message also states whether the action occurred on a master or backup unit. Note that because the part of a message stating the source of an action is the same in all such messages, it is not included in the messages listed here.

For a complete list of ScreenOS log messages, refer to the Message Log Reference for ScreenOS 5.1.0.


**Interface**

These messages relate to WAN interfaces.

**Notification (00009)**

<table>
<thead>
<tr>
<th>Message</th>
<th>&lt;interface_option&gt; for interface &lt;interface&gt; has been changed to &lt;new_value&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>An admin has changed the value of an interface option (such as clocking, hold time up/down, BERT algorithm/error rate/period, build out, byte encoding, and so on)</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>

| Message | Admin status for interface <interface> has been changed to { enable | disable } |
|---------|------------------------------------------------------------------------------|
| Meaning | An admin enabled or disabled the interface. |
| Action  | No recommended action |

**Cisco HDLC**

These messages relate to the Cisco HDLC encapsulation protocol.

**Alert (00087)**

<table>
<thead>
<tr>
<th>Message</th>
<th>Cisco-HDLC detected loop &lt;number&gt; times on interface &lt;interface&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>A link loop (when the sender receives the same keepalive packet it sent out) has been detected on the interface.</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>

**Notification (00062)**

<table>
<thead>
<tr>
<th>Message</th>
<th>Set interface &lt;interface&gt; encap as cisco-hdlc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>An admin configured Cisco HDLC encapsulation on the specified interface.</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message</th>
<th>unset interface &lt;interface&gt; encap from cisco-hdlc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>An admin removed Cisco HDLC encapsulation on the specified interface.</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>
### Message Descriptions

<table>
<thead>
<tr>
<th>Message</th>
<th>Meaning</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISCO-HDLC keepalive is enabled on interface <code>&lt;interface&gt;</code></td>
<td>The specified interface is able to send keepalive packets. This is the default behavior.</td>
<td>No recommended action</td>
</tr>
<tr>
<td>CISCO-HDLC keepalive interval is changed from <code>&lt;number1&gt;</code> to <code>&lt;number2&gt;</code> on interface <code>&lt;interface&gt;</code></td>
<td>An admin changed the interval at which the specified interface sends keepalive packets.</td>
<td>No recommended action</td>
</tr>
<tr>
<td>CISCO-HDLC keepalive down count value is changed from <code>&lt;number&gt;</code> to <code>&lt;number&gt;</code> on interface <code>&lt;interface&gt;</code></td>
<td>An admin changed the number of consecutive times that the interface must fail to receive a keepalive before the link is considered to be down.</td>
<td>No recommended action</td>
</tr>
<tr>
<td>CISCO-HDLC keepalive up count value is changed from <code>&lt;number&gt;</code> to <code>&lt;number&gt;</code> on interface <code>&lt;interface&gt;</code></td>
<td>An admin changed the number of consecutive times that the interface must receive a keepalive before the link is considered to be up.</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>

### Notification (00561)

<table>
<thead>
<tr>
<th>Message</th>
<th>Meaning</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISCO-HDLC is up</td>
<td>The protocol is up or down on the specified interface.</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>
Frame Relay

These messages relate to the Frame Relay encapsulation protocol.

Alert (00086)

Message |fr/lmi|: <interface>: LMI link down due to errors over threshold (n392)
Meaning Local Management Interface is down on the specified interface because the number of errors encountered reached the configured DTE error threshold (default is 3).
Action No recommended action

Notification (00060)

Message |fr/lmi|: <interface>: DTE | DCE
Meaning The specified interface is configured for DTE or DCE operation.
Action No recommended action

Message |fr/lmi|: <interface>: LMI : set to enable | disable
Meaning An admin enabled or disabled LMI on the interface.
Action No recommended action

Message |fr/lmi|: <interface>: LMI : set to ANSI | ITU
Meaning An admin configured the LMI type on the interface as ANSI or ITU.
Action No recommended action

Message |fr/lmi|: <interface>: LMI: set <parameter> to <value>
Meaning An admin configured the indicated LMI parameter.
Action No recommended action

Message |fr/lmi|: <interface>: Set | Del the DLCI for the interface
Meaning An admin configured the DLCI for the interface.
Action No recommended action

Message |fr/lmi|: <interface>: Inverse arp enabled | disabled
Meaning An admin enabled or disabled inverse ARP on the interface.
Action No recommended action
**Notification (00559)**

Message: [fr/lmi]: <interface> LMI status change to DOWN | UP
Meaning: The LMI status has changed to down or up.
Action: No recommended action

Message: [fr/lmi]: <interface> dlci <id> status change to N(new) | D(delete) | A(Active) | I(inactive)
Meaning: The specified DLCI status has changed, as indicated.
Action: No recommended action

**Multilink Frame Relay**

These messages relate to the Multilink Frame Relay encapsulation protocol.

**Alert (00085)**

Message: [mlfr/lip]: <interface> detected loop <number> times
Meaning: A link loopback was detected for the indicated number of times.
Action: No recommended action

Message: [mlfr/lip]: the bid <string> in the ADD_LINK packet from link <link> is inconsistent with the received bid <string> on the bundle <bundle>
Meaning: An invalid bundle ID was detected in the received ADD_LINK packet.
Action: Check the bundle ID configuration at the local and remote endpoints.

**Notification (00061)**

Message: [mlfr/cfg]: set interface <interface> encap as mlfr-uni-nni
Meaning: An admin configured the specified interface for Multilink Frame Relay encapsulation.
Action: No recommended action

Message: [mlfr/cfg]: unset interface <interface> encap from mlfr-uni-nni
Meaning: An admin removed Multilink Frame Relay encapsulation from the specified interface.
Action: No recommended action
Message: [mlfr/cfg]: set MLFR bundle-id as <bundle-id> for multilink interface <interface>
Meaning: An admin configured a bundle link identifier for the specified multilink interface.
Action: No recommended action

Message: [mlfr/cfg]: unset MLFR bundle-id as the name of multilink interface <interface>
Meaning: An admin removed the bundle link identifier from the specified multilink interface.
Action: No recommended action

Message: [mlfr/cfg]: set MLFR drop-timeout as <number> for multilink interface <interface>
Meaning: An admin configured the drop timeout for the specified multilink interface.
Action: No recommended action

Message: [mlfr/cfg]: unset MLFR drop-timeout to 0 (disable) for multilink interface <interface>
Meaning: An admin disabled drop timeout for the specified multilink interface.
Action: No recommended action

Message: [mlfr/cfg]: set MLFR minimum-links as <number> for multilink interface <interface>
Meaning: An admin configured the minimum number of links for the specified multilink interface.
Action: No recommended action

Message: [mlfr/cfg]: unset MLFR minimum-links to default (1) for multilink interface <interface>
Meaning: An admin reset the minimum number of links for the specified multilink interface to the default (1).
Action: No recommended action

Message: [mlfr/cfg]: set lip hello-timer as <number>(s) for bundle link <bundle>
Meaning: An admin configured the rate at which hello messages are sent for the specified multilink interface.
Action: No recommended action
Message [mlfr/cfg]: unset lip hello-timer to default <number> (s) for bundle link <bundle>
Meaning An admin reset the rate at which hello messages are sent on the specified multilink interface to the default (10 milliseconds).
Action No recommended action

Message [mlfr/cfg]: set lip fragment-threshold as <number> for bundle link <bundle>
Meaning An admin configured the maximum size for packet payloads for the specified multilink interface.
Action No recommended action

Message [mlfr/cfg]: unset bundle link <bundle> lip fragment-threshold to <mtu>
Meaning An admin reset the maximum size for packet payloads for the specified multilink interface to the default (MTU size of the physical link).
Action No recommended action

Message [mlfr/cfg]: set lip acknowledge-retries as <number> for bundle link <bundle>
Meaning An admin configured the number of retransmission attempts after the acknowledge timer expires for the specified multilink interface.
Action No recommended action

Message [mlfr/cfg]: unset lip acknowledge-retries to default <number> for bundle link <bundle>
Meaning An admin reset the number of retransmission attempts after the acknowledge timer expires for the specified multilink interface to the default (2 times).
Action No recommended action

Message [mlfr/cfg]: set lip acknowledge-timer as <number> for bundle link <bundle>
Meaning An admin configured the maximum period to wait for an acknowledgement for the specified multilink interface.
Action No recommended action

Message [mlfr/cfg]: unset lip acknowledge-timer to default <number> for bundle link <bundle>
Meaning An admin reset the maximum period to wait for an acknowledgement for the specified multilink interface to the default (4 milliseconds).
Action No recommended action
### Notification (00560)

<table>
<thead>
<tr>
<th>Message</th>
<th>[mlfr/cfg]: add link <code>&lt;interface&gt;</code> to bundle <code>&lt;bundle&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>An admin added the specified interface to the multilink interface.</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message</th>
<th>[mlfr/cfg]: delete link <code>&lt;interface&gt;</code> from bundle <code>&lt;bundle&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>An admin removed the specified interface from the multilink interface.</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>

| Message | [mlfr/lip]: `<link>` LIP FSM: `{ Invalid state | Add_sent | Ack_rx | Add_rx | Up | Idle_pending | Idle | Down | Idle_idle } -> `{ Invalid state | Add_sent | Ack_rx | Add_rx | Up | Idle_pending | Idle | Down | Idle_idle }` by event `{ Invalid event | ADD_LINK(valid) | ADD_LINK(invalid) | ADD_LINK_ACK | ADD_LINK_REJ | HELLO | HELLO_ACK | REMOVE_LINK | REMOVE_LINK_ACK | T_HELLO_EXP | T_ACK_EXP_Retry | T_ACK_EXP_no_Retry | PH_DEACTIVATE | PH_ACTIVATE | PH_DATA | BL_ACTIVATE | BL_DEACTIVATE | BL_DATA } |
|---------|-----------------------------------------------|
| Meaning | The indicated event has changed the Link Integrity Protocol state (the previous and new states are shown). |
| Action  | No recommended action |

<table>
<thead>
<tr>
<th>Message</th>
<th>[mlfr/lip]: link interface <code>&lt;interface&gt;</code> LIP is up at bundle <code>&lt;bundle&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>Link Interface Protocol is up on the specified link interface in the bundle.</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message</th>
<th>[mlfr/lip]: link interface <code>&lt;interface&gt;</code> LIP is down at bundle <code>&lt;bundle&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>Link Interface Protocol is down on the specified link interface in the bundle.</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message</th>
<th>[mlfr/lip]: change bundle <code>&lt;bundle&gt;</code> physical status to up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>The specified bundle is up.</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message</th>
<th>[mlfr/lip]: change bundle <code>&lt;bundle&gt;</code> physical status to down</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>The specified bundle is down.</td>
</tr>
<tr>
<td>Action</td>
<td>No recommended action</td>
</tr>
</tbody>
</table>
PPP

These messages relate to Point-to-Point Protocol (PPP) encapsulation.

**Notification (00017)**

| Message | Notification (00017) 
| PPP settings were changed |
| Meaning | An admin reconfigured PPP. |
| Action | No recommended action |

**Alert (00088)**

| Message | Alert (00088) 
| PPP control packet queue on <interface> takes on too many | normal number packets. |
| Meaning | The “too many” message is generated when the queued packet number is too large. The “normal number” message is generated when the number returns back to a normal level. |
| Action | If the “too many” message appears, check the peer or other task for abnormal operation. |

| Message | Alert (00088) 
| PPP on <interface> detects loopback. |
| Meaning | PPP found a loopback on the specified interface. |
| Action | Check to see why the loopback is occurring. |

**Notification (00063)**

| Message | Notification (00063) 
| PPP profile <profile_name> is created | deleted. |
| Meaning | Ad admin has created or deleted a PPP profile with the specified name. |
| Action | No recommended action |

| Message | Notification (00063) 
| PPP profile <profile_name> changes authentication type to chap | pap | chap pap | any | none |
| Meaning | An admin changed the authentication method in the specified profile. |
| Action | No recommended action |

| Message | Notification (00063) 
| PPP profile <profile_name> changes local-name to “ <string> ” |
| Meaning | An admin changed the local name in the specified profile. |
| Action | No recommended action |
Message  PPP profile <profile_name> changes secret to "<string>".
Meaning   An admin changed the password in the specified profile.
Action    No recommended action

Message  PPP profile <profile_name> enable | disable passive mode chap.
Meaning   An admin enabled or disabled passive mode in the specified profile.
Action    No recommended action

Message  PPP profile <profile_name> sets [not] to use static ip.
Meaning   An admin set the use of a static IP address in the specified profile.
Action    No recommended action

Message  PPP profile <profile_name> sets netmask <ip_addr>.
Meaning   An admin set a netmask in the specified profile.
Action    No recommended action

Message  PPP set | unset encapsulation ppp | mlppp for interface <interface>.
Meaning   An admin set or unset PPP or MLPPP encapsulation for the specified interface.
Action    No recommended action

Message  PPP bind | unbind profile <profile_name> for interface <interface>.
Meaning   An admin bound or unbound a profile to the specified interface.
Action    No recommended action

Message  PPP enable | disable short sequence number for interface <bundle>.
Meaning   An admin set or unset the use of a 12-bit sequence header format in MLPPP packets for the specified multilink interface.
Action    No recommended action

Message  PPP set MRRU <number> for interface <bundle>.
Meaning   An admin set a new maximum received reconstructed unit size for the specified multilink interface.
Action    No recommended action

Message  PPP add | delete interface <interface> into | from bundle <bundle>.
Meaning   An admin added or deleted an interface to or from the specified bundle.
Action    No recommended action
**Notification (00562)**

**Message**  
PPP protocol on interface `<interface>` is up | down, local IP: `<ip_addr1>`, peer IP: `<ip_addr2>`

**Meaning**  
PPP is up or down; the local and peer IP addresses are shown.

**Action**  
No recommended action

**Message**  
PPP updates interface `<interface>`'s L3 MTU to `<number>`.

**Meaning**  
Based upon the results of PPP negotiation, the interface's MTU is updated to the specified number.

**Action**  
No recommended action

**Message**  
PPP updates interface `<interface>`'s IP to `<ip_addr>`.

**Meaning**  
PPP updated the interface’s IP address to the assigned address.

**Action**  
No recommended action

**Message**  
PPP on `<interface>` resets LCP for `<reason>`.

**Meaning**  
PPP has reset the Link Control Protocol because of one of the following reasons:
- IPCP finished
- LCP finished
- The profile was updated
- The Hostname was updated
- LCP failed to come up after negotiation
- NCP failed to come up after negotiation
- A profile was not obtained after NCP
- The IP address could not be modified after NCP
- The host route could not be set
- An admin changed the interface’s IP address
- An admin changed the interface’s MTU

**Action**  
Check the specified reason.

**Message**  
PPP member `<interface>` joins bundle `<bundle>` successfully.

**Meaning**  
The interface successfully joined the specified bundle after LCP.

**Action**  
No recommended action
| Message | PPP member `<interface>` fails to join bundle `<bundle>` for `<reason>`.
| Meaning | The interface was not able to join the specified bundle for one of the following reasons:
- No empty member entry is available
- Either side does not negotiate the MRRU
- The joining member carries a different EPD
- The peer joining member carries a different MRRU
- The peer joining member carries a different SSN flag
- The local joining member carries a different MRRU
- The local MRU is greater than the local MRRU
| Action | Check the specified reason. Make sure both sides of the link are using acceptable parameters.

| Message | PPP bundle `<bundle>` is up | down and then brings up | down bundle NCP.
| Meaning | The specified bundle is up or down, and brings up or down NCP.
| Action | No recommended action

| Message | PPP LCP on interface `<interface>` is up | down.
| Meaning | LCP state on the specified interface changed to up or down.
| Action | No recommended action

| Message | PPP authentication state on interface `<interface>`: `<state>`.
| Meaning | PPP authentication state on the specified interface is one of the following:
- Peer failed to authenticate itself
- Peer authenticated itself successfully
- Local failed to authenticate itself
- Local authenticated itself successfully
| Action | If either the peer or local failed to authenticate itself, check the user name and password configured on both sides.

| Message | PPP on interface `<interface>` is terminated by missing too many echo reply.
| Meaning | The local side sent many Echo-Requests without receiving a reply, so it terminated and then reset the PPP session.
| Action | Check to see why the peer failed to reply to the Echo-Requests.

| Message | PPP on interface `<interface>` is terminated by receiving Terminate-Request.
| Meaning | The peer sent a request to terminate the PPP session.
| Action | No recommended action
WAN

These messages relate to WAN Physical Interface Modules (PIMs).

Critical (00091)

Message: WAN card <slot-number> is not functioning properly and will be restarted.

Meaning: The WAN card in the specified slot is restarting.

Action: No recommended action
### Appendix A

## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADM</strong></td>
<td>Add-drop multiplexer.</td>
</tr>
<tr>
<td><strong>AIS</strong></td>
<td>Alarm indication signal.</td>
</tr>
<tr>
<td><strong>AMI</strong></td>
<td>Automatic Mark Inversion.</td>
</tr>
<tr>
<td><strong>ANSI</strong></td>
<td>American National Standards Institute.</td>
</tr>
<tr>
<td><strong>ARP</strong></td>
<td>Address Resolution Protocol.</td>
</tr>
<tr>
<td><strong>B8ZS</strong></td>
<td>8 bits zero suppression.</td>
</tr>
<tr>
<td><strong>BECN</strong></td>
<td>Backward Explicit Congestion Notification.</td>
</tr>
<tr>
<td><strong>BER</strong></td>
<td>Bit error rate. The ratio of error bits to total number of bits received in a transmission, usually expressed as 10 to a negative power.</td>
</tr>
<tr>
<td><strong>BERT</strong></td>
<td>Bit error rate testing.</td>
</tr>
<tr>
<td><strong>bundle</strong></td>
<td>An aggregation of multiple physical links.</td>
</tr>
<tr>
<td><strong>bundle link</strong></td>
<td>A physical link in a bundle.</td>
</tr>
<tr>
<td><strong>CAS</strong></td>
<td>Channel Associated Signaling.</td>
</tr>
<tr>
<td><strong>CHAP</strong></td>
<td>Challenge Access Authentication Protocol.</td>
</tr>
<tr>
<td><strong>Cisco HDLC</strong></td>
<td>Cisco High-level Data Link Control protocol. Proprietary Cisco encapsulation for transmitting LAN protocols over a WAN. HDLC specifies a data encapsulation method on synchronous serial links by means of frame characters and check sums. Cisco HDLC enables the transmission of multiple protocols.</td>
</tr>
<tr>
<td><strong>CLI</strong></td>
<td>Command Line Interface.</td>
</tr>
<tr>
<td><strong>CRC</strong></td>
<td>Cyclic redundancy check.</td>
</tr>
<tr>
<td><strong>CSU/DSU</strong></td>
<td>Channel Service Unit/Data Service Unit.</td>
</tr>
<tr>
<td><strong>CTS</strong></td>
<td>Clear to send.</td>
</tr>
<tr>
<td><strong>DCD</strong></td>
<td>Data carrier detect.</td>
</tr>
<tr>
<td><strong>DCE</strong></td>
<td>Data circuit-terminating equipment. This equipment provides switching services in the WAN and is typically owned and managed by the service provider.</td>
</tr>
</tbody>
</table>
**DLCI** Data-link connection identifier. In Frame Relay, DLCIs are assigned to separate customer traffic.

**DS0** A transmission rate of 64 Kbps. DS0 is the base for the digital signal X series.

**DS1** Digital signal 1, also known as a T1 interface.

**DS3** Digital signal 3, also known as a T3 interface.

**DSR** Data set ready.

**DTE** Data terminal equipment. RS-232 interface that the SSG device uses to exchange information with a serial device. This equipment is the terminating point for a specific network and is typically located on the customer premises.

**DTR** Data transmit ready.

**E1 interface** Physical WAN interface for transmitting signals in European digital transmission (E1) format. The E1 signal format carries information at a rate of 2.048 Mbps and can carry 32 channels of 64 Kbps each.

**encapsulation type** Type of protocol header in which data is wrapped for transmission.

**ESF** Extended superframe.

**FDL** Facilities data link.

**FEAC** Far-end alarm and control.

**Frame Relay** WAN protocol that operates over a variety of network interfaces, including serial, T1/E1, and T3/E3. Frame Relay allows private networks to reduce costs by sharing facilities between the end-point switches of a network managed by a Frame Relay service provider.

**HA** High Availability.

**IP** Internet Protocol.

**ISO** International Organization for Standardization.

**ITU-T** Telecommunication Standardization Sector of the International Telecommunications Union (ITU-T) that is the primary international body for cooperative standards for telecommunications equipment and systems. It was formerly known as the CCITT.

**LAN** Local area network.

**LCP** Link Control Protocol.

**LIP** Link Integrity Protocol.

**LIU** Line interface unit.

**LMI** Local Management Interface.

**LOS** Loss of signal.
MLFR  Multilink Frame Relay. Protocol that allows multiple physical links to be aggregated into a single logical link across a Frame Relay network.

MLPPP  Multilink Point-to-Point Protocol. Allows you to bundle multiple PPP links into a single logical unit. It can be used to better utilize bandwidth and also has the advantages of reduced latency and improved fault tolerance.

MRRU  Maximum received reconstructed unit.

MTU  Maximum transmission unit. Maximum or largest segment size that a network can transmit.

multilink interface  Virtual interface that provides access to a bundle.

NNI  Network-to-Network Interface. The interface between Frame Relay devices where both devices are in a private network or in a public network.

NRZ  Non-return to zero. Simple high/low voltage transmission method.

NRZI  Non-return to zero inverted. A form of digital transmission that ensures sending and receiving clocks are synchronized.

PAP  Password Authentication Protocol.

PIM  Physical Interface Module. Contains ports that allow network connections.

POP  Point of presence.

PPP  Point-to-Point Protocol. Link Layer protocol that provides multiprotocol encapsulation. PPP is used for Link Layer and Network Layer configuration.

PSN  Packet-switched network.

PVC  Permanent Virtual Circuit, which is identified by a DLCI. Allows separate flows of traffic to be multiplexed onto a single data link for transmission across a Frame Relay network.

RTS  Request to send.

RX  Receive.

security zone  A collection of one or more network segments requiring the regulation of inbound and outbound traffic via policies.

serial interface  Physical interface for transmitting data between computing devices. An SSG device has two types of serial interfaces:

- Asynchronous serial interface—RJ-45 console/modem port that can be connected to either a console terminal or a workstation running a terminal-emulation application, or to an asynchronous modem.

- Synchronous serial interface—Port that transmits packets at speeds up to 8 Mbps on cables that support V.35, V.21, EIA 530, RS-232, and RS-449 protocols. You cannot use this serial interface to connect a console.

SSG  Secure Services Gateway.
SF  Superframe.

SLARP  Serial Line Address Resolution Protocol.

SSG  Secure Services Gateway.

T1 interface  Physical WAN interface for transmitting digital signals in the T-carrier system; used in North America and Japan.

T3 interface  Physical WAN interface for transmitting digital signals in the T-carrier system; used in North America and Japan. T3 signals are formatted like T1 signals but carry information at a higher rate. T3 is also called DS3.

TIA/EIA  Telecommunications Industry Association/Electronics Industry Alliance.

TX  Transmit.

UNI  User-to-Network Interface. Interface between Frame Relay devices where one device is in a public network and the other device is in a private network.

UNI/NNI  User-to-Network Interface/Network-to-Network Interface.

Voh  High-level output voltage.

Vol  Low-level output voltage.

VPN  Virtual private network.

WAN  Wide area network.

WDM  Wavelength-division multiplexer.

WebUI  Web User Interface.
Index

B
BERT properties
  T3 interfaces, usage guidelines ........................................... 33
  bert-algorithm option
    T3 interfaces, usage guidelines ........................................... 33
  bert-error-rate option
    T3 interfaces, usage guidelines ........................................... 33
  bert-period option
    T3 interfaces, usage guidelines ........................................... 33

C
C-bit parity mode ........................................................... 31
cbit-parity option
  usage guidelines .............................................................. 31
checksums See frame checksums command
  set interface ........................................................................ 82
compatibility-mode option
  T3 interfaces, usage guidelines ........................................... 32
conventions
  WebUI ................................................................................ xvi
CSU compatibility
  T3 interfaces ...................................................................... 32

D
data inversion ..................................................................... 22
  disabling
    keepalives ......................................................................... 40
dici option
  usage guidelines .................................................................. 42

E
E1 interfaces
  data inversion ..................................................................... 22
  frame checksum ................................................................... 21
  framing mode ....................................................................... 22
  idle cycle flag .................................................................... 22
  loopback capability ............................................................ 23
  start-end flag ....................................................................... 22
  time slots .............................................................................. 22
encapsulation option
  physical interfaces
    usage guidelines ................................................................ 38

F
far-end alarm and control signal See FEAC response
  fcs option
    E1 interfaces
      usage guidelines .............................................................. 21
    T3 interfaces
      usage guidelines .............................................................. 30
  FEAC response ..................................................................... 31
  feac-loop-respond option
    usage guidelines .............................................................. 31
frame checksums
  E1 interfaces ..................................................................... 21
  T3 interfaces ..................................................................... 30
Frame Relay connections
  DLCIs .................................................................................. 42
Frame Relay encapsulation
  keepalives ........................................................................... 40
  physical interfaces ............................................................... 38
framing option
  E1 interfaces
    usage guidelines .............................................................. 22

H
HDLC payload scrambling
  T3 interfaces ..................................................................... 31

I
idle-cycle-flag option
  E1 interfaces
    usage guidelines .............................................................. 22
  T3 interfaces
    usage guidelines .............................................................. 30
invert-data option
  usage guidelines .............................................................. 22

K
keepalives
  Frame Relay, disabling ....................................................... 40
  keepalives option
    usage guidelines .............................................................. 40

L
line buildout
  T3 interfaces ..................................................................... 31
lmi-type option
usage guidelines ................................................. 41, 47
long-buildout option
usage guidelines .................................................. 31
loopback capability
  E1 interfaces ............................................................ 23
loopback option
  E1 interfaces
    usage guidelines .................................................. 23

M
mtu option
  Frame Relay encapsulation
    usage guidelines .................................................. 39, 46
MTUs
  protocol MTUs .................................................. 39, 46

N
no-keepalives option
usage guidelines .................................................. 40

P
payload-scrambler option
  T3 interfaces
    usage guidelines .................................................. 31
physical interfaces
  BERT properties .................................................. 33
  C-bit parity mode .................................................. 31
  CSU compatibility .................................................. 32
data inversion .................................................. 22
FEAC response .................................................. 31
frame checksums
  See frame checksums
Frame Relay encapsulation
  usage guidelines .................................................. 38
framing modes .................................................. 22
HDLC payload scrambling ........................................... 31
idle cycle flag .................................................. 22, 30
keepalives .................................................. 40
line buildout .................................................. 31
loopback capability
  E1 interfaces .................................................. 23
start-end flag .................................................. 22, 30
time slots
  fractional E1 interfaces ........................................... 22
protocol MTUs
  Frame Relay encapsulation ........................................... 39, 46

S
set commands
  interface .................................................. 82
start-end flag
  E1 interfaces .................................................. 22
  T3 interfaces .................................................. 30
start-end-flag option
  E1 interfaces
    usage guidelines .................................................. 22

T
T3 interfaces
  usage guidelines .................................................. 30
subrate option .................................................. 32

T3 interfaces
  BERT properties .................................................. 33
  C-bit parity mode .................................................. 31
  CSU compatibility .................................................. 32
  FEAC response .................................................. 31
  frame checksum .................................................. 30
  HDLC payload scrambling ........................................... 31
  idle cycle flag .................................................. 30
  line buildout .................................................. 31
  start-end flag .................................................. 30
time slots
  E1 interfaces .................................................. 22
timeslots option
  E1 interfaces
    usage guidelines .................................................. 22

W
WebUI conventions .................................................. xvi