MCN: 0081793-01
Version 7.0
Rev. November 22, 2002

CopperEdge®

CopperCraft Reference and MIB Definitions

CopperEdge 200 DSL Concentrator

Copper Mountain Networks, Inc.

San Diego Facility:
10145 Pacific Heights Boulevard
San Diego, California 92121

Palo Alto Facility:
1850 Embarcadero Road
Palo Alto, California 94303

Important Information on Networking Models

Beginning with Release 7.0, the former HDIA and CopperVPN networking models have been superseded by a new, more capable, more secure, and much easier-to-use netModel called CopperVPN+ (referred to in the software as simply CopperVPN). The functionality of the older netmodels continues to be supported for backward compatibility, but those configurations, when saved, are upgraded to the new CopperVPN netModel and format automatically. This document contains numerous references to the older netModels, and no attempt has been made to delete these. For high-level (overview) information in using the new CopperVPN netmodel, see the Version 7.0 Release notes, and the section on CopperVPN in the Installation and Operating Guide for your system.

V 7.0: 0081793-01
Document Rev. A
Release Date: June 12, 2002

Limited Warranty

Copper Mountain Networks, Inc. (“Copper Mountain”) warrants the CopperEdge™ 200 to be free from defects in materials or workmanship for a period of one (1) year from the date of shipment from Copper Mountain's factory. Should your CE200 fail during the warranty period, Copper Mountain will, at its option and as its sole and exclusive obligation under this warranty, repair or replace it with a like product, which may include new or refurbished parts or components. This warranty is extended only to the original purchaser and only covers failures due to defects in materials and workmanship which occur during normal use during the period of the warranty. It does not cover damage which occurs in shipment or failures resulting from misuse, negligence, accident, improper storage, installation or testing, unusual electrical stress, fire, lightning, other environmental hazards, unauthorized attempts at repair, operation inconsistent with published electrical and environmental specifications, or if the Product was maintained in a manner other than described in this document, or if the serial number or other identifications markings have been altered, removed or rendered illegible. Expendable components such as batteries or cabling external to the unit are not covered by this warranty.

For specific terms and conditions of the product warranty and Copper Mountain's obligations thereto, please refer to the warranty section of your purchase agreement.

In order to exercise your rights to repairs under this warranty, you must first contact Copper Mountain at 888-611-4266 (for calls made from U.S. locations only) or 858-410-7100 (for all calls made from outside the U.S.) to obtain authorization (including tracking number) and instructions for return of the product(s).

EXCEPT FOR THE EXPRESS WARRANTIES SET FORTH HEREIN AND IN THE APPLICABLE PURCHASE AGREEMENT, COPPER MOUNTAIN MAKES NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY AND HEREBY EXPRESSLY DISCLAIMS ANY AND ALL SUCH WARRANTIES, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE PRODUCTS PURCHASED. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT, WHETHER MADE BY COPPER MOUNTAIN OR OTHERWISE, SHALL BE DEEMED TO BE A WARRANTY FOR ANY PURPOSE OR GIVE RISE TO ANY LIABILITY OF COPPER MOUNTAIN.

The laws in some localities do not allow limitations on the warranty period so the above exclusions or limitations may not be applicable. This warranty provides specific legal rights. You may have other rights with respect to this product, depending on local laws.

Software License Agreement

Use of this Product is subject to the Software License Grant contained in the purchase agreement used to originally purchase this Product from Copper Mountain.

Except as expressly permitted by applicable law, Customer or any third party to which the software may be licensed as permitted under the original purchase agreement, may not alter, modify, adapt or translate, create any derivative works based on the software, reverse engineer, decompile, disassemble or otherwise attempt to derive the human-readable source code for the software licensed with this Product.

* * * * *

FCC Information

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions:

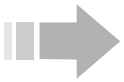
(1) This device may not cause harmful interference, and

(2) This device must accept any interference received, including interference that may cause undesired operation.

Changes or modifications to this device not expressly approved by the party responsible for compliance could void the user's authority to operate this device.

W A R N I N G

This equipment has been tested and found to comply with the limits for a Class "A" digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause interference to radio communications. Unprotected operation of this type of commercial equipment in a residential area is likely to cause harmful interference which the user would be required to correct



N O T E

The CopperEdge™ 200 was FCC verified under test conditions that included the use of shielded I/O cables and connectors on certain system components. To be in compliance with FCC regulations, you must use properly installed shielded cables and connectors on all connections to the System Control Module, and the V.35 and DS3 Frame WAN Modules. Shielding is not required on cables to DSL port connectors (LJ1 through LJ8 on the rear of the equipment chassis), the alarm status connector (JA2 on the rear of the chassis), or on the DC power input connections.

10145 Pacific Heights Blvd.
Suite 100
San Diego, CA 92121 USA

Declaration of Conformity

We, Copper Mountain Networks, Inc., declare under our sole responsibility that the following product to which this declaration relates is in conformity with the Essential Requirements and Harmonised Standards identified below.

1999/5/EC of the European Parliament and of the Council relating to Radio & Telecommunications Terminal Equipment. *The mutual recognition of conformity with this directive is based on compliance with the following Essential Requirements*:

89/336/EEC, EMC Directive

EN 55022:1998, Limits and methods of measurement of radio interference characteristics of Information Technology Equipment (ITE)

EN 55024:1998, Information Technology Equipment - Immunity Characteristics - Limits and Methods of Measurement

73/23/EEC, Low Voltage Directive

EN60950:1992, Safety of Information Technology Equipment, Including Electrical Business Equipment, Including amendments A1:1993, A2:1993, A3:1995, A4:1997, A11:1997

PRODUCT

Manufacturer	Copper Mountain Networks, Inc.
Trade Name/Model Number	CopperEdge 200 DSL concentrator, CE200

Alternate Trade Names/Model Numbers-Identical to that shown above except for Trade Name/Model Number.

- CopperEdge 200 made for Lucent Technologies by Copper Mountain Networks, Inc.

Year of First Issue 2000

Tom Lavka, Member of Technical Staff
Typed Name and Title of Representative

+01.858.410.7110, tlavka@coppermountain.com
Telephone Number & E Mail address

Signature of Manufacturer's Representative

May 15, 2000
Date

Form Copper Mountain: EU D of C

Issue: 05/15/00, Rev B

Industry Canada CS-03 Telecommunications Notice

The Industry Canada label identifies certified equipment. This certification means that the equipment meets certain telecommunications network protective operational and safety requirements in the appropriate Terminal Equipment Technical Requirements documents. The Department does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. The customer should be aware that compliance with the above conditions may not prevent degradation of service in some situations.

Repairs to certified equipment should be coordinated by a representative designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines, and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

C A U T I O N

Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

The Ringer Equivalence Number (REN) of this device is 0.6.

The Ringer Equivalence Number (REN) assigned to each terminal device provides an indication of the maximum number of terminals allowed to be connected to a telephone interface. The termination of an interface may consist of any combination of devices subject only to the requirement that the sum of the Ringer Equivalence Numbers of all the devices does not exceed 5.

This equipment uses Canadian CA11A Jacks.

Table of Contents

List of Figures

Revision History

Document MCN & Date	Release	Summary of Changes
0081793-01, 3/22/01	7.0	Limited Availability release.
0081793-01, 6/12/02	7.0	General Availability release
11/22/02	DDC	Corrected cmVclTable aaltype options and updated manual cover and all dates in front matter.

System Software and Applicability

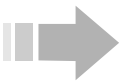
This document applies to CopperEdge 200 systems delivered under Hardware/Software Release 7.0.

Document Conventions

The following conventions are used throughout this document:

this bold-face font	Indicates commands that you type in order to command and control the *CopperEdge*. Examples of typed input are preceded by the system prompt, shown as "`Craft>`" or "`System>`"
`This fixed-space font`	Indicates output from the system that displays on your screen.
This italic font	Emphasizes new terms, names, titles, or trademarked words.

Throughout this document, you will encounter examples of configurations or commands showing link- or user-specific information such as IP addresses, MAC addresses, etc. Unless otherwise specified, all such data is fictitious and is provided for illustrative purposes only.



N O T E

Information or instructions to which you should pay particular attention.

C A U T I O N

Information alerting you to a hazard, either to personnel or to the systems and equipment.

C A U T I O N

Information alerting you to an electrostatic discharge hazard which could damage equipment or cause the loss of stored information.

W A R N I N G

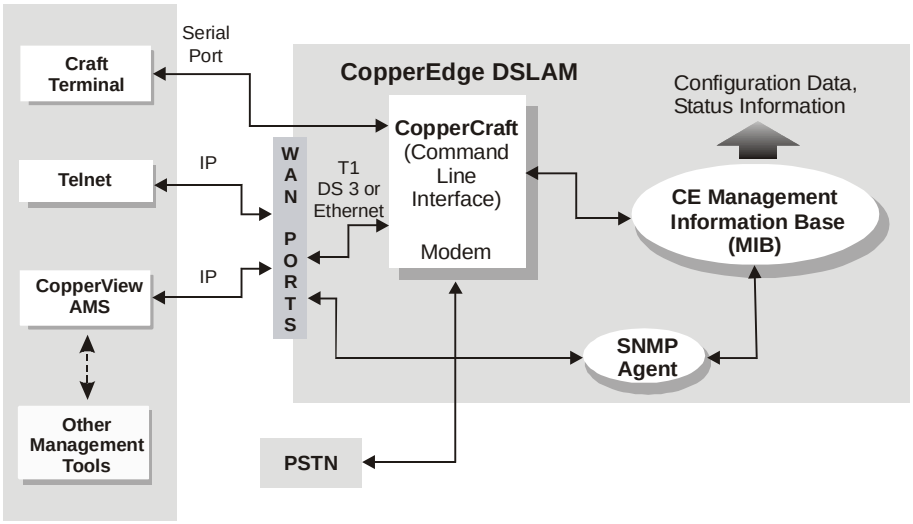
Information alerting you to a situation that could result in damage to the network, and/or violation of local or national laws.

CopperEdge 200 DSL Concentrator

Chapter 1
Overview

Three models for configuring and managing the CE200 systems are available. You can use any or all of them. They are:

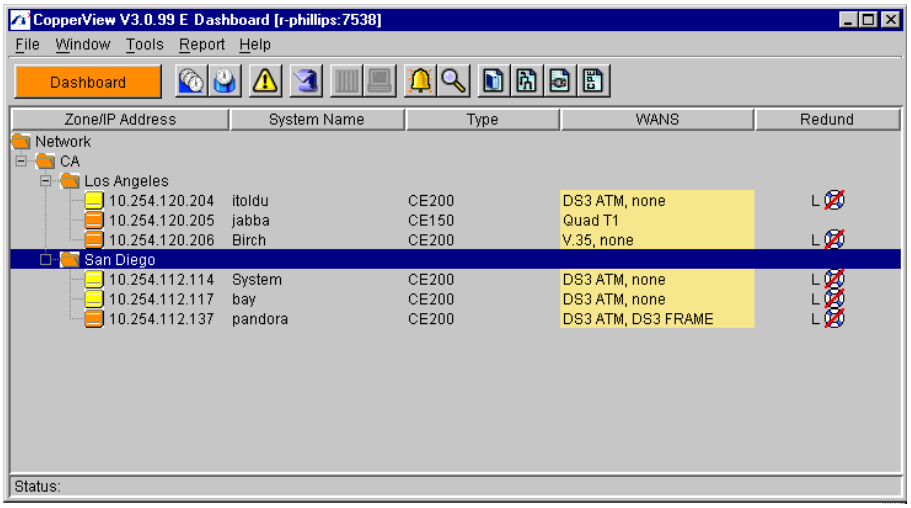
- Sessions employing the CopperView™ Access Management System (AMS) or other SNMP-based network management systems. The AMS contains a CopperView Element Manager™ (EM) system.
- Sessions employing the CopperCraft™ command line interface (CLI) along with a LAN connection and Telnet connection to the CE200 from a remote station.
- Sessions employing the CopperCraft command line interface along with a direct connection to the serial port on the Control Module of the CE200 from a terminal or a terminal emulator.



Status/Configuration Data Flow

CopperView AMS

The *CopperView* AMS uses the Simple Network Management Protocol (SNMP) and allows you to organize a large group of *CopperEdge* DSLAMs into a series of smaller groups or zones.



The CopperView Access Management System (AMS)

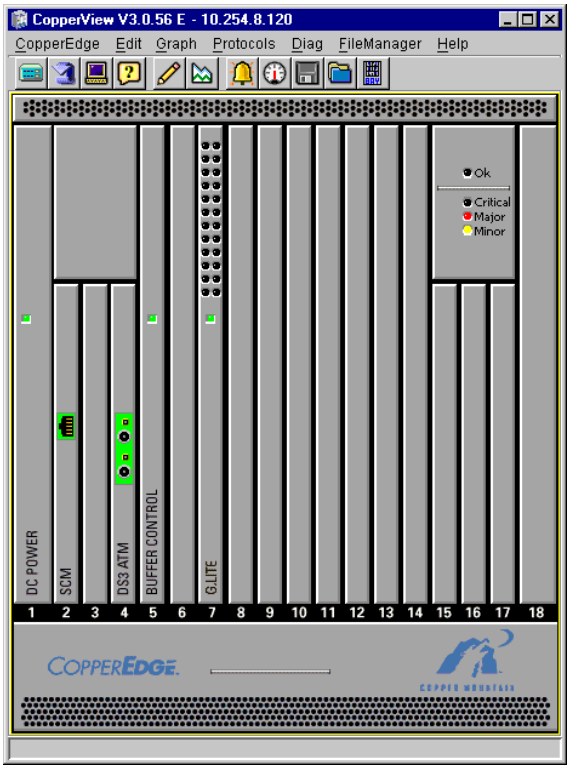
Once the DSLAMs have been divided into management zones, you can do the following:

- Monitor the operation of each zone by quickly glancing at its color coding. You can represent normal operations, warning conditions, and error conditions with user-configurable color codes.
- Control the backup of configurations for DSLAMs by selecting all of the DSLAMs in a zone or by selecting individual ones within the zone.
- Retrieve DSL port statistics on a cumulative basis, or configure the system to automatically collect DSL port statistics at selected intervals.

In addition to offering a global view of all of the CopperEdge DSLAMs in a zone, the CopperView AMS allows you to access each DSLAM using a complementary system called the *CopperView* Element Manager™ (EM).

CopperView EM

The *CopperView* Element Manager (EM) provides a graphical interface to easily control, configure and monitor the CE200. In *Chassis View* the EM displays the front panel of the CE200 you are controlling, including the installed WAN and DSL modules.



The CopperView Element Manager (EM)

By clicking on one of the ports of a DSL module, you are able to configure it for a new customer, look at parameters set for an existing customer, or monitor the traffic that has been going over the interface at selected intervals. You can also set up loopback and troubleshooting tests.

CopperCraft Command Line Interface

The *CopperCraft* command line interface provides the same functions for configuring new customers, displaying parameters already set for existing customers, and monitoring the traffic across interfaces such as the *CopperView* EM. *CopperCraft* also allows you to set up loopback and troubleshooting tests. The CLI is made up of a series of object groups based on the following documents:

- SNMP MIBs of RFC-1213 and RFC-1573 (MIB-II)
- If-MIB of RFC 2863
- Frame Relay MIB of RFC-2115
- DS1 MIB of RFC-2495

- DS3 MIB of RFC-2496
- ADSL Line MIB of RFC-2662 and ADSL Supplemental MIB (draft)
- ATM MIBs of RF-2514 and RF-2515
- SONET MIB of RFC-2558
- HDSL2/SHDSL MIB (ADSL MIB Working Group)
- ATM IMA MIB (ATM Forum af-phy-0086-001)
- Copper Mountain proprietary MIBs

All of the applicable MIBs for any version of the CE200 are provided as part of the corresponding software release package. For more information on standard MIBs, the source documents (RFCs) are freely available from a number of World Wide Web sites. See for example the Internet Encyclopedia web site at http://www.freesoft.org or the web site of the Internet Engineering Task Force (IETF) at http://www.ietf.org

For more information about the proprietary Copper Mountain MIBs, contact the Copper Mountain Technical Assistance Center. All of the object groups making up the CLI are listed alphabetically in Chapter 3.

Using the CopperCraft CLI

When connected to the network, only qualified personnel should be permitted to operate the CE200. To qualify as an operator of the CE200 through the Craft Interface, it's not necessary that you become an expert on the OSI layered network architecture. However, you should have a working knowledge of the various communications protocols on which the MIB groups have been built, as well as a working knowledge of the basic configuration procedures. For instance, familiarity with the following will enable you to become proficient with the CLI:

- Networking models
- IP addressing and subnetting
- Bridging, routing, and switching
- Procedures for creating virtual circuits
- Procedures for configuring SDSL/IDSL ports
- Procedures for configuring ADSL (G.lite or G.dmt) ports
- Procedures for configuring VoDSL networks

Many books address the first two topics. The *CopperEdge 200 Installation and Operating Guide* discusses the last five topics.

Do not attempt to set or modify any of the parameters on a CopperEdge system that is in service if you are unsure of the effect. Either leave the factory default in place, defer the adjustment to a more qualified operator, or contact the Copper Mountain Technical Assistance Center.

Chapter 2
Using the *CopperCraft* Interface

To access the CopperEdge Management Information Base (MIB) from the CopperCraft command line interface (CLI), you must use a text-based command language that simulates the SNMP machine protocol. But while standard SNMP consists of five message types (Set, Get, Getnext, Get Response, and Trap), the CopperCraft CLI only uses only the Set, Get, and Getnext commands. In addition, the CLI has four other commands: Getall, Help, Find, and Ping.

The CopperCraft commands allow you to configure and manage a CE200 system, either through a Telnet session or with a directly connected terminal. Besides the basic commands, however, you should become familiar with these additional features to make your use of the control and monitor functions easier and more efficient:

- the Line Editor
- the Permanent Interface Identifier
- SNMP command structure
- CopperCraft Help
- Command object alias
- the truncate function

CopperCraft Line Editor

Command strings for SNMP object groups can be very short or very long, especially when you are setting a series of new parameters in an object group. Here is an example of a short Getall command, which returns data about all of the boards in a CE200 chassis:

```
getall cmboard
```

Here is an example of a longer Set command, which sets the parameters on an ATM quality of service table:

```
set cmcircuitparam [6] rowstatus=createandgo
       servicecategory=nrtvbr pcr=104000 scr=604
       mbs=4 cdv=unspecified
```

The line editor function on the CopperCraft CLI lets you recall as many as 20 previous commands from the system's memory buffer. You can move through a multi-line command and edit it before you press Enter and send the command to the system.

To display the list of valid Line Editor commands, press **ESC ?** at the system prompt.

```
^D            Delete current character
DEL           Delete current character
^H            Backspace
^E            End of line
^A            Beginning of line
ESC F         Forward one word
ESC B         Backward one word
^K            Delete to end of line
^U            Delete to beginning of line
^U^K          Delete entire line
^L            Redisplay current line
Left Arrow    Cursor left
Right Arrow   Cursor right
Up Arrow      Scroll up through history
^P            Scroll up through history
Down Arrow    Scroll down through history
^N            Scroll down through history
```

Control characters are shown with a caret (^). Escape sequences are indicated by the ESC prefix.

If you are using a terminal emulator, set it to VT100 mode. If a full command will not fit on a single line, an arrow character (< or >) appears at either end of the line indicating Scroll mode. Use the arrow keys to move to the end of the line; the next 10 characters will be displayed.

Permanent Interface Identifier (PII)

Fully populated CE200s can have over 200 physical interfaces, including DSL ports, WAN ports, and Ethernet ports. They can have a much larger number of virtual circuits (up to 976). All ports and circuits must be uniquely identifiable so they can be individually configured and acted upon. To organize and keep track of these hundreds of interfaces, the CopperEdge uses a system of P*ermanent Interface Identifiers* (PIIs).

The PII consists of up to four elements, listed serially in a specific order: c.s.p.v, where:

c = Chassis (shelf) number

s = Slot number (1 to 18) where the target module resides (see table below)

p = Port number of the target interface

v = Virtual circuit number (when used)

Slot No.	CE200 Module
1	DC Power Module
2	System Control Module
3-4	V.35, ATM, Frame Relay, Quad T1, OC-3c/STM-1 (Single or Multimode), T1/E1 IMA
5	Buffer Control Module
6-13	SDSL, IDSL, ADSL (G.dmt and G.lite), G.SHDSL, or DS1/DSL Modules
14	Redundant Buffer Control Module
15	Redundant System Control Module
16-17	Redundant WAN Modules
18	DC Power Module

Ports in WAN modules are numbered top to bottom, 1 to *n*, depending on the number of physical ports on the module.

Examples:

- A PII entered as 1.2.1 identifies the physical interface at the CE200 in shelf 1, the System Control Module in slot 2, and port 1 (the *Net 1* Ethernet port).
- A PII such as 1.3.1.24 would refer to the CE200 chassis 1, the WAN module in slot 3, port 1, and virtual circuit 24.
- A PII such as 1.7.24 identifies the CE200 chassis 1, the DSL module in slot 7, and physical port 24.

If you are using an SNMP manager to configure the CE200, you may be required to use a slightly different PII format. In that case, the identifier is entered without periods to separate the segments, and the entire PII must be filled, using leading zeros as necessary.

Example:

To enter a PII of 1.3.1.24 using the SNMP manager format, you would type: `103010024`

SNMP Command Structure

Command strings directed to SNMP object groups must contain all of their required identity elements, and the elements must be presented in the correct order. If any of the elements are missing or out of order, the CE200 will not be able to process the command, and will return an error message.

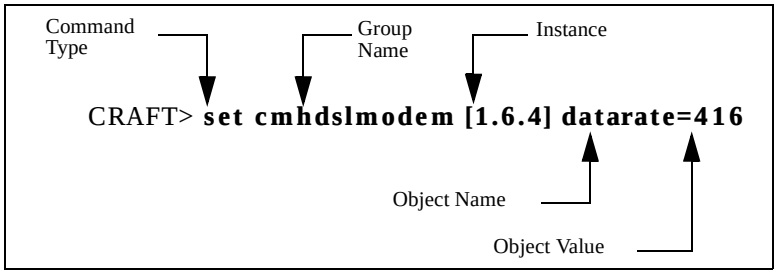
Command strings for SNMP object groups must include the following two elements:

- a command type
- an object group name

In addition, command strings may require:

- an instance (see below)
- one or more object names
- an object value for each object name

In the following example of a typical Set command, a DSL port is being configured with a data rate of 416 Mbps. The port is the fourth one on the module in slot 6 slot of the chassis.



Example of a Set Command

When you enter a command, the elements must follow the order and the rules listed below:

- Command type: *set*, *get*, *getnext*, *getall*, *help*, *find*, *ping*.
- Always enter the command type first, separated from the group name by a space.
- Group name: *cmSystem*, *cmIface*, *cmADSLModem*, etc.
- Always enter a group name for the Set, Get, Getnext, Getall, and Find commands. Group names are listed alphabetically in Chapter 3.
- Any Instance required by this group: *IP address*, *PII*, etc.
- The term instance refers to all of the index items that uniquely identify a row in a table in the MIB database. If a group has more than one index, all of them are required to form the instance.
- The instance, when required, is enclosed in brackets [like these]. If the entry requires more than one index, enter them in the same order as displayed in the *help objects* listing or in this guide. Multiple indexes are separated with a comma, but enclosed in a common set of brackets: [index1, index2].
- Object name: *NetModel*, *IpAddr*, *NetMask*, etc.
- Enter an object name. Separate the object name from the instance by a space. Entering one or more object names in a *Get* command is useful if you want to see specific objects rather than all objects in a group.
- Object value: An object name must be accompanied by a value when performing a *Set* command.

In the command string, indicate an object's value with an equal sign (=) between the name and the value, as: **netmodel=coppervpn**. Do *not* use spaces between object names and object values.

In one Set command, you can enter multiple objects with their values. Separate each object/value with spaces, as shown:

```
CRAFT> set cmiface [1.6.1] netmodel=vwan destpii=1.3.1.500
       encapsulationtype=rfc1483 cmcpcompatible=yes
```

You can enter multiple *Get* commands on the same line by separating them with a semicolon, as in this example:

```
CRAFT> get cmiface [1.6.4]; get iftable [1.3.1.109]
```

If you mistakenly issue a command that is inconsistent with the physical configuration of the CE200 (such as a request to view a frame-relay parameter at an ATM interface), the system may return unexpected or inconsistent messages.

Coppercraft Help Function

The CopperCraft CLI has an extensive Help function to aid in configuring and monitoring the CE200. The Help function has three levels:

- A listing of the basic commands on the CLI.
- A listing of all of the object group names, with any aliases, recognized by the CLI.
- A listing of all of the objects in each group, some with short descriptions.

Basic Help Commands

While the most frequently used Craft commands are *Set* and *Get*, a 12 different commands are actually available. If you enter the word *help*, the following list displays:

```
help
help alarms
help ChangeSCMType
help elog
help find
help get
help getall
help getnext
help GROUPS
help LCRestart
help OBJECTS group_name
help SCMRestart
help set
```

Descriptions of each of the 12 commands are available through the Help utility. Each description lists the proper syntax, and a brief explanation of the command.

For example, entering the command, *help lcrestart*, returns the following information.

```
help lcrestart
SYNTAX:  LCRestart shelf.slot
  Restarts the LC card in the given shelf and slot.
  The operator must type the whole command.
```

Listing of Help Object Groups

To list the names of all of the available object groups, enter the command **help groups**. The names are listed alphabetically and include any recognized aliases for the group names (in parentheses).

```
help groups
adslAtucChanPerfDataTable ADSL ATUC Channel
            Performance Statistics (atucch)
adslAtucPerfDataExtTable ADSL ATUC Extended
Performance Statistics (atucpfext)
adslAtucPerfDataTable    ADSL ATUC Performance
                         Statistics (atucpf)
adslAtucPhysTable        ADSL ATUC Physical Layer
                         Parameters (atucph)
...etc....
```

Description of Help Object Groups

Descriptions of the objects contained within each MIB table are available through the Help function. Each description lists the full names of the objects in the group and how they are used. It also indicates whether they are read-only (RO) or read-write (RW), and, if writable, lists the available options or ranges for the value. In the listing of objects, an asterisk (*) precedes any index object(s) for the group.

Example:

```
help cmcircuitparam
Object           Description
                    Access, Type, Format
-------------------------------------------------
*Index           Index. Range 1 .. 255
                    RO, Decimal Integer
RowStatus        Status of this row
                    RW, Enumerated type
                     VALUES: Active, CreateAndGo, Destroy
ServiceCategory  Service Category
                    RW, Enumerated type
                       VALUES: rtVBR, nrtVBR, ubr
PCR              Peak Cell Rate in cells/s
                    RW, Decimal Integer
```

```
SCR             Sustained Cell Rate in cells/s
                      RW, Decimal Integer

MBS             Maximum Burst Size in # of cells
                      RW, Decimal Integer

CDV             Cell Delay Variation
                      RW, Enumerated type

                VALUES: unspecified, minimum,
                      nominal
```

Other CopperCraft Commands

Besides Help, additional *CopperCraft* commands include: *Set*, *Get*, *Getall*, *Getnext*, and *Find*. A Ping utility, and various restart and logging options are also supported via the *CopperCraft* interface.

Set

The Set command allows you to assign the values for specified objects in a scalar or table group, such as the maximum and minimum transmit and receive rates for *cmADSLModem*.

The basic syntax is:

```
set mib_group [instance] object1=value1
       object2=value2
```

Get

The *Get* command allows you to display information about a specific object group, or up to five specific objects within a group.

The basic syntax to see all objects in the group is:

```
get mib_group [instance]
```

The basic syntax to see only specific objects in the group is:

```
get mib_group [instance] object1...object5
```

Getall

Getall allows you to review the status of all objects for all instances of a specified group at the same time. The output may be a condensed version of the output for a *Get* command.

The basic syntax is:

```
getall mib_group
```

Since the list could have hundreds of entries, the screen will fill up and stop at four or five entries. To view additional entries, press any key except Esc. Continue paging until you find the entries you want or until you reach the end of the list. To quit the listing at any time, press Esc.

You can refine a Getall query by specifying up to five objects on a single command line:

```
getall mib_group object1...object5
```

You can specify an instance to see a list of all objects starting with that instance:

```
getall mib_group [instance]
```

You can also specify an instance and up to five objects on the command line:

getall *mib_group* [*instance*] *object1...object5*

Examples:

To review the configuration of interfaces in the CE200, including all WAN, DSL, and VC interfaces, enter the following command:

getall cmiface

The system will list all of the instances, starting with the System Control Module in slot 2.

To list all instances of cmIface with their corresponding IfIndex numbers and GroupNames, enter the following command:

getall cmiface ifindex groupname

To list all interfaces with a PII of 1.8.1 or higher, with their IP addresses and net masks, enter the following command:

getall cmiface [1.8.1] ipaddr netmask

Getnext

The Getnext command lists all objects for the *next* instance of the specified group:

getnext *mib_group*

As with the Get command, you can specify up to five specific objects in the Getnext command:

getnext *mib_group object1...object5*

Getnext can also be used after a Getall command in which you specified an instance. If you press Esc to quit the Getall listing and last instance displayed is 1.3.1.16, the Getnext command displays all objects for the next instance, 1.3.1.17.

Examples:

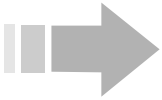
If you issue a Get command to look at the configuration of a specific WAN VC using cmIface, such as 1.3.1.16:

get cmiface [1.3.1.16]

and then you want to see similar information about several VCs following VC 16. The Getnext command will display the complete cmIface table for VC 17 (or the next assigned VC number).

getnext cmiface

Subsequent Getnext commands would display the complete cmIface tables for VC 18, VC 19, and so on.



N O T E

The Getall command truncates the field description response if it is 20 characters or more on length. Thus, the response to a getall cmBoard query may return the info that slot five is a BufferControlModule but omit an important bit of information from the full description (BufferControlModule2).

Find

The *find* command provides an easy way of selectively querying the database based on common criteria. For example, you can compile lists of interfaces for use in troubleshooting and record-keeping. There are a number of ways you can use the find command as shown in the following examples.

To specify up to three objects and their values for all instances:

```
find mib_group object1=value1...object3=value3
```

To find values of 1 to 3 objects starting with a specific instance:

```
find mib_group [instance] object1=value1...
       object3=value3
```

You can include up to five additional object names after the three objects and values to refine the output. The system will return only the values for the instance and the object names for those entries that matched the specified criteria.

To specify up to three objects/values for all instances, and display up to five object names, the basic syntax is:

```
find mib_group object1=value1...object3=value3
       object_name1...object_name5
```

To specify up to three objects/values starting with a specific instance, and display up to five object names, the basic syntax is:

```
find mib_group [instance] object1=value1...
       object3=value3 object_name1...object_name5
```

Find commands are not boolean expressions, but a command with multiple selection criteria is a fixed logical AND expression; every row that matches the first, AND the second, AND the third criterion is listed in response to the query.

Examples:

To list all DSL interfaces routed to a specific DestPII, you must enter a specific value for DestPII:

```
find cmiface destpii=1.3.1.39
```

In this case, the CE200 will return the entire cmIface table for each PII that matched the criterion (destpii=1.3.1.39).

To list the DSL interfaces that match specified selection criteria for instance 1.8.2 and higher, enter the following:

```
find cmiface [1.8.2] netmodel=ip
       encapsulationtype=rfc1483 CMCPCompatible=yes
```

In this case, the CE200 will return the entire cmIface table for each PII starting with 1.8.2, that matched the criteria (netmod=ip encap=rfc1483 CMCPCompatible=yes).

To list only the DestPIIs for the DSL interfaces in the previous example, enter the object name after the objects and values:

```
find cmiface [1.8.2] netmodel=ip
       encapsulationtype=rfc1483 CMCPCompatible=yes
       destpii
```

Ping

From the CopperCraft interface, you can *ping* external IP addresses—but only through the CopperEdge's IP stack; not to VWAN, CrossConnect, HDIA, or CopperVPN networks or destinations.

The syntax is:

ping *ip_address*

By default, the target IP address will be pinged four times. You can also specify the ping count and the delay between pings in milliseconds (min=100, max=5000). For example, to ping a certain IP address 6 times with 1000 milliseconds between each ping:

```
ping 206.14.101.24 6 1000
```

To end the ping process at any time, press any key.

The ping utility is only accessible through CopperCraft; it is not supported in SNMP.

Shortcuts and Special Cases

The CopperCraft interface has a number of features built into it for using special characters and alternate commands.

Text Entries

Generally, the CopperCraft CLI is not case-sensitive, but you should type in directory paths in lower case letters. Radius authentication, and all operator names and passwords for Radius, are case-sensitive. In contrast, operator names on the CE200 are not case-sensitive, but, passwords on the CE200, with the exception of the first letter, *are case-sensitive.*

You will note that, throughout this document, many of the group and object names are presented with one or more uppercase letters within the object or group name (e.g. FarEndAddr). Such mixed upper-case lower-case style is intended to make the string easier to read and understand at a glance. You need not use the mixed upper-/lowercase style when entering commands.

When entering text, character strings cannot contain backslash (\), angle brackets (< and >), or apostrophe (‘) characters. Also, do not use the pound sign (#). It will be interpreted as a comment and the rest of the command will be ignored.

Truncated Commands

Any CopperCraft command word, and any MIB group or object name can be truncated up to the point of its shortest unique string. For example, instead of entering the full command:

```
getall cmfrcircuit
```

you can enter the short forms of the command and group name:

```
geta cmfrc
```

Short forms for all supported MIB groups are listed in a quick reference table on page 22.

Aliases

Many MIB groups and objects also have a unique alias that the system will recognize. For example, instead of entering this full command:

```
geta cmadslmodem
```

You can enter the alias:

```
geta amod
```

Group and object aliases are listed in the Online help, below their full names in the Summary Description tables listed in Chapter 3, and in a quick reference table on page 22.

Related Object Groups in the MIB

By convention in this volume, all of the MIB groups are presented in alphabetical order. Many of the groups containing similar functions have similar names, and therefore they appear together on the list. However, not all groups that have similar or related functions are adjacent to each other. The names of many object groups include prefixes that denote their source MIBs, such as ATM, DSX1, Sonet, etc.

For example, object groups pertaining to ADSL (G.lite or G.dmt) that are contained in the ADSL MIB have names like adslAtucPhysTable and adslAturPhys Table. But other groups that deal with ADSL functions have names like cmAdslModem and cmAdslPerf. Even though their functions are related, the groups are separated alphabetically. Some of these related groups include:

- *Trap* and *Alarm* groups
- *DSL* groups for monitoring performance on DSL ports
- *ADSL* groups for configuring and monitoring ADSL (g.lite and g.dmt) groups
- *Dsx1* groups for monitoring performance on DSX1 WAN ports
- *Dsx3* groups for monitoring performance on DS3 WAN ports
- *Sonet* groups for monitoring performance on OC-3c/ STM-1 ports
- Other groups which have functionally relatives include H*DSL2/GSHDSL, IMA, IMUX (Bundle* and *Endpoint* groups) and *Loopback*.

cmAlarm Groups (Performance Thresholds)

The two alarm groups, cmAlarm and cmAlarmTable, neither generate nor report on alarms on the CE200. Rather, they allow you to configure the conditions under which a special kind of trap event, called a *Threshold Crossing Alert* (*TCA),* is generated.

TCAs are diagnostic tools, useful for reporting on specific areas of data performance on various types of DSL and WAN ports. If the incidents you are interested in (such as a worrisome level of transmit underruns on a particular DSL port) exceed the number you configure as a threshold, the CE200 will generate a TCA and then it will send it to the configured trap destinations.

The cmAlarm and cmAlarmTable groups are modeled on similar groups in the Remote Monitor (RMON) MIB. The objects in cmAlarm allow you to set the sampling intervals and then turn on the TCA function. The objects in cmAlarmTable allow you to select the objects to be monitored and also to set both rising and falling thresholds. They also allow you to specify the PII on which the monitoring for each object will be done.

Sampling Intervals

By default, the performance-monitoring interval for all cmDSL, dsx1, and dsx3 groups is set to 15 minutes. The interval is configurable through the cmAlarm SamplingInterval object for 5, 15, or 60 minutes. *[Note that the default sampling interval for Sonet groups is also 15 minutes, but in the current release that value is fixed and cannot be otherwise configured.]*

If a configurable interval is set for a period of less than 15 minutes, the groups that normally collect and display a full day's performance statistics will only display 96 intervals (96 times the number of minutes set in the SamplingInterval object). For example, if the interval is set to 5 minutes, statistics will only be displayed for 8 hours (96 intervals of 5 minutes each).

In commands directed to CPE units, the interval period is fixed at one hour, except for the cmCpeDataPortInterval group, which uses a 15-minute interval.

cmDSL Groups

The six object groups in this category display data on traffic over the DSL ports on the CE200. These groups apply only to physical ports, not logical ones. IMUX bundles, for example, are not supported. Certain IMUX bundle performance indicators are available through the IfTable.

All of the cmDSL groups display information about the data performance at the specified port, and all six have nearly identical objects. The assortment of similar groups allows you to easily compare performance using different time snapshots.

- *cmDSLInterval*—Displays information about a DSL port for a specified 15-minute interval in the past 24 hours.
- *cmDSLCurrent*—Displays information about a specified DSL port during the current (in-progress) 15-minute interval.
- *cmDSLTodays*—Displays statistics for the specified DSL port from the midnight just past, up to the end of the most recently completed interval.

- *cmDSLUpTime*—Displays cumulative total of statistics for the specified DSL port since the previous board reset/restart.
- *cmDSLYesterdays*—Displays the total for each parameter for the 24-hour period from midnight yesterday, up to the midnight just past.
- *cmDSL24Hr*—Displays cumulative totals for each parameter for the full 24 hours preceding the current interval.

By default, the performance-monitoring Interval for all cmDSL groups is 15 minutes. The Interval is configurable through the cmAlarm SamplingInterval object for 5, 15, or 60 minutes. For more information, see "Sampling Intervals" on page 16.

ADSL Groups

Groups in this category apply to both G.dmt modules and G.lite modules. The objects not only display data on traffic over the DSL ports on the CE200, but also allow you to configure the ports, setting upstream and downstream rates, line types, overhead rates, and interleave delay. The groups are:

- adslAtucChanPerfDataTable
- adslAtucChanPerfDataExtTable
- adslAtucChanPerfDataTable
- adslAtucPerfDataTable
- adslAtucPhysTable
- adslAturPerfDataExtTable
- adslAturPhysTable
- cmAdslChanTable
- cmAdslModemTable
- cmAdslPerfTable

Dsx1 Groups

Groups in this category display data on traffic over the "Quad T1" WAN, DS1/E1 IMA or DS1 DSL interfaces on the CE200.

- *Dsx1Config*—Allows you to configure the DS1 interface, indicating line type, line coding, line length, transmit clock source, loopback status, and other parameters.
- *Dsx1CurrentTable, dsx1IntervalTable, dsx1TotalTable*—Allow you to monitor the status of DS1 links to and from the CE200. The only difference between the groups is the time interval.
- *Dsx1FarEndCurrent* Table, *dsx1FarEndIntervalTable, dsx1FarEndTotalTable*—Allow you to monitor the DS1 link. The only difference between the groups is the time interval.

Far-end statistics from the dsx1FarEnd groups are available only for lineType=ClearChannel. For M23Multiplex lines, the displayed response (all zeros) does not represent actual data.

By default, the performance-monitoring interval for all cmDsx1 groups is 15 minutes. The interval is configurable through the cmAlarm SamplingInterval object for 5, 15, or 60 minutes. For more information, see “Sampling Intervals” on page 16.

Dsx3 Groups

The seven object groups in this category display data on traffic over the DS3 (or WAN) ports on the CE200.

- *Dsx3Config*—Allows you to configure the DS3 interface, indicating line type, line coding, line length, transmit clock source, loopback status, and other parameters.
- *Ddsx3CurrentTable*, *dsx3IntervalTable*, *dsx3TotalTable*—Allow you to monitor the status of DS3 WAN links to and from the CE200. The only difference between the groups is the time interval.
- *Dsx3FarEndCurrent* Table, *dsx3FarEndIntervalTable*, *dsx3FarEndTotalTable*—Allow you to monitor the DS3 WAN link. The only difference between the groups is the time interval.

Far-end statistics from the dsx3FarEnd groups are available only for lineType=ClearChannel. For M23Multiplex lines, the displayed response (all zeros) does not represent actual data.

By default, the performance-monitoring Interval for all cmDsx3 groups is 15 minutes. The Interval is configurable through the cmAlarm SamplingInterval object for 5, 15, or 60 minutes. For more information, see “Sampling Intervals” on page 16.

Sonet Groups

The 12 object groups in this category display performance statistics for the specified OC-3c/STM-1 interfaces at both the near and far ends of the Sonet line, path and section, for both the current and specified 15-minute monitoring interval.

Two other groups, *SonetMediumTable*, and *SonetSESThreshSet* allow you to set and display configuration information and error thresholds.

Loopback Groups

The loopback groups are diagnostic tools that allow you to test either a WAN or DSL interface or a WAN or DSL line, verifying its integrity and performance. Several different groups are available:

- the cmLoop and cmLoopHist groups
- the cmSDSLTest group
- the dsx1Config and dsx3Config groups

The cmLoop group allows you to perform tests on either SDSL and IDSL ports and lines. The cmLoopHist group stores the results of loopback tests and allows them to be displayed.

The cmSDSLTest group allows you to collect and view statistics on an SDSL line.

The dsx1Config and dsx3Config groups allow you to perform various of DS1 and DS3 circuits.

cmTrap Groups

A number of groups deal with *events* and *alarms*, all of which are captured as SNMP *traps*. The groups are:

- cmTrapAlarmTable
- cmTrapDestinationTable
- cmTrapEventConfigTable
- cmTrapEventFilterTable
- cmTrapEventTable
- cmTrapFilteredLogTable
- cmTrapSummary
- cmTrapTrapFilterTable
- cmTrapTypeTable

All of the cmTrap groups deal with three basic categories of events:

- Events that cause an alarm to be recorded
- Events that cause an alarm to clear
- Typical operating events, such as logins, logouts, status changes, etc.

For details about CopperEdge events and alarms, see the companion Installation and Operating Guide volume to this document.

Operating Status

Objects that describe the operating status of the elements in the CE200 (systems, boards, ports, links) appear in several different object groups.

Elements	Description	Object Groups
System	A system is always Enabled.	Displayed in cmSystem and cmState as *OperState*.
Shelf	A shelf is always Enabled.	Displayed in cmShelf as *Operability*. Displayed in cmState as *OperState*
Board	A board is Enabled if any port is enabled; otherwise a board is Disabled.	Displayed in cmBoard and cmState as *OperState*.
Port	A port is a physical connection to an entity outside of a shelf (DSL ports and WAN ports). A port is Enabled if connected with an external device. It is Disabled if not connected to an external device. For instance, a DSL port is Disabled if not trained with its CPE. A port is Testing if it is running a loopback test.	Displayed in cmHDSLModem, cmIDSLModem, cmV35, and cmState as *OperState*. Displayed in ifTable as *OperStatus*.
Link	A link is a logical connection for exchanging data between the CE200 and an external device. Both a DSL and a WAN port can provide multiple links (PVCs). A link is Enabled if its port or VC is enabled and is able to pass data between the CE200 and a far-end device. It is Testing if its interface is currently running a loopback test; otherwise its link is Disabled.	Displayed in cmAtmVcl and ifTable as *OperStatus*. Displayed in cmFrCircuit, cmFrDlcmi, and cmState as *OperState*.
	Two MIB II Frame Relay tables are special cases: • *frCircuit* — Configuring a frame relay DLCI in this group creates an entry in the cmState table with an ObjClass (specific class) of FrameRelayPVC. The circuit is enabled if the DLCI is active; otherwise it is disabled. Testing is not a valid state for PVCs. • *frDLCMI* — Although entries in the frDLCMI table have the ObjClass (specific class) of FrameRelayLink, they are not links. Rather, they describe the LMI status of the frame relay service provided at a WAN port.	Displayed in frCircuit as *State*. Displayed in frDLCMI as *State*.
CPE	A CPE is enabled if it can pass data. Otherwise, the CPE is disabled.	Displayed in cmCpeBoard, cmCpeDataPort, cmCpeHDSLModem, cmCpeIDSLModem, and cmCpeVoicePort as *OperState*.
Support	A support element is a Power Module or a Fan Assembly. It is Enabled if it is plugged into the shelf and operating within its normal parameters. It is Disabled if it is plugged in but not operating normally (that is, if it is bypassed). If a Power Module is absent, it does not appear in the cmState table. A disconnected Fan Assembly shows up as Enabled in cmState OperState.	Displayed in cmState as *OperState*.

Chapter 3
CopperCraft MIB Definitions

This chapter consists of a reference listing of the SNMP objects (MIB definitions) that can be used to configure and manage the CE200. The groups are derived from the following Management Information Base documents:

- SNMP MIBs of RFC-1213, RFC 2863 (IF MIB), and RFC-1573 (MIB-II).
- Frame Relay MIB of RFC-2115. Groups from this MIB all begin with *fr*.
- DS1 MIB of RFC-2495. Groups from this MIB begin with *dsx1*.
- DS3 MIB of RFC-2496. Groups from this MIB begin with *dsx3*.
- SONET MIB of RFC2558. Groups from this MIB begin with *Sonet*.
- ADSL Line MIB of RFC-2662 and ADSL Supplemental MIB (draft). Groups from these MIBs begin with *adsl*.
- HDSL2/GSHDSL MIB (ADSL Working Group)
- IMA MIB (ATM Forum af-phy-0086-001)
- ATM MIBs of RF-2514 and RF-2515. Groups from this MIB begin with *atm*.
- Copper Mountain proprietary MIBs. These groups begin with *cm*.

Support for specific MIB groups and their objects will vary for each version of the CopperEdge software. In general, the changes are the result of incorporating additional features and functions in the CE200. So the version number on your reference guide should always match the version number of the software in your CE200. Note also that neither the existence of a group or object in the software, nor its listing in this guide should be considered as necessarily indicating that it is *supported* (i.e., fully tested and warranted) in the software. If in doubt, check the Release Notes for the software version you are using and/or contact Copper Mountain Customer Service.

Command Object Groups

Each listing for the MIB object groups includes the following information: the group name, its alias (if available), its description, the command syntax, a sample command, and a table describing the objects in the group. This information is provided in the following format.

Sample_Group_Name
(alias: xxxxx)

A description of the MIB object group and any other relevant information.

Syntax: The syntax of the command.

Example: One or more sample commands, including a Get command, and, if the object group has a writable object, a Set command.

The sample commands show the shortest form of the group name, either the alias or the truncated form.

For a quick reference list of the MIB groups with their truncated forms and aliases, see the table that follows this section.

Note: Important information about using this group.

Objects:

RO MIB_Object alias: xxxxx	Read-only objects that cannot be set or changed are shaded.
RW MIB_Object alias: xxxxx	Read-write objects that can be set or changed are not shaded. The description will usually provide the range of values for numeric input, or the list of valid options for alphabetic input. It may also give the default value, if applicable.

MIB Group Names and Shortcuts

Here is a quick reference of each MIB group name, its short (truncated) form, and its alias. The examples in the MIB definitions on the following pages show the shortest form of the group name.

MIB Group Name	Short Form	Alias
adslAtucChanPerfDataTable	adslatucc	atucch
adslAtucPerfDataExtTable	adslatucperfdatae	atucpfext
adslAtucPerfDataTable	adslatucperfdatat	atucpf
adslAtucPhysTable	adslatucph	atucph
adslAturPerfDataExtTable	adslaturpe	aturpfext
adslAturPhysTable	adslaturph	aturph
atmInterfaceConfTable	atmi	
atmMIBObjects	atmm	
atTable	att	

MIB Group Name	Short Form	Alias
cmActiveSessionTable	cmac	
cmADSLChanTable	cmadslc	achan
cmADSLModemTable	cmadslm	amod
cmADSLPerfTable	cmadslp	aper
cmAlarm	cmalarm	
cmAlarmTable	cmalarmt	cmon
cmAtmIfExtTable	cmatmi	
cmAtmPerfTable	cmatmperft	
cmAtmPerfIntervalTable	cmatmperfi	
cmAtmVclTable	cmatmv	
cmBoardTable	cmbo	
cmBundleTable	cmbu	
cmCircuitParamTable	cmci	
cmConnect	cmco	
cmCPEBoardTable	cmcpeb	cpeb
cmCpeDataPortIntervalTable	cmcpedataporti	cpedint
cmCpeDataPortTable	cmcpedataportta	cped
cmCpeDataPortTotalTable	cmcpedataportto	cpedtot
cmCpeDSLIntervalTable	cmcpeds	cmhdslperf
cmCpeEthernetIntervalTable	cmcpee	cmethernet
cmCpeGshdslModemTable	cmcpeg	cpeshmod
cmCpeHDSLModemTable	cmcpeh	
cmCpeIADTable	cmcpeia	cpeiad
cmCpeIDSLModemTable	cmcpeid	cpeidsl
cmCpeLogTable	cmcpel	
cmCpePlugAndPlayTable	cmcpep	cpepnp
cmCpeT1InterfaceTable	cmcpet1	cpet1
cmCpeTollBridgeTable	cmcpeto	cpetoll
cmCpeVoicePortIntervalTable	cmcpevoiceporti	cpevint
cmCpeVoicePortTable	cmcpevoiceportta	cpev
cmCpeVoicePortTotalTable	cmcpevoiceportto	cpevtot
cmDHCPTable	cmdh	
cmDS3AtmTable	cmds3	
cmDSL24HrTable	cmdsl2	dsl24
cmDSLCurrentTable	cmdslc	dslcur
cmDSLIntervalTable	cmdsli	dslint
cmDSLTodaysTable	cmdslt	dsltod
cmDSLUpTimeTable	cmdslu	dslup
cmDSLYesterdaysTable	cmdsly	dslyes

MIB Group Name	Short Form	Alias
cmDsx1ConfigTable	cmdsx	cmds1cfg
cmEndPointConfigTable	cmendpointc	epconfig
cmEndPointPortTable	cmendpointp	epport
cmFile	cmfile	
cmFilterTable	cmfilt	
cmFrCircuitTable	cmfrc	
cmFrDlcmiTable	cmfrd	
cmFrf5DlcmiTable	cmfrf	
cmGroupFilterTable	cmgroupf	
cmGroupTable	cmgroupt	cmgrp
cmGshdslModemTable	cmgs	shmod
cmHDSLModemTable	cmh	
cmIDSLBoardTable	cmidslb	
cmIDSLModemTable	cmidslm	
cmIfaceTable	cmif	
cmInterfaceExtTable	cminterfacee	
cmInterfaceOptionsTable	cminterfaceo	
cmLoop	cmloop	
cmLoopHistTable	cmlooph	
cmMacTable	cmmac	
cmMaint	cmmaint	
cmMaintCmdTable	cmmaintc	
cmMemberTable	cmme	
cmNameTable	cmn	
cmOperatorTable	cmo	
cmParamSummaryTable	cmpa	
cmProxyArpTable	cmpr	
cmRadius	cmr	
cmSDSLTestTable	cmsd	
cmServiceClass	cmse	sclass
cmShelfTable	cmsh	cmchassis
cmSonetSdhTable	cmso	cmsnt
cmStateTable	cmst	
cmSubIfaceTable	cmsu	
cmSystem	cmsy	
cmTracePii	cmtrac	
cmTrapAlarmTable	cmtrapa	tralarm
cmTrapDestinationTable	cmtrapd	trdest
cmTrapEventConfigTable	cmtrapeventc	treconfig

MIB Group Name	Short Form	Alias
cmTrapEventFilterTable	cmtrapeventf	trefilter
cmTrapEventTable	cmtrapeventt	trevent
cmTrapFilteredLogTable	cmtrapf	trfiltered
cmTrapSummary	cmtraps	trsum
cmTrapTrapFilterTable	cmtraptr	trtfilter
cmTrapTypeTable	cmtrapty	trtype
cmTrunkConnectionTable	*Not supported in this release.*	
cmTrunkIfTable	*Not supported in this release.*	
cmTrunkTopoTable	*Not supported in this release.*	
cmV35Table	cmv3	
cmVbridgeTable	cmvb	
cmVpnGroupTable	cmvpng	
cmVpnRouteTable	cmvpnr	
cmVpnSubnetTable	cmvpns	
cmWANMemberTable	cmw	
dsx1ConfigTable	dsx1co	ds1cfg
dsx1CurrentTable	dsx1cu	ds1ncur
dsx1FarEndCurrentTable	dsx1farendc	ds1fcur
dsx1FarEndIntervalTable	dsx1farendi	ds1fint
dsx1FarEndTotalTable	dsx1farendt	ds1ftot
dsx1IntervalTable	dsx1i	ds1nint
dsx1TotalTable	dsx1t	ds1ntot
dsx3ConfigTable	dsx3co	ds3cfg
dsx3CurrentTable	dsx3cu	ds3ncur
dsx3FarEndCurrentTable	dsx3farendc	ds3fcur
dsx3FarEndIntervalTable	dsx3farendi	ds3fint
dsx3FarEndTotalTable	dsx3farendt	ds3ftot
dsx3IntervalTable	dsx3i	ds3nint
dsx3TotalTable	dsx3t	ds3ntot
frCircuitTable	frc	frpvc
frDlcmiTable	frd	frlink
frErrTable	fre	
frGlobals	frg	
frSystem	frs	
hdsl2Shdsl15MinTable	hdsl2shdsl15	shint
hdsl2Shdsl1DayTable	hdsl2shdsl1d	sh1dint
hdsl2ShdslEndPointCurrTable	hdsl2shdslendpointc	shcurr
hdsl2ShdslEndPointMaintTable	hdsl2shdslendpointm	shmaint
hdsl2ShdslInventoryTable	hdsl2shdsli	shinv

MIB Group Name	Short Form	Alias
hdsl2ShdslUnitMaintTable	hdsl2shdslu	shumaint
icmp	ic	
if	if	
ifStackTable	ifs	
ifTable	ift	
ifXTable	ifx	
imaGroupMappingTable	imagroupm	
imaGroupNumber	imagroupn	
imaGroupTable	imagroupt	
imaLinkTable	imalinkt	
ip	ip	
ipAddrTable	ipa	
IPNetToMediaTable	ipn	
IpRouteTable	ipr	
Snmp	sn	
SonetFarEndLineCurrentTable	sonetfarendlinec	sntflcur
SonetFarEndLineIntervalTable	sonetfarendlinei	sntflint
SonetFarEndPathCurrentTable	sonetfarendpathc	sntfpcur
SonetFarEndPathIntervalTable	sonetfarendpathi	sntfpint
SonetLineCurrentTable	sonetlinec	sntnlcur
SonetLineIntervalTable	sonetlinei	sntnlint
SonetMediumTable	sonetm	sntmed
SonetPathCurrentTable	sonetpathc	sntnpcur
SonetPathIntervalTable	sonetpathi	sntnpint
SonetSectionCurrentTable	sonetsectionc	sntnscur
SonetSectionIntervalTable	sonetsectioni	sntnsint
SonetSESThreshSet	sonetses	sntthresh
System	sy	
Tcp	tcp	
TcpConnTable	tcpc	
Udp	udp	
UdpTable	udpt	

adslAtucChanPerfDataTable
(alias: atucch)

This group displays received and transmitted block statistics for a near-end or central office channel on an ADSL port (G.lite or G.dmt). Since ADSL modules on the CE200 currently support only a single channel, you do not need to specify a channel.

You can see how many blocks have been received and transmitted over the port since the ADSL module was reset. You can also see how many corrected and uncorrectable blocks have been received on the port.

Syntax: *command* adslatucchanperfdata *[ifindex]*

You only need to specify the PII of the ADSL port.

Example: To query port 3, slot 8 for its history of received and transmitted blocks:

```
get atucch [1.8.3]
Group: adslAtucChanPerfDataTable
Instance: [1.8.3.0]
IfIndex              =  1.8.3.0
ReceivedBlks         =  344340144
TransmittedBlks      =  2756828256
CorrectedBlks        =  51
```

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the ADSL port.
ReceivedBlks	A count of all encoded blocks or ADSL superframes received since the ADSL module was last reset.
TransmittedBlks	A count of all encoded blocks transmitted since the ADSL module was last reset.
CorrectedBlks	A count of the corrected blocks received since the ADSL module was last reset.

(*) indicates object used as an index.

adslAtucPerfDataExtTable
(alias: atucpfext)

This group displays Fast Retrain and Errored Second statistics for a near-end or central office port on an ADSL module (G.lite or G.dmt). It is an extension to or supplement for the adslAtuc-Perf-Data table.

You can see the number of fast retrains that succeeded and the number of fast retrains that did not succeed. A fast retrain occurs when conditions on the line change. Either they get better or they get worse. In response, the ADSL module in the CE200 adjusts its transmitting speed for the port, increasing the speed when conditions get better and decreasing the speed when they get worse.

In addition, you can also see the number of Severely Errored Seconds (SES) and a count of Unavailable Seconds (UAS).

- An SES is a one-second period during which at least 18 CRC-8 anomalies appear or at least one LOS, SEF, or LPR defect appears. For more info, see *adslAtucPerfDataTable* on page 29.
- A UAS is a count of one-second intervals where a line is unavailable. The line becomes unavailable after 10 continuous SESs appear; it becomes available after 10 continuous seconds without an SES.

The counts for Severely Errored Seconds and Unavailable Seconds typically occur right before or right after conditions on a line change. When the number of SESs and UASs gets too large, the modems on both ends of the line must retrain at lower speeds.

Syntax: *command* adslatucperfdataext *[ifindex]*

The only valid query is to a port; you cannot query a VC.

Example: To query a port for Fast Retrains and Errored Seconds history:

```
get atucpfext [1.8.1]
Group: adslAtucPerfDataExtTable
Instance: [1.8.1.0]
IfIndex              =  1.8.1.0
StatFastR            =  0
StatFailedFastR      =  0
StatSesL             =  3
StatUasL             =  0
```

Note: In the current release, the counter for Fast Retrain (StatFastR) and the counter for Failed Fast Retrain (StatFailedFastR) will always be set to zero since FastRetrain is not supported.

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the ADSL port.
StatFastR	A count of Fast Retrains since the G.lite module was last reset. G.dmt does not support this object. **Note:** In this release, this counter will always be zero.
StatFailedFastR	A count of Failed Fast Retrains since the G.lite module was last reset. G.dmt does not support this object. **Note:** In this release, this counter will always be zero.
StatSesL	A count of Severely Errored Seconds (SES) since the ADSL module was last reset. An SES is a period, one second in length, during which at least 18 CRC-8 anomalies appear or at least one LOS, one SEF, or one LPR defect appears.
StatUasL	A count of Unavailable Seconds (UAS) since the ADSL module was last reset. A UAS is a count of 10 one-second intervals, during which 10 SESs appear. A line becomes unavailable after 10 continuous SESs; it becomes available after 10 continuous seconds without an SES.

(*) indicates object used as an index.

adslAtucPerfDataTable
(alias: atucpf)

This group displays cumulative data about performance problems on a near-end or central office port on an ADSL module (G.lite or G.dmt). Information includes loss of framing, loss of signal, loss of link, and loss of power.

In contrast to the objects in the adslAtucPhys group, which only tell you that a port is currently experiencing a framing loss, a signal loss, a link loss, or a power loss, the adslAtucPerfData group tells you how many times since the ADSL module has been reset that each of the events has occurred. It also provides a count of critical errors, such as CRC problems, and a count of successful and unsuccessful initialization attempts.

Syntax: *command* adslatucperfdata *[ifindex]*

The only valid query is to a port; you cannot query a VC.

Example: To query port 1, slot 8:

```
get atucpf [1.8.1]
Group: adslAtucPerfDataTable
Instance: [1.8.1.0]
IfIndex              =  1.8.1.0
Lofs                 =  2
Loss                 =  0
Lols                 =  0
Lprs                 =  1
ESs                  =  6
Inits                =  1679
```

Note: In the current release, the counter for Loss of Power (Lprs) will always be at zero because CE200 hardware does not yet support messaging for the software counter.

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the ADSL port.
Lofs	The number of Loss of Framing (LOF) failures since the ADSL module was last reset.
Loss	The number of Loss of Signal (LOS) failures since the ADSL module was last reset.
Lols	The number of Loss of Link (LOL) failures since the ADSL module was last reset.
Lprs	The number of Loss of Power (LOP) failures since the ADSL module was last reset. The number will always be zero.
ESs	The number of Errored Seconds (ES) since the ADSL module was last reset. An ES is a count of one-second intervals with one or more CRC anomalies, or one or more LOS or SEF anomalies.
Inits	The number of line initialization attempts since the ADSL module was last reset. It includes both successful and failed attempts.

(*) indicates object used as an index.

adslAtucPhysTable
(alias: atucph)

This group displays status information for a near-end or central office port on an ADSL module (G.lite or G.dmt). Information includes Signal-to-Noise Ratio (SNR), power attenuation, power output, line status, and the highest possible data rate.

You can check the status on an ADSL port to determine the current SNR on the downstream line and the amount of power output by the port. You can also determine if problems exist with framing, initialization sequences, and linkup from the CE200's point of view. Finally, you can determine if there is a CPE on the other end of the line.

Syntax: *command* adslatucphys [*ifindex*]

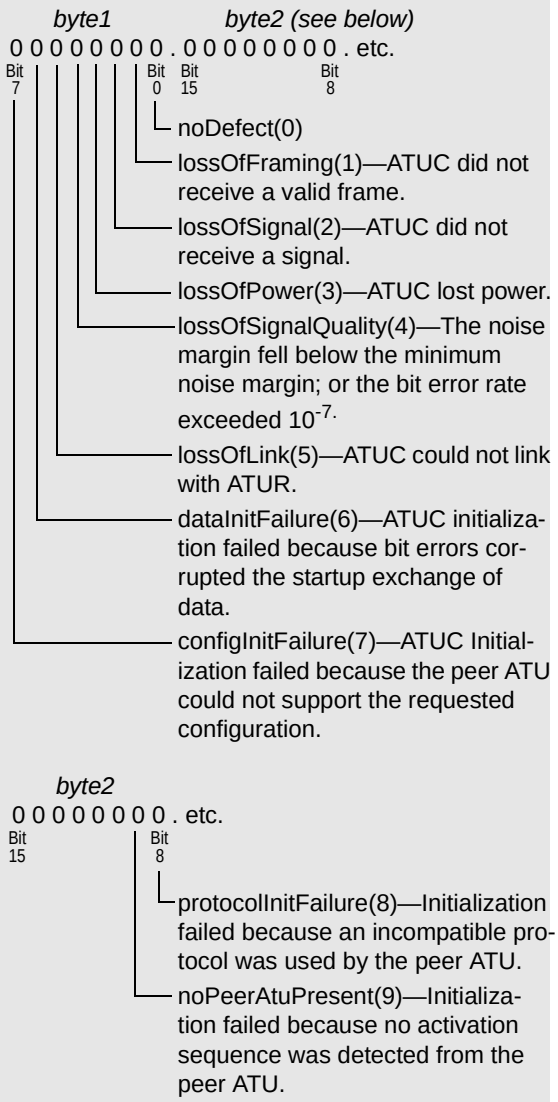
The only valid query is to a port; you cannot query a VC.

Example: To query port 1, slot 8 about its current status:

```
get atucph [1.8.1]
Group: adslAtucPhysTable
Instance: [1.8.1.0]
IfIndex             =  1.8.1.0
InvSerialNumber     =  0123456789
InvVendorID         =  0x39
InvVersionNumber    =  0x1
CurrSnrMgn          =  -5
CurrAtn             =  20
CurrStatus          =  LossOfFraming(1)+NoPeerAtuPresent(9)
CurrOutputPwr       =  144
CurrAttainableRate  =  0
```

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the ADSL port.
InvSerialNumber	The vendor specific serial number. The format is an octet string typically displayed as ASCII characters. If any characters are not printable, the string is displayed in hex format. This object is not currently used. See the SerialNumber object in *cmBoardTable* on page 61.
InvVendorID	The binary vendor identification number. The format is an octet string typically displayed as ASCII characters. If any characters are not printable, the string is displayed in hex format.
InvVersionNumber	The vendor specific version number. The format is an octet string typically displayed as ASCII characters. If any characters are not printable, the string is displayed in hex format.
CurrSnrMgn	The noise margin on this port measured in tenths of dB: -640 to 640.
CurrAtn	The difference in total power, measured in tenths of dB transmitted by the CPE and received on this port: 0 to 630.

CurrStatus	The current state of the ATU-C line. The bitmap of possible values is:  For Get and Getnext requests, the format is *bitStr0(bitId0)+bitStr1(bitId1)*..., where *bitString* is the ASCII text and BitID is the corresponding bit number. For Getall requests, the format is *bit1+bit2...* . An empty string (“ “) means that no bits are set.
CurrOutputPwr	The total output power transmitted by this ATU measured in tenths of dB during the last activation: -310 to 310.
CurrAttainableRate	The maximum currently attainable data rate in bps: 0 to 6144000. This value is the highest data rate the port can attain, not the current data rate on the line. It is greater than or equal to the current line rate.

(*) indicates object used as an index.

adslAturPerfDataExtTable
(alias: aturpfext)

This group displays performance information for a far-end or CPE port on an ADSL module (G.lite or G.dmt). It is an extension to or supplement for the adslAturPerfData table. In this release, however, the adslAturPerfData table itself is not available.

You cannot see the number of fast retrains that succeeded or did not succeed here. The CE200, not the CPE, records this information. However, you can see the number of Severely Errored Seconds (SES) and a count of Unavailable Seconds (UAS).

- An SES is a one-second period during which at least 18 CRC-8 anomalies appear or at least one LOS, one SEF, or one LPR defect appears. For more information, see *adslAtucPerfDataTable* on page 29.
- A UAS is a count of one-second intervals where a line is unavailable. The line becomes unavailable after 10 continuous SESs appear; it becomes available after 10 continuous seconds without an SES.

The counts for Severely Errored Seconds and Unavailable Seconds typically occur right before or right after conditions on a line change. When the number of SESs and UASs gets too large, the modems on both ends of the line must retrain at lower speeds.

Syntax: *command* adslaturperfdataext [*ifindex*]

The only valid query is to a port; you cannot query a VC.

Example: To query the CPE on the line attached to port 1, slot 8:

```
get aturpfext [1.8.1]
Group: adslAturPerfDataExtTable
Instance: [1.8.1.0]
IfIndex              =  1.8.1.0
StatSesL             =  1
StatUasL             =  0
```

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the ADSL port.
StatSesL	A count of Severely Errored Seconds (SES) since the ADSL module was last reset. An SES is a period, one second in length, during which at least 18 CRC-8 anomalies appear or at least one LOS, one SEF, or one LPR defect appears.
StatUasL	A count of Unavailable Seconds (UAS) since the ADSL module was last reset. A UAS is a count of 10 one-second intervals, during which 10 SESs appear. A line becomes unavailable after 10 continuous SESs; it becomes available after 10 continuous seconds without an SES.

(*) indicates object used as an index.

adslAturPhysTable
(alias: aturph)

This group status information for a far-end or CPE port on an ADSL module (G.lite or G.dmt). Information includes the serial number and version number, current signal-to-noise ratio, power attenuation, power output, line status, and attainable data rate.

You can check the status of a CPE port to determine the current SNR on the upstream line and the amount of power output by the CPE. You can also see if problems exist with framing and loss of signal from the CPE's point of view.

Syntax: *command* adslaturphys *[ifindex]*

The only valid query is to a port; you cannot query a VC.

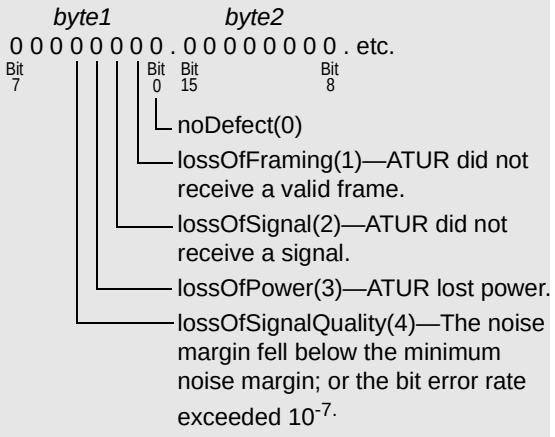
Example: To query the CPE on the line attached to port 1, slot 7:

```
get aturph [1.7.1]
Group: adslAturPhysTable
Instance: [1.7.1.0]
IfIndex              =  1.7.1.0
InvSerialNumber      =  ""
InvVendorID          =  ""
InvVersionNumber     =  ""
CurrSnrMgn           =  0
CurrAtn              =  0
CurrStatus           =  LossOfFraming(1)
CurrOutputPwr        =  0
CurrAttainableRate   =  0
```

'Note: Since object values are retrieved from the CPE, all values may not be available or correct at all times.

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the ADSL port.
InvSerialNumber	The vendor specific serial number. The format is an octet string typically displayed as ASCII characters. If any characters are not printable, the string is displayed in hex format.
InvVendorID	The binary vendor identification number. The format is an octet string typically displayed as ASCII characters. If any characters are not printable, the string is displayed in hex format.
InvVersionNumber	The vendor specific version number. The format is an octet string typically displayed as ASCII characters. If any characters are not printable, the string is displayed in hex format.
CurrSnrMgn	The noise margin on this port measured in tenths of dB: -640 to 640.
CurrAtn	The difference in total power, measured in tenths of dB transmitted by the CPE and received on this port: 0 to 630.

CurrStatus	The current state of ATU-R line. The bitmap of possible values is:  For Get and Getnext requests, the format is *bitStr0(bitId0)+bitStr1(bitId1)*..., where *bitString* is the ASCII text and BitID is the corresponding bit number. For Getall requests, the format is *bit1+bit2*... . An empty string (“ “) means that no bits are set.
CurrOutputPwr	The total output power transmitted by this ATU measured in tenths of dB during the last activation: -310 to 310.
CurrAttainableRate	The maximum currently attainable data rate in bps: 0 to 6144000. This value is the highest data rate the CPE can attain, not the data rate on the line at present. It is greater than or equal to the current line rate.

(*) indicates object used as an index.

atmInterfaceConfTable

This group has objects that focus on the number and type of virtual path and virtual channel connections on an ATM interface. It applies to either ADSL (G.lite or G.dmt) or WAN interfaces.

The table is important because it gives you an instant view of the maximum number of VCs allowed on a port and the number of VCs currently configured on the port. The maximum number allowed on a WAN port is 976; the maximum number allowed on a G.lite or G.dmt port is 8.

To see all of the VCs that have been configured on the CE200, along with their VPI and VCI settings, use the command **`geta cmatmvcl`**. For more information, see *cmAtmPerfIntervalTable* on page 55.

Syntax: *command* atminterfaceconf *[ifindex]*

The only valid query is to a port; you cannot query a VC.

Example: To query WAN port 1, slot 3 about the maximum number of VCs allowed and the number of VCs currently configured:

```
get atmi [1.3.1]
Group: atmInterfaceConfTable
Instance: [1.3.1.0]
IfIndex              =  1.3.1.0
MaxVpcs              =  0
MaxVccs              =  976
ConfVpcs             =  0
ConfVccs             =  15
MaxActiveVpiBits     =  4
MaxActiveVciBits     =  9
CurrentMaxVpiBits    =  4
CurrentMaxVciBits    =  9
SubscrAddress        =
```

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the ADSL or ATM WAN port.
MaxVpcs alias: vpcs	The maximum number of Virtual Path Connections at this interface. The value is always 0. The CE200 does not support VPCs on either ADSL or WAN ports.
MaxVccs alias: vccs	The maximum number of Virtual Channel Connections supported by this interface. For WAN ports, the number is 976. For ADSL ports, the number is 8.
ConfVpcs alias: cfvpcs	The number of Virtual Path Connections currently in use at this interface. The value is always 0. The CE200 does not support VPCs.
ConfVccs alias: cfvccs	The number of Virtual Channel Connections currently in use at this Interface. For WAN ports: 0 to 976. For ADSL ports: 0 to 8.
MaxActiveVpiBits alias: actvpi	Maximum number of active Virtual Path Identifier bits configured at this interface. Value is fixed at 4.
MaxActiveVciBits alias: actvci	Maximum number of active Virtual Channel Identifier bits configured at this interface. Value is fixed at 9.
CurrentMaxVpiBits alias: curvpi	Maximum number of Virtual Path Identifier bits that can be currently used at this Interface. Value is fixed at 4.
CurrentMaxVciBits alias: curvci	The maximum number of Virtual Channel Identifier bits that can be currently used at this Interface. The value is always 9.
SubscrAddress	The subscriber's ID assigned by the service provider. The format is *byte1.byte2*..., where bytes are displayed as hex numbers. If no address assigned, the address length is zero.

(*) indicates object used as an index.

atmMIBObjects

Use this group to determine the next available instance number when setting up a new row in the cmCircuitParamTable. The group allows you to set up as many as 255 different configurations for both Real Time Variable Bit Rate and non-Real Time Variable Bit Rate classes of service and then apply them to Virtual Circuits.

Each configuration must have a unique index number, which is its sequential row number in the cmCircuitParamTable. You cannot set up a new configuration without the instance number for the row. You can determine the number to use for the new row either by entering a *getall cmcircuitparam* and noting the number of the last row in the (often lengthy) response, or simply enter: *get atmmibobjects*.

Syntax: *command* atmmibobjects

Example: Since two rows with values already exist in the cmCircuitParam table, the number this command will return is 3.

```
get atmm
Group: atmMIBObjects
TrafficDescrParamInd =  3
```

Objects:

TrafficDescrParamInd	The next available cmCircuitParamIndex number to be used when creating a new row: 0 to 255. A value of 0 indicates that the cmCircuitParam table is full.

atTable

This group displays the address translation table for the specified interface, indexed by the interface's IfIndex and IP address.

Syntax: *command* attable *[ifindex, physaddress]*

Example: To display address translation for an interface at port 1, slot 2:

```
get att [1.2.1.0, 10.254.8.2]
Group: atTable
Instance: [1.2.1.0, 10.254.8.2]
IfIndex              =  1.2.1.0
PhysAddress          =  0.a0.cc.53.1c.cb
NetAddress           =  10.254.8.2
Type                 =  Dynamic
```

Objects:

*IfIndex	Permanent Interface Identifier (PII) of the interface.
PhysAddress	The physical address (typically, the MAC address) associated with this interface. The format is *byte1.byte2*..., where bytes are displayed as hex numbers.
*NetAddress	The IP address associated with this interface.
Type	The type of mapping: Other, Invalid, Dynamic, Static

(*) indicates object used as an index.

cmActiveSessionTable

The cmActiveSession table is only accessible if you have the Security privilege level. It displays information about all of the active operating sessions—that is, all operators currently logged into the system. It also allows you to terminate any active Craft, FTP, or SNMP session that another operator opened.

You can log into the CE200 through one or more of the following methods:

- Telnet—up to four sessions per System Control Module
- Serial—one session per System Control Module
- FTP—up to four sessions per System Control Module
- SNMP—the number of sessions depends on the bandwidth

Sometimes an operator may forget to log out and a script may still be running, preventing the CE200 from automatically logging the session out after a timeout expires. You can terminate any Craft, FTP, or SNMP sessions, whether they were opened locally or through the Radius authentication server.

Terminating a session stops any application that may be running and logs the operator out of the session. If the operator has multiple sessions on the same CE200, they are not affected. The event log will list the logout event as due to a termination of the session.

The cmActiveSession table will not let you accidentally terminate your own session. If you enter your session's instance, the following message will be displayed:

```
set cmact [1985] row=destroy
Delete failed
Group: cmActiveSessionTable
```

If your system is set up with a redundancy complex, you can only terminate sessions on the same side (Preferred or Backup) as your current session. That is, if your current session is on the Preferred side, you cannot terminate a session on the Backup side.

See also *cmOperatorTable* on page 155 and *cmRadius* on page 159.

Syntax: *command* cmactivesession *[sessionid]*

Example 1: To list all active sessions:

```
geta cmact
SessionID        OperatorName        Context         Privilege
IpAddress        StartTime           IdleTimeout     RowStatus

Instance: [1982]
1982             private             SNMP            Provision
10.64.20.162     18 day 18 hour 33 m 900             Active

Instance: [1983]
1983             ce200               Telnet          Security
10.64.20.249     18 day 18 hour 36 m 900             Active

Instance: [1984]
1984             public              SNMP            View
10.64.40.66      18 day 18 hour 36 m 900             Active

Instance: [1985]
1983             ce200               Telnet          Security
10.64.20.234     18 day 18 hour 36 m 900             Active
```

Example 2: To display information for active session 1983:

```
get cmact [1983]
Group: cmActiveSessionTable
Instance: [1983]
SessionID            =  1983
OperatorName         =  ce200
Context              =  Telnet
Privilege            =  Security
IpAddress            =  10.64.20.249
StartTime            =  18 day 18 hour 36 min 27.0 sec
                        (2000/11/18-09:19:39)
IdleTimeout          =  900
RowStatus            =  Active
```

Example 3: To terminate active session 1983:

```
set cmact [1983] row=destroy
Set successful
```

Note: SNMP entries are preserved in the cmActiveSession table even after their idle time-outs expire. If an SNMP operator tries to access the CE through the same session after timeout, the system drops the old table entry and creates a new one. Unlike the Craft interface, the operator is not required to log in again.

Objects:

*SessionID	The index number identifying he current database session for this active operator.
OperatorName Alias: Name	The name that the operator used to log in for this session.
Context	The context the operator used to log in to the system: Serial SNMP Telnet FTP Shell Modem HTTP—Not supported in this release.
Privilege	The privilege level assigned to this operator: View Monitor Provision Security
IPAddress	The IP address of the connected terminal for this session.
StartTime	The elapsed time since the system was reset, and the calendar date/time when the operator logged in to the system. The elapsed time is expressed as *day hour min sec*. This is the value of the SysUpTime counter. The calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.
IdleTimeout	The time in seconds that the terminal can be idle before being automatically logged out. A default minimum of 900 (15 minutes) is enforced on any external server.
RowStatus alias: state	The current status of the row (operator session) in the cmActiveSession table: Active or Destroy. Use Destroy to terminate another operator's session and delete it from the table.

(*) indicates object used as an index.

cmADSLChanTable
(alias: achan)

Use this group to configure channels for ADSL ports (both G.dmt and G.lite). In this release, each port supports eight VCs, and one channel for both downstream and upstream lines.

Syntax: *command* cmadslchan *[index]*

The only valid query is to a port; you cannot query a VC.

Example 1: To query port 1, slot 8:

```
get achan [1.8.1]
Group: cmADSLChanTable
Instance: [1.8.1.0]
Index                =  1.8.1.0
TxLineType           =  InterleavedOnly
RxLineType           =  InterleavedOnly
TxFastOverhead       =  onePercent
RxFastOverhead       =  onePercent
TxInterleaveDelay    =  delay32MS
RxInterleaveDelay    =  delay16MS
TxInterleaveCorrTime =  time2MS
RxInterleaveCorrTime =  time1MS
TxMaxDelay           =  delay2MS
RxMaxDelay           =  delay2MS
CurrTxR              =  0
CurrRxR              =  0
CurrTxS              =  0
CurrRxS              =  0
CurrTxD              =  0
CurrRxD              =  0
CurrTxDelay          =  notAvailable
CurrRxDelay          =  notAvailable
```

Example 2: To change the downstream channel buffer delay and buffer correction settings for port 1, slot 8:

```
set achan [1.8.1] txintdelay=delay16ms txintcorr=time1ms
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the ADSL port.
TxLineType	The line type of the downstream channel: FastOnly (Default for G.dmt) InterleavedOnly (Default: for G.lite)
RxLineType	The line type of the upstream channel: FastOnly (Default for G.dmt) InterleavedOnly (Default for G.lite)
TxFastOverhead	For G.dmt, with InterleavedOnly line type, this object is the downstream redundancy overhead option. With variable rate, overhead of onePercent is best. With fixed rate and large SNR margin, a larger overhead is better. For G.dmt, with FastOnly Tx line type, set this object to disable. Not applicable to G.lite: notApplicable (0) fiftyPercent (1) twentyfivePercent (2) twelvePercent (3) sixPercent (4) threePercent (5) twoPercent (6) onePercent (7) disable (Fhex) Default: onePercent for InterleavedOnly G.dmt; disable for FastOnly G.dmt; and notApplicable for G.lite.

RxFastOverhead	For G.dmt, with InterleavedOnly line type, this object is the upstream Reed-Solomon buffer redundancy overhead option. With variable rate, overhead of onePercent is best. With fixed rate and large SNR margin, a larger overhead is better. For G.dmt, with FastOnly Rx line type, set this object to disable. Not applicable to G.lite: notApplicable (0) fiftyPercent (1) twentyfivePercent (2) twelvePercent (3) sixPercent (4) threePercent (5) twoPercent (6) onePercent (7) disable (Fhex) Default: onePercent for InterleavedOnly G.dmt; disable for FastOnly G.dmt; and notApplicable for G.lite.
TxInterleaveDelay alias: txintdelay	For G.dmt, with InterleavedOnly line type, this object is the downstream Reed-Solomon buffer interleave delay option. For optimal correction of random errors, use a ratio of 16:1 between delay and correction time. Interleaving has no effect on random errors, only bursts. A larger ratio offers less protection against random noise; a smaller ratio uses too much overhead. For G.dmt, with FastOnly line type, set object to *disable*. Not applicable to G.lite. notApplicable (0) delay1MS (2) delay2MS (3) delay4MS (4) delay8MS (5) delay16MS (6) delay32MS (7) delay64MS disable (Fhex) Default: delay32MS for InterleavedOnly. In the above list US means microseconds; MS is milliseconds
RxInterleaveDelay alias: txintdelay	For G.dmt, with InterleavedOnly line type, this object is the downstream interleave delay option. For optimal correction of random errors, use a ratio of 16:1 between delay and correction time. Interleaving has no effect on random errors, only bursts. A larger ratio does not offer as much protection against random noise; a smaller ratio uses too much overhead. For G.dmt, with FastOnly line type, set this object to disable. Not applicable to G.lite: notApplicable (0) delay1MS (2) delay2MS (3) delay4MS (4) delay8MS (5) delay16MS (6) delay32MS (7) delay64MS disable (Fhex) Default: delay16MS for InterleavedOnly G.dmt; disable for FastOnly G.dmt; and notApplicable for G.lite.

TxInterleaveCorrTime alias: txintcorr	For G.dmt, with InterleavedOnly line type, this object is the downstream interleave correction option. For optimal correction of random errors, use a ratio of 16:1 between delay and correction time. Interleaving has no effect on random errors, only bursts. A larger ratio does not offer as much protection against random noise; a smaller ratio uses too much overhead. For G.dmt, with FastOnly line type, set this object to disable. Not applicable to G.lite: notApplicable (0) time64US (2) time125US (3) time250US (4) time500US (5) time1MS (6) time2MS (7) time4MS (8) disable (Fhex) Default: time2MS for InterleavedOnly G.dmt; disable for FastOnly G.dmt; and notApplicable for G.lite.
RxInterleaveCorrTime alias: rxintcorr	For G.dmt, with InterleavedOnly line type, this object is the upstream interleave correction option. For optimal correction of random errors, use a ratio of 16:1 between delay and correction time. Interleaving has no effect on random errors, only bursts. A larger ratio does not offer as much protection against random noise; a smaller ratio uses too much overhead. For G.dmt, with FastOnly line type, set this object to disable. Not applicable to G.lite: notApplicable (0) time64US (2) time125US (3) time250US (4) time500US (5) time1MS (6) time2MS (7) time4MS (8) disable (Fhex) Default: time1MS for InterleavedOnly G.dmt; disable for FastOnly G.dmt; and notApplicable for G.lite.
TxMaxDelay	For the G.lite card, the maximum delay allowed from the Dmt framing (S) and interleaving (D) in the downstream direction. Not applicable for full-rate G.dmt. notApplicable (0) delay1MS (2) delay2MS (3) delay4MS (4) delay8MS (5) delay16MS (6) Default: Delay2MS for G.lite; not applicable for full-rate G.dmt.
RxMaxDelay	For the G.lite card, the maximum delay allowed from the Dmt framing (S) and interleaving (D) in the upstream direction. Not applicable for full-rate G.dmt. notApplicable (0) delay1MS (2) delay2MS (3) delay4MS (4) delay8MS (5) delay16MS (6) Default: Delay2MS for G.lite; not applicable for full-rate G.dmt.
CurrTxR	Number of redundant bytes (R) per Reed-Solomon code word in the downstream direction. Default: None for G.dmt; none for G.lite
CurrRxR	Number of redundant bytes (R) per Reed-Solomon code word in the upstream direction. Default: None for G.dmt; none for G.lite.

CurrTxs	Number of DMT symbols (S) per Reed-Solomon code word in the downstream direction. Default: 1 for full-rate G.dmt FastOnly; none for G.dmt InterleavedOnly; and none for G.lite.
CurrRxS	Number of DMT symbols (S) per Reed-Solomon code word in the upstream direction. Default: 1 for G.dmt FastOnly; none for G.dmt InterleavedOnly; and none for G.lite.
CurrTxD	Interleave depth (D) in the downstream direction. Default: for G.dmt FastOnly; none for G.dmt InterleavedOnly; and none for G.lite.
CurrRxD	Interleave depth (D) in the upstream direction. Default: for G.dmt FastOnly; none for G.dmt InterleavedOnly; and none for G.lite.
CurrTxDelay	Current delay from the dmt framing (S) and interleaving (D) in the downstream direction, computed as S*D/4. delay250US (2) delay500US(3) delay1MS (4) delay2MS (5) delay4MS (6) delay8MS (7) delay16MS (8) delay32MS (9) delay64MS (10) Default: delay250US for G.dmt FastOnly; none for G.dmt InterleavedOnly; and none for G.lite.
CurrRxDelay	Current delay from the dmt framing (S) and interleaving (D) in the upstream direction, computed as S*D/4. delay250US (2) delay500US(3) delay1MS (4) delay2MS (5) delay4MS (6) delay8MS (7) delay16MS (8) delay32MS (9) delay64MS (10) Default: delay250US for G.dmt FastOnly; none for G.dmt InterleavedOnly; and none for G.lite.

(*) indicates object used as an index.

Notes:

1. When TxLineType=FastOnly and RxLineType=FastOnly, and you are setting the buffer redundancy overhead option of the downstream and upstream fast channels:
   - When using a variable rate, an overhead of 1% gives the best performance.
   - When using a fixed rate with a large SNR margin, a larger overhead gives better performance.
2. When TxLineType=InterleavedOnly and RxLineType=FastOnly InterleavedOnly, and you are setting the TxInterleave and RxInterleave objects:
   - For optimal correction of random errors, we suggest a ratio of 16:1 between delay and correction. Interleaving has no impact on random errors; only on bursts.
   - A ratio larger than 16:1 is not as effective, and a ratio less than 16:1 uses so much overhead the benefits of correction may be offset by the effort to send the redundant bytes.

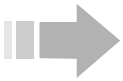
cmADSLModemTable
(alias: amod)

Use this group to configure ADSL ports (G.lite or G.dmt) on the CE200. The most important settings are the Signal to Noise Ratios (SNR) and data rates both upstream and downstream.

There are three important settings for SNR on both the upstream and downstream: minimum SNR, maximum SNR, and target SNR. The higher you set the target SNR setting above the default, the more restrictions you place on frequencies that can be used in transmitting. Setting the target SNR too high may force the CPE to retrain at lower data rates.

There are two important settings for data rate on the upstream and downstream portions of the line: minimum rate and maximum rate. The **upstream** rates are the **Received** rates, that is, the rates at which data is received by the module from the subscriber (*MinRxRate* and *MaxRxRate*); the **downstream** rate is the rate at which the module **transmits** data over the DSL link to the subscriber (*MinTxRate* and *MaxTxRate*). Beginning with general Release 7.0, the maximum settings for these rates have been increased as listed in the table below.

Remember to save the configuration file after you change parameters (enter **`set cmsystem command=save`**). If the ADSL module restarts and you have not saved the configuration, your changes will be lost.



N O T E

Although the CE200, through the cmAtmVcl group, allows you to place up to eight VCs on a G.lite or G.dmt port, the data rates must be set on the port, not on the VC. The same is true of Signal-to-Noise settings.

Syntax: *command* cmadslmodem *[index]*

The only valid query is to a port; you cannot query a VC.

Example 1: To change the upstream and downstream target SNR settings for port 1, slot 7 on a G.lite module:

```
set amod [1.7.1] upstrtar=6 downstrtar=6
```

Example 2: To view the changes you made on the G.lite module:

```
get amod [1.7.1]
Group: cmADSLModemTable
Instance: [1.7.1.0]
Index                =  1.7.1.0
ConfigLineCoding     =  G.lite
ActualLineCoding     =  G.lite
UpStrTargetSnrMgn    =  6
DownStrTargetSnrMgn  =  6
UpStrMaxSnrMgn       =  30
DownStrMaxSnrMgn     =  30
UpStrMinSnrMgn       =  0
DownStrMinSnrMgn     =  0
MaxTxRate            =  2336000
MinTxRate            =  64000
```

```
CurrTxRate            =  0
PrevTxRate            =  0
MaxRxRate             =  512000
MinRxRate             =  32000
CurrRxRate            =  0
PrevRxRate            =  0
HEC                   =  Enable
Scrambling            =  Enable
Ghs                   =  Enable
Command               =  None
PMDState              =  16
Debug                 =  ""
SuccessfulTrains      =  0
FailedTrainingAtt     =  176347
CpeResponses          =  0
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the ADSL port.
ConfigLineCoding	For G.dmt, the configured ADSL line coding is ITU-T G.992.1. For G.lite, the configured ADSL line coding is ITU-T G.992.2.
ActualLineCoding	For G.dmt, the actual ADSL line coding is always ITU-T G.992.1; for G.lite, the actual ADSL line coding is always ITU-T G.992.2. In this release, these setting are always the same as the setting for ConfigLineCoding.
UpStrTargetSnrMgn	The upstream target signal to noise ratio margin: 0 to 15 in 1 dB steps. For G.lite, the default is 4. For G.dmt, the default is 6. **Note:** Setting the TargetSnr too high may decrease the data rate.
DownStrTargetSnrMgn	The downstream target signal to noise ratio margin: 0 to 15 in 1 dB steps. For G.lite, the default is 4. For G.dmt, the default is 6. **Note:** Setting the TargetSnr too high may decrease the data rate.
UpStrMaxSnrMgn	The upstream maximum signal to noise ratio margin: 0 to 31 in 1 dB steps. For G.lite, the default is 30. G.dmt does not support this object; the value is always 0.
DownStrMaxSnrMgn	The downstream maximum signal to noise ratio margin: 0 to 31 in 1 dB steps. For G.lite, the default is 30. G.dmt does not support this object; the value is always 0.
UpStrMinSnrMgn	The upstream minimum signal to noise ratio margin: 0 to 31 in 1 dB steps. For G.lite, the default is 0. G.dmt does not support this object; the value is always 0.

DownStrMinSnrMgn	The downstream minimum signal to noise ratio margin: 0 to 31 in 1 dB steps. For G.lite, the default is 0. G.dmt does not support this object; the value is always 0.
MaxTxRate	The configured maximum transmit rate for channels, in 32000 bps steps. For G.lite: 64000 to 2336000. For G.dmt: 32000 to 8064000. The MinTxRate and MaxTxRate objects can be set to the same value, but MinTxRate cannot be greater than MaxTxRate.
MinTxRate	The configured minimum transmit rate for channels, in 32000 bps steps. For G.lite: 64000 to 2336000. For G.dmt: 32000 to 8064000. The MinTxRate and MaxTxRate objects can be set to the same value, but MinTxRate cannot be greater than MaxTxRate.
CurrTxRate	The actual transmit rate in 32000 bps steps. For G.lite: 64000 to 2336000. For G.dmt: 32000 to 8064000.
PrevTxRate	The previous transmit rate at the time of the last rate change, in 32000 bps steps. For G.lite: 64000 to 2336000. For G.dmt: 32000 to 8064000.
MaxRxRate	The configured maximum receive rate for channels, in 32000 bps steps. For G.lite: 32000 to 512000. For G.dmt: 32000 to 1024000. The MinRxRate and MaxRxRate objects can be set to the same value, but MinRxRate cannot be greater than MaxRxRate.
MinRxRate	The configured minimum receive rate for channels, in 32000 bps steps. For G.lite: 32000 to 512000. For G.dmt: 32000 to 1024000. The MinRxRate and MaxRxRate objects can be set to the same value, but MinRxRate cannot be greater than MaxRxRate.
CurrRxRate	The actual receive rate in 32000 bps steps. For G.lite 32000 to 512000. For G.dmt: 32000 to 1024000.
PrevRxRate	The previous receive rate at the time of the last rate change, in 32000 bps steps. For G.lite: 32000 to 512000. For G.dmt: 32000 to 1024000.

HEC	The ATM cell Header Error Correction control: Disable or Enable (the default).
Scrambling	The ATM cell payload scrambling control: Disable or Enable (the default). For G.dmt, the only option is Enable.
Ghs	The G.hs protocol control: Disable or Enable (the default). When disabled, the modem bypasses the G.hs message exchange. For G.dmt, the only option is Enable.
Command	The retraining commands: None Retrain FastRetrain—In this release, initiates a full retrain.
PMDState	For factory use only.
Debug	For factory use only.
SuccessfulTrains	For G.dmt, the number of successful training attempts since the module was powered up. This object is not supported by G.lite.
FailedTrainingAtt	For G.dmt, the number of unsuccessful training attempts since the module was powered up. This object is not supported by G.lite.
CpeResponses	For G.dmt, the number of CPE responses to the training attempts. This object is not supported by G.lite.

(*) indicates object used as an index.

cmADSLPerfTable
(alias: aper)

This group displays ATM traffic information for an ADSL port (G.lite or G.dmt). Information includes the total number of ATM cells transmitted and received, and the number of HEC errors, overruns, and discards accumulated since the module was last reset

The cmADSLPerf group provides information about traffic in ATM cells, unlike the adslAtucChanPerfData group, which provides information about blocks or ADSL superframes sent between the two modems on the line. ATM cell overruns and PDU discards on the CE200 are rare, but RxHecErrors and RemoteHecErrors, along with the RxCrc8Errors and RemoteCrc8Errors, increase whenever conditions on the line are changing and the two modems on the line are being forced to retrain, either at slightly faster or slightly lower data rates.

The cmADSLPerf group, used in conjunction with the adslAtucPerfData group, gives you a clear indication of when conditions on the line are changing. The cmADSLPerf and adslAtucPerfData groups, used in conjunction with the adsl-AtucPerfDataExt group, give you a clear indication of how the modems are adjusting to changing conditions.

Syntax: *command* cmadslperf *[index]*

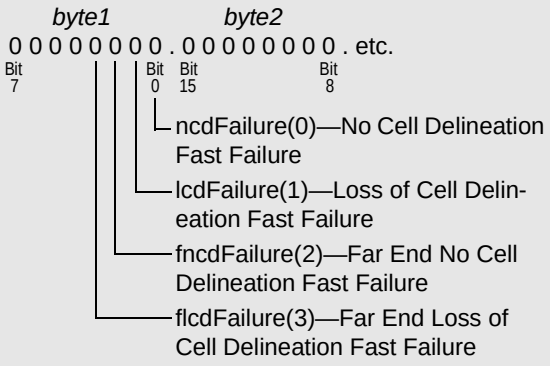
The only valid query is to a port; you cannot query a VC.

Example: *To query port 1, slot 8 about ATM cell transmissions and errors:*

```
get aper [1.8.1]
Group: cmADSLPerfTable
Instance: [1.8.1.0]
Index                =  1.8.1.0
TTxCells             =  0
RxCells              =  5741938
RxCellOverruns       =  0
RxPDUDiscards        =  0
RxHecErrors          =  26
RemoteHecErrors      =  0
RxCrc8Errors         =  20
RemoteCrc8Errors     =  0
RemoteCorrectedBlks  =  0
ATMStatus            =
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the ADSL port.
TxCells	The count of all ATM cells sent to the CPE over the port since the module was last reset.
RxCells	The count of all ATM cells received from the CPE over the port since the module was last reset.
RxCellOverruns	The count of cell overruns on the port since the module was last reset. Overruns are rare.
RxPDUDiscards	The count of PDUs dropped because of congestion on the SCM module or the buffer module.
RxHecErrors	The count of all upstream HEC errors on data sent by the CPE to the CE200. They typically occur during retrains.
RemoteHecErrors	The A count of all downstream HEC errors on data sent by the CE200 to the CPE. They typically occur during retrains.
RxCrc8Errors	The count of upstream CRC8 errors on data sent by the CPE to the CE200.They typically occur during retrains.
RemoteCrc8Errors	The count of downstream CRC8 errors on data sent by the CE200 to the CPE. They typically occur during retrains.
RemoteCorrected-Blks	The count of blocks or ADSL superframes sent by the CE200 and corrected by the CPE.

ATMStatus	The description of the ATM path status. The bitmap of possible values is:  For Get and Getnext requests, the format is *bitStr0(bitId0)+bitStr1(bitId1)*..., where *bitString* is the ASCII text and BitID is the corresponding bit number. For Getall requests, the format is *bit1+bit2...* . An empty string (“ “) means that no bits are set.

(*) indicates object used as an index.

cmAlarm

This is one of two groups—cmAlarm and cmAlarmTable—that allow you to configure conditions under which a special kind of trap event, called a Threshold Crossing Alert (TCA), will be generated. For a description of the different types of conditions that you can monitor, see *cmAlarmTable* on page 49.

The cmAlarm and cmAlarmTable groups are derived from similar groups in the Remote Monitor (RMON) MIB. They have kept their name in the Copper Mountain Monitor (CMON) performance monitoring segment of the Copper Mountain MIB.

The cmAlarm group allows you to perform two actions: first, to set the length of the interval to be used in monitoring the DSL and DS3 ports and their VCs; second, to enable all of the interfaces of the CE200 to measure and record TCAs.

Notes about Intervals:

By default, the performance-monitoring interval for all cmDSL, dsx1, and dsx3 groups is 15 minutes. The interval is configurable through the SamplingInterval object for 5, 15, or 60 minutes. If the interval is set for a period of less than 15 minutes, the groups designed to collect and display a full day’s worth of performance statistics will only display 96 intervals (96 times the number of minutes set in the SamplingInterval object). For example, if the interval is set to 5 minutes, statistics will only be displayed for 8 hours (96 intervals of 5 minutes each).

In commands directed to CPE units, the interval period is fixed at one hour, except for the cmCpeDataPortInterval group, which uses a 15-minute interval.

Syntax: *command* cmalarm *object_1 object_2*

Example 1: To turn on TCA functionality and set the monitoring interval period to five minutes:

```
set cmalarm sw=enabled sa=5
```

Example 2: To see how cmAlarm has been set:

```
get cmalarm
Group: cmAlarm
SamplingInterval     =  5
Switch               =  enabled
```

Objects:

SamplingInterval	The length of the interval (in minutes) to be used in collecting CE200 data performance statistics: 5 (the minimum) 15 (the default) 60
Switch	The functional status of the TCA: Enabled4(the default) or Disabled.

cmAlarmTable
(alias: cmon)

This is one of two groups—cmAlarm and cmAlarmTable—that allow you to configure conditions under which a special kind of trap event, called a Threshold Crossing Alert (TCA), will be generated and then written to the cmTrapAlarmTable.

To establish a TCA, you select a specific performance object from an array of statistics that are already being measured, such as received overruns or minimum noise margin on a DSL port, or Current Errored Seconds on a DS1 interface. Then you determine how many of these occurrences should be tolerated within a fixed real-time interval, and set that value as a rising or a falling threshold.

The *Variable* object indicates each condition you can monitor. The *RisingThreshold* and *FallingThreshold* allow you to set the value which, when crossed, will cause a TCA to be generated and an alarm to be written to the cmTrapAlarmTable. You must configure each performance object and its threshold separately for each applicable interface on the CE200.

When you set a new TCA or alarm condition, you must give it an identifying number. The system will support up to 1,000 active thresholds at any given time. Note that each index number for a TCA setting must be unique (it can be any integer from 1 to 65,535). If a duplicate index number is assigned, its previous TCA setting will be overwritten.

Also, if you use the object, Status=create, and the entry already exists, the command will not overwrite the entry. Instead, the system will indicate that the index is already in use.

Syntax: *command* cmalarmtable *[index] object_1...object_4*

Example 1: To configure a new TCA numbered 101:

```
set cmon [101] status=create variable=cmdslcurrenttable.
rxcrcerrors.109010000 rising=4
```

The threshold of interest is defined as received CRC errors (RxCRCErrors) occurring in the current DSL performance table (cmDSLCurrentTable), at shelf 1, slot 9, port 1 (109010000) at a value of 4 (rising=4). This command will cause a trap to be generated at the end of an interval in which the number of CRC errors reaches or goes beyond four errors.

Example 2: To display the configuration of TCA 101:

```
get cmon [101]
Group: cmAlarmTable
Instance: [101]
Index                =  101
Interval             =  900
Variable             =  cmDSLCurrentTable.RxCRCErrors.109010000
SampleType           =  Absolute
Value                =  0
StartupAlarm         =  Rising
RisingThreshold      =  4
FallingThreshold     =  0
Status               =  Valid
```

Notes:

1. You need not enter the entire configuration as a single Set command as shown in the example. If you first assign the index number and create the entry, you can enter the other parameters with separate commands.
2. You can enter the Variable using its SNMP Object Identifier (OID) rather than the logical names. In this case, the object in the Set command of the above example would read:

   ```
   variable=1.3.4.1.4.1.1996.10.7.2.1.8.109010000
   ```

Objects:

*Index	An integer designating the index of this TCA configuration: 1 to 65535.
Interval	The length of the sampling interval in seconds: 300, 900, 3600

Variable

Object to be monitored. The components of the object are listed in the following order, separated by periods (.).

xxxTable.PerfObject.PII

xxxTable choices are:

cmDSLCurrent	dsx1Current
dsx1FarEndCurrent	dsx3Current
dsx3FarEndCurrent	sonetSectionCurrent
sonetLineCurrent	sonetFarEndLineCurrent
sonet PathCurrent	sonetFarEndPathCurrent
hdsl2ShdslEndPointCurrent	

PerfObject choices for cmDSLCurrent:

TxUnderruns	RxTruncated
RxBig	RxAborts
RxAlignmentErrors	RxCRCErrors
RxOverruns	RxValidFrames
TxValidFrames	RxNoBuffers
PortUpTime	RxValidOctets
TxValidOctets	MinNoiseMargin

PerfObject choices for dsx1Current and dsx1FarEndCurrent:

ESs	SESs
SEFs	UASs
CSSs (dsx1Current)	PCVs
LESs	BESs
LCVs (dsx1Current)	

PerfObject choices for dsx3Current:

PESs	PSESs
SEFs	UASs
LCVs	PCVs
LESs	CCVs
CESs	CSESs

PerfObject choices for dsx3FarEndCurrent:

CurrentCESs	CurrentCSESs
CurrentCCVs	CurrentUASs

PerfObject choices for sonetSectionCurrent:

ESs	SESs
SEFs	CVs

PerfObject choices for sonetLineCurrent, sonetFarEndLineCurrent, sonet PathCurrent, and sonetFarEndPathCurrent

ESs	SESs
CVs	UASs

PerfObject choices for hdsl2ShdslEndPointCurrent:

Atn	SnrMgn
15MinES	15MinSES
15MinCRCanomalies	15MinLOSWS
15MinUAS	

Note: You must type the PII portion of the Variable object in SNMP format:

- Do not use periods to separate the segments.
- Fill the entire field, including any leading zeros (except the initial shelf designation, which does not take a leading zero).

For example, 106040000 indicates shelf 1, slot 6, port 4, VC 0000.

SampleType	Type of increment recognized by the sample: Absolute.
Value	The absolute value of the specified performance object at the end of the last interval.
StartupAlarm	The type of threshold that triggers the TCA: Rising Falling, RisingOrFalling
RisingThreshold	0 to 2147483647 When the current sample of the specified variable reaches a level that is equal to or greater than the numeric value specified in this object, it will trigger a Threshold Crossing Alert. If the value is 0, monitoring is halted.
FallingThreshold	-128 to 127 When the current sample of the specified variable (currently supported only by MinNoiseMargin) reaches a level that is equal to or less than the numeric value specified in this object, it will trigger a Threshold Crossing Alert.
Status	The status of this row in the table: Valid—In a Get command, the row is actively being used by the CE200. In a Set command, overwrites an existing row. Create—Creates a new row that will be immediately used by the CE200. This value prevents an existing row from being overwritten. Creating—Not supported in this release. Invalid—Deletes a row. **Note:** Another way to overwrite an existing row is to not use the Status object in the Set command.

(*) indicates object used as an index.

cmAtmIfExtTable

This group allows you to administratively disable or re-enable OAM for an ATM WAN port. Note that the default setting is enabled.

Syntax: *command* cmatmifext *[pii]*

Example: To enable OAM for port 1, slot 4:

```
get cmatmi [1.4.1]

Group: cmAtmIfExtTable
Instance: [1.4.1]
PII                 =  1.4.1
OAMAdminState       =  enabled
```

Objects:

*PII	The Permanent Interface Identifier of the port.
OAMAdminState	Set the OAM administrative state for the port: disabled or enabled. The default is enabled.

(*) indicates object used as an index.

Note:

1. To enable OAM for a VC:
   a) Set the AdminStatus to Up for both the port and VC.
   b) Check that the OAMAdminState object in the cmAtmIfExtTable is enabled for the port.
   c) Enable OAM for the VC using the OAMAdminState object in the cmAtmVclTable.

   For more detailed instructions, see Chapter 5, “Advanced Configuration,” in the *CopperEdge 200 Installation and Operating Guide*.

cmAtmPerfTable

This group displays ATM traffic information for an ATM ATUC port (G.dmt, G.lite or G.shdsl). Information includes the total number of ATM cells transmitted and received, and the number of HEC errors, overruns, and discards accumulated since the module was last reset

Syntax: *command* cmatmperftable *[index]*

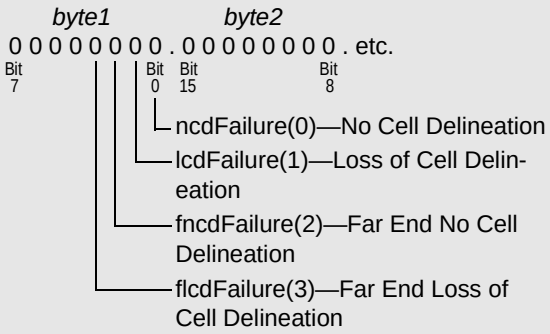
The only valid query is to a port; you cannot query a VC.

Example: *To query port 1, slot 8 about ATM cell transmissions and errors:*

```
get cmatmperft [1.8.1]
Group: cmATMPerfTable
Instance: [1.8.1.0]
Index                =  1.8.1.0
TTxCells             =  0
RxCells              =  5741938
RxCellOverruns       =  0
RxHecErrors          =  26
RemoteHecErrors      =  0
ATMStatus            =
Valid Intervals      =
Invalid Intervals    =
       .
       .
       .
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the port.
TxCells	The count of all ATM cells sent toward the CPE over the port since the module was last reset.
RxCells	The count of all ATM cells received from the CPE over the port since the module was last reset.
RxCellOverruns	The count of cell overruns on the port since the module was last reset. Overruns are rare.
RxHecErrors	The count of all upstream HEC errors on data received from the CPE.
RemoteHecErrors	The A count of all downstream HEC errors on data sent over this interface to the CPE.

ATMStatus	The description of the ATM path status. The bitmap of possible values is:  An empty string (“ “) means that no bits are set.
ValidIntervals	Number of previous 15-minute intervals during which valid data was collected in the past 24 hours: 0 to 96.
InvalidIntervals	Number of intervals in the range from 0 up to the value of cmADSLPerfValidIntervals for which no data is available. Typically 0 except where data for some intervals are not available, as in proxy situations.
Curr15MinTimeElapsed	Number of seconds elapsed in the present interval
Curr15MinTxCells	Number of downstream cells transmitted in the present interval
Curr15MinRxCells	Number of upstream cells received in the present interval
Curr15MinRxCellOverruns	Number of upstream cells discarded in the present interval
Curr15MinRxHecErrors	Number of upstream Header Error Control (HEC) errors in the present interval
Curr15MinRemoteHecErrors	Number of remote HEC errors in the present interval
Curr1DayTimeElapsed	Number of seconds elapsed since the start of the current 1-day interval
Curr1DayTxCells	Number of downstream cells transmitted since the start of the current 1-day interval as measured at *cmAdslPerfCurr1DayTimeElapsed*
Curr1DayRxCells	Number of upstream cells received since the start of the current 1-day interval as measured at *cmAdslPerfCurr1DayTimeElapsed*
Curr1DayRxCellOverruns	Number of upstream cells discarded since the start of the current 1-day interval as measured at *cmAdslPerfCurr1DayTimeElapsed*
Curr1DayRxHecErrors	Number of upstream HEC errors since the start of the current 1-day interval as measured at *cmAdslPerfCurr1DayTimeElapsed*
Curr1DayRemoteHecErrors	Number of remote HEC errors since the start of the current 1-day interval as measured at *cmAdslPerfCurr1DayTimeElapsed*

Prev1DayMoniSecs	Amount of time in the previous 1-day interval over which the performance monitoring information was actually counted.
Prev1DayTxCells	Number of downstream cells transmitted during the most recently completed 24-hour period
Prev1DayRxCells	Number of upstream cells received during the most recently completed 24-hour period
Prev1DayRxCellOverruns	Number of upstream cells discarded during the most recently completed 24-hour period
Prev1DayRxHecErrors	Number of upstream HEC errors during the most recently completed 24-hour period
Prev1DayRemoteHecErrors	Number of remote HEC errors during the most recently completed 24-hour period

(*) indicates object used as an index.

cmAtmPerfIntervalTable

This group displays one row for each ATM ATUC link for any 15-minute interval you select over the previous 1-day period. Parameters displayed are similar to those of the previous table (*cmATMPerfTable*), and includes total number of ATM cells transmitted and received, and the number of HEC errors, overruns, and discards counted for the selected interval.

Syntax: *command* cmatmperfinterval *[index, Number]*

The only valid query is to a port; you cannot query a VC.

Example: To query port 1, slot 8 about its ATM performance:

```
get cmatmperfint [1.8.1]
Group: cmATMPerfIntervalTable
Instance: [1.8.1.0]
Index                =  1.8.1.0
Number               =  1
TTxCells             =  0
RxCells              =  5741938
RxCellOverruns       =  0
RxHecErrors          =  26
RemoteHecErrors      =  0
ATMStatus            =
Valid Intervals      =
Invalid Intervals    =
       .
       .
       .
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the port.
*Number	The number of the selected 15-minute interval: 1 to 96, where 1 is the most recently completed interval, and 96 is the interval furthest away in time.
TxCells	The count of all ATM cells sent toward the CPE over the port during the specified interval.
RxCells	The count of all ATM cells received from the CPE over the port during the specified interval.

RxCellOverruns	The count of cells discarded on the port during the specified interval
RxHecErrors	The count of upstream HEC errors on data received from the CPE during the specified interval.
RemoteHecErrors	The A count of the downstream HEC errors on data sent over this interface to the CPE during the specified interval.
ValidData	Indicates whether or not the data for the selected interval are valid: true (1) or false (2)
TimeTagBegin	Calendar time at which the specified interval began
TimeTagEnd	Calendar time when the specified interval completed

(*) indicates object used as an index.

cmAtmVclTable

This group allows you to establish, configure, and manage both WAN and DSL ATM Virtual Channel Links (VCLs). It allows you to configure VPIs, VCIs, and other traffic parameters for each of the links either on the WAN or DSL side.

The Operation Administration and Maintenance (OAM) functions allow you to monitor end-to-end conditions for transmitting and receiving data over multiple segments of the WAN link. When the OAM function receives Alarm (AIS) messages and Remote Defect (RDI) messages, it allows you to send loopback requests to the device at the other endpoint, and receive loopback requests from the device. The OAM functions also allow you to send loopback response messages between endpoints of the link. By tracking the loopback messages over link segments, you can isolate faults in the link.

The TransmitTrafficDescrIndex object allows you to attach Service Category and Quality of Service (QoS) parameters to VCs. The parameters can be attached at this time only to VCs on the WAN interface. To learn about QoS parameters and indexes, see *cmCircuitParamTable* on page 66. To view all of the indexes in the system, along with parameter settings, enter the following command: **`getall cmcircuitparam`**.

On an ADSL module (G.lite or G.dmt), you can set 8 VCs on each port. On a WAN port, you can set a maximum of 976 VCs.

Syntax: *command* cmatmvcl *[pii] object_1...object_5*

The Virtual Link number is an arbitrary, user-assigned integer from 16 through 991 that is used to index CE200 VCLs. The Virtual Link number is unrelated to the values of VPI or VCI.

Example 1: To configure VCL 101 for port 1, slot 4:

```
set cmatmv [1.4.1.101] rowstat=createandgo vpi=2 vci=11
adminstatus=up transmittraffic=2
```

Example 2: To display the configuration of VCL 101:

```
get cmatmv [1.4.1.101]
Group: cmAtmVclTable
Instance: [1.4.1.101]
PII                   =  1.4.1.101
Vpi                   =  2
Vci                   =  11
AdminStatus           =  Up
OperStatus            =  Up
LastChange            =  12 day 1 hour 44 min 54.0 sec
                         (2000/09/28-14:44:46)
AalType               =  Aal5
Aal5CpcsTransmitSduS  =  1600
Aal5CpcsReceiveSduSi  =  1600
RowStatus             =  Active
TransmitTrafficDescr  =  2
OAMState              =  adminDown(0)
OAMAdminState         =  disabled
OAMAutoLBState        =  disabled
OAMManualLBCmd        =  none
OAMManualLBCmdStatus  =  none
OAMLBInterval         =  5
OAMLBTimeOut          =  15
OAMTxAISCells         =  0
OAMRxRDICells         =  0
OAMRxAISCells         =  0
OAMTxRDICells         =  0
OAMTxLBRequestCells   =  0
OAMRxLBResponseCells  =  0
OAMRxLBRequestCells   =  0
OAMTxLBResponseCells  =  0
OAMRxUnsupportedCell  =  0
OAMTxDiscards         =  0
OAMRxDiscards         =  0
```

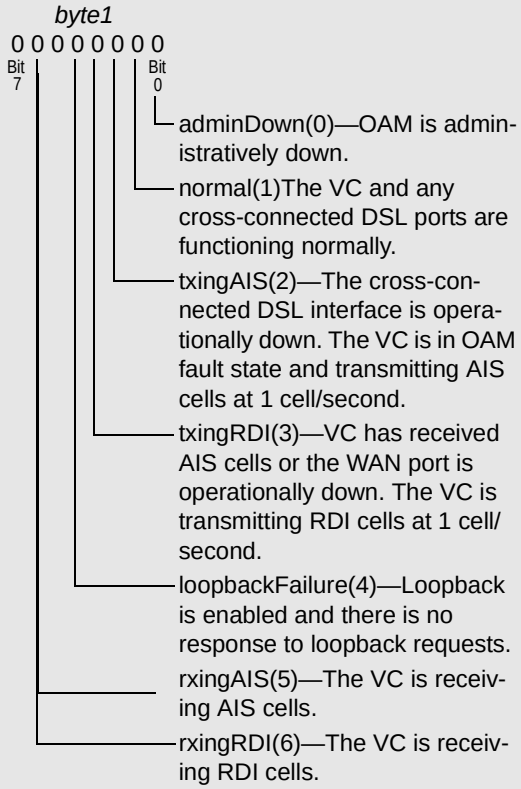
Notes:

1. For WAN links, the VPI/VCI combination of 0/511 used for loopback testing is automatically set when the applicable cmLoop test is configured (using *cmLoop on page 142*).
2. DSL ports using 1483 encapsulation require the VC number 528.

Objects:

*PII	The Permanent Interface Identifier for an ATM circuit on either an ADSL or a WAN port. For G.lite or G.dmt ports, use only 16 to 23 and 528. For a WAN port, only 16 to 991 are valid.
Vpi	The Virtual Path Identifier: 0 to 15. The combination of vpi=0 and vci=511 is reserved for ATM Link Management and loopback testing on the WAN ATM ports.
Vci	The Virtual Channel Identifier: 32 to 511. The combination of vpi=0 with vci=511 is reserved for ATM Link Management and loopback testing.
AdminStatus	The administrative state of the VCL: Up or Down (the default).
OperStatus	The current operational state of the VCL: Up or Down. For ATM WAN VCs, this can also reflect the operational state of the cross-connected VC, and vice versa. For a complete definition of OperStatus as it relates to this group, see *Operating Status* on page 20.

LastChange	The elapsed time since the system was reset, and the calendar date/time when the operator logged in to the system. The elapsed time is expressed as *day hour min sec*. This is the value of the SysUpTime counter. The calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.
AalType	For VCCs that terminate at the specified interface, the type of ATM Adaptation Layer on this interface: AAL5 (default) – Normal setting for variable rate, connection-oriented packet data. Unknown – Used for Cell Relay (aal type irrelevant) Other – For AAL5 trailer suppression. When applied to ATM WAN VCs, or to Aal5Transmit and Receive SDU sizes, *Other* indicates that none of the standard encapsulations is used on the VC.
Aal5CpcsTramsmitSdu-Size alias: tsdu	For VCCs that terminate at the specified interface, and which use AAL5, this object lists the maximum supported size (octets) of the Common Part Convergence Sublayer Service Data Unit in the transmit direction. The default is 1600.
Aal5CpcsReceiveSdu-Size alias: rsdu	For VCCs that terminate at the specified interface, and which use AAL5, this object lists the maximum supported size (cctets) of the Common Part Convergence Sublayer Service Data Unit in the receive direction. The default is 1600.
RowStatus	The operational state of a row in the cmAtmVcl table: Active—In a Get command, the row is actively being used by the CE200. In a Set command, overwrites an existing row. CreateAndGo—Creates a new row that will be immediately used by the CE200. It prevents an existing row from being overwritten. Destroy (alias: delete)—Deletes a row.
TransmitTrafficDescrIndex	The index to the cmCircuitParam table: 1 to 255. Index 1 is for Unspecified Bit Rate (UBR), and cannot be erased. All other indexes and classes of service can be erased or destroyed using the cmCircuitParam group.

OAMState	For WAN VCs only, the status of the link as per the OAM cells. The bitmap of possible values is:  For Get and Getnext requests, the format is *bitStr0(bitId0)+bitStr1(bitId1)*..., where *bitStr* is the ASCII text and *bitId* is the bit number. For Getall requests, the format is *bitId1+bitId2*... . An empty string (“ “) means that no bits are set.
OAMAdminState	For WAN VCs only, set OAM for the VC: disabled or enabled. The default is disabled. Before you set this object to enabled for the VC, check the cmAtmIfExtTable and be sure the OAMAdminState object is set to enabled for the port. If OAM is disabled in cmAtmIfExtTable, OAM cells are not generated by the CE, any received OAM cells are discarded, and the OAMAutoLBState object has no impact.
OAMAutoLBState	For WAN VCs only, control the periodic generation of loopback request cells: disabled or enabled. The default is disabled. This object does not control responses to loopback requests received from the network. Use this object to continuously monitor the status of a VC connection.

OAMManualLBCmd	For WAN VCs only, controls the manual generation of loopback request cells: none (default), f5End2End, f5Segment This object does not control responses to loopback requests received from the network. It has no impact if the port or VC is administratively disabled, or OAM is administratively disabled at the port or VC level.
OAMManualLBCmdStatus	For WAN VCs only, the status of the last manual loopback request transmitted: none inProgress succeeded failed
OAMLBInterval	For WAN VCs only, controls how often loopback cells are generated when the OAMLBAdminState object is set to enabled: 1 to 999 seconds. (Default=seconds)
OAMLBTimeOut	For WAN VCs only, the timeout for transmitted loopback cells: 1 to 999 seconds. (Default is 18 seconds)
OAMTxAISCells	For WAN VCs only, the number of AIS cells transmitted. This object does not change if the OAMAdminState object is disabled.
OAMRxRDICells	For WAN VCs only, number of RDI cells received.
OAMRxAISCells	For WAN VCs only, the number of AIS cells received.
OAMTxRDICells	For WAN VCs only, the number of RDI cells transmitted. This object does not change if the OAMAdminState object is disabled.
OAMTxLBRequestCells	For WAN VCs only, the number of loopback request cells transmitted. This object does not change if the OAMAdminState or OAMLBAdminStateobjects are disabled
OAMRxLBResponseCells	For WAN VCs only, the number of LoopbackResponse cells received in response to the LoopbackRequest cells transmitted.
OAMRxLBRequestCells	For WAN VCs only, the number of loopback request cells received. When a cell is received, the CE responds with a LoopbackResponse cell.
OAMTxLBResponseCells	For WAN VCs only, the number of loopback response cells transmitted in response to LoopbackRequest cells received. This object does not change if the OAMAdminState object is disabled
OAMRxUnsupportedCell	For WAN VCs only, the number of OAM cells received that are not processed by the CE. Unsupported cells include F4 cells, F5 segment level cells, Continuity check cells, and Performance Monitoring OAM cells.
OAMTxDiscards	For WAN VCs, number of OAM cells discarded.
OAMRxDiscards	For WAN VCs only, the number of OAM cells received but discarded.

(*) indicates object used as an index.

Note:

1. To enable OAM for a VC:
   a) Set the AdminStatus to Up for both the port and VC.
   b) Check that the OAMAdminState object in the cmAtmIfExtTable is enabled for the port.
   c) Enable OAM for the VC with the OAMAdminState object. For detailed instructions, see Chapter 5, “Advanced Configuration,” in the *Copper-Edge 200 Installation and Operating Guide*.
2. If the OAMAutoLBState is enabled, F5 End-End loopback request cells are sent periodically. If no response is received within the OAMLBTimeOut, the OAMState object changes to loopbackFailure.
3. When the OAMManualLBCmd object is set to f5End2End or f5Segment, a loopback request is sent. If a response is received, no more requests are generated. If no response is received, loopback request cells are generated as specified by the OAMLBInterval object, to the maximum set in the OAML-BTimeOut object.

cmBoardTable

This group displays information about the specified module. The configurable object, Command, provides the control mechanism for a soft restart of an individual module.

Syntax: *command* cmboard *[index]*

When entering the PII for the index, you can omit the port number (or leave it at zero) to indicate the entire module.

Example: To display information for the module in slot 2:

```
get cmbo [1.2]
Group: cmBoardTable
Instance: [1.2.0.0]
Index                =  1.2.0.0
ObjectClass          =  SystemControlModule3
OperState            =  Enabled
HwType               =  SystemControlModule3
HwVersion            =  R 1.0
SwVersion            =  5.0.59
PromVersion          =  4.0.46
Role                 =  Primary
ClusterRole          =  Master
UpTime               =  0 day 23 hour 43 min 8.0 sec
                        (2001/01/10-11:29:35)
NumPorts             =  5
FileName             =  P:/ce200/scm
FileDate             =  Jan 5 2002, 03:13:51
ConfigChange         =  0
Command              =  None
SerialNumber         =  S6A4500GC_00564611X4
Information          =  RAM 256Mb;Flash 64Mb;CMN_BIOS 04.00.46;
                        BRD_BIOS 02.00.02
```

Objects:

*Index alias: idx	The Permanent Interface Identifier (PII) of the module. Since this indicates the board, port numbers other than 0 are ignored.

ObjectClass alias: class	The type of module in the Board class: Board SystemControlModule SDSL10xModule-1 SDSL10xModule SDSL30xModule SDSL30xModule-24 IDSLModule-24 BufferControlModule BufferControlModule2 V.35-WAN DS3FR-WAN DS3ATM-WAN QuadT1-WAN G.liteModule-24 SystemControlModule3 ADSLMultiModeModule-24A ADSLMultiModeModule-24B DualT1-WAN SDSLModule-8 T1Module-12 IMAWANModule SONETSingleMode-WAN SONETMultiModeWAN GSHDSLATMModule-24 GSHDSLPacketModule-24‘
OperState alias: oper	The current operational state of this module: Unknown Disabled Enabled Testing For a complete definition of OperState as it relates to this group, see *Operating Status* on page 20.
HWType alias: type	The type of module: SystemControlModule BufferControlModule BufferControlModule2 SDSL10xModule-1 SDSL10xModule SDSL30xModule SDSL30xModule-24 IDSLModule-24 V.35-WAN DS3FR-WAN DS3ATM-WAN QuadT1-WAN G.liteModule-24 SystemControlModule3 ADSLMultiModeModule-24A ADSLMultiModeModule-24B SDSLModule-8—Not supported in this release. DualT1-WAN—Not supported in this release. T1Module-12 IMAWANModule SONETSingleMode-WAN SONETMultiMode-WAN GSHDSLModule-24 GSHDSLATMModule-24
HWVersion	The hardware version in the format T M.N, where: T = release type M = major_rev N = minor_rev For example, R 1.0.
SWVersion alias: swver	The version number of the downloaded software in the format M.N.B, where: M = major revision N = minor revision B = build number For example, 2.111.4.
PromVersion alias: promver	The version number of the PROM software in the format M.N.B, where: M = major revision N = minor revision B = build number For example, 2.10.33.

Role	The role of the module in the system: Primary or Secondary—for System Control Modules Active—for other modules
ClusterRole	The role of the System Control Module in a single-shelf system. It will always display Master for both the Preferred and Backup System Control Modules.
UpTime	The elapsed time since this module was powered up, and the calendar date/time when this event occurred. The elapsed time is expressed as *day hour min sec*. The calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.
NumPorts alias: ports	The number of physical ports on this module.
FileName	The name of the file downloaded to this module.
FileDate	The date and time the downloaded file was compiled (or "built").
ConfigChange	Not supported in this release; always displays 0.
Command alias: cmd	When used in a Get statement (such as **get cmbo [*pii*]**), shows the last command issued. When used in a Set statement (such as **set cmbo [*pii*] command=restart**), issues a subsequent value as a command for the module to perform. Command values are None and Restart. Restart is not applicable to V.35, DS3 ATM WAN, or Buffer Control modules. **Note:** A Restart command directed to a module that does not support an individual module restart (e.g. DS1/E1 IMA Module) may return the message, `Set successful`, but the command will be ignored.
SerialNumber	The serial number of the specified module (**** is displayed if there is no serial number).
Information	An optional text string providing additional information about the module.

(*) indicates object used as an index.

cmBundleTable

Use this group to configure and display information about specific groups of ports on the CE200 that have been bundled for connection to an IMUX CPE, such as a CopperRocket 202 (SDSL) or 212 (IDSL), or a compatible device made by a third-party manufacturer. You can configure bundles with four DSL member ports, but, currently, the CE200 supports only two-port bundles.

When configuring a bundle, assign the PII of each DSL port to a member PII in the bundle, set the bundle's slot number to 51, and assign an arbitrary port number between 1 and 63. All bundles must be assigned slot number 51—it designates a bundle. Also, each bundle's port number must be unique.

Ports in a bundle do not need to be on the same DSL module, but all ports in a bundle must be on the same CPE. Therefore, all ports in a bundle must have the same end point ID. You can use cmEndPointPort to sort all of the member ports on the CE200 by their end point IDs. CmEndPointPort will also list the bundle PII for each port. If you have assigned ports to the wrong bundles, cmEndPointPort will help you find them.

When creating IMUX bundles, do not preconfigure the individual DSL member ports with cmIface or cmSubIface before assigning them to a bundle. If you try to assign configured ports to an IMUX bundle, the cmBundle command will fail, and the system will return an error message. If this happens, use cmIface to reset the DSL member port that is causing the error to NetModel=None. Configuration of NetModel and the resultant forwarding mode can only be done by configuring the bundle.

Note, however, that you can configure the information fields in cmIface for the member ports.

Syntax: *command* cmbundle *[pii] object_1...object_5*

The *pii* (*c.51.n*) is a variation of the usual PII format, where *c* is the shelf number, *51* designates the PII as a bundle, and *n* is any number from 1 to 63 that you assign to identify this bundle.

Example 1: To create a bundle and assign two member ports:

```
set cmbu [1.51.10] m1pii=1.10.9 m2pii=1.10.10
```

Example 2: To display the configuration of the bundle you just created:

```
get cmbu [1.51.1]
Group: cmBundleTable
Instance: [1.51.1.0]
PII                 =  1.51.10.0
RowStatus           =  Active
Member1PII          =  1.10.9.0
Member1EndPointID   =  0.60.58.1.4a.1d.0.0.0.0.0.0.0.0.0.0
Member1Status       =  Active
Member2PII          =  1.10.10.0
Member2EndPointID   =  0.60.58.1.4a.1d.0.0.0.0.0.0.0.0.0.0
Member2Status       =  Active
Member3PII          =  0.0.0.0
Member3EndPointID   =
Member3Status       =  None
Member4PII          =  0.0.0.0
Member4EndPointID   =
Member4Status       =  None
```

Example 3: To remove member port 2 from the bundle, set it to 0:

```
set cmbu [1.51.1] m2pii=0
```

Objects:

*PII	The Permanent Interface Identifier of a specified bundle of DSL ports in *c.b.n* format, where *c* is the shelf, *51* is the bundle designator, and *n* is an arbitrary, user-assigned integer between 1 and 63.

RowStatus	The operational state of a row in the cmBundle table: Active—In a Get command, the row is actively being used by the CE200. In a Set command, creates a new row. Destroy (alias: delete)—Deletes a row. **Note:** Configuring a bundle automatically creates the corresponding row with a default RowStatus of Active.
Member1PII alias: m1pii	The PII of the first DSL port assigned to this bundle.
Member1EndPointID alias: m1epid	The unique identifier of the IMUX CPE to which the Member1PII is connected. The format is *byte1.byte2...*, where bytes are displayed as hex numbers. If the bundle is configured correctly, the end point ID is the same for all members of the bundle.
Member1Status m1stat	The status of this member of the IMUX bundle: None—No DSL port is configured for this member. Active—A connected CPE is trained and ready to communicate. WaitForAdd—The DSL port is configured, and the link is trained, but the system is waiting for the CPE to report.
Member2PII alias: m2pii	The PII of the second DSL port assigned to this bundle.
Member2EndPointID alias: m2epid	The unique identifier of the CPE to which the Member2PII is connected. The format is *byte1.byte2...*, where bytes are displayed as hex numbers. If the bundle is configured correctly, the end point ID is the same for all members of the bundle.
Member2Status m2stat	The status of this member of the IMUX bundle: None—No DSL port is configured for this member. Active—A connected CPE is trained and ready to communicate. WaitForAdd—The DSL port is configured, and the link is trained, but the system is waiting for the CPE to report.
Member3PII alias: m3pii	The PII of the third DSL port assigned to this bundle. Bundles of more than two ports are not supported in this release.

Member3EndPointID alias: m3epid	The unique identifier of the CPE to which the Member2PII is connected. The format is *byte1.byte2...*, where bytes are displayed as hex numbers. If the bundle is configured correctly, the end point ID is the same for all members of the bundle. Bundles of more than two ports are not supported in this release.
Member3Status m3stat	The status of this member of the IMUX bundle: None Active WaitForAdd Bundles of more than two ports are not supported in this release.
Member4PII alias: m4pii	The PII of the fourth DSL port assigned to this bundle. Bundles of more than two ports are not supported in this release.
Member4EndPointID alias: m4epid	The unique identifier of the CPE to which the Member2PII is connected. The format is *byte1.byte2...*, where bytes are displayed as hex numbers. If the bundle is configured correctly, the end point ID is the same for all members of the bundle. Bundles of more than two ports are not supported in this release.
Member4Status alias: m4stat	The status of this member of the IMUX bundle: None Active WaitForAdd;Bundles of more than two ports are not supported in this release.

(*) indicates object used as an index.

cmCircuitParamTable

This group's objects focus on controlling upstream traffic on WAN VCs. You can set up as many as 255 classes of service, setting parameters to suit your applications. When defining a new WAN VC through the cmAtmVcl group, its Class of Service is entered using the index created in cmCircuitParam group. The table provides three basic classes of service:

- Real Time Variable Bit Rate (rtVBR)—reserve it for voice applications
- Non-real time Variable Bit Rate (nrtVBR)—reserve it for conferencing applications
- Unspecified Bit Rate (UBR)—reserve it for data applications.

RtVBR packets have priority over packets from the other two classes; nrtVBR packets have priority over UBR packets. But each time you configure a new rtVBR group and enter new values for its four settings—Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), Maximum Burst Size (MBS), and Cell Delay Variation (CDV)—you are creating a new subclass of service. Similarly, each time you configure a new nrtVBR group and enter new values for its three set-

tings—PCR, SCR, and MBS—you are creating a new subclass of service.

The UBR class requires no configuring and therefore cannot be subdivided. It has an index number of 1 and is the default setting for WAN VCs created through the cmATMVcl group.

Syntax: *command* cmcircuitparam *[paramindex]*

Example 1: To create a new category with index number 6, in the nrtVBR class, with a peak cell rate of 104000 cells/sec., a sustainable cell rate of 604 cells/sec., and a maximum burst size of 4:

```
set cmci [6] rowstatus=createandgo servicecat=nrtvbr pcr=104000
scr=604 mbs=4 cdv=unspecified
```

Example 2: To view the new category:

```
get cmci [6]
Group: cmCircuitParamTable
Instance: [6]
Index                =  6
RowStatus            =  CreatAndGo
ServiceCategory      =  nrtVBR
PCR                  =  104000
SCR                  =  604
MBS                  =  1024
CDV                  =  unspecified
```

Note: Due to hardware (SAR) limitations the system will round up the requested SCR to the next highest achievable rate. The achievable rate results from a complex relationship between SCR and MBS, and cannot be defined simply. The most useful technique is to read-back the SCR after configuration to find out the actual value that the system assigned. Note that for Cross-Connect netmodel, upstream ATM devices may be configured with original requested SCR, to achieve traffic policing as desired. This feature does not provide Rx traffic policing.

Objects:

*Index	The class of service. It must be linked to the cmATMVcl table: 1 to 255. The number 1 is always the UBR class; it cannot be deleted. All other indexes (or classes and subclasses) can be deleted.
RowStatus	Operational state of this row in cmCircuitParamtable: Active—In a Get command, the row is actively being used by the CE200. In a Set command, overwrites an existing row. CreateAndGo—Creates a new row that will be immediately used by the CE200. This value prevents an existing row from being overwritten. Destroy (alias: delete)—Deletes a row. You cannot create a new UBR row, destroy a row currently in use by a virtual circuit, or delete the fixed UBR entry with an Index of 1.
ServiceCategory	rtVBR—a real time variable bit rate, used primarily for voice applications. nrtVBR—a non-real time variable bit rate, used primarily for video conferencing applications. ubr—an unspecified bit rate, used primarily for data applications.

PCR	The Peak Cell Rate in cells/second: 0 to 110,000; if *PhysicalType* is set for *oc3stm1*, the allowable range is 0 to 354,000. The cell rate must accommodate the Maximum Burst Size. The 110,00 value is slightly greater than the DS3 data rate of 45 Mbps. For both the rtVBR class and the nrtVBR class, you must set the value. The PCR must be less than line rate but greater than the SCR.
SCR	Sustainable Cell Rate in cells/second: 0 to 110,000; if *PhysicalType* is configured as *oc3stm1*, then the allowable range is 0 to 354,000. The cell rate can rise above this value to accommodate the Maximum Burst Size, but the cell rate can never rise above Peak Cell Rate. For both the rtVBR class and the nrtVBR class, you must set the value. The SCR must be less than the PCR, which must be less than the line rate.
MBS	Maximum Burst Size (cells): 0 to 2048. MBS is the maximum number of cells that can be transmitted at PCR while still conforming to the SCR. Setting determines the amount of burstiness allowed on the VC, i.e., after brief periods of inactivity on a VC, up to MBS cells can be sent at up to the PCR to "catch up" to the average rate defined by the Sustainable Cell Rate. If you don't explicitly set a value for MBS, one will be generated based on the values of PCR and SCR. Note that if the values of PCR and SCR are large, the value of MBS will also be large and that value will persist, even if the values of PCR and SCR are reconfigured to lower numbers. To use a lower MBS value, you must explicitly set one.
CDV	The Cell Delay Variation through the DSLAM (not end-to-end): unspecified, minimum, nominal. Unspecified means rtVBR data has no priority over other data crossing the CE200. Minimum means rtVBR data always has priority. Nominal means rtVBR data sometimes has priority.
PhysicalType	Specifies the type of WAN module for which the entry is being configured: none, ds3, oc3stm1, t1e1ima

(*) indicates object used as an index.

cmConnTable

The cmConn table is a double indexed table which provides you the capability (using either the CopperCraft CLI or the CopperView Element Manager) of retrieving a listing of which DSL PIIs are connected to a specified WAN VC or Ethernet port. Because many networking models may involve aggregation of a number of DSL PIIs onto a single WAN VC or Ethernet port, this feature can be very useful in provisioning. As always, a DSL PII may denote a DSL port or a VC on a DSL port depending on the application and netmodel.

Syntax: *command* cmconn *[instance]*
While it is possible to view entries in the cmConnTable one at a time, the most practical method of using the connection table would be to cause it to generate a list. This is most easily done using the find or getnext command.

Example:

```
getn cmconn [1.3.1.101, 0]

Group: cmConnTable
Instance: [1.3.1.101, 1.6.1.21]
PII1             =  1.3.1.101
PII1OperStatus   =  Up
PII2             =  1.6.1.21
PII2OperStatus   =  Up
```

Note: Every interface that appears in the connection table occurs in two places; once as Pii1 and once as Pii2.

To refer to a T1/E1 IMA group, always use the “virtual” PII reference for the IMA group. Do *not* use individual link PIIs to configure an IMA group.

Objects:

*Pii1	PII of the first index object. The PII can be any physical or logical interface in the system.
Pii1OperStatus	State of the interface specified as Pii1: Up, Down
*Pii2	PII of the second index object. The PII can be any physical or logical interface in the system.
Pii2OperStatus	State of the interface specified as Pii2: Up, Down

(*) indicates object used as an index.

cmCPEBoardTable
(alias: cpeb)

This group displays information about a specified CPE. It also provides the control mechanism for a soft restart of a CPE, and for loading, updating, or erasing software on an individual CPE.

As shown in the following examples, cmCpeBoard commands must be explicitly directed to the CPE as distinguished from the CE200 port at the same PII location. For IMUX CPEs, cmCpeBoard commands may be directed to the PII of any of its member ports with identical results.

Syntax: *command* cmcpeboard *[cpe:index] object_1*

Where “cpe:” indicates a command for the CPE.

Example 1: To query a CPE connected to port 9, slot 10:

```
get cpeb [cpe:1.10.9]
Group: cmCPEBoardTable
Instance: [CPE:1.10.9]
Index                =  CPE:1.10.9
ObjectClass          =  CPE-SDSL
OperState            =  Disabled
HwType               =  CPE-SDSL
HwVersion            =  R 1.0
SwVersion            =  3.0.172
PromVersion          =  2.20.64
Role                 =  Active
UpTime               =  6 day 19 hour 40 min 46.0 sec
                        (2000/08/29-17:31:23)
NumPorts             =  2
FileName             =  CPE5_B.BIN
FileDate             =  Aug  7 2000, 23:52:22
ConfigChange         =  0
Command              =  None
SerialNumber         =  0.60.58.1.4a.1d
VendorDescription    =  CR202 IMUX - SDSL DSU
GroupMap             =  3a.40.39.38.82.83.84
ManagementOptions    =  1
```

Example 2: To restart the CPE connected to port 9, slot 10:

```
set cpeb [cpe:1.10.2] command=restart
```

Notes:

1. You can use a command from the cmCpeBoard group to download software to an individual CPE as described in the table below. To download software to multiple CPEs, it is generally more efficient to designate DSL physical ports that are eligible for bulk CPE download with cmEndPointConfig, and then use the cmMaintCmd group to execute the BulkDownload on all eligible CPEs at the same time.
2. If you need to download software to an individual CPE, note that the 1.568 Mbps data rate is not hard-coded in the CPE PROM. Thus, the following special procedure must be performed. (If you upgrade in bulk as described in note 1, any necessary speed adjustments are performed automatically and this procedure is not necessary.)
   a) Set the SDSL module port to any data rate lower than 1568000 bps. This will retrain the CPE.

      set cmhdslmodem [*pii*] datarate=784

   b) Issue the Upgrade command to the CPE:

      set cmcpeboard [cpe:*pii*] command=upgrade

   c) When the upgrade is complete, set the SDSL module data rate to any supported rate.
3. You should expect that any command directed to a CPE, or directly affecting a CPE (such as cmHDSLModem data rate), may cause the CPE to retrain.
4. If the octet for the cmCpePlugAndPlay table is not present in the GroupMap object, the voice VC on your IAD (Integrated Access Device) will not work. You will need to update the software on the IAD.

Objects:

*Index alias: idx	The Permanent Interface Identifier (PII) of one of the two bundled interfaces. It includes the PII of the DSL port to which the CPE is connected.
ObjectClass alias: class	The type of the connected CPE (CPE class): CR — CR201-10x CR201-30x — CR201-SDSL CR201IDSL — CPE-SDSL CPE-IDSL — Netopia-SDSL CPE-T1 — CPE-GSHDSL

OperState alias: oper	The current operational state of this CPE: Unknown, Disabled, Enabled, Testing For a complete definition of OperState as it relates to this group, see *Operating Status* on page 20.
HWType alias: type	The type of CPE hardware: CR, CR201-10x, CR201-30x, CR201-SDSL, CR201IDSL, CPE-SDSL, CPE-IDSL, Netopia-SDSL, CPE-T1
HWVersion alias: hwver	The hardware version in the format T M.N, where: T = release type M = major_rev N = minor_rev For example, T 2.1.
SWVersion alias: swver	The downloaded software version in the format M.N.B, where: M = major revision N = minor revision B = build number For example, 3.0.130.
PromVersion alias: promver	The PROM software version in the format M.N.B, where: M = major revision N = minor revision B = build number For example, 3.0.97.
Role	The role of the module in the system: Active Primary Secondary
UpTime	The elapsed time since this module was powered up, and the calendar date/time when the event occurred. The elapsed time is expressed as *day hour min sec*. The calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.
NumPorts alias: ports	The number of physical ports in this module.
FileName	The name of the file downloaded to this module.
FileDate	The date and time the downloaded file was compiled (or “built”).
ConfigChange	The calendar date and time of the last configuration change (not supported).

Command alias: cmd	When used in a Get statement (such as **get cpeb [*pii*]**), shows the last command issued. When used in a Set statement (such as **set cpeb [*pii*] command=restart**), issues a subsequent value as a command for the module to perform. Command values are: None Restart Upgrade Erase Unknown
SerialNumber	The serial number (MAC address) of this CPE.
VendorDescription	The conforming CPE will print the vendor's text description of the CPE. For example, CMN CR201-SDSL.
GroupMap	On a conforming CPE, an octet string that indicates which CE200 MIB groups it supports. The format is *byte1.byte2.byte3*..., where bytes are displayed as hex numbers. For example, 3a.40.39.38.91.92.93.94.9b.a0. Bytes indicate the corresponding MIB groups as: Hex 38 = Decimal 56 = cmCpeHDSLModem Hex 39 = Decimal 57 = cmCpeDSLInterval Hex 3a = Decimal 58 = cmCpeEthernetInterval Hex 40 = Decimal 64 = cmCpeBoard Hex 61 = Decimal 97 = cmCPEIDSLModem Hex 82 = Decimal 130 = cmCpeDataPort Hex 83 = Decimal 131 = cmCpeDataPortInterval Hex 84 = Decimal 132 = cmCpeDataPortTotal Hex 91 = Decimal 145 = cmCpeIAD Hex 92 = Decimal 146 = cmCpeVoicePort Hex 93 = Decimal 147 = cmCpeVoicePortInterval Hex 94 = Decimal 148 = cmCpeVoicePortTotal Hex 9b = Decimal 155 = cmCpePlugAndPlay Hex a0 = Decimal 160 = cmCpetollBridge Hex b3 = Decimal 179 = cmCpeGshdslModem
ManagementOptions	The CPE support for the software downloaded from the CE200: 1 = Downloads are supported by this CPE. 0 = Downloads not supported by this CPE.

(*) indicates object used as an index.

cmCpeDataPortIntervalTable
(alias: cpedint)

This group displays statistics for traffic between an IMUX SDSL CPE, such as a CopperRocket 202, and a downstream device, such as a LAN-based router. Note that the data port is *not* a DSL port, but an interface on the downstream side of the CPE.

The CPE stores statistics (such as received/transmitted frames and discards) in RAM for the most recent 96 intervals, set at 15 minutes each. You can review statistics for any specified interval in the past 24 hours.

Syntax: *command* cmcpedataportinterval *[cpe:pii, intervalnumber]*

Where "cpe:" indicates a command for the CPE. You must specify the PII of one of the DSL physical ports connected to the IMUX CPE. CPE data port statistics cannot be retrieved by specifying the bundle PII. To find the physical ports that comprise the bundle, use cmBundle to list the Member PIIs.

The *intervalnumber* tells the unit how far back in time to look. If the current interval is less than 15 minutes, it has a value of 1. To see statistics from 15 minutes ago, enter 2; to see statistics from 40 minutes ago, enter 3; and so on.

Example: To query the CPE connected to port 9, slot 10 on its performance for the 15-minute interval before the current interval:

```
get cpedint [cpe:1.10.9,2]
Group: cmCpeDataPortIntervalTable
Instance: [CPE:1.10.9, 2]
Index                =  CPE:1.10.9
IntervalNumber       =  2
RxFrames             =  10
RxOctets             =  20
RxDiscards           =  0
RxErrors             =  0
TxFrames             =  5
TxOctets             =  10
TxDiscards           =  0
TxErrors             =  0
```

Objects:

*Index alias: idx	The Permanent Interface Identifier (PII) of one of the two bundled DSL ports connected to the CPE of interest.
*IntervalNumber	The number of the 15-minute interval in the past 24 hours: 1 to 96; where 1 is the current interval. The count starts from when the CPE was powered on. For example, if it has been powered on for 3 hours and 8 minutes, interval 1 shows statistics for the current 8 minutes, and interval 3 shows statistics from 38 minutes ago.
RxFrames	The number of frames received in this interval.
RxOctets	The number of octets received in this interval.
RxDiscards	The number of receive frames discarded in this interval.
RxErrors	The number of error frames received in this interval.
TxFrames	The number of frames transmitted in this interval.
TxOctets	The number of octets transmitted in this interval.
TxDiscards	The number of transmit frames discarded (instead of being transmitted) in this interval.
TxErrors	The number of error frames transmitted in this interval

(*) indicates object used as an index.

cmCpeDataPortTable
(alias: cped)

This group allows you to configure and display the current communication control settings on the CPE data port associated with a specified DSL port. The CPE data port is on the downstream (LAN) side of an IMUX SDSL CPE, such as a CopperRocket 202. Typically, the CPE data port connects to a router to provide multiple users on a LAN with access to the CPE. The data port is *not* a DSL port.

Each CopperRocket 202 includes a CPE data port and a table entry containing information about the port. The cmCpeDataPort group also supports third-party IMUX SDSL CPE, provided they are equipped with an EIA 530 or a compatible V.35 data port.

Syntax: *command* cmcpedataporttable *[cpe:pii] object_1...object_5*

Where “cpe:” indicates a command for the CPE.

To query an IMUX CPE for information about settings on its CPE Data Port, you can specify the PII for any of the DSL member ports connected to the CPE. For example, if the IMUX bundle on the CE200 includes SDSL ports 1.9.4 and 1.11.6, you can query the data port on the CPE using either of the two PIIs. Using either port as the instance will allow configuration of the same parameters or return the same information.

Example: To query the CPE connected to port 10, slot 10 for its configuration:

```
get cped [cpe:1.10.10]
Group: cmCpeDataPortTable
Instance: [CPE:1.10.10]
Index                =  CPE:1.10.10
Type                 =  Unknown
ForceDSR             =  Disable
IgnoreDTR            =  Yes
IgnoreRTS            =  Yes
TerminalTiming       =  Disable
TM                   =  Off
LL                   =  Off
RL                   =  Off
DTR                  =  Off
RTS                  =  Off
DSR                  =  Off
CTS                  =  Off
DCD                  =  Off
TxClockSource        =  Internal
DTEDCE               =  DCE
CRC                  =  CRC-CCITT
ClockRate            =  0
AdminState           =  Enabled
OperState            =  Disabled
LoopbackConfig       =  noLoop
LoopbackStatus       =  noLoop
```

Objects:

*Index	The Permanent Interface Identifier (PII) of one of the DSL ports to which the CPE is connected.
Type	The type of data port: EIA-530 V.35 Unknown

ForceDSR	The Data Set Ready (DSR): Enable or Disable (the default). When enabled, Data Set Ready is forced to On regardless of the value of OperState. When disabled, no DSR setting is enforced.
IgnoreDTR	The Data Terminal Ready (DTR) signal line: Yes (the default) or No. When set to Yes, the CPE will not monitor the DTR signal line. When set to No, the CPE will de-assert DCD if the DTR signal is Off.
IgnoreRTS	The Ready To Send signal line: Yes (the default) or No. When set to Yes, the CPE will not monitor the RTS signal line. When set to No, the CPE will de-assert DCD if the RTS signal is Off.
Terminal Timing	The clock: Enable or Disable (the default). When enabled, the CPE will use the SCTE clock from the DTE. When disabled, the CPE will use its own clock.
TM	The Test Mode: On or Off. TM is On when the local loopback is On. Otherwise, it is Off.
LL	The Local Loopback status: On or Off. The local loopback must be generated by a DTE
RL	The Remote Loopback status: On or Off.
DTR	The Data Terminal Ready current status: On or Off.
RTS	The Ready To Send current status: On or Off.
DSR	The Data Set Ready current status: On or Off. It is On if the value of OperState=Enable or Testing.
CTS	The Clear to Send current status: On or Off.
DCD	The Data Carrier Detect current status: On or Off.f
TxClockSource	The Transmit Clock Source: External or Internal.
DTEDCE	The current functional role of the CPE Data Port: DTE or DCE.
CRC	The type of CRC used on packets. Currently, only CRC-CCITT is supported.
ClockRate	The current clock rate in bps: 0 to 3136000. The default for loopback is 1544000.
AdminState	The administrative state of the CPE Data Port: Unknown Enabled (the default) Disabled

OperState	The operational state of the CPE Data Port: Disable Enable Testing Unknown For a complete definition of the OperState object as it relates to this group, see *Operating Status* on page 20.
LoopbackConfig	The configuration of the local loopback: noLoop (the default) or localLoop. This object is not supported in this release.
LoopbackStatus	The status of the local loopback: noLoop (the default) or localLoop. This object is not supported in this release

(*) indicates object used as an index.

cmCpeDataPortTotalTable
(alias: cpedtot)

This group displays cumulative statistics for traffic between an IMUX SDSL CPE, such as a CopperRocket 202, and a downstream device, such as a LAN-based router. Note that the data port is *not* a DSL port, but an interface on the *downstream* side of the CPE.

The CPE stores statistics (such as received/transmitted frames and discards) in RAM for the most recent 96 intervals, set at 15 minutes each. You can review statistics for the cumulative total of all 96 stored intervals (24 hours).

Syntax: *command* cmcpedataporttotal *[cpe:pii]*

Where “cpe:” indicates a command for the CPE.

CPE data port statistics cannot be retrieved by specifying the bundle PII; you must specify the PII of one of the DSL physical ports connected to the IMUX CPE as the instance. To find the physical ports that comprise the bundle, use cmBundle to list the Member PIIs.

Example: To query the CPE connected to port 10, slot 10 for cumulative traffic statistics:

```
get cpedtot [cpe:1.10.10]
Group: cmCpeDataPortTotalTable
Instance: [CPE:1.10.10]
Index                =  CPE:1.10.10
RxFrames             =  10
RxOctets             =  20
RxDiscards           =  0
RxErrors             =  0
TxFrames             =  5
TxOctets             =  10
TxDiscards           =  0
TxErrors             =  0
```

Objects:

*Index	The Permanent Interface Identifier (PII) of one of the DSL ports connected to the CPE of interest.
RxFrames	The number of frames received in this interval.

RxOctets	The number of octets received in this interval.
RxDiscards	The number of receive frames discarded in this interval.
RxErrors	The number of error frames received in this interval.
TxFrames	The number of frames transmitted in this interval.
TxOctets	The number of octets transmitted in this interval.
TxDiscards	The number of transmit frames discarded (instead of being transmitted) in this interval.
TxErrors	The number of error frames transmitted in this interval

(*) indicates object used as an index.

cmCpeDSLIntervalTable

This group displays stored statistics (such as transmitted and received overruns/underruns and valid frames) collected on the DSL interface (SDSL, IDSL, ADSL, or T1) of a CPE. The CPE stores statistics in RAM for the most recent 24 intervals, set at one hour each. You can review statistics for any specified interval in the past 24 hours.

On a regular CPE or an IAD CPE, you are measuring all of the traffic on the DSL interface. It does not matter if the port on the CR201 has multiple data VCs serving multiple hosts, or if the port on the IAD CPE has a data VC serving multiple hosts and a voice VC serving multiple phone lines. Statistics for all VCs on a single DSL port are merged together.

To retrieve statistics for traffic over each of the VCs on a single DSL port, use the frCircuit group (page 226).

Syntax: *command* cmcpedslinterval [cpe:*index, number*]

Where “cpe:” indicates a command for the CPE.

The *number* tells the unit how far back in time to look. If the current interval is less than one hour, it has a value of 1. To see statistics from the most recent full hour, enter 2; to see statistics from eight hours ago, enter 9; and so on.

Example: To query the CPE connected to port 2, slot 11, on its performance one hour before the current interval:

```
get cmcpeds [cpe:1.11.2, 2]
Group: cmCpeDSLIntervalTable
Instance: [CPE:1.11.2, 2]
Index                =  CPE:1.11.2
Number               =  2
TxUnderruns          =  0
RxTruncated          =  0
RxBig                =  0
RxAborts             =  0
RxAlignmentErrors    =  0
RxCRCErrors          =  0
RxOverruns           =  0
RxValidFrames        =  4669
TxValidFrames        =  4486
RxNoBuffers          =  0
```

```
RxValidOctets        =  1049810
TxValidOctets        =  1056741
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DSL port to which the CPE is connected.
*Number	The number of the one-hour interval in the past 24 hours: 1 to 24, where 1 is the current interval. The count starts from when the CPE was powered on. For example, if it has been powered on for 3 hours and 8 minutes, interval 1 shows statistics for the 8 minutes of the current hour, and interval 3 shows statistics from 2 hours 30 minutes ago.
TxUnderruns	The number of transmit underruns.
RxTruncated	The number of frames truncated due to a receiver overrun.
RxBig	The number of frames discarded because they were too large.
RxAborts	The number of aborted frames received.
RxAlignmentErrors	The number of unaligned frames received.
RxCRCErrors	The number of frames received with CRC errors.
RxOverruns	The number of frames discarded because of a receiver overrun.
RxValidFrames	The number of valid frames received.
TxValidFrames	The number of valid frames transmitted.
RxNoBuffers	The number of received frames discarded due to lack of buffers.
RxValidOctets	The number of valid octets received.
TxValidOctets	The number of valid octets transmitted.

(*) indicates object used as an index.

cmCpeEthernetIntervalTable

This group displays stored statistics for the Ethernet interface of a CPE. The CPE stores statistics in RAM for the most recent 24 intervals, set at one hour each. You can review statistics for any specified interval in the past 24 hours.

On a regular CPE or an IAD CPE, you are measuring all of the traffic on the Ethernet interface. It includes traffic from multiple hosts on a LAN connected to a CR201 or a CR408. But, on a CR408, data for the voice VC does not enter on the Ethernet interface.

Syntax: *command* cmcpeethernetinterval *[cpe:index, intervalnumber]*

Where “cpe:” indicates a command for the CPE.

The *intervalnumber* tells the unit how far back in time to look. If the current interval is less than one hour, it has a value of 1. To see statistics from the most recent full hour, enter 2; to see statistics from eight hours ago, enter 9; and so on.

Example: To query the CPE connected at port 2, slot 11 on its performance during the third interval (two hours back in time):

```
get cmcpee [cpe:1.11.2, 3]
Group: cmCpeEthernetIntervalTable
Instance: [CPE:1.11.2, 3]
Index                =  CPE:1.11.2
IntervalNumber       =  3
IntUnknowns          =  0
RxOK                 =  4411
RxNoBufs             =  0
RxMiss               =  0
RxSkip               =  0
RxRunt               =  0
RxExtraData          =  0
RxCRCError           =  0
RxAlignError         =  0
RxDribbleBits        =  0
TxTooBig             =  0
TxNotReady           =  0
TxLossCRS            =  0
TxSQEErr             =  0
TxOutOfWindow        =  0
TxJabber             =  0
Tx16Collisions       =  0
TxCollisions         =  0
TxOK                 =  4594
```

Objects:

*Index alias: idx	The Permanent Interface Identifier (PII) of the DSL port to which the CPE is connected.
*IntervalNumber	The number of the one-hour interval in the past 24 hours: 1 to 24, where 1 is the current interval. The count starts from when the CPE was powered on. For example, if it has been powered on for 3 hours and 8 minutes, interval 1 shows statistics for the 8 minutes of the current hour, and interval 3 shows statistics from 2 hours 30 minutes ago.
IntUnknowns	The number of unknown interrupt events.
RxOK	The number of successful receptions.
RxnoBufs	Number of times the CPE buffer could not receive.
RxMiss	The number of times chip could not receive a frame.
RxSkip	The number of implied received skips.
RxRunt	The number of runt (incomplete) frames received.
RxExtraData	The number of times extra data was received.
RxCrcError	The number of invalid CRCs received.
RxAlignError	The number of received frame alignment errors.
RxDribbleBits	The number of extra bits received.
TxTooBig	The number of times the requested transmit length was too big.
TxNotReady	The number of times the transmitter was not ready.
TxLossCRS	The number of transmissions that lost carrier sense.

TxSQEErr	The number of SQE errors.
TxOutOfWindow	The number of late collisions.
TxJabber	The number of transmissions that were greater than 26 msecs.
Tx16Collisions	The number times when there were 16 collisions for a single frame (the same packet collided 16 times and was then discarded).
TxCollisions	The number of transmissions that had collisions.
TxOK	The number of successful transmissions.

(*) indicates object used as an index.

cmCpeGshdslModemTable
(alias: cpeshmod)

This read-only group displays information about the G.SHDSL CPE connected to this interface.

A related table, *cmGshdslModem*, gives similar statistics for the G.Shdsl modules in the CE200. To set the data rate for the port on the CE200 and for the line leading to the CPE, use the appropriate object in the cmGshdslModem group.

Syntax: *command* cmcpegshdslmodem *[index]*

Example: To query the CPE connected to port 2, slot 11 for its statistics:

```
get cpeshmod [cpe:1.11.2]
Group: cmCpeGSHDSLModemTable
Instance: [CPE:1.11.2]
IfIndex              =  CPE:1.11.2
OperState            =  Enabled
DataRate             =  1568
TxPower              =  3
RxGain               =  1
SqMean               =  3
SwVersion            =  1
```

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the DSL port to which the HDSL CPE is connected.
OperState	Current operational state of the interface: Unknown Enabled Disabled Testing
DataRate	The operating data rate of this interface (kbps in 64 kbps increments): 192 256 320 384 448 512 576 640 704 768 832 896 960 1024 1088 1152 1216 1280 1344 1408 1472 1536 1600 1664 1728 1792 1856 1920 1984 2048 2112 2176 2240 2304 None Unknown
TxPower	A decimal integer indicator of the Transmit power.

RxGain	Current CPE receiver gain setting.
SqMean	Integer indicating the current mean signal quality.
SwVersion	The Software Version of the CPE's G.shdsl chip

(*) indicates object used as an index

cmCpeHDSLModemTable

This group displays stored statistics for data rate, signal level, noise margins, timing, and other functions on the DSL interface of a regular CPE (such as a CR201) or on the DSL interface of an IAD CPE (such as a CR408).

A similar table, cmHDSLModem, gives the same statistics for the CE200. You must set the data rate for the port on the CE200 and for the line leading to the CPE with the appropriate object in the cmHDSLModem group. Then the object for data rate in the cmCpeHDSLModem group is able to pick up the data rate from the cmHDSLModem group.

Syntax: *command* cmcpehdslmodem *[cpe:index]*

Where "cpe:" indicates a command for the CPE.

Example: To query the CPE connected to port 2, slot 11 for its statistics:

```
get cmcpeh [cpe:1.11.2]
Group: cmCpeHDSLModemTable
Instance: [CPE:1.11.2]
Index                =  CPE:1.11.2
OperState            =  Enabled
DataRate             =  1568
Type                 =  Remote
AnalogAGC            =  6LevelAGC
Enable               =  TRUE
TxGain               =  0
TestMode             =  NoTest
ReverseTipRing       =  TRUE
SignalLevel          =  21
DCOffset             =  0
Attenuation          =  6
NoiseMargin          =  31
TimingRecovery       =  2
BPStatus             =  216
BPSwMajorVersion     =  2
BPSwMinorVersion     =  0
BPHwVersion          =  19
BpStage              =  79
AAGCValue            =  1
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DSL port to which the HDSL CPE is connected.
OperState	Current operational state of the interface: Unknown Enabled Disabled Testing For a complete definition of OperState as it relates to this group, see *Operating Status* on page 20.

DataRate	The configured data rate, in Kbps, for this port: 1568 1040 784 416 320 208 160 None Unknown
Type	The Bit Pump terminal type: Central or Remote.
AnalogAGC‡	The AGC level, if it is available: NoAGC 4LevelAGC 2LevelAGC 6LevelAGC
Enable	The Port Status Indicator: TRUE or FALSE.
TxGain	The value of the transmit gain.
TestMode	The operating test mode: NoTest ExternalAnalogLoopback DigitalNearLoopback DigitalFarLoopback TxIsolatedPlus3Pulse TxIsolatedPlus1Pulse TxIsolatedMinus1Pulse TxIsolatedMinus3Pulse Continuous4LevelTx Continuous2LevelTx SetNominalVCXOFrequency SetMinimumVCXOFrequency SetMaximumVCXOFrequency InternalAnalogLoopback IsolatedAnalogLoopback Continuity NearEndLoopProfile
ReverseTipRing	The Reverse Tip/Ring polarity on a received signal: TRUE or FALSE.
SignalLevel‡	The average absolute value of an ADC input signal.
DCOffset‡	The average DC offset per ADC sample.
Attenuation‡	`The overall signal power attenuation.`
NoiseMargin	The noise margin of the receiver.
Timing Recovery‡	Eight MSbs of the timing recovery control word.
BPStatus	The Bit-Pump status bits.
BPSwMajorVersion	The Bit-Pump major software version.
BPSwMinorVersion	The Bit-Pump minor software version.
BPHwVersion	The Bit-Pump hardware type and version.
BPStage‡ alias: Stage	The Bit-Pump software internal stage number.
AAGCValue‡	The current value of 3 (LSB) AAGC control bits.

‡ The object is accessible through CopperCraft only (no SNMP access).

(*) indicates object used as an index.

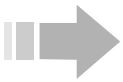
cmCpeIADTable

This group displays data about current and stored IP addresses, and the number of voice ports and voice connections possible on an IAD CPE.

The object, Command, allows a user in a central office to erase the two IP addresses saved in memory on an IAD after assigning new addresses in cmDHCP. It erases the IP address of the IAD (SavedIpAddr) and the IP address of the voice gateway (SavedCAIPAddr).

The function sets the two saved IP addresses to zero, but leaves the two current IP addresses as they were. Later, when you restart the IAD using either cmCPEBoard or ifTable, the CPE will ask the CE200 to send it two new IP addresses: one for itself and one for its voice gateway. The IAD saves the new addresses in the SavedIpAddr and SavedCAIpAddr fields, and also writes them to the CurrentIpAddr and CurrentCAIpAddr fields. Now the IAD can function on a new subnet with a new voice gateway. The IAD will always ask, through DHCP, for the two IP addresses after the DSL line is retrained or the IAD has been restarted.

The objects, LogAction, LogType, LogLevel, and LogDuration, allow you to log debug messages to the cmCpeLogTable when performing diagnostics for an IAD CPE.



N O T E

If you forget to set the IP address for a CR408 IAD in the cmDHCP table, the voice VC on your IAD will not work. Even though you set the IP address of the voice VC in the cmIface table, and the IP address of the voice gateway with cmInterfaceOptions table, you must also set the IP address of the IAD in the cmDHCP table.

Syntax: *command* cmcpeiad *[cpe:index]*

Where “cpe:” indicates a command for the CPE.

Example 1: To query the CPE connected to port 7, slot 10:

```
get cmcpeia [cpe:1.10.7]
Group: cmCpeIADTable
Instance: [CPE:1.10.7]
Index                =  CPE:1.10.7
CurrentIpAddr        =  0.0.0.0
CurrentCAIpAddr      =  0.0.0.0
SavedIpAddr          =  10.0.2.6
SavedCAIpAddr        =  10.0.2.1
NumVoicePorts        =  8
NumConnections       =  16
Command              =  None
TOSByte              =  0
LogAction            =  None
LogType              =  ALL(0)
LogLevel             =  INFO(0)
LogDuration          =  30
TotalMsgsLogged      =  0
Trace1Mask           =  0
Trace2Mask           =  0
```

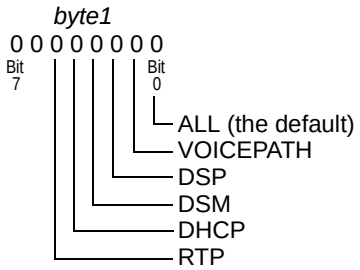
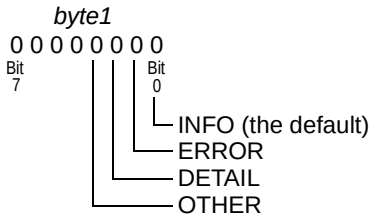
Example 2: To set the saved IP address on an IAD to zero:

```
set cmcpeia [cpe:1.10.13] command=savedipaddrerase
```

Remember to reboot the IAD, using cmCpeBoard, so the new IP addresses will be sent from the CE200.

Objects:

*Index alias: pii	The Permanent Interface Identifier (PII) of the DSL port to which the CPE is connected.
CurrentIpAddr	The IP address of the voice IP port on the IAD. It was received from the CE200 system by a DHCP request.
CurrentCAIpAddr	The IP address of the main call agent. It was received from the CE200 by a DHCP request.
SavedIpAddr	The IP address of the voice IP Port on the IAD. It was received from the CE200 system by a DHCP request and was saved in flash memory. This address may be different from the CurrentIpAddr if the flash memory has been updated but the IAD has not been restarted.
SavedCAIpAddr	The IP address of the main call agent. It was received from the CE200 by a DHCP request and was saved in flash memory. This address may be different from the CurrentCAIpAddr if the flash memory has been updated but the IAD has not been restarted.
NumVoicePorts	The number of voice ports on the IAD.
NumConnections	The number of voice connections that can be connected to the IAD. Two connections per voice port are required for the IAD to support Call Hold and Call Transfer.
Command	There are three possible commands. None (the default) SavedIPAddrErase SavedCAIPAddrErase Use SavedIPAddrErase to set the SavedIPAddr to zero. When the IAD reboots, it will request a new CurrentIPAddr from the DHCP function and save the address in NVRAM. Use SavedCAIPAddrErase to set the SavedCAIPAddr to zero. When the IAD reboots, it will request a new CurrentCAIPAddr from the DHCP function and save the address in NVRAM.
TOSByte	The TOS (Type Of Service) byte that the IAD will use to send voice data and signaling. This value is set by the upstream Call Agent.
LogAction	The type of debug action to be performed: None (the default) StartDebug—Start logging debug messages. StopDebug—Stop logging debug messages. DeleteDebug—Delete stored debug messages.

LogType	The type of debug messages that the cmCpeLog table receives. The bitmap of possible values is:  For Get and Getnext requests, the format is *bitStr0(bitId0)+bitStr1(bitId1)*..., where *bitStr* is the ASCII text and *bitId* is the corresponding bit number. For Getall requests, the format is *bitId1+bitId2...* . An empty string (“ “) means that no bits are set.
LogLevel	Level of message that the cmCpeLog table receives. The bitmap of possible values is:  For Get and Getnext requests, the format is *bitStr0(bitId0)+bitStr1(bitId1)*..., where *bitStr* is the ASCII text and *bitId* is the corresponding bit number. For Getall requests, the format is *bitId1+bitId2...* . An empty string (“ “) means that no bits are set.
LogDuration	The time to stop logging debug messages, in minutes. The default is 30 minutes. When the LogDuration minutes have passed, the LogAction object is automatically set to StopDebug.
TotalMsgsLogged	The total number of messages logged.
Trace1Mask	The first Trace mask, for logging purposes.
Trace2Mask	The second Trace mask, for logging purposes.

cmCpeIDSLModemTable
(alias: cpeidsl)

This group displays the performance/characteristics of the connected IDSL CPE at the CPE side of the specified IDSL interface. A similar table, cmIDSLModem, is used to set and monitor characteristics for the CO side of the IDSL interface.

Syntax: *command* cmcpeidslmodem *[cpe:index]*

Where "cpe:" indicates a command for the CPE.

Example: To query the CPE connected to port 3, slot 11 on its performance:

```
get cpeidsl [cpe:1.11.3]
Group: cmCpeIDSLModemTable
Instance: [CPE:1.11.3]
Index                =  CPE:1.11.3
OperState            =  Enabled
DataRate             =  128
State                =  activated
Debug                =  0
ActivationInterval   =  5
TerminationMode      =  networkTerminated
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DSL port to which the CPE is connected.
OperState	The current operational state of the interface: Unknown Disabled Enabled Testing For a complete definition of OperState as it relates to this group, see *Operating Status* on page 20.
DataRate	The configured data rate, in Kbps, for this interface: 64 128 144
State	The current state of the IDSL interface: none inactive activating activated activatedCPEDetected
Debug	The value of the factory Debug setting.
ActivationInterval	The IDSL port activation interval setting. See *cmBoardTable* on page 61 for details.
TerminationMode	The current ISDN Termination Mode: lineTerminated or networkTerminated. IDSL ports are LT except Port 1 when it is used to acquire network timing.

(*) indicates object used as an index.

cmCpeLogTable

This group displays voice trace information (ASCII trace strings) from the IAD CPE. You can use it when debugging voice problems on the IAD.

To use this group, turn on message logging in the cmCpeIADTable. Messages will then be entered in the cmCpeLogTable. For information about how to use both of these tables, see "CPE Message Logging Table" in Chapter 9, Troubleshooting, of the *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmcpel *[cpe:index]*

Where "cpe:" indicates a command for the CPE.

Example 1: To see all message entries in the log table:

```
geta cmcpelog
Index              MsgNbr             MsgType             TimeStamp
Message

Instance: [CPE:1.13.2, 1]
CPE:1.13.2          1                  DHCP                587565
Sending DHCP request

Instance: [CPE:1.13.2, 2]
CPE:1.13.2          2                  DHCP                587575
Got a DHCP Reply

Instance: [CPE:1.13.2, 3]
CPE:1.13.2          3                  DHCP                587575
ipaddr 10.10.1.5

Instance: [CPE:1.13.2, 4]
CPE:1.13.2          4                  DHCP                587575
iad_hdia_enable is

Instance: [CPE:1.13.2, 5]
CPE:1.13.2          5                  DHCP                587575
caipaddr 10.10.1.1

No more instances
```

Example 2: To see details for entry 4 from the above list:

```
get cmcpelog [cpe:1.13.2,4]
Group: cmCpeLogTable
Instance: [CPE:1.13.2, 4]
Index                =  CPE:1.13.2
MsgNbr               =  4
MsgType              =  DHCP
TimeStamp            =  587575
Message              =  iad_hdia_enable is 1
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the port to which the IAD CPE is connected.
*MsgNbr	The number of the debug output line on the IAD CPE, up to a maximum of 1000 entries.
MsgType	The type of debug message: None (the default) VOICEPATH DSP' DSM DHCP RTP
TimeStamp	The timestamp of the debug message in milliseconds. The default is 0.
Message	The actual debug message, up to 255 characters. The default is " ".

(*) indicates object used as an index.

cmCpePlugAndPlayTable

This group contains objects that reference objects in other tables. For example, values from VoiceGatewayType, VoiceGateway-TypeID, MaxVoiceChannels, and CpeJitterTarget are obtained from the cmInterfaceOptions table.

When a CE200 needs to send the basic plug and play parameters to an IAD, it collects the data from the cmCpeInterfaceOptions and other tables, populates the cmCpePlugAndPlay table, and then sends the data from the latter to the IAD. The IAD saves the data to local storage and then uses it to update other tables.

The CE200 will set a minor alarm and send an SNMP trap if a CPE Plug and Play update fails. For more information, see the *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmcpeplugandplay *[cpe:pii]*

Where "cpe:" indicates a command for the CPE.

Example: To query the CPE connected to port 2, slot 6:

```
get cmcpep [cpe:1.6.2]
Group: cmCpePlugAndPlayTable
Instance: [CPE:1.6.2]
PII                  =  CPE:1.6.2
VoiceGatewayType     =  2
CpeJitterTarget      =  5
MaxVoiceChannels     =  8
CpeDataFunction      =  LAN-extension
DataEncapsulation    =  rfc1483
VoiceGatewayDesc     =  CopperCom
```

Objects:

*PII	The Permanent Interface Identifier of the DSL port to which the CPE is connected.
VoiceGatewayType	When set on a WAN VC, the ID number of the voice gateway upstream of the CE200: 1 to 255. The default is 1 (for None). This ID is obtained from the VoiceGatewayType or VoiceGatewayTypeId objects in the cmInterfaceOptions table. For the description of the gateway, see the VoiceGatewayDesc object below. All DSL ports/VCs that use this WAN VC as their destPii in the cmIface group will inherit this VoiceGatewayType for their CPE PlugAndPlay parameters.
CpeJitterTarget	The target CPE line jitter in milliseconds for real time traffic data (voice traffic): 0 to 255 ms. The default is 0. The CPE can use this value to calculate the fragment size for frame fragmentation of NRT (data) traffic. A zero value means that the CPE should not attempt to control jitter (that is, the CPE should not attempt to fragment data traffic).
MaxVoiceChannels	The maximum number of voice lines the CPE will allow: 0 to 255. The default is 0 if the CPE is not configured for voice. The value 255 indicates that there is no restriction.

CpeDataFunction	The function the CPE will perform for NRT (data) traffic: Unknown (default), LAN-extension, DSU All netmodels other than Cross-Connect must use LAN-extension; Cross-Connect must use DSU.
DataEncapsulation	The type of encapsulation used on this interface: None (the default) rfc1483 rfc1490 HDLC PPP-HDLC rfc1973 Q922 Q922-1490 FRF5 IP-1490 ATM rfc2364-llc rfc2364-null For the VWAN, IP, or CopperVPN netmodels, all Copper Mountain CPEs except the CR202 (IMUX SDSL) use rfc1483. Third-party CPEs and the CopperRocket 202 use rfc1490. The Cross-Connect netmodel can use any of the other listed types, but if the interface is an ATM VC, you will probably use rfc1483. See Appendix A in the *CopperEdge 200 Installation and Operating Guide* for a list of supported encapsulation types. DSL voice VCs (on DSL VCID=22) must specify encapsulation of IP-1490. No other cmIface rows can use IP-1490. Only WAN VCs can use FRF5.
VoiceGatewayDesc	The text description of the type of voice gateway that corresponds to the VoiceGatewayTypeID in the cmInterfaceOptions table. All descriptions are stored in CPE software, not the CE software. If an older CPE does not have the descriptions that correspond to IDs 1 through 5, the CE will supply them. You can use this object to verify that the configuration of the VoiceGatewayTypeId is correct.

(*) indicates object used as an index.

cmCpeT1InterfaceTable
(alias: cpet1)

This group displays configuration information about the T1 interface on the IAD CPE, such as the type and length of the T1 line, transmit clock source, and attenuation of the transmit signal. This information is stored in the CPE, not in the CE200.

Syntax: *command* cpet1 *[cpe:index]*

Where “cpe:” indicates a command for the CPE.

Example: To query the CPE connected to port 2, slot 13:

```
get cpet1 [cpe:1.13.2]
Group: cmCpeT1InterfaceTable
Instance: [CPE:1.13.2]
Index                =  CPE:1.13.2
LineType             =  ESF
```

```
LineCoding            =  B8ZS
TransmitClockSource   =  LoopTiming
LineLength            =  LongHaul
TxAttenuation         =  0dB
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the T1 port to which the CPE is connected.
LineType	The type of line that is implementing the T1 circuit. The only value is ESF.
LineCoding	The method of Zero Code Suppression that is used on this interface. The only value is B8ZS.
TransmitClockSource	The timing source to be used by the interface. The only value is LoopTiming—the recovered receive clock used on the circuit.
LineLength	The length of the DS1 line in meters. The only value is LongHaul.
TxAttenuation	The attenuation on the transmit signal of the DS1 interface when configured for Long Haul: 0dB -7.5dB -15dB

(*) indicates object used as an index.

cmCpeTollBridgeTable
(alias: cpetoll)

This table displays information about each telephone port on an IAD CPE (such as a CR408, CR508, or CR508T). The CPE must be served by a TollBridge TB200 gateway on the WAN side of CE200.

You can use this table for diagnostic purposes. It indicates the current received and transmitted messages on the telephone line on the port (such as AIS, RLCF, LO, and LC), and the current state of a call (such as Ring, GS_Idle, OnHook, DialBreak, or Talk).

Syntax: *command* cpetoll *[cpe:index]*

Where "cpe:" indicates a com3mand for the CPE.

Example: To query voice port 1 on the IAD CPE connected to port 2, slot 13:

```
get cpetoll [cpe:1.13.2,1]
Group: cmCpeTollBridgeTable
Instance: [CPE:1.13.2, 1]
Index                =  CPE:1.13.2
PortNumber           =  1
RxABCDBit            =  4
TxABCDBit            =  1
CallState            =  0
RxABCDValue          =  LO
TxABCDValue          =  AIS
CallStateValue       =  InitLocked
```

Objects:

*Index	PII of the port to which the IAD CPE is connected.
*PortNumber	The analog voice port number on the IAD: 1 to 8.
RxABCDBit	The received ABCD bit value for the IAD port.
TxABCDBit	The transmitted ABCD bit value for the IAD port.
CallState	The internal state of the call for the IAD port.
RxABCDValue	The explanation of the RxABCDBit object: None, Ring, AIS, RLCF, Yellow, LO, LC, INVALID **Note:** The explanation is not a one to one correlation with the RxABCDBit object.
TxABCDValue	The explanation of the TxABCDBit object: None Ring AIS RLCF Yellow LO LC INVALID **Note:** The explanation is not a one to one correlation with the TxABCDBit object.
CallStateValue	The explanation of the CallState object: InitLocked LS_Idle GS_Idle OnHook Ring Ring_Ground ExecRing PostRing DialBreak DialMake Talk PreTalk FwDisk PostFwDisk WaitFwDisk PostPreTalk HwFail **Note:** The explanation is not a one to one correlation with the CallStateBit object.

(*) indicates object used as an index.

cmCpeVoicePortIntervalTable
(alias: cpevint)

This group displays usage statistics for each telephone port on the IAD CPE for a particular time interval. It shows how many times the phone on the port went from on-hook to off-hook, and how many times a connection occurred where voice packets were transmitted. It also shows the amount of time during an interval when people were talking.

The CPE stores statistics in RAM for the most recent 96 intervals, set at 15 minutes each. You can review statistics for any specified interval in the past 24 hours.

Syntax: *command* cmcpevoiceportinterval *[cpe:index, portnumber, intervalnumber]*

Where "cpe:" indicates a command for the CPE.

The *intervalnumber* tells the unit how far back in time to look. If the current interval is less than 15 minutes, it has a value of 1. To see statistics from 15 minutes ago, enter 2; to see statistics from 40 minutes ago, enter 3; and so on.

Example: To query voice port 1 on the CPE connected to port 2, slot 10 for statistics from the fifth interval (one hour back in time):

```
get cpevint [cpe:1.10.2, 1, 5]
Group: cmCpeVoicePortIntervalTable
Instance: [CPE:1.10.2, 1, 5]
Index                =  CPE:1.10.2
PortNumber           =  1
IntervalNumber       =  5
NumOffHookTransition =  1
NumActiveCalls       =  0
TotalActiveCallTime  =  3
```

Objects:

*Index alias: idx	The Permanent Interface Identifier (PII) of the DSL port to which the CPE is connected.
*PortNumber	The number of the port on the CPE.
*IntervalNumber	The number of the 15-minute interval in the past 24 hours: 1 to 96; where 1 is the current interval. The count starts from when the CPE was powered on. For example, if it has been powered on for 3 hours and 8 minutes, interval 1 shows statistics for the current 8 minutes, and interval 3 shows statistics from 38 minutes ago.
NumOffHookTransitions	The number of times that the phone connected to this port changed from OnHook to OffHook.
NumActiveCalls	The number of connections attached to this port that actively transmitted voice packets.
TotalActiveCallTime	The total time in seconds that connections attached to this port actively transmitted voice packets.

(*) indicates object used as an index.

cmCpeVoicePortTable
(alias: cpev)

This group displays configuration data and the status of each telephone port on the IAD CPE. The table indicates not only the IP address of the voice gateway for the selected port, but also the type of voice compression and voice companding used, and the current status of the telephone line on the port.

The table allows an administrator to find out whether a telephone line is in use. It allows someone who is troubleshooting the voice subnet to find out whether the voice gateway is up and functioning appropriately.

Syntax: *command* cmcpevoiceporttable *[cpe:index, portnumber]*

Where "cpe:" indicates a command for the CPE.

Example: To query voice port 1 on the CPE connected to port 2, slot 10 for its configuration data and status:

```
get cpev [cpe:1.10.2,1]
Group: cmCpeVoicePortTable
Instance: [CPE:1.10.2, 1]
Index                =  CPE:1.10.2
PortNumber           =  1
VoiceCAIpAddr        =  192.168.28.1
CallAgentStatus      =  Down
VoiceCompression     =  ADPCM
VoiceCompanding      =  uLaw
HookState            =  OnHook
OperState            =  Disabled
TxGain               =  -29
RxGain               =  -29
```

Objects:

*Index alias: pii	The Permanent Interface Identifier (PII) of the DSL port to which the CPE is connected.
*PortNumber	The number of the voice port connected to the IAD.
VoiceCAIpAddr	The IP address of the upstream Call Agent. The Call Agent will send its IP address to the IAD in the NCS RQNT message. The address is stored in the IAD's NVRAM.
CallAgentStatus	The state of the connection between the IAD and the upstream gateway: Down or Up.
VoiceCompression	The type of voice compression: PCM or ADPCM.
VoiceCompanding	The type of voice companding: aLaw or uLaw.
HookState	The state of the phone connected to the IAD port: OnHook or OffHook.
OperState	The operating state of the port: Disabled (the default) or Enabled. The MaxVoiceChannels setting in the cmCpePlug-AndPlay group determines whether the port is Enabled or Disabled. For a complete definition of OperState as it relates to this group, see *Operating Status* on page 20.

TxGain	The Voice TxGain of this port: 0 to -6 dbm of gain to be added to the transmitted signal. The default is 0.
RxGain	The Voice TxGain of this port: 0 to -6 dbm of gain to be added to the received signal. The default is 0.
SignalType	The type of voice signal: None GroundStart LoopStart

(*) indicates object used as an index.

cmCpeVoicePortTotalTable
(alias: cpevtot)

This group displays usage statistics for each telephone port on the IAD CPE for the period since power up. It indicates how many times the phone on the port went from on-hook to off-hook, and how many times a connection occurred where voice packets were transmitted over the CPE.

Syntax: *command* cmcpevoiceporttotal *[cpe:index, portnumber]*

Where "cpe:" indicates a command for the CPE.

Example: To query voice port 2 on the CPE connected to port 2, slot 11 for statistics for the period since power up:

```
get cpevtot [cpe:1.11.2, 2]
Group: cmCpeVoicePortTotalTable
Instance: [CPE:1.11.2, 2]
Index                =  CPE:1.11.2
PortNumber           =  2
NumOffHookTransition =  65
NumActiveCalls       =  26
TotalActiveCallTime  =  330
```

Objects:

*Index alias: idx	The Permanent Interface Identifier (PII) of the DSL port to which the CPE is connected.
*PortNumber	The number of the voice port on the CPE.
NumOffHookTransitions	The number of times that the phone connected to this port changed from OnHook to OffHook.
NumActiveCalls	The number of connections attached to this port that actively transmitted voice packets.
TotalActiveCallTime	The total amount of time in seconds that connections attached to this port actively transmitted voice packets.

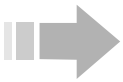
(*) indicates object used as an index.

cmDHCPTable

Use this group to configure subscriber ports (either a DSL physical port or an IMUX bundle) and CPEs for Dynamic Host Configuration Protocol (DHCP). In DHCP, a server assigns an on-demand IP address, which the requesting CPE or host on a premise LAN generally only uses for the duration of a session. The server can be either the CE200 or a service provider's DHCP server.

An IAD, such as the CR408, uses the DHCP table only to obtain the IP address for its voice VC. It currently does not use values from the NetMask, DefaultRouter, or DNSserver objects, although you can set these values. The voice VC on the IAD requires only one IP address, so it always uses the default net mask of 255.255.255.255. However, the netmask configured in the cmDHCP table will be sent to the CPE in the DHCP response.

Depending on the netmodel and CPE, the CE200 will either respond to DHCP requests arriving on the DSL interface, or relay or forward them upstream for assignment by an upstream DHCP server. For more information about DHCP processing, see Chapter 5 in the *CopperEdge 200 Installation and Operating Guide*.



N O T E

If you forget to set the IP address for the IAD in cmDHCPTable, the voice VC on your IAD will not work. Even though you set the IP address of the voice VC in cmIface and the IP address of the voice gateway in cmInterfaceOptions, you must also set the IP address of the IAD in cmDHCPTable.

Syntax: *command* cmdhcp *[pii]*

Example 1: To configure subscriber port 1, slot 9:

```
set cmdh [1.9.1] ip=208.101.66.4 netm=255.255.255.0
```

To display that configuration:

```
get cmdh [1.9.1]
Group: cmDHCPTable
Instance: [1.9.1.0]
PII                   =  1.9.1.0
RowStatus             =  Active
IpAddress             =  208.101.66.4
NetMask               =  255.255.255.0
DefaultRouter         =  0.0.0.0
DNSServer             =  0.0.0.0
Function              =  DHCPRespond
ServerIPAddr          =  0.0.0.0
CircuitID             =  CMTN-1.9.1.0
```

Example 2: To set the IP address for the voice VC on an IAD, include the VC number in the PII:

```
set cmdh [1.6.13.22] ipaddress=10.0.21.2
```

You do not need to enter a value for Netmask or for RowStatus.

Example 3: To configure VC 1. port 7, slot 10 to relay DHCP requests:

```
set cmdh [1.10.7.1] funct=dhcprelay serverip=172.24.121.20 cir=cmtn-
1.10.7.1
Set Successful
```

To display that configuration:

```
get cmdh [1.10.7.1]
Group: cmDHCPTable
Instance: [1.10.7.1]
PII                  =  1.10.7.1
RowStatus            =  Active
IpAddress            =  208.101.66.4
NetMask              =  255.255.255.0
DefaultRouter        =  0.0.0.0
DNSServer            =  0.0.0.0
Function             =  DHCPRelay
ServerIPAddr         =  172.24.121.20
CircuitID            =  CMTN-1.10.7.1
```

Note:

1. After configuring a port for DHCP, briefly bring the interface down and then back up (iftable [pii] adminstat=down/up). This will force the CPE to retrain, at which time it will issue a DHCP request.
2. Cisco routers may reject DHCP request packets with the message, “Inconsistent relay information. Relay information option exists, but giaddr is zero.” By default, Cisco equipment does not trust DHCP packets in which the 'giaddr' field is all zeros. To correct, add the following global statement to the Cisco router configuration: ip dhcp relay information trust-all.

Objects:

*PII	PII of the DSL port or DSL VC.
RowStatus	The operational state of a row in the cmDHCP table: Active or Destroy. Active—In a Get command, the row is actively being used by the CE200. In a Set command, creates a new row. Destroy (alias: delete)—Deletes a row. **Note:** Configuring any object automatically creates the corresponding row.
IpAddress	The IP address that the CE200 sends in response to DHCP requests from the CPE.
NetMask	The net mask that the CE200 sends in response to DHCP requests from the CPE.
DefaultRouter	The default router’s IP address, sent to the CPE when a DHCP request is received over the DSL link. Only non-zero values will be sent in responses.
DNSServer	The DNS server’s IP address, sent to the CPE when a DHCP request is received over the DSL link. Only non-zero values will be sent in responses.

Function	The DHCP function the CE200 is to perform (Respond, Relay, or Forward): If the netmodel is IP: DHCPRespond—The default. For CMCP CPEs, the CE200 acts as a DHCP server by responding to DHCP requests from the CPE. For premise LAN hosts and non-CMCP CPEs, the CE200 relays the requests to an upstream DHCP server. DHCPRelay—The CE200 relays DHCP requests directly to an upstream DHCP server, whose IP address must be entered in the ServerIPAddr object. If the netmodel is Cross-Connect, HDIA, VWAN, or CopperVPN: DHCPRespond—The default. For CMCP CPEs, the CE200 acts as a DHCP server by responding to DHCP requests from the CPE. For premise LAN hosts and non-CMCP CPEs, the CE200 forwards the requests to an upstream DHCP router. DHCPForward—The CE200 forwards DHCP requests to an upstream DHCP router, which then sends the requests to the appropriate DHCP servers.
ServerIPAddr	The IP address of the upstream DHCP server to which the CE200 sends relayed DHCP requests. The default is 0.0.0.0. This object is required for the Relay mode. It overrides the DHCP server IP address that may already be specified in the received DHCP request.
CircuitID	The unique identifier of each DSL port and VC. It must begin with an alpha character. The default is *SystemName-PII*. You can change this to another alphanumeric identifier, but it must start with an alpha character. The CircuitID is used in DHCP requests that are received from directly connected CPEs and are forwarded or relayed to an upstream DHCP server. It is inserted in both the Relay Agent Information and the Client Identifier. The DHCP server can then identify the source of the DHCP request and implement a configuration assignment accordingly.

(*) indicates object used as an index.

cmDS3AtmTable

Use this group to configure and display certain attributes specific to DS3 ATM. It applies only to the DS3 ATM WAN port, not to VCs or links.

Syntax: *command* cmds3atm *[pii]*

Example 1: To configure port 1, slot 4:

```
set cmds3 [1.4.1] parity=enable scr=enable cell=hec
```

Example 2: To display the configuration of port 1, slot 4:

```
get cmds3 [1.4.1]
Group: cmDS3AtmTable
Instance: [1.4.1.0]
PII                  =  1.4.1.0
CbitParityEnable     =  enable
CellPayloadScramblin =  enable
CellMapping          =  hec
```

Objects:

*PII	The Permanent Interface Identifier of the port.
CbitParityEnable alias: Parity	Controls C-bit parity checking on the DS3 ATM interface: enable (the default) or disable.
CellPayloadScrambling alias: Scrambling	Controls payload scrambling on the DS3 ATM interface: enable or disable (the default).
CellMapping alias: Cell	Selects the physical-layer data format for the DS3 ATM interface: hec—Header Error Correction (the default) plcp—Physical Layer Convergence Protocol

(*) indicates object used as an index.

cmDSL24HrTable
(alias: dsl24)

This group displays the cumulative totals from a specified DSL port (SDSL, IDSL, ADSL, or T1) for the full 24 hours preceding the current interval. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

Statistics for the member ports of IMUX bundles are calculated, however they are not summed for the bundles in the cmDSL24Hr table. Use the ifTable group to see certain IMUX bundle performance indicators (page 242).

The cmDSL24Hr table is one of six cmDSL groups that are available for retrieving stored performance statistics from the DSL ports on the CE200. For more information about these groups, see *cmDSL Groups* on page 16.

Syntax: *command* cmdsl24hr *[index]*

Example: To query port 21, slot 6 for its summary statistics from the past 24 hours:

```
get dsl24 [1.6.21]
```

```
Group: cmDSL24HrTable
Instance: [1.6.21.0]
Index                =  1.6.21.0
TxUnderruns          =  0
RxTruncated          =  0
RxBig                =  0
RxAborts             =  0
RxAlignmentErrors    =  0
RxCRCErrors          =  0
RxOverruns           =  0
RxValidFrames        =  6915
TxValidFrames        =  49479
RxNoBuffers          =  0
MinNoiseMargin       =  -1
RxValidOctets        =  345750
TxValidOctets        =  2358325
PortUpTime           =  85500
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DSL interface.
TxUnderruns	The number of transmit underruns.
RxTruncated	The number of frames truncated due to a receiver overrun.
RxBig	The number of frames discarded because they were too large.
RxAborts	The number of aborted frames received.
RxAlignmentErrors	The number of unaligned frames received.
RxCRCErrors	The number of frames received with CRC errors
RxOverruns	The number of frames discarded due to receiver overrun.
RxValidFrames	The number of valid frames received.
TxValidFrames	The number of valid frames transmitted.
RxNoBuffers	The number of received frames discarded due to lack of buffers
MinNoiseMargin	The minimum noise margin encountered during the interval. It is not measurable for IDSL, so IDSL ports will always display 0.
RxValidOctets	The number of octets received in valid frames over the interface during the interval.
TxValidOctets	The number of octets transmitted over the interface during the interval.
PortUpTime	The number of seconds the interface was enabled during the interval.

(*) indicates object used as an index.

cmDSLCurrentTable
(alias: dslcur)

This group displays information about a specified DSL port (SDSL, IDSL, ADSL, or T1) during the current interval. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

IMUX bundles are not supported; use the ifTable to see certain IMUX bundle performance indicators.

The CmDSLCurrent table is one of six cmDSL groups that are available for retrieving stored performance statistics from the DSL ports on the CE200. For more information about these groups, see *cmDSL Groups* on page 16.

Syntax: *command* cmdslcurrent *[index]*

Example: To query port 21, slot 6 for statistics from its current interval:

```
get dslcur [1.6.21]
Group: cmDSLCurrentTable
Instance: [1.6.21.0]
Index                =  1.6.21.0
TxUnderruns          =  0
RxTruncated          =  0
RxBig                =  0
RxAborts             =  0
RxAlignmentErrors    =  0
RxCRCErrors          =  0
RxOverruns           =  0
RxValidFrames        =  0
TxValidFrames        =  0
RxNoBuffers          =  63
MinNoiseMargin       =  50
RxValidOctets        =  0
TxValidOctets        =  0
PortUpTime           =  456
```

Objects:

*Index	Permanent Interface Identifier of the DSL interface.
TxUnderruns	The number of transmit underruns.
RxTruncated	The number of frames truncated due to a receiver overrun.
RxBig	The number of frames discarded because they were too large.
RxAborts	The number of aborted frames received.
RxAlignmentErrors	The number of unaligned frames received.
RxCRCErrors	The number of frames received with CRC errors
RxOverruns	The number of frames discarded due to receiver overrun.
RxValidFrames	The number of valid frames received.
TxValidFrames	The number of valid frames transmitted.
RxNoBuffers	The number of received frames discarded due to lack of buffers
MinNoiseMargin	The minimum noise margin encountered during the interval. It is not measurable for IDSL, so IDSL ports will always display 0.
RxValidOctets	The number of octets received in valid frames over the interface during the interval.
TxValidOctets	The number of octets transmitted over the interface during the interval.
PortUpTime	The number of seconds the interface was enabled during the interval.

(*) indicates object used as an index.

cmDSLIntervalTable
(alias: dslint)

This group displays performance statistics for a specified DSL port (SDSL, IDSL, ADSL, or T1) during a specified interval in the past 96 intervals. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

IMUX bundles are not supported; use the ifTable to see certain IMUX bundle performance indicators.

The cmDSLInterval table is one of six cmDSL groups that are available for retrieving stored performance statistics from the DSL ports on the CE200. For more information about these groups, see *cmDSL Groups* on page 16.

Syntax: *command* cmdslinterval *[index, number]*

The *number* tells the unit how far back in time to look. The most recently completed interval has a value of 1. To see statistics from 30 minutes ago when the interval is set to 15 minutes, enter 2; to see statistics from five hours ago when the interval is set to 60 minutes, enter 5; and so on.

Example: To query port 21, slot 6 for its performance statistics from the second interval:

```
get dslint [1.6.21,2]
Group: cmDSLIntervalTable
Instance: [1.6.21.0, 2]
Index                =  1.6.21.0
Number               =  2
TxUnderruns          =  0
RxTruncated          =  0
RxBig                =  0
RxAborts             =  0
RxAlignmentErrors    =  0
RxCRCErrors          =  0
RxOverruns           =  0
RxValidFrames        =  0
TxValidFrames        =  0
RxNoBuffers          =  63
MinNoiseMargin       =  50
RxValidOctets        =  0
TxValidOctets        =  0
PortUpTime           =  900
TimeTagBegin         =  2000/07/07-08:30:02
TimeTagEnd           =  2000/07/07-08:45:01
```

Note: The cmDSLInterval group does not include the current, in-progress interval. To see statistics for the current interval, use the cmDLScurrent group.

Objects:

*Index	The Permanent Interface Identifier of the interface.
*Number	Number of the interval: 1 to 96, with 1 the most recent full interval, and 96 is the oldest interval.
TxUnderruns	The number of transmit underruns.
RxTruncated	Number of frames truncated due to receiver overrun.
RxBig	Number of frames discarded as too large.
RxAborts	The number of aborted frames received.
RxAlignmentErrors	The number of unaligned frames received.

RxCRCErrors	The number of frames received with CRC errors
RxOverruns	The number of frames discarded due to receiver overrun.
RxValidFrames	The number of valid frames received.
TxValidFrames	The number of valid frames transmitted.
RxNoBuffers	Received frames discarded due to lack of buffers
MinNoiseMargin	The minimum noise margin encountered during the interval. It is not measurable for IDSL, so IDSL ports will always display 0.
RxValidOctets	The number of octets received in valid frames over the interface during the interval.
TxValidOctets	The number of octets transmitted over the interface during the interval.
PortUpTime	The number of seconds the interface was enabled during the interval.
TimeTagBegin	The calendar time when the interval began.
TimeTagEnd	The calendar time when the interval ended.

(*) indicates object used as an index.

cmDSLTodaysTable
(alias: dsltod)

This group displays the cumulative statistics for a specified DSL port (SDSL, IDSL, ADSL, or T1) from midnight last night to the end of the most recently completed interval. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

IMUX bundles are not supported; use the ifTable to see certain IMUX bundle performance indicators.

The cmDSLTodays table is one of six cmDSL groups that are available for retrieving stored performance statistics from the DSL ports on the CE200. For more information about these groups, see *cmDSL Groups* on page 16.

Syntax: *command* cmdsltodays *[index]*

Example: To query port 21, slot 6 for its cumulative statistics from midnight last night to the current interval:

```
get dsltod [1.6.21]
Group: cmDSLTodaysTable
Instance: [1.6.21.0]
Index                =  1.6.21.0
TxUnderruns          =  0
RxTruncated          =  0
RxBig                =  0
RxAborts             =  41539
RxAlignmentErrors    =  79856
RxCRCErrors          =  89623
RxOverruns           =  0
RxValidFrames        =  0
TxValidFrames        =  586
RxNoBuffers          =  24027
MinNoiseMargin       =  32
```

```
RxValidOctets          =  0
TxValidOctets          =  30472
PortUpTime             =  271
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DSL interface.
TxUnderruns	The number of transmit underruns.
RxTruncated	The number of frames truncated due to a receiver overrun.
RxBig	The number of frames discarded because they were too large.
RxAborts	The number of aborted frames received.
RxAlignmentErrors	The number of unaligned frames received.
RxCRCErrors	The number of frames received with CRC errors
RxOverruns	The number of frames discarded due to receiver overrun.
RxValidFrames	The number of valid frames received.
TxValidFrames	The number of valid frames transmitted.
RxNoBuffers	The number of received frames discarded due to lack of buffers
MinNoiseMargin	The minimum noise margin encountered during the interval. It is not measurable for IDSL, so IDSL ports will always display 0.
RxValidOctets	The number of octets received in valid frames over the interface during the period.
TxValidOctets	The number of octets transmitted over the interface during the period.
PortUpTime	The number of seconds the interface was enabled during the period.

(*) indicates object used as an index.

cmDSLUpTimeTable
(alias: dslup)

This group displays cumulative statistics for a specified DSL (SDSL, IDSL, ADSL, or T1) port since the previous system restart. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

IMUX bundles are not supported; use the ifTable to see certain IMUX bundle performance indicators.

The cmDSLUpTime table is one of six cmDSL groups that are available for retrieving stored performance statistics from the DSL ports on the CE200. For more information about these groups, see *cmDSL Groups* on page 16.

Syntax: *command* cmdsluptime *[index]*

Example: To query port 21, slot 6 for its cumulative statistics from the last system restart to the current interval:

```
get dslup [1.6.21]
Group: cmDSLUpTimeTable
Instance: [1.6.21.0]
Index                =  1.6.21.0
TxUnderruns          =  0
RxTruncated          =  0
RxBig                =  0
RxAborts             =  41539
RxAlignmentErrors    =  79856
RxCRCErrors          =  89623
RxOverruns           =  0
RxValidFrames        =  0
TxValidFrames        =  586
RxNoBuffers          =  24090
MinNoiseMargin       =  -32
RxValidOctets        =  0
TxValidOctets        =  30472
PortUpTime           =  271
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DSL interface.
TxUnderruns	The number of transmit underruns.
RxTruncated	The number of frames truncated due to a receiver overrun.
RxBig	The number of frames discarded because they were too large.
RxAborts	The number of aborted frames received.
RxAlignmentErrors	The number of unaligned frames received.
RxCRCErrors	The number of frames received with CRC errors
RxOverruns	The number of frames discarded due to receiver overrun.
RxValidFrames	The number of valid frames received.
TxValidFrames	The number of valid frames transmitted.
RxNoBuffers	The number of received frames discarded due to lack of buffers
MinNoiseMargin	The minimum noise margin encountered during the interval. It is not measurable for IDSL, so IDSL ports will always display 0.
RxValidOctets	The number of octets received in valid frames over the interface during the interval.
TxValidOctets	The number of octets transmitted over the interface during the interval.
PortUpTime	The number of seconds the interface was enabled during the interval.

(*) indicates object used as an index.

cmDSLYesterdaysTable
(alias: dslyes)

This group displays the cumulative statistics for a specified DSL (SDSL, IDSL, ADSL, or T1) port for the 24-hour period from midnight yesterday to midnight last night. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

IMUX bundles are not supported; use the ifTable to see certain IMUX bundle performance indicators.

The cmDSLYesterdays table is one of six cmDSL groups that are available for retrieving stored performance statistics from the DSL ports on the CE200. For more information about these groups, see *cmDSL Groups* on page 16.

Syntax: *command* cmdslyesterdays *[index]*

Example: To query port 21, slot 6 for its cumulative statistics from midnight yesterday to midnight last night:

```
get dslyes [1.6.21]
Group: cmDSLYesterdaysTable
Instance: [1.6.21.0]
Index                = 1.6.21.0
TxUnderruns          =  0
RxTruncated          =  0
RxBig                =  0
RxAborts             =  0
RxAlignmentErrors    =  0
RxCRCErrors          =  0
RxOverruns           =  0
RxValidFrames        =  0
TxValidFrames        =  0
RxNoBuffers          =  0
MinNoiseMargin       =  0
RxValidOctets        =  0
TxValidOctets        =  0
PortUpTime           =  0
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DSL interface.
TxUnderruns	The number of transmit underruns.
RxTruncated	The number of frames truncated due to a receiver overrun.
RxBig	The number of frames discarded because they were too large.
RxAborts	The number of aborted frames received.
RxAlignmentErrors	The number of unaligned frames received.
RxCRCErrors	The number of frames received with CRC errors
RxOverruns	The number of frames discarded due to receiver overrun.
RxValidFrames	The number of valid frames received.
TxValidFrames	The number of valid frames transmitted.
RxNoBuffers	The number of received frames discarded due to lack of buffers

MinNoiseMargin	The minimum noise margin encountered during the interval. It is not measurable for IDSL, so IDSL ports will always display 0.
RxValidOctets	The number of octets received in valid frames over the interface during the period.
TxValidOctets	The number of octets transmitted over the interface during the period.
PortUpTime	The number of seconds the interface was enabled during the period.

(*) indicates object used as an index.

cmDsx1ConfigTable
(alias: cmds1cfg)

This group contains configuration objects for DS1 interfaces (Quad T1 WAN, DS1/E1 IMA, or DS1 DSL) that are not included in the standard DS1 MIB. Two of the objects, *RxFdlLoopbackCmdProc* and *PortDownConfig*, are not supported by the Quad T1 WAN interface. The *CellPayloadScrambling* object is supported only on the DS1/E1 IMA Module.

The cmDsx1Config table is important because the DS1 line is a twisted pair, not a coaxial cable, so it is more sensitive to interference. You need to take into consideration the length of the line when setting the transmit pulse template and dB setting.

For this release, the table has been enhanced with the addition of a new object, *FDL*, which you will use when setting up or tearing down a custom FDL loopback.

In this release, the only known-valid values for the FDL object are 18 and 36. Use 18 to set up the loop between the CopperEdge and the device at the other end. Use the second value, 36, to reset (or tear down) the loop between the CE and the device at the other end. To properly perform a custom FDL loopback test, you must configure/display *cmDsx1ConfigTable*, *Dsx1ConfigTable*, and the *cmLoopTable*.

Setting up a custom FDL loop and transmitting test patterns should be done only by experienced network administrators. Erroneous transmission of test patterns over the network and/or allowing test patterns to reach the wrong equipment could result in damage or service interruptions to the network.

Syntax: *command* cmdsx1config *[lineindex]*

Quad T1 WAN modules can only be installed in slots 1.3, 1.4, 1.15, and 1.16.

Example: To query port 7, slot 10:

```
get cmdsx [1.10.7]
Group: cmDsx1ConfigTable
Instance: [1.10.7.0]
LineIndex            =  1.10.7.0
LineLength           =  0-133feet
LongHaulTxAttenuatio =  0dB
RxFdlLoopbackCmdProc =  enabled
PortDownConfig       =  generateAis
```

Objects:

*LineIndex	The Permanent Interface Identifier (PII) of the DS1/E1 interface.

LineLength	Physical length of the DS1 connection at this interface in feet. This sets the transmit-pulse template. Valid options are: 0-133feet 133-266feet 266-399feet 399-533feet (Default) 533-655feet longHaul Note: This object **not applicable to E1** interfaces.
LongHaulTxAttenuation	Attenuation of the transmit signal on this interface when configured for LineLength=longHaul: 0 dB (default) -7.5 dB -15 dB -22.5 dB If LineLength is configured for one of the short haul distances, this object is ignored.
RxFdlLoopbackCmdProc	Allows the port to process and act on FDL-based loopback commands: enabled (default) disabled (FDL commands ignored). **Note:** This object is **not applicable to E1** interfaces, and is not supported by Quad T1 WAN modules.
PortDownConfig	When the T1 port is down, the output of the port is either silent (no spectrum) or generating an AIS signal. The two settings are: noOutput generateAis (default) **Note:** This object is not supported by Quad T1 WAN modules.
FdlCode	This parameter defines the first byte of the FDL code, sent when *sendCode* on the *dsx1ConfigTable* is set to *OtherPattern*. Values must be divisible by 2. Value range is: 0 - 126. The usual value for setup is 18. This value is used with many SmartJacks. Other devices may require other values for setup. For reset (or tear-down) the most widely used value is 36. This value is used with many SmartJacks. Other devices may require other values for tear-down or reset. **Not supported in** Quad T1 WAN module.
CellPayloadScrambling	Enable or disable payload scrambling. Default: Disable Note: This object **applies to DS1/E1 IMA only**, and is not supported by other DS1 modules.

(*) indicates object used as an index.

cmEndPointConfigTable
(alias: epconfig)

This group allows you to specify which CPEs are eligible for bulk downloading of software. The DownloadStatus object provides information about the process of downloading new software to the CPE. Other objects provide information about the starting and ending times for the process.

Syntax: *command* cmendpointconfig *[pii] object_1 object_2*

To specify an IMUX bundle, you must enter the PII of one or more of the DSL physical ports in the bundle. Bundle PIIs (such as 1.51.2) are not supported by cmEndPointConfig

Example 1: To set the BulkDownload eligibility for the CPE connected to port 16, slot 9:

```
set epconfig [1.9.16] bu=enabled
```

RowStatus is Active by default. You only need to specify this object when you want to delete a row (rowstatus=destroy).

Example 2: To query port 16, slot 9 for its configuration:

```
get epconfig [1.9.16]
Group: cmEndPointConfigTable
Instance: [1.9.16.0]
PII                 =  1.9.16.0
RowStatus           =  Active
BulkDownload        =  Enabled
DownloadStatus      =  None
DownloadStartTime   =  ""
DownloadEndTime     =  ""
```

Objects:

*PII	The PII of the physical DSL port to which this configurations applies
RowStatus	The operational state of a row in the cmEndPoint-Config table: Active (the default)—In a Get command, the row is actively being used by the CE200. In a Set command, creates a new row. Destroy (alias: delete)—Deletes a row. **Note:** Configuring any object automatically creates the corresponding row.
BulkDownLoad	Specifies whether the port is eligible for bulk download of CPE software: Enabled or Disabled (the default).

DownLoadStatus	The status of the Bulk Download *relative to this IMUX CPE*: None—No BulkDwnload command has been executed since powerup. Pending—The BulkDownload command has been issued, this CPE is eligible, but its download has not started. InProgress—The download to this CPE has started but is not complete. Failed—The download to this CPE has failed. Completed—The download to this CPE has successfully completed. NotDownloadable—This CPE does not support software download.
DownloadStartTime	The time the download started.
DownloadEndTime	The time the download completed or failed.

(*) indicates object used as an index.

cmEndPointPortTable
(alias: epport)

This group displays information about the IMUX CPE connected to the specified SDSL/IDSL port. Each IMUX CPE is identified with a unique EndPointID, usually derived from the CPE's MAC address.

Although information is available for every SDSL/IDSL port served by the CE200, cmEndPointPort is especially useful for reviewing the configuration of IMUX DSL bundles and for troubleshooting multilink configuration problems.

Each entry in the cmEndPointPort group is identified by a BasePII (the PII of the DSL port you entered as part of the instance) and an arbitrary Number assigned by the CE200. If the IMUX CPE is served by more than one link (such as an IMUX bundle), the Number object is uniquely associated with a DSL port (PartnerPII) connected to the same end point CPE as the specified (BasePII) DSL port. For the first instance of the Number (Number=1) in an IMUX bundle, and for IMUX CPEs served by a single link, the BasePII and PartnerPII are the same.

N O T E

If you assign DSL ports to a bundle not associated with a trained and connected IMUX CPE, those ports' end point IDs will return zero and the number of matching end points will not be counted. If an IMUX CPE is later connected and trained, the CE200 will set an alarm and send a trap if the end point ID of the second port to train does not match the end point ID of the first.

Syntax: *command* cmendpointport *[basepii, number]*

Example 1: To list all CPEs connected to all ports starting with port 10, slot 10:

```
geta epport [1.10.10,0]
BasePII           Number    EndPointID           PartnerPortPII
PartnerBundlePII  NumMatchingEndPoint

Instance: [1.10.10.0, 1]
1.10.10.0         1         0.60.58.1.4a.1d.0.0  1.10.10.0
1.51.1.0          2

Instance: [1.10.10.0, 2]
1.10.10.0         2         0.60.58.1.4a.1d.0.0  1.10.9.0
1.51.1.0          2

Instance: [1.10.11.0, 1]
1.10.11.0         1         0.0.0.0.0.0.0.0.0.0  1.10.11.0
0.0.0.0           1
```

Example 2: To see information about the end point CPE and ports assigned to the same bundle as the specified DSL port (port 10, slot 10, and the first port in this bundle):

```
get epport [1.10.10,1]
Group: cmEndPointPortTable
Instance: [1.10.10.0, 1]
BasePII             =  1.10.10.0
Number              =  1
EndPointID          =  0.60.58.1.4a.1d.0.0.0.0.0.0.0.0.0.0
PartnerPortPII      =  1.10.10.0
PartnerBundlePII    =  1.51.1.0
NumMatchingEndPoint =  2
```

Example 3: To see the next port in the same bundle as shown in example 2:

```
getn epport
Group: cmEndPointPortTable
Instance: [1.10.10.0, 2]
BasePII             =  1.10.10.0
Number              =  2
EndPointID          =  0.60.58.1.4a.1d.0.0.0.0.0.0.0.0.0.0
PartnerPortPII      =  1.10.9.0
PartnerBundlePII    =  1.51.10.0
NumMatchingEndPoint =  2
```

Note: The EndPointID object will return all zeros if the port is not trained, or if a CPE that does not support IMUX is connected and trained.

Objects:

*BasePII	The Permanent Interface Identifier of the DSL port.
*Number	The number arbitrarily assigned by the DSLAM to the port reporting the same end point ID as the base PII. For example, the second port on a 2-port IMUX CPE, with both lines trained, will have the number 2.
EndPointID	The end point ID reported by both the CPE connected to the BasePII and the CPE connected to the PartnerPortPII. The format is *byte1.byte2...*, where bytes are displayed as hex numbers.
PartnerPortPII	The Permanent Interface Identifier of the port connected to a CPE reporting the same end point ID as the BasePII. If Number=1, then PartnerPortPII is the same as BasePII.
PartnerBundlePII	The Permanent Interface Identifier of the bundle in which the PartnerPortPII is configured. If a partner port is not assigned to a bundle, this field is all zeros (an IMUX bundle PII has the format 1.51.n, where n is a number from 1 to 63).

NumMatchingEndPoint	The total number of trained DSL ports with the same end point ID as the BasePII port. In this release, this value is always 1 (for single link) or 2 (for multilink).

(*) indicates object used as an index.

cmFile

This group allows you to perform two major functions: configure/display data about the remote file server holding new CPE code; configure/enable the Syslog output.

For the first function, cmFile allows you to indicate the remote file server on which CPE download code is stored, including its name and IP address, the FTP user name and password, and the directory path where files are stored. For the second function, cmFile allows you to configure Syslog output and route it to a Unix Syslog server. Two read-only objects also provide information about the location of the system boot files, and the path to the system software in the CopperEdge memory.

Syntax: *command* cmfile *object_1...object_8*

Example 1: To send Syslog messages to a server with address 209.14.14.4:

```
set cmfile syslogaddr=209.141.14.4
```

Example 2: To display information about the current server:

```
get cmfile
Group: cmFile
BootDevice          =  Flash
FSName              =  ce200
FSAddr              =  0.0.0.0
FSUsername          =  ""
FSPassword          =  ****
FSDir               =  ""
FLASHDir            =  P:/ce200/
BootFSAddr          =  0.0.0.0
BootFSUsername      =  user
BootFSPassword      =  ****
BootFSDir           =  ""
SysLogAddr          =  209.141.14.4
SysLogPort          =  514
SysLogFacility      =  1
SysLogPriority      =  6
```

Objects:

BootDevice	The name of the default device for obtaining files (flash or file server).
FSName	The name of the remote file server.
FSAddr	The IP address of the external file server.
FSUserName	The user name to be used for remote file server FTP operations.
FSPassword	The password to be used for remote file server FTP operations.
FSDir	The base directory name for all FTP requests to the file server. The CE200 appends any required subdirectory name, based on the same file and directory structure as used in its flash.

FlashDir	The root directory path for this system's files in the flash file system.
BootFSAddr	The IP address of the file server specified as the Boot file server (Boot FS). For internal use only.
BootFSUsername	The username to be used when retrieving files from the Boot FS. For internal use only.
BootFSPassword	The password to be used when retrieving files from the Boot FS. For internal use only.
BootFSDir	The base directory on the Boot FS where the Copper Mountain files are stored. For internal use only.
SysLogAddr	The IP address of the server to which Syslog messages are to be sent. Configuring a "non-zero" IP address enables the Syslog function; entering SyslogAddr=0 disables the function.
SysLogPort	The UDP port number of the Syslog server. The default is 514, which is the standard designation for Unix Syslog servers.
SysLogFacility	The Client Facility Code of this CE200: 0 to 99, where 1 is the default (1 is the standard code for "user"). Other codes have other standard uses, such as e-mail and news.
SysLogPriority	The Syslog level assigned to all messages from this CE200: 0 to 7, where 6 is the default (the standard designation for "information"). Lower numbers denote increasing levels of urgency (0=Emergency). Note that there is no benefit to setting the priority to any higher level; the setting has no effect on the number or type of messages logged by the CopperEdge.

Notes

1. The File Server referred to in FSName, FSAddress, etc., tells the CE200 where to look for software stored on a remote machine. In this release, remote file server applies only to nonessential software, such as software for IMUX-capable CPEs. This file server is not the one configured in cmMaintCmd for backup/restore of system Config files.
2. The file server referred to in SysLogAddr and SysLogPort tells the CE200 where to send SysLog messages. The default values of the four Syslog objects have been carefully selected to provide diagnostic information without disruption to other services. We recommend that you deviate from the defaults only in cases of operational necessity.
3. Make sure that the Syslog server's IP address is reachable through the route table of the client CopperEdge units.
4. When you enable the CE200 as a Syslog client, it immediately transmits a message to the specified server address to introduce itself. To verify your configuration, check the server's Syslog file to confirm receipt of a message originating at the IP address of each CE200.

cmFilterTable

Use this group to configure and display IP filters and their parameters. Note that filters can be applied to individual interfaces/VCs or to IMUX bundles using their bundle PIIs.

Syntax: *command* cmfilter *[index, filternumber] object_1...object_11*

Example 1: To add a new filter to the list for the DSL interface at port 3, slot 9:

```
set cmfilt [1.9.3, 3] func=activate srca=208.101.66.4
```

The number 3 specifies that this is the third filter in the list.

Example 2: To query port 3, slot 9 for the configuration of the fourth filter in the list:

```
get cmfilt [1.9.3, 4]
Group: cmFilterTable
Instance: [1.9.3.0, 1]
Index                =  1.9.3.0
FilterNumber         =  4
Function             =  ACTIVE
SrcMask              =  0.0.0.0
SrcAdrs              =  208.101.66.4
DstMask              =  0.0.0.0
DstAdrs              =  0.0.0.0
IpProtocol           =  0
SrcPortCompare       =  ANY
SrcPort              =  0
DstPortCompare       =  ANY
DstPort              =  0
Action               =  BLOCK
```

Note:

1. To filter IP packets destined for the CE200 specify the system using its virtual PII: 1.0.0.0
2. The cmFilter table for the CE200 includes a static filter that will immediately discard any ICMP Redirect messages destined for the CE200.
3. To apply filtering to a T1/E1 IMA group, use the “virtual” PII reference for the IMA group. Do *not* use individual link PIIs to configure an IMA group.

Objects:

*Index alias: ifx	Permanent Interface Identifier (PII) of the DSL interface.
*FilterNumber alias: fnum	The integer assigned to this filter; its place in the filter list for the DSL interface.
Function	Function of this command (activate or delete) for the specified PII
SrcAdrs	Source IP address; the address associated with the incoming IP packet
SrcMask	Source subnet mask applied as a bitwise AND to the Source IP address.
SrcAdrsCompare	Criterion against which the Source IP Address is measured: *Any, EQ* (default)*, LT, GT, NE*) ***Caution: “Any” and “NE” sound alike***
DstAdrs	Destination IP address; the IP address of the interface to which the packet is directed.
DstMask	Destination subnet mask applied as a bitwise AND to the destination IP address

DstAdrsCompare	Criterion with which the Destination IP Address is measured: *Any, EQ* (default), *LT, GT, NE*) ***Caution: "Any" and "NE" sound alike***
IpProtocol alias: proto	IP Protocol Identifier (to match against the IP Protocol IDs of incoming messages): 0 = IP 1= ICMP 2 = IGMP 3 = GGP 6 = TCP 17 = UDP
SrcPort	The IP Port number of the source of the packet
SrcPortCompare alias: srcpc	Criterion with which the Source Port number is compared: *Any, EQ* (default), *LT, GT, NE*) ***Caution: "Any" and "NE" sound alike***
DstPort	The IP Port number of the packet's Destination
DstPortCompare alias dstpc	Criterion with which the Destination Port number is compared: *Any, EQ* (default), *LT, GT, NE*) ***Caution: "Any" and "NE" sound alike***
Action	Action taken on messages that match the filter: *pass, block, chain, redirect*
RedirectPii	Outgoing interface PII for *Action=Redirect*. *Redirect* is valid only for packets arriving at the CE from subscriber interfaces (not WAN side interfaces). The outgoing interface must be a WAN-side interface (not a subscriber interface). Redirect messages to subscriber interfaces are automatically discarded.
NumMatches	The number of packets that match the rules imposed by this filter. The object is writable to allow you to reset it (Num=0). This counter increments independently for each rule in a chain.

(*) indicates object used as an index.

cmFrCircuitTable

Like the RFC-1315 MIB group, frCircuit, this Copper Mountain group allows you to display information about specific virtual circuits. Settings and integer values apply only to specified WAN or DSL VCs, or to IMUX bundles which are functioning as VCs.

This table is also used to control and monitor the per-VC rate limiting function on SDSL, DS1/DSL, and G.SHDSL interfaces[1], in both the upstream and downstream directions. Note, however, that rate limiting includes only the user payload and IP/Ethernet headers. It does not include RFC 1483/1490 overhead, data link encapsulation (e.g. Q.922 for frame relay interfaces and ATM cell overhead for ATM interfaces), or physical layer overhead such as control channel information, framing bits, or bit stuffing.

There is no enforcement of Committed Information Rate (CIR) on DSL VCs. Since there is no enforcement of CIR on the receive side of WAN VCs, the CE200 has no basis for assigning or honoring

1. Rate limiting is not supported on IDSL, ADSL or G.Lite modules, nor on IMUX bundles.

bandwidth commitments on the transmit side of the corresponding DSL VCs. For more information on congestion management, see *fr-CircuitTable* on page 226.

Syntax: *command* cmfrcircuit *[ifindex]*

Example: To configure rate limiting on a VC of 512 kbps upstream, and 1.024 Mbps downstream:

```
set cmfrc [1.6.3.528] RxRateLimit=512000 TxRateLimit=1024000
Set Successful

set cmfrc [1.6.3.528]
Group: cmFrCircuitTable
Instance: [1.6.3.528]
IfIndex              =  1.6.3.528
Dlci                 =  528
OperState            =  Enabled
FramesDroppedTxUnava =  0
FramesDroppedRxUnava =  0
FramesDroppedExcess  =  0
RxRateLimit          =  512000
RxCurrentRateLimit   =  512000
RxBurstSize          =  51200
RxDiscardsDueToLimit =  0
TxRateLimit          =  1024000
TxCurrentRateLimit   =  1024000
TxBurstSize          =  102400
TxDiscardsDueToLimit =  0
```

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the ifIndex object from the corresponding entry in the IfTable. CE200 VCs are layered onto existing entries in the IfTable.
Dlci	The DLCI for this virtual circuit
OperState	The operational state of the VC: Unknown Disabled Enabled Testing OperState is Enabled when the following conditions are met; otherwise the PVC is Disabled. • The cmFrDlcmi OperState is enabled for the frame relay link. • The frCircuit State object for this PVC is Active.
OperState (cont'd)	• If LMI (Local Management Interface) is enabled on the link, the frame relay DCE or NNI peer reports this PVC status as Active. • If this PVC is configured as NetModel=Cross-Connect, the PVC to which the cmFrCircuit OperState object is mapped is Enabled.
FramesDroppedTx-Unavail	The number of transmit frames dropped on this circuit due to one of the following conditions: • The link is down. • The VC is inactive. • The Discard Eligible frame is seen during congestion.
FramesDroppedRx-Unavail	The number of receive frames dropped on this circuit due to the link being down or the VC being inactive.
FramesDroppedExcess	The number of frames dropped on this circuit because the hold queue was full.

RxRateLimit	Specifies the rate (in bps) to which the upstream data flow will be limited. Valid entries are 0 and 64000 through 8064000. A value of 0 (the default) indicates that no rate limiting will be done.
RxCurrentRateLimit	Lists the upstream data-rate limit (in bps) currently in effect. A value of 0 means no rate-limiting is in effect. If RxRateLimit is greater than the trained rate, then rate limiting is automatically disabled and CurrentRateLimit will be reset to 0.
RxBurstSize	Upstream burst size in bits. Valid values are 12000 (MTU of 1500 Bytes) up to the value of RxRateLimit. Default value is 10% of RxRateLimit. This value will likely need to be adjusted to a higher value in order to prevent excessive discards, and thus retransmissions for TCP applications.
RxDiscardsDueToLimit	The number of dropped packets in the upstream direction due to exceeding the rate limit.
TxRateLimit	Specifies the rate (in bps) to which the downstream data flow will be limited. Valid entries are 0 and 64000 through 8064000. A value of 0 (the default) indicates that no rate limiting will be done.
TxCurrentRateLimit	Lists the downstream data-rate limit (in bps) currently in effect. A value of 0 means no rate-limiting is in effect. If TxRateLimit is greater than the trained rate, then rate limiting is automatically disabled and CurrentRateLimit will be reset to 0.
TxBurstSize	Downstream burst size in bits. Valid values are 12000 (MTU of 1500 Bytes) up to the value of TxRateLimit. Default value is 10% of TxRateLimit. This value will likely need to be adjusted to a higher value in order to prevent excessive discards, and thus retransmissions for TCP applications.
TxDiscardsDueToLimit	The number of dropped packets in the upstream direction due to exceeding the rate limit.

(*) indicates object used as an index.

cmFrDlcmiTable

Like the RFC-1315 MIB Group, frDlcmi, this group displays information relating to the Frame Relay Data Link Connection Management Interface. All settings and values in cmFrDlcmi group apply to the entire Frame Relay physical interface and all of its associated PVCs, whether over a WAN link, a DSL line, or an IMUX bundle.

Syntax: *command* cmfrdlcmi *[ifindex] object_1...object_5*

Example 1: To disable the frame relay interface at port 1, slot 3:

```
set cmfrd [1.3.1] adminstate=disabled
```

Example 2: To query port 1, slot 3 for its configuration:

```
get cmfrd [1.3.1]
Group: cmFrDlcmiTable
Instance: [1.3.1.0]
IfIndex              = 1.3.1.0
OperState            =  Disabled
AdminState           =  Disabled
LMIMode              =  dte
DceN392              =  3
DceN393              =  4
DceT392              =  15
FirstDLCI            =  0
RcvUnknownDLCI       =  0
LMITxStatusEnquiries =  0
LMITxStatusResponses =  0
LMIRxStatusEnquiries =  0
LMIRxStatusResponses =  0
LMIRxUpdates         =  0
UnknownLMIMessagesRc =  0
LostLMISequences     =  0
```

Note: Always disable the link using cmFrDlcmi (admin=dis) before changing the LMI (Local Management Interface) state or any other link attributes. Modifying link attributes is service-affecting.

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the physical port.
OperState	The operational state of the frame relay link: Unknown Disabled Enabled Testing OperState is Enabled when all of the following conditions are met; otherwise the state is Disabled. • The physical port ifTable OperStatus is up. • The cmFrDlcmi AdminState object is enabled. • If LMI is enabled on the link, the LMI protocol layer is up.
AdminState	The status of the frame relay link: Unknown Disabled Enable If disabled, any VCs connected to this port are also disabled.
LMIMode	If LMI polling is used, how is it being performed: dce—the interface is being polled dte—the interface is polling nni—the interface is both polling and being polled For DSL VCs, LMIMode is always dce.

DceN392	The maximum number of unreceived or invalid status inquiries detected before the interface will be declared down: 1 to 10. The default is 3.
DceN393	The number of polling intervals over which the error threshold (DCEN392) is counted. For example, if the DCEN392 setting is 3 and the DCEN393 setting is 4, the function must collect 3 unreceived or invalid status inquiries over 4 intervals to declare the interface down: 1 to 10. The default is 3.
DceT392	The value of the polling verification timer in units of seconds. The value is the maximum amount of time the DCE will wait to receive a Status Inquiry message: 5 to 30 seconds.
FirstDLCI	This object is not supported in this release.
RcvUnknownDLCI	The number of frames received with unknown DLCI
LMITxStatusEnquiries	The total number of LMI status inquiries sent by the DTE on this interface.
LMITxStatusResponses	The total number of LMI status responses sent by the DCE on this interface.
LMIRxStatusEnquiries	The total number of LMI status inquiries received by the DCE on this interface.
LMIRxStatusResponses	The total number of LMI status Responses received from the DCE on this interface.
LMIRxUpdates	The number of LMI asynchronous status updates received by the DTE on this interface.
UnknownLMIMessages-Rcvd	The number of Unknown LMI messages received on this interface.
LostLMISequences	The number of times a loss of sequencing in received LMI messages was detected by the DTE on this interface.

(*) indicates object used as an index.

cmFrf5DlcmiTable

Use this group to set the LMI (Local Management Interface) control and display statistical parameters associated with running LMI over a WAN VC as defined in FRF.5.

Rows in this table are automatically created when WAN VCs with an encapsulation type of FRF5 are created, and automatically deleted when the encapsulation type is changed from FRF5 to something else. To configure a WAN VC for FRF.5, an ATM module must exist in the corresponding slot.

The FRF.5 feature applies only to WAN VCs configured with a netmodel of Cross-Connect and encapsulation type of FRF5.

Syntax: *command* cmfrf5dlcmi *[ifindex] object_1*

The *ifindex* represents a logical VC, not the physical port.

Example 1: To enable FRF.5 LMI on VC 120, port 1, slot 4

```
set cmfrf5 [1.4.1.120] st=q933
```

Example 2: To query port 1, slot 4 for the FRF.5 configuration of VC 120:

```
get cmfrf5 [1.4.1.120]
Group: cmFrf5DlcmiTable
Instance: [1.4.1.120]
IfIndex              = 1.4.1.120
State                =  Q933-Annex-A-BiDir
LastRcvdPVCState     =  disabled
DteT391              =  180
DteN391              =  1
DceT392              =  200
DceN392              =  3
DceN393              =  4
RcvUnknownDLCI       =  0
LMITxStatusEnqs      =  902
LMITxStatusRsps      =  0
LMIRxStatusEnqs      =  0
LMIRxStatusRsps      =  0
UnknownLMIMsgsRcvd   =  0
LostLMISequences     = 0
```

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the WAN VC. It distinguishes the FRF.5 entries from each other since they use the same physical PII.
State	Specify whether the Q.933 Annex-A LMI procedures over the WAN VC are to be activated: noLmiConfigured (the default) Q933-Annex-A-BiDir
LastRcvdPVCState	The remote FR PVC state as reported in the most recently received Q.933-Annex A full status message: unknown disabled enabled The value of “unknown” is displayed if:cmFrf5DlcmiSate is set to noLmiConfigured; or the CE200 is unable to determine the state from the received full status message.
DteT391	The status inquiry interval: the number of seconds between successive status inquiry messages.
DteN391	The number of status inquiry intervals before a full status inquiry.
DceT392	The polling verification timer in seconds. The system stops the timer when it receives a STATUS ENQUIRY message from the polling device, and starts the timer when it responds with a STATUS message. If the timer expires, it indicates an error in the polling process.
DceN392	The status inquiry error threshold: the maximum number of unanswered Status Enquiries the equipment is to accept before declaring the interface down.
DceN393	The number of status polling intervals over which the error threshold is counted. This is the number of Monitored Events.

RcvUnknownDLCI	The number of frames with an unknown DLCI received on this interface.
LMITxStatusEnqs	The number of LMI status enquiries sent by the FRF.5 process on this interface.
LMITxStatusRsps	The number of LMI status responses sent by the FRF.5 process on this interface.
LMIRxStatusEnqs	The number of LMI status enquiries received by the FRF.5 process on this interface.
LMIRxStatusRsps	The number of LMI status responses received by the FRF.5 on this interface.
UnknownLMIMsgsRcvd	The number of unknown LMI messages received by the FRF.5 on this interface.
LostLMISequences	The number of times a loss of sequencing in received LMI messages was detected by the FRF.5 process on this interface.

(*) indicates object used as an index.

cmGroupFilterTable

This MIB group, derived from and essentially identical to the *cmFilter* MIB group, is used to display or configure IP filters applicable to CopperVPN groups. The index objects are the *VPNGroupPii* and the sequentially assigned integer designating the *FilterNumber*.

Example 1:

```
getn cmgroupfilter

                    Group: cmGroupFilterTable
                    Instance: [1.3.1.16, 1]
                    FilterNumber=  1
                    Function=  ACTIVE
                    SrcAdrs=  10.254.8.33
                    SrcMask=  255.255.255.248
                    SrcAdrsCompare=  EQ
                    DstAdrs=  0.0.0.0
                    DstMask=  0.0.0.0
                    DstAdrsCompare=  ANY
                    IpProtocol=  0
                    SrcPort=  0
                    SrcPortCompare=  ANY
                    DstPort=  0
                    DstPortCompare=  ANY
                    Action=  PASS
                    RedirectPII=  0.0.0.0
                    NumMatches=  0
```

Example 2: In this (partial) example, the cmGroupFilter applies to the CopperVPN group interface at location 1.3.1.30. The number 4 means that the specified filter is the fourth in the list for the group interface.

```
set cmgroupfilter [1.3.1.30, 4] function=active srcadrs=n.n.n.n
```

This command adds a new filter, number four in the list for the specified interface.

Note: The *Index* and *FilterNumber* objects are read-only in the sense that, once in existence, their value can't be changed, but you must specify them to create or delete a filter entry.

Objects:

*Index alias: Ifx	The PII of the VPNGroup to which this filter applies
*FilterNumber alias: fnum	A sequential integer you assign to the filter when it is created, designating its place in the filter list
Function	Function of this command (activate or delete) for the specified PII
SrcAdrs	Source IP address; the address associated with the incoming IP packet
SrcMask	Source subnet mask applied as a bitwise AND to the Source IP address.
SrcAdrsCompare	Criterion against which the Source IP Address is measured: *Any, EQ* (default)*, LT, GT, NE*) ***Caution: "Any" and "NE" sound alike***
DstAdrs	Destination IP address; the IP address of the interface to which the packet is directed.
DstMask	Destination subnet mask applied as a bitwise AND to the destination IP address

DstAdrsCompare	Criterion with which the Destination IP Address is measured: *Any, EQ* (default), *LT, GT, NE*) ***Caution: "Any" and "NE" sound alike***
IpProtocol alias: proto	IP Protocol Identifier (to match against the IP Protocol IDs of incoming messages): 0 = IP 1= ICMP 2 = IGMP 3 = GGP 6 = TCP 17 = UDP
SrcPort	The IP Port number of the source of the packet
SrcPortCompare alias: srcpc	Criterion with which the Source Port number is compared: *Any, EQ* (default), *LT, GT, NE*) ***Caution: "Any" and "NE" sound alike***
DstPort	The IP Port number of the packet's Destination
DstPortCompare alias dstpc	Criterion with which the Destination Port number is compared: *Any, EQ* (default), *LT, GT, NE*) ***Caution: "Any" and "NE" sound alike***
Action	Action taken on messages that match the filter: *pass, block, chain, redirect*
RedirectPii	Outgoing interface PII for *Action=Redirect*. *Redirect* is valid only for packets arriving at the CE from subscriber interfaces (not WAN side interfaces). The outgoing interface must be a WAN-side interface (not a subscriber interface). Redirect messages to subscriber interfaces are automatically discarded.
NumMatches	The number of packets that match the rules imposed by this filter. The operator can write a value (typically 0) to reset it. This counter increments independently for each rule in a chain.

(*) indicates object used as an index.

cmGroupTable
(alias: cmgrp)

This group contains the user-defined list of CE200 group names. Assigning a unique name to a group of interfaces, and hence to the users associated with those interfaces, would allow a carrier to associate individual end-users with a common group, (for example, to facilitate consolidating a number of users onto a single corporate billing account). Using cmGroup, you can create, rename, or delete groups.

The cmGroup table contains only the list of current group names. Specific interfaces are assigned to their groups using cmMember.

Syntax: *command* cmgroup *[grpname] object_1 object_2*

Example 1: To configure a new group:

```
set cmg [tomco] row=createandgo
```

Example 2: To display the status of a group:

```
get cmg [tomco]
Group: cmGroupTable
Instance: [tomco]
Name                  =  tomco
NewGrpName            =  ""
RowStatus             =  Active
```

Example 3: To rename an existing group:

```
set cmg [tomco] newgrp=jerryco
```

Objects:

*Name	The name assigned to this group. Up to 32 characters can be in a group name.
NewGrpName	New name of an existing group, as shown in example 3.
RowStatus	The operational state of a row in the cmGroup table: Active—In a Get command, the row is actively being used by the CE200. In a Set command, overwrites an existing row. CreateAndGo—Creates a new row that will be immediately used by the CE200. This value prevents an existing row from being overwritten. Destroy (alias: delete)—Deletes a row. Use this object to create and delete groups.

(*) indicates object used as an index.

cmGSHDSLModemTable
(alias: shmod)

Use this group to configure and display information about the performance/characteristics of a G.SHDSL port on the CE200 side of the specified interface. Displayed data represent an instantaneous view of the parameters; interval time is not specified. A similar table, *cmCpeGSHDSLModem*, is used to view characteristics for the CPE side of the interface.

The MaxDataRate object sets the desired data rate for ports on the G.SHDSL modules. If the loop performance degrades, the Rate-Adaptive function will successively step the data rate on the affected port down to the next slowest rate (generating a trap and minor alarm with each step), until it reaches a sustainable rate.

Settings for the *cmGSHDSLModem* group are saved as part of the system Config file, and their settings are maintained even if power is cycled or removed.

Syntax: *command* cmgshdslmodem *[index] object_1 ... object_4*

Remember, this command is for control and management of the CO side (the line card) of the interface. To display the modem data from the CPE side, use cmCpeGshdslModem (page 80).

Example 1: To set the maximum data rate for port 11, slot 6:

```
set shmod [1.6.11] maxdatarate=2304000
```

Example 2: To query port 11, slot 6 for its configuration:

```
get shmod [1.6.11]
Group: cmGSHDSLModemTable
Instance: [1.6.11.0]
Index           =  1.6.11.0
MaxDataRate     =  2304000
CurrDataRate    =  2112000
TransmissionMode =  1
RemoteEnabled   =  disabled
HostMode        =  central
```

```
DSLMode           =  GSHDSL
SuccessfulTrains  =  1
FailedTrainingAt  =  0
CpeResponses      =  2
ModemCommand      =  None
Debug             =  00.00.00.00
MinDataRate       =  192000
CurrSNRDown       =   0
CurrSNRMarginDown=   6
MinimumSNRMarginD=   2
CurrSNRUp         =   0
CurrSNRMarginUp   =   6
MinimumSNRMArginU=   4
ConfPSD           =  symmetric
CurrPSD           =  symmetric
ConfTrainMode     =  rateadaptive
CurrTrainMode     =  rateadaptive
TxPower           =   0
PBOVal            =   0
RxGain            =   0
```

Objects:

*Index	PII of the G.shdsl physical port
MaxDataRate	Enter the maximum data transmission rate (Rate Adaptive Mode) or the desired transmission rate (Fixed Rate mode) for this port, in bps, 192 kbps to 2.304 Mbps in 64 kb increments: 192000 256000 320000 384000 448000 512000 576000 640000 704000 768000 832000 896000 960000 1024000 1088000 1152000 1216000 1280000 1344000 1408000 1472000 1536000 1600000 1664000 1728000 1792000 1856000 1920000 1984000 2048000 2112000 2176000 2240000 2304000 (default)
CurrDataRate	The attempted rate (if training) or the actual rate (if trained) of the specified port. Options are the same as in *MaxDataRate* above, plus: *None* and *Unknown*
TransmissionMode	Sets data transmission characteristics (power, modulation, etc.) to those used in your geographic region: 1 (default) or 2. 1 = Annex A (North America); 2 = Annex B (Europe)
RemoteEnabled	Indicates status of EOC message initiation on the CPE connected to this port: In this release, status is always *Disabled.*
HostMode‡	Configures the modem interface as either STU-C (CO side) or STU-R (remote [CPE] side) to enable back-to-back testing of G.SHDSL Modules without introducing the variable resulting from the use of CPE. Values: *central* (default) or *remote*
DSLMode	Indicates whether the interface is configured for G.SHDSL or SHDSL operation. Only G.SHDSL is supported in this release.
SuccessfulTrains	The number of successful trains for this port. Counter is zero at powerup and resets to zero whenever data rate is changed or the module is restarted.
FailedTrainingAttempts	The number of unsuccessful attempts by this port to train. Counter is zero at powerup and resets to zero whenever data rate is changed or module restarted.

CpeResponses	The number of CPE responses seen to training sequences detected on this port. Responses may or may not result in successful trains.Counter is zero at powerup and resets to zero whenever data rate is changed or the module is restarted.
ModemCommand	Use this object to force a retrain at the specified port. Only option is *retrain*
Debug‡	For factory use only.
MinDataRate	In *RateAdaptive* mode you can configure minimum acceptable data transmission rate for this port in bps. Valid range is 192000 (Default) to 2304000. In *FixedRate* mode, this object can't be set, but will automatically assume the value of *MaxDataRate*.
CurrSNRDown	Displays the current SNR in the downstream direction for this G.SHDSL port in dB (-127 to 128).
CurrSNRMarginDown	The current SNR margin (in dB) in the downstream direction for this G.Shdsl port (-10 to 21)
MinimumSNRMarginDown	In *RateAdaptive* mode specifies the worst-case target SNR margin in the downstream direction for this G.SHDSL port. Valid range:0 to 10. Default value: 2. In *FixedRate* mode, this object remains at 0.
CurrSNRUp	Displays the current SNR in the upstream direction for this G.Shdsl port (-127 to 128 dB)
CurrSNRMarginUp	The current SNR margin (in dB) in the upstream direction for the specified G.Shdsl port (-10 to 21).
MinimumSNRMarginUp	In *RateAdaptive* mode specifies the worst-case target SNR margin in the upstream direction for this port. Valid range: -10 to 21. Default value: 4. In *FixedRate* mode, this object remains at 0.
ConfPSD	Selects the Power Spectral Density mask configuration for this G.SHDSL port. Only the default (*Symmetric*) setting is supported in this release.
CurrPSD	Displays the Power Spectral Density mask configuration for the selected port.
ConfTrainMode	Selects the Training Mode for this G.SHDSL link as either *Fixed* rate (default) or *RateAdaptive*.
CurrTrainMode	Displays the Training Mode for the selected port.
TxPower	Displays transmit power (±0.1 dBm increments)
PBOVal	The Power Backoff value for the selected port. The PBOVal, when added to the TxPower should equal the maximum power available for the port with the given configuration (for Annex A symmetric, this value would be 13.5 dBm; for Annex B with rate >2.048 Mbps, the value would be 14.5 dbm)
RxGain	Receiver gain value, in dB

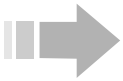
‡ Indicates object accessible through CopperCraft only (no SNMP access).

(*) indicates object used as an index.

cmHDSLModemTable

Use this group to configure and display information about the performance/characteristics of an SDSL port on the CE200 side of the specified interface. Displayed data represent an instantaneous view of the parameters; interval time is not specified. A similar table, cmCpeHDSLModem, is used to set and monitor characteristics for the CPE side of the SDSL interface.

The DataRate object sets the data rate for ports on the SDSL modules. If the loop performance degrades, the Rate-Adaptive SDSL function will successively step the data rate on the affected port down to the next slowest rate (generating a trap and minor alarm with each step), until it reaches the configured minRate object. The DataRate object in the cmCpeHDSLModem group is read-only; connected CPEs will follow the configured rate of their corresponding SDSL ports on the CE200.



N O T E

The DataRate object in the cmHDSLModem group does not apply to IMUX bundles. To set the speed of an IMUX bundle, you must pre-set the data rate on each of its DSL Member ports before assigning the ports to a bundle. The aggregate data rate of the DSL Member ports becomes the data rate of the bundle. For best performance in a bundle, set the data rate of all of its ports to the same speed.

The cmHDSLModem DataRate and MinRate configurations are saved as part of the system Config file, and their settings are maintained even if power is cycled or removed.

The CPECode object can be used to preconfigure the interface for optimum interoperability with different types of SDSL CPEs.

Syntax: *command* cmhdslmodem *[index] object_1 ... object_4*

Remember, this command is for control and management of the CO side (the DSL module) of the interface. To display the HDSL Modem data from the CPE side, use cmCpeHDSLModem (page 81).

Example 1: To set the data rate for port 11, slot 6:

```
set cmhd [1.6.11] datarate=1568
```

Example 2: To query port 11, slot 6 for its configuration:

```
get cmhd [1.6.11]
Group: cmHDSLModemTable
Instance: [1.6.11.0]
Index                =  1.6.11.0
OperState            =  Unknown
DataRate             =  1568
CurrentRate          =  784
MinRate              =  None
Command              =  None
CpeCode              =  0
Type                 =  Central
AnalogAGC            =  6LevelAGC
Enable               =  TRUE
```

```
TxGain                =  0
TestMode              =  NoTest
ReverseTipRing        =  FALSE
SignalLevel           =  86
DCOffset              =  -1
Attenuation           =  220
NoiseMargin           =  -32
TimingRecovery        =  0
BPStatus              =  1
BPSwMajorVersion      =  2
BPSwMinorVersion      =  0
BPHwVersion           =  34
BpStage               =  22
AAGCValue             =  4
SuccessfulTrains      =  0
FailedTrainingAttemp  =  11777
CpeResponses          =  0
```

Note: The 1.568 Mbps rate is valid only for ports on 24-port SDSL 30X modules.*:*

Objects:

*Index alias: idx	The Permanent Interface Identifier (PII) of the DSL physical port.
OperState	The current operational state of the interface: Unknown — Disabled Enabled — Testing For a complete definition of OperState as it relates to this group, see *Operating Status* on page 20.
DataRate	The configured data rate, in Kbps, for this port: 1568 — 1040 784 — 416 320 — 208 160 — None Unknown
CurrentRate	The current operating rate, in Kbps, for this port: 1568 — 1040 784 — 416 320 — 208 160 — None Unknown
MinRate	The minimum data rate allowed on this port under the rate-adaptation fallback function: 1568 — 1040 784 — 416 320 — 208 160 — None Unknown
Command	When used in a Get statement (such as **get cmhd [*pii*]**), shows the last command issued. When used in a Set statement (such as **set cmhd [*pii*] command=retrain**), issues a subsequent value as a command for the module to perform. Command values are None and Retrain.

CPECode	The integer code corresponding to the type of CPE connected to this port. It describes certain operating characteristics of the CPE (line polarity, di-bit order, preactivation signaling, etc.) to enable the CE200 to efficiently exchange data with it. All CopperCompatible CPEs currently use Code 0, but Copper Mountain may assign additional codes to specific CPE types if required.
Type	The Bit Pump terminal type: Central or Remote.
AnalogAGC‡	Indicates if AGC is available and which level: NoAGC 4LevelAGC 2LevelAGC 6LevelAGC
Enable	The Port Status Indicator: TRUE or FALSE.
TxGain	The value of the transmit gain.
TestMode	The operating test mode: NoTest ExternalAnalogLoopback DigitalNearLoopback DigitalFarLoopback TxIsolatedPlus3Pulse TxIsolatedPlus1Pulse TxIsolatedMinus1Pulse TxIsolatedMinus3Pulse Continuous4LevelTx Continuous2LevelTx SetNominalVCXOFrequency SetMinimumVCXOFrequency SetMaximumVCXOFrequency InternalAnalogLoopback IsolatedAnalogLoopback Continuity NearEndLoopProfile
ReverseTipRing	The Reverse Tip/Ring polarity on a received signal: TRUE or FALSE.
SignalLevel‡	The average absolute value of an ADC input signal.
DCOffset‡	The average DC offset per ADC sample.
Attenuation‡	The overall signal power attenuation.
NoiseMargin	The noise margin of the receiver: -32 to +32. Any values between +10 and +32 will provide a clean loop. A value below +10 typically causes the line to behave erratically.
Timing Recovery‡	Eight MSbs of the timing recovery control word.
BPStatus‡	The Bit-Pump status bits.
BPSwMajorVersion	The Bit-Pump major software version.
BPSwMinorVersion	The Bit-Pump minor software version.
BPHWVersion	The Bit-Pump hardware type and version.
BPStage‡	The Bit-Pump software internal stage number.
AAGCValue‡	The current value of 3 (LSB) AAGC control bits.

SuccessfulTrains	The number of successful trains for this port. Counter is zero at powerup and resets to zero whenever data rate is changed or manually set by operator command
FailedTrainingAttempts	The number of unsuccessful attempts by this port to train. Counter is zero at powerup and resets to zero whenever data rate is changed or manually set by operator command
CpeResponses	The number of CPE responses seen to training sequences detected on this port. Responses may or may not result in successful trains.Counter is zero at powerup and resets to zero whenever data rate is changed or manually set by operator command

‡ The object is accessible through CopperCraft only (there is no SNMP access).

(*) indicates object used as an index.

cmIDSLBoardTable

This group displays timing information about a specified IDSL module in addition to the information provided through cmBoard.

Syntax: *command* cmidslboard *[index] object_1*

When entering the PII for the index, you can omit the port number (or leave it at zero) to indicate the entire module.

Example: To query the IDSL module in slot 11:

```
get cmidslb [1.11]
Group: cmIDSLBoardTable
Instance: [1.11.0.0]
Index                =  1.11.0.0
TimingMode           =  internallyTimed
TimingStatus         =  internallyTimed
TimingAvailable      =  no
```

Note: Only one IDSL module in a CE200 can be designated as a NetworkTimedMaster, and only one IDSL module in a CE200 can be designated as a NetworkTimed-Backup.

Objects:

*Index	The Permanent Interface Identifier (PII) of the module. Since this indicates the board, port numbers other than 0 are ignored.

TimingMode alias: mode	The configured timing mode for the module: internallyTimed—The module generates its own clock and distributes it to all ports on this module. This is the default. locallyTimed—The module acquires the network clock from its Port No. 1 and distributes it to all other ports on this module. networkTimedMaster—The module acquires the clock and distributes it per this protocol: 1) Acquire the network clock from Port 1 and distribute it to all ports on this module and across the backplane for use by other modules. If this alternative fails, then... 2) Acquire the clock from the backplane (provided by the NetworkBackup module). If this alternative fails, then... 3) Provide internal timing to all ports on this module. NetworkTimedBackup—The module is designated as the backup “Master” in case the designated master is unable to supply the clock. Clocking for this module is performed per this protocol: 1) Acquire the clock from the CE200 backplane. If there is no backplane clock available (the Master module is not providing one), then...
TimingMode (continued)	2) Become the Master: acquire the clock from Port 1 and distribute it locally to all ports on this module and across the backplane. If this alternative fails, then... 3) Provide internal timing to all ports on this module. As soon as the configured Master is again able to supply clock, the backup module relinquishes that role and returns to its “normal” condition. networkTimedSlave—The module acquires the clock from the CE200 backplane. If the backplane clock is absent, the module automatically reverts to InternallyTimed.
TimingStatus alias: status	The timing mode that is actually operating or the specified module (rather than the user-configured setting shown above): internallyTimed locallyTimed networkTimedMaster—The module is acquiring the clock from Port 1 and distributing it to all of its ports and across the CE200 backplane. networkTimedBackup—The module is acquiring the clock from the CE200 backplane. networkTimedSlave—The module is acquiring the clock from the CE200 backplane.

TimingAvailable alias: avail	For and IDSL module configured as a NetworkTimedMaster or NetworkTimedBackup, this object indicates whether a valid clock signal is present at Port 1: Yes—The module is a NetworkTimedMaster or NetworkTimedBackup and the clock is available at Port 1. No—The conditions required for Yes are not satisfied.

(*) indicates object used as an index.

cmIDSLModemTable

This group allows you to configure and display information about the performance/characteristics of an IDSL port on the CE200 side of a specified IDSL interface. A similar table, cmCpeIDSLModem, is used to set and monitor characteristics for the CPE side of the IDSL interface.

Set the DataRate object to select the speed of the specified IDSL port. On the CPE side, the DataRate object in the cmCpeIDSLModem group is read-only; connected IDSL CPEs will follow the configured rate of their corresponding ports on the CE200.

The other user-configurable objects are CPECode, used to preconfigure the IDSL interface for optimum interoperability with different types of IDSL CPE, and ActivationInterval, which in case the IDSL port does not successfully train on the CPE, sets the amount of time between retries. Once the CPE is trained, the State object indicates ActivatedCpeDetected. The Debug object will also respond to Set commands, but should not be used unless you are directed to do so by Copper Mountain Customer Support.

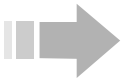
Syntax: *command* cmidslmodem *[index] object_1...object_4*

Example 1: To set the data rate for port 4, slot 11:

cmIfaceTable

Use this group to set and display the configuration table for the specified physical or protocol interface (location, device, circuit).

All Ethernet ports, DSL ports, DSL data VCs, DSL voice VCs, WAN Ports, WAN VCs, and IMUX bundles are accessible to the cmIface group. But the object group is subject to frequent revision to accommodate new features as they are added to the CE200. So, it is important that when you upgrade your software, you have the manuals or other documentation to match.



N O T E

The cmIface table is a table for configuring all interfaces on the CE200 *system, but be aware of these caveats: cmIface allows you to configure unused interfaces, to preconfigure DSL interfaces in advance of CPE installation, and to leave many important physical or logical interfaces unconfigured. To view all installed interfaces, use the ifTable group.*

Syntax: *command* cmiface *[pii]*

For WAN VCs or DSL voice VCs (DLCI 22), the full PII includes the DLCI or VC number.

Notes:

1. When configuring an interface with cmIface, the system will accept Set commands for objects that are otherwise irrelevant. But the irrelevant setting has no effect, and assuming you supply valid input for objects essential to the configuration, the system will operate normally.

   For example, if you intend to configure an interface for the VWAN netmodel and you correctly configure the other cmiface objects, but also add an (unnecessary) IP address, the configuration will be listed in the table, even though the IP address will be ignored and data traffic will pass over the interface. Conversely, if you wish to configure an interface for Policy IP routing, and you configure the correct EncapsulationType and a destPII, but leave the NetModel object set to None, some of the settings may still be retained, but the interface will not pass traffic.

2. To ensure interfaces are configured correctly, always verify the actual settings (get cmiface), even on automatically generated interfaces; the default settings may not always be the appropriate ones for your operation.

3. When creating IMUX bundles, do not attempt to configure the individual DSL Member ports. If you try to assign configured ports to an IMUX bundle, the cmBundle command will fail and the system will return an error message. If this happens, reset the netmodel to None for the DSL member port causing the error. Assigning a netmodel and the resultant forwarding mode can only be done by configuring the bundle.

4. Similarly, in referring to a T1/E1 IMA group, use the “virtual” PII reference for the IMA group. Do *not* use individual link PIIs to configure an IMA group.

5, For the HDIA (High Density IP Access) netmodel, several limitations apply to the way that you can enter values. Also, the limitations are different depending on whether it is a DSL port or a WAN port that you are configuring. When configuring a data VC on a WAN port serving a series of IAD CPEs, you must enter a value in FarEndAddr for the default router. When configuring a voice VC on a WAN port, you must enter a value in FarEndAddr for the voice gateway. But for both the voice VC and the data VC on a WAN port, you must leave the DestPII set to zero.

6. You must explicitly enter an EncapsulationType when configuring a new cmIface entry; the object will no longer self-configure based on the interface’s other settings.

7. Character strings for Name and AdditionalInfo objects may include a-z, A-Z, 0-9, plus the following characters: ! @ % & * - _ ( ) + = : . , / ' < > ? and <space>. Other characters are reserved for future use.

Example 1: To display the cmIface table for virtual circuit (Frame Relay DLCI/ ATM Virtual Link number) 17, port 1, slot 3:

```
get cmif [1.3.1.17]
Group: cmIfaceTable
Instance: [1.3.1.17]
PII                 =  1.3.1.17
IfIndex             =  1.3.1.17
Name                =  ""
GroupName           =  ""
AdditionalInfo      =  ""
NetModel            =  VWAN
IpAddr              =  0.0.0.0
NetMask             =  0.0.0.0
MacAddr             =  ff.ff.ff.ff.ff.ff
BurnedInMacAddr     =  ff.ff.ff.ff.ff.ff
FarEndAddr          =  0.0.0.0
DestPII             =  0.0.0.0
CMCPCompatible      =  No
EncapsulationType   =  rfc1490
FwdMode             =  VWAN-bridge
Pix                 =  283
ServiceClass        =  None
```

Example 2: To set the far-end address for another VC as part of a policy-routed frame relay link:

```
set cmif [1.3.1.44] netmodel=ip farend=192.168.99.2
```

The far-end address always points to the *upstream* device.

Example 3: To set the range of IP addresses for voice VC 21, port 1, slot 3 on the IAD:

```
set cmif [1.3.1.21] netmodel=hdia ipaddr=10.0.21.0
netmask=255.255.255.192 farend=10.0.21.1 encaps=rfc1483
```

The IP address is for the IP subnet; it is not an address for the WAN VC. The voice gateway must be on the same IP subnet.

Example 4: To set the destination IP address of voice VC 22 at DSL port 13, slot 6 on the DSLAM to point to voice VC 21 from example 3:

```
set cmif [1.6.13.22] netmodel=hdia ipaddr=10.0.21.2
netmask=255.255.255.255 destpii=1.3.1.21 encaps=ip-1490
```

The voice VC on the IAD must be on the same subnet as the voice gateway, and the DestPII must point to the voice VC on the WAN.

Objects:

*PII	The Permanent Interface Identifier of the protocol index.
IfIndex	The Permanent Interface Identifier (PII) as listed in the IfTable.
Name	A name for this interface (up to 31 characters of text, which must begin with an alpha character). The name you assign for the interface will be added to the cmNameTable.
GroupName	A text string identifying the user group to which this interface will be assigned. Use the command **getall cmGroup** to display a listing of currently valid GroupNames.
AdditionalInfo	Up to 128 characters of miscellaneous user-defined text.

NetModel alias: Model	The Internetworking model used on this interface: None — IP VWAN — Cross-Connect CopperVPN — HDIA (input accepted but model not supported) Setting an interface's netmodel to None causes all of its other forwarding objects to revert to 0, and the interface is effectively "downed."
IpAddr	For the IP netmodel, this is the IP address for the interface, if needed. For a netmask of all 1s, the IP address is the only legal premise IP address. For any other netmask, the IP address is the first of a contiguous range of IP addresses. The range size is given by the netmask. **Note:** This object defines a range of IP addresses, not a subnet.
NetMask alias: nm	For the IP or VWAN netmodels, this is the interface subnet mask address, if needed.
MacAddr alias: ma	The working MAC address used by this DSL interface. The value displayed may be the same as the BurnedInMacAddr object. The format is *byte1.byte2.byte3...*, where bytes are displayed as hex numbers. For example, 0.60.58.1.b4.5c.
BurnedInMacAddr alias: bim	The permanent MAC address of the connected CPE (if it has one, and if the CPE is CMCPCompatible). This object serves as a reference when a different value appears in the macAddr object, as discussed on the previous page. The format is *byte1.byte2.byte3...*, where bytes are displayed as hex numbers. For example, 0.60.58.1.b4.5c.
FarEndAddr alias: fa	For the IP, or CopperVPN netmodels, this is the IP address that identifies the upstream end point of this interface. For HDIA WAN VCs, this is the IPaddress that identifies the upstream IP router or voice gateway. Note that if the DestPII is the Ethernet port on the System Control module, the FarEndAddr must be a separate IP address, but it must be on the same subnet as the Ethernet Port.
DestPII alias: dp	The full PII of the destination (upstream) WAN interface to which the DSL interface (CopperVPN, IP policy routed, HDIA, or VWAN interfaces) will forward packets.

CMCPCompatible	For DSL links only, this object identifies whether the connected CPE recognizes Copper Mountain's proprietary Internal Control Protocol for management of CPEs: Yes and No. For other types of interfaces, this object has no meaning and will always be No. If a Copper Mountain CPE is connected to this DSL interface, CMCP *must* be set to Yes. For other CPE makes, CMCP can be No or Yes. If in doubt, check the CPE documentation
EncapsulationType	The type of encapsulation used on this interface: None (default) rfc1483 rfc1490 HDLC PPP-HDLC rfc1973 Q922 Q922-1490 FRF5 IP-1490 ATM rfc2364-llc rfc2364-null IP-1483 MAC-1483 MAC-1490 For the VWAN and IP netmodels, all Copper Mountain CPEs except the CR202 (IMUX SDSL) use rfc1483, while third-party CPEs and the CopperRocket 202 use rfc1490. The Cross-Connect netmodel can use any of the other listed types, but if the interface is an ATM VC, you will probably use rfc1483. See Appendix A in the *CopperEdge 200 Installation and Operating Guide* for a list of supported encapsulation types. For CopperVPN+, IP-1483 means *routed* RFC1483 and MAC-1483 means *bridged* RFC1483 encapsulation. Similarly, IP-1490 is the routed-mode RFC1490 encapsulation. DSL voice VCs (on DSL VCID=22) must specify encapsulation of IP-1490. No other cmIface rows can use IP-1490. Only WAN VCs can use FRF5.
FwdMode	The packet forwarding mode used to route packets received from this interface: None Full-IP IP-Policy VWAN-point-to-point VWAN-bridge IP-Policy-to-Ethernet CopperVPN CopperVPNAuto HDLC-VC-payload VC-VC-payload PPP-HDLC-1973 FRF8-1490-1483 PER-VC FRF5 ppp-translation HDIA ppp-transparent
Pix	The ifIndex number used by the ifTable.

ServiceClass alias: sc	On DSL interfaces, the current service class of this DSL port: None A B C D (the default) See also *cmServiceClass* on page 164.

(*) indicates object used as an index.

cmInterfaceExtTable

This group displays the reason why a Frame Relay DSL VC or ATM WAN VC is operationally down. It complements the MIB-II ifTable and includes information specific to the Copper Mountain interface.

Syntax: *command* cminterfacee *[index]*

Example 1: To see all WAN VC or DSL VC failures:

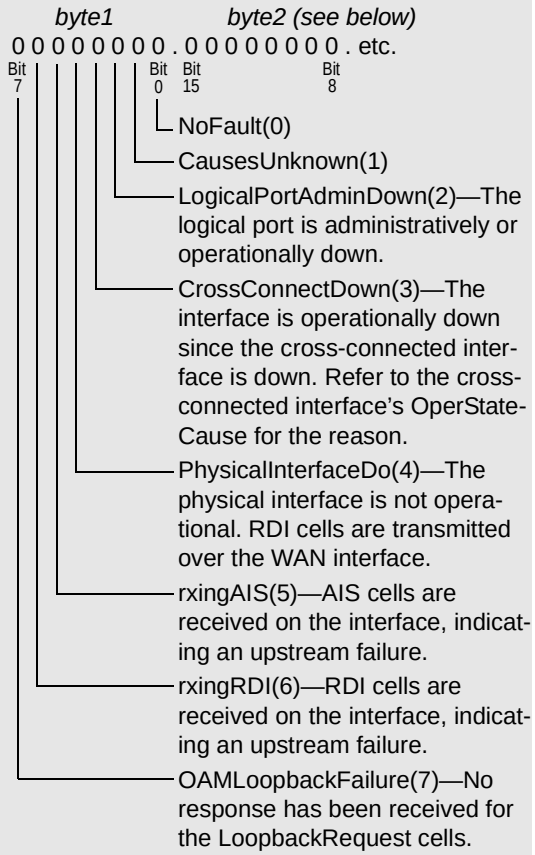
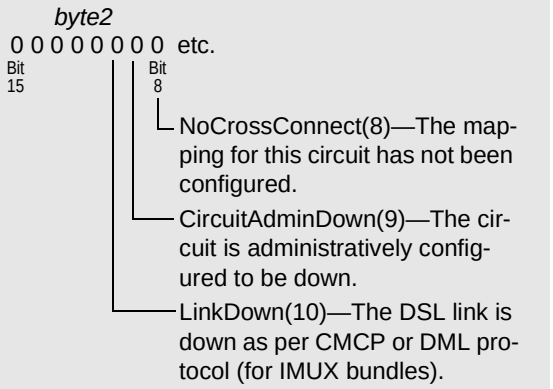
```
geta cminterfacee
Index                OperStateCause

Instance: [1.3.1.9]
1.3.1.9              9

Instance: [1.3.1.16]
1.3.1.16             3+7
```

Example 2: To query VC 16, port 1, slot 3:

```
get cminterfacee [1.3.1.16]
Group: cmInterfaceExtTable
Instance: [1.3.1.16]
Index                =  1.3.1.16
OperStateCause       =  CrossConnectDown(3)+OAMLoopbackFailure(7)
```

Objects:

*Index	Permanent Interface Identifier RO, PII, shelf.slot.port.virtualId

OperStateCause	The reason why the interface is operationally down. The bitmap of possible values is: 
OperStateCause (continued)	 For Get and Getnext requests, the format is *bitStr0(bitId0)+bitStr1(bitId1)*..., where *bitStr* is the ASCII text and *bitId* is the corresponding bit number. For Getall requests, the format is *bitId1+bitId2*... . An empty string (“ “) means that no bits are set

(*) indicates object used as an index.

cmInterfaceOptionsTable

Use this group to configure and display parameters for a DSL port or a WAN VC. For WAN VCs, three objects apply: VoiceGatewayType, VoiceGatewayAddr, and VoiceGatewayTypeId. For DSL ports, three objects apply: MaxVoiceChannels, JitterSpeedTradeoff, and JitterControlStatus.

Syntax: *command* cminterfaceoptions *[pii]*

Example 1: To configure the voice gateway for VC 21 on port 1, slot 3:

```
set cmint [1.3.1.21] voicegatewaytypeid=4
voicegatewayipaddr=10.0.21.1
```

Or,

```
set cmint [1.3.1.21] voicegatewaytype=tollbridge
voicegatewayipaddr=10.0.21.1 voicegatewaytypeid=4
```

The voice gateway must be on the same subnet as all of the voice VCs on the IADs. Also, if you use the VoiceGatewayType object, you must also set the VoiceGatewayTypeId to the same type.

Example 2: To query port 1, slot 3 about VC 21:

```
get cmint [1.3.1.21]
Group: cmInterfaceOptionsTable
Instance: [1.3.1.21]
PII                 =  1.3.1.21
VoiceGatewayType    =  TollBridge
VoiceGatewayIpAddr  =  10.0.21.1
MaxVoiceChannels    =  255
JitterSpeedTradeoff =  FullSpeedDataOnly
JitterControlStatus =  Uncontrolled
RowStatus           =  Active
VoiceGatewayTypeID  =  4
```

Example 3: To set a maximum number of voice channels:

```
set cmint [1.6.13] maxvoicechannels=8
```

Although the default setting for MaxVoiceChannels is 255, a system administrator would likely limit customers to the number of lines they pay for.

Notes:

1. You can use either the VoiceGatewayType object or the VoiceGatewayTypeId object, or both objects. If you use both objects, they must be set to the same type or the CPE will reject it.
2. Controlling jitter, or the amount of delay occurring between packets, should be done only on DSL ports carrying real time (RT) or voice packets.

Objects:

*PII	The Permanent Interface Identifier of the DSL port connected to the CPE or WAN VC. PIIs for WAN VCs are used to configure VoiceGatewayType and VoiceGatewayIpAddr. PIIs for DSL Ports are used to configure MaxVoiceChannels.

VoiceGatewayType alias: gwtype	Supported by WAN VCs only. The type of voice gateway upstream of the CE200: None (default), CopperCom, JetStream, TollBridge, PathStar, Other All DSL VCs that use this WAN VC as their destPii will inherit this VoiceGatewayType for their CPE PlugAndPlay parameters.
VoiceGatewayIpAddr alias: gwipaddr	Supported by WAN VCs only. The IP address of the voice gateway that is upstream from the WAN interface. The default is 0.0.0.0.
MaxVoiceChannels	Supported by DSL ports only. The maximum number of voice lines that the CPE will allow: 0 to 255. The default is 255.
JitterSpeedTradeoff alias: jspeed	Supported by DSL ports only. The settings for jitter on the specified DSL port: FullSpeedDataOnly—The default MaxJitterAndSpeed—Not supported OptJitterAndSpeed—Not supported MinJitterAndSpeed FullSpeedDataOnly means no jitter control. It is used only on ports with NRT traffic (data traffic). MinJitterAndSpeed is the only option for controlling jitter in this release.
JitterControlStatus alias: jstatus	Supported by DSL ports only. The status of jitter control: Uncontrolled or Controlled. This object changes to Controlled after a CPE accepts PlugAndPlay parameters and the CE200 has configured itself.
RowStatus	The operational status of a row in the cmDHCP table: Active or Destroy. Active—In a Get command, the row is actively being used by the CE200. In a Set command, creates a new row. Destroy (alias: delete)—Deletes a row. **Note:** Configuring any object automatically creates the corresponding row.
VoiceGatewayTypeId	Supported by WAN VCs only. The type of voice gateway upstream of the CE200: 1 to 255, where 1 = None 2 = CopperCom 3 = JetStream 4 = TollBridge 5 = PathStar 6 to 255 = Voice gateways unknown to the CE. Starting with 6, the text descriptions are stored in the CPE, not in the CE200. All DSL ports and VCs that use this WAN VC as their destPII in the cmIfaceTable will inherit this type for the CPE plug-and-play parameters.

(*) indicates object used as an index.

cmLoop

The *cmLoopTable* allows you to set up and run DSL loopback tests in real time on lines configured to run SDSL, IDSL, ADSL, and T1 signals. It also allows you to set up and run WAN loopback tests in real time over DS1 and DS3 modules, including both ATM and Frame Relay. In this release, the table has been enhanced with the addition of four new T1 test patterns to the *Type* object: *ds1All-0s*, *ds1All-1s*, *ds1QRS*, and *ds13in24*. You may use these values when setting up and running custom FDL loops for ports on T1 line modules, or when running standard loops for the same ports. Note that these apply to loopbacks on T1 Line Module ports, and ***not*** to the Quad TI WAN module.

The cmLoop is one of two groups (cmLoop and cmLoopHist) that are available as diagnostic tools for verifying the integrity and performance of SDSL (including BER) and IDSL (including IDSL EOC and Block Error Rate) circuit loops.

The cmLoop and cmLoopHist groups also support the SCM-WAN-Generator test, which sends packets over the loopbacks set in the dsx1Config and dsx3Config groups (page 204 and page 215, respectively). Procedures for setting up loopback tests are described in the *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmloop *object_1...object_6*

Example: To run a loopback test for port 2, slot 10:

```
set cmloop inter=1.10.2 type=eoc act=start dur=120
```

Notes:

1. After sending test packets or bits, you can use the TxCount and RxError objects to view the results of the test.
2. In the responses for Type and TermCode, Lc indicates Line Card (a DSL module), Cpe indicates Customer Premise Equipment, Mio stands for Multiplexed I/O (a WAN port), and Loc and Rem indicate Local and Remote.
3. IDSL EOC loopbacks will not run on interfaces in which the value of IDSLModemTerminationMode=NT (port 1 on network-timed IDSL modules).
4. Loopback tests will not run on IMUX bundles. DSL ports must be unbundled (use cmBundle and set the MemberPII to 0.0.0) before loopback testing. When testing is complete, reinstate the port as a member of the bundle.

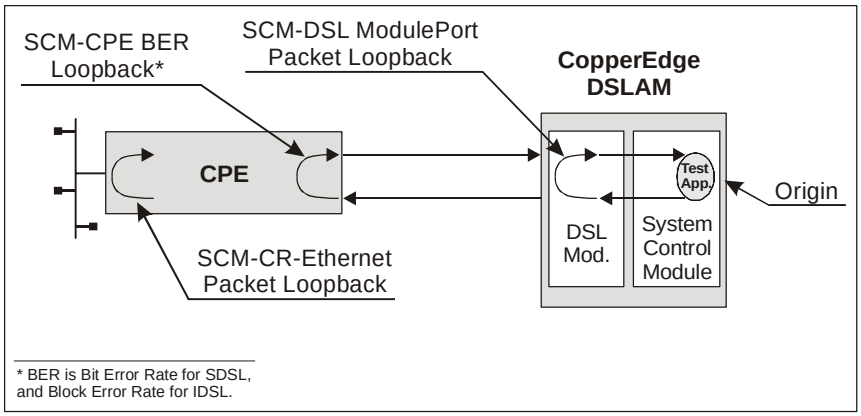
Objects:

ID	The unique identifier for the test in progress. It is created automatically when you start a loopback test.
Interface	The Permanent Interface Identifier (PII) of the interface.

Type	Type of test: None ds1All-0s—Transmits all zeros through the specified port on a DS1 Line Module. ds1All-1s—Transmits all ones through the specified port on a DS1 Line Module. ds1QRS—Transmits all Quasi-Random Signal (QRS) test pattern through the specified port on a DS1 Line Module. ds13in24—Transmits 3 in 24 through the specified port on a DS1 Line Module. scm-DSLModulePort—Applies to SDSL, IDSL, and T1 modules. scm-CR-Eth—Applies to SDSL and IDSL modules. It does not apply to T1. scm-CPE-BER—Applies to SDSL, IDSL, and T1. scm-WAN-Generator—Applies to all DS1 and DS3 modules. scm-DSLModuleEOC (alias: eoc)—Applies only to IDSL modules.
Action	The action to perform on the test: Init Start Stop
Duration	Length of time, in seconds, the test will run.
Status	Current status of selected test: Idle InProgress Done
TimeElapsed	The elapsed time in seconds since start of the test.
TermCode	The reason the test was terminated: InProgress SendError DurationExpired OperStop NoLcResponse NoCpeResponse LcInitError LcReportError CpeInitError CpeStartError LcStartError CpeDisabled LcDisabled NoLinkStateChg Idle MioInitError MioDisabled NoMioResponse BERCountOverflow LoopNotPresent
TxCount	In packet-based tests, the number of Tx packets. In SDSL BER tests, the number of Tx SDSL bits. In IDSL BER tests, the number of IDSL blocks.
RxErrors	In packet-based tests, the number of Rx packet errors. In BER tests, the number of SDSL Rx bit errors or IDSL Rx block errors.
OwnerString	The operator running the test (not enforced).

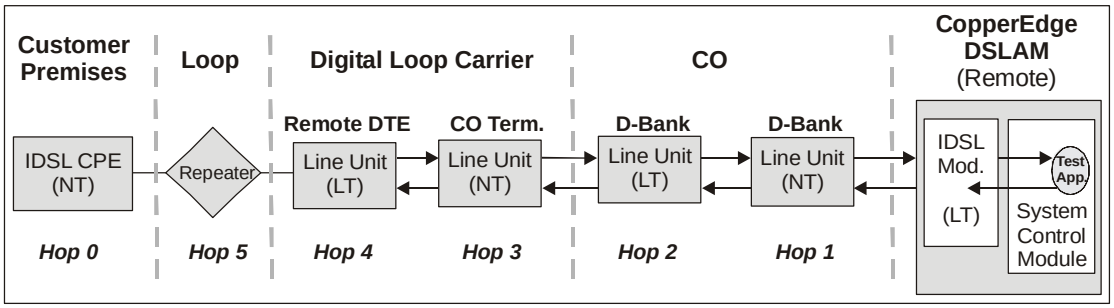
TestParameter1 alias: p1	The test-specific parameter or subtype. For EOC loopbacks, the address (hop number) corresponding to the network element to be placed in loopback mode: 0 to 6. Address 0 always refers to the IDSL CPE. Addresses 1 to 6 refer to the network elements from 1 to 6 hops downstream of the IDSL port specified by the object, Interface.
ErrorInfo	Additional error information: None IllegalSubtype UnsupportedTestType ModemIsNT ModemUntrained EOCAddrTooBig

The following example shows the DSL subscriber-side loopbacks supported with the cmLoop group.



IDSL Subscriber-Side Loopbacks

The EOC Loopback test can aid in isolating IDSL line problems by testing the various segments between the CE200 and the CPE that comprise an IDSL loop.



IDSL Loop with Multiple Network Elements

The cmLoop and cmLoopHist groups also support a slightly different type of test, the SCM-WAN-Generator. This option, available with all WAN module types, is used to test a data path when a loopback path can be established by other means (loopback controlled by another device). In this mode, the port will transmit packets and count the number of packet errors in returned packets.

DS3 loopbacks are illustrated on page 215. DS1 loopbacks are illustrated on page 204.

cmLoopHistTable

This group allows you to retrieve information about the ten most recent SDSL or IDSL WAN loopback tests run on the system. Loopback tests for the DS1/DS3 modules and DS1/DS3 WAN links are controlled separately through the dsx1Config and dsx3Config groups (page 204 and page 215, respectively).

The cmLoopHist is one of two groups (cmLoop and cmLoopHist) that are available as diagnostic tools for verifying the integrity and performance of SDSL (including BER) and IDSL (including IDSL EOC and Block Error Rate) circuit loops.

The cmLoop and cmLoopHist groups also support the SCM-WAN-Generator test, which sends packets over the loopbacks set in the dsx1Config and dsx3Config groups (page 204 and page 215, respectively). Procedures for setting up loopback tests are described the *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmloophist *[id]*

Example: To display loopback information for test 2:

```
get cmlooph [2]
Group: cmLoopHistTable
Instance: [2]
ID                  =  2
Interface           =  1.10.1.0
Type                =  scm-DSLModuleEOC
Action              =  Start
Duration            =  120
Status              =  Done
TimeElapsed         =  0
TermCode            =  LcInitError
TxCount             =  0
RxErrors            =  0
OwnerString         =  ""
TestParameter1      =  0
ErrorInfo           =  None
```

Objects:

*ID	An integer representing the unique identifier for the test in progress or the history entry.
Interface	The Permanent Interface Identifier (PII) of the interface.
Type	The type of test: None scm-DSLModulePort scm-CR-Eth scm-CPE-BER scm-WAN-LocalDigital (alias: LoopA) scm-WAN-LocalLine (alias: LoopB) scm-WAN-RemoteLine (alias: LoopC) scm-WAN-Generator scm-DSLModuleEOC (alias: EOC) ds1All-0s ds1All-1s ds1QRS ds13ini24
Duration	The amount of time, in seconds, the test will run.
Status	The status of the test: Idle InProgress Done

TimeElapsed	The elapsed time, in seconds, since the start of the test.
TermCode	The reason the test was terminated: InProgress DurationExpired NoLcResponse LcInitError CpeInitError LcStartError LcDisabled Idle MioDisabled BERCountOverflow SendError OperStop NoCpeResponse LcReportError CpeStartError CpeDisabled NoLinkStateChg MioInitError NoMioResponse LoopNotPresent
TxCount	In packet-based tests, the number of Tx packets. In SDSL BER tests, the number of Tx SDSL bits. In IDSL BER tests, the number of IDSL blocks.
RxErrors	In packet-based tests, the number of Rx packet errors. In BER tests, the number of SDSL Rx bit errors or IDSL Rx block errors.
OwnerString	The operator running the test (not enforced).
TestParameter1 alias: p1	The test-specific parameter or subtype. For EOC loopbacks, the address (hop number) corresponding to the network element to be placed in loopback mode: 0 to 6. Address 0 always refers to the IDSL CPE. Addresses 1 to 6 refer to the network elements from 1 to 6 hops downstream of the IDSL port specified by the object, Interface.
ErrorInfo	For EOC loopbacks, the errors or inconsistencies in specifying the command or its intended target: None IllegalSubtype UnsupportedTestType ModemIsNT ModemUntrained EOCAddrTooBig

(*) indicates object used as an index.

cmMacTable

This group displays information about the Ethernet host device (CPE) associated with a WAN interface that is also a member of a virtual bridge group (FwdMode=VWAN Bridge).

Each row in the MAC table displays the MAC address of the host device's Ethernet interface, the amount of time the entry will stay in the current MAC table, and the PII of the CopperEdge DSL port connected to the host device.

Syntax: *command* cmmac *[pii, index]*

Example: To query VC 28, port 1, slot 3:

```
get cmmac [1.3.1.28, 1]
Group: cmMacTable
Instance: [1.3.1.28]
PII                 =  1.3.1.28
Index               =  1
MAC                 =  0.a0.cc.53.1c.cb
TTL                 =  30
DslPII              =  1.6.2.0
```

Objects:

*PII	The Permanent Interface Identifier of the WAN-VC uplink for this VWAN bridge group.
*Index	An arbitrary number assigned to this member of this bridge group.
MAC	MAC address of host device's ethernet interface. The format is *byte1.byte2...*, where bytes are displayed as hex numbers.
TTL	Time To Live—The time, in minutes, that the entry will stay in the MAC table unless it is refreshed by a message to or from the host device to which this MAC address is assigned.
DslPII	PII of this host's DSL interface.

(*) indicates object used as an index.

cmMaint

The single object in this group allows you to specify the character that will be used to delimit records in the Bulk Statistics file, which is configured and managed through the cmMaintCmd group. See *cmMaintCmdTable* on the next page.

Syntax: *command* cmmaint *object_1*

Example 1: To set record delimiter as X:

```
set cmmaint statsdelim=X
```

Example 2: To see which character is being used as a record delimiter:

```
get cmmaint
Group: cmMaint
StatsDelimiter       =  X
```

Objects:

StatsDelimiter	The character to be used when separating records in the bulk statistics file.

cmMaintCmdTable

This group allows you to manage the following system administration and maintenance tasks:

- Upload collected bulk statistics from the CE200 to a remote server, either automatically at scheduled intervals or on demand.
- Perform scheduled or on-demand backups of the Config file (config.tgz or config.txt)to a remote server.
- Recover from memory or storage malfunctions by downloading a recent config file from the remote server.
- Perform a bulk download of software to eligible CPEs with a single command. Eligible CPEs are those that have been configured using *cmCpeEndPointConfig*.

The Statistics feature provides a way to upload DSL performance statistics from the cmDSLInterval group automatically (StatsAuto) at scheduled intervals. Statistics reported are those from the specified time period. Statistics can be uploaded immediately (that is, without waiting for a programmed interval) using the manual programming (StatsManual) function. In either case, you can only upload data that was collected within the last six hours.

Use the ConfigBackup command to upload the Config file from the CE200's flash memory. Backups can be set to occur automatically at preset times, but the backup transaction is only executed if needed; that is if the identical configuration has not previously been uploaded, or if it has changed since the last time a backup file was uploaded. The stored configuration can then be downloaded using the ConfigRestore object.

The file servers configured in cmMaintCmd tell the CE200 where to send backups of the configuration file for safe storage, and where to look for the file if needed to restore a lost or damaged system configuration. This file server is not the one configured with cmFile for download of CPE or other operating code that may be stored remotely.

The BulkCPEDownload command runs the bulk download of software (either an upgrade or a download to regulate connected CPE) to all eligible CPEs as configured in cmEndPointConfig (page 109).

C A U T I O N

*Do **not** try to edit the Config file with any off-line (non-CE200) application. While it may appear to be in text format, the Config file cannot be reliably revised or updated with a text editor. If the file should become corrupted, routing information may be adversely affected, with unintended effects on service.*

Syntax: *command* cmmaintcmd *[command] object_1...object_10*

Example 1: Complete configuration of cmMaintCmd requires that most of the writable objects be assigned a value. To reduce the required complexity of any single command, you can issue multiple *Set* commands, provided you begin by specifying the command and creating the row. The following lines are an example of the complete set *cmMaintCmd* input for a *StatsAuto* command.

```
set cmmaintc [statsauto] rows=createandwait
Set Successful
Missing IP address; Missing StartTime; Missing Recurrence

set cmmaintc [statsauto] recur=1 start=99/04/01-00:00
set cmmaintc [statsauto] primaryip=206.71.190.4
set cmmaintc [statsauto] secondaryip=206.71.190.5
set cmmaintc [statsauto] directory=/ce200data/sys1
set cmmaintc [statsauto] basefilename=stats_lima
set cmmaintc [statsauto] username=anonymous pass=" "
set cmmaintc [statsauto] rowstatus=active


get cmmaintc [statsauto]
Group: cmMaintCmdTable
Instance: [statsAuto]
Command              =  statsAuto
RowStatus            =  Active
Recurrence           =  1
ReportingPeriod      =  1
StartTime            =  2000/06/15-00:00:00
PrimaryIpAddr        =  206.71.190.4
SecondaryIpAddr      =  206.71.190.5
Directory            =  /cd200data/sys1
BaseFileName         =  stats_lima
UserName             =  anonymous
Password             =  ****
PreviousTime         =  ""
NextTime             =  2000/06/15-01:00:00
CommandStatus        =  None
StatusText           =  ""
```

Example 2: To view the last configured command of the specified type:

```
get cmmaintc [configbackup]
Group: cmMaintCmdTable
Instance: [configBackup]
Command               =  configBackup
RowStatus             =  NotInService
Recurrence            =  0
ReportingPeriod       =  0
StartTime             =  2000/07/19-00:00:00
PrimaryIpAddr         =  10.64.20.180
SecondaryIpAddr       =  0.0.0.0
Directory             =  d:/backup
BaseFileName          =  Config%N_%T
UserName              =  amsuser
Password              =  ****
PreviousTime          =  ""
NextTime              =  ""
CommandStatus         =  None
StatusText            =  ""
```

Notes:

1. When reinstating a backed-up configuration using the ConfigRestore option, note that changes to the actual operating configuration made subsequent to the last backup will not be reflected in the restored configuration.
2. The cmMaintCmd group does not include support for Daylight Saving Time. If possible, you should adhere to standard time for this locality year round.
3. To stop an in-progress BulkCpeDownload, issue the command again, but with RowStatus=NotInService.

Objects:

*Command	Type of command being configured: statsAuto statsManual configBackup configRestore bulkCPEDownload.
RowStatus	The operational status of a row in the cmMaintCmd table: Active—In a Get command, the row is actively being used by the CE200. In a Set command, overwrites an existing row or activates a row that displays NotInService. Create—Creates a new row that will be immediately used by the CE200. CreateAndWait—Creates a new row but does not activate it; the row will not be used until RowStatus is set to Active. Use this value to create rows that need further configuration before they go active. Destroy (alias: delete)—Deletes a row. NotInService—In a Get command, the row exists but is not being used by the CE200. In a Set command, creates a new row or deactivates a row that is currently Active. NotReady—The row exists, but not all required parameters have been set.

Recurrence	The number of hours between occurrences of this command. For statsAuto, this number is limited to 6 and, if necessary, the ReportingPeriod will be modified to match the value of Recurrence. For configRestore and statsManual, the value of this object is ignored. For bulkCPEDownload, Recurrence must be 0.
ReportingPeriod	The number of hours of data to include in the uploaded file: from 1 to 6. This object is configurable only for the statsManual command
StartTime	The applicable CalendarTime for this command to begin: *yyyy/mm/dd-hh:mm* (seconds are not required and will be ignored). For configBackup, configRestore, and bulkCPE-Download, the time the command will first be executed. If StartTime is in the past and Recurrence is 0, the command will be executed immediately.
PrimaryIpAddr	The IP Address of the primary file-server site for file transfer. If no IP Address is entered, no transfer will be attempted to the primary, and no event will be generated. This object is ignored for bulkCPEDownload.
SecondaryIpAddr	The IP Address of the secondary file-server site for file transfer. If no IP Address is entered, no transfer will be attempted to the secondary. If there is no secondary IP address and transfer to primary fails, the CommandStatus will indicate “failed”. This object is ignored for bulkCPEDownload.
Directory	A character string (from 1 to 63 characters) specifying the directory on the file server where uploaded files should be stored. This object ignored for bulkCPEDownload.

BaseFileName	The file name that the CE200 will use when generating an upload file (StatsAuto, StatsMan, ConfigBackup). The CE200 automatically appends the interval start time to the filename before transferring the file. For ConfigRestore, displays the full name of the saved configuration file to be downloaded from the server. This object is ignored for bulkCPEDownload. If the basefilename contains format specifiers, the following substitutions occur: %%% is placed in the output %IInserts the management IP address (such as 192_115_17_2) in the output %NInserts the string returned by db_system_name in the output %TInserts the 13-character Calendar time (YYYYMMDD-HHMM) in the output No other combinations are valid. The %T specifier can be used only once in the string. If it is not used in the base_name string, the formatted time value is appended to the output string. That is, %N_ is exactly equivalent to %N_%T.
BaseFileName (continued)	Since there is no validation of the resulting file name, you can create a file name that certain FTP file servers will not accept, and the upload will fail when you try to open the FTP connection.
UserName	The user name for the CE200 to use when establishing an FTP session with an external file server. The default is Anonymous. This object ignored for bulkCPEDownload.
Password	The password that the CE200 will use when establishing an FTP session with an external file server. This object ignored for bulkCPEDownload.
PreviousTime	The calendar time of the last attempt to run the specified command. If the command is ConfigBackup, the time is only updated when the upload is actually attempted (that is, if the saved configuration has actually changed since the last attempt).
NextTime	The calendar time when the next automatic operation will be attempted. If the command is not automatic (statsManual), this field will be empty.

CommandStatus	The status of the last command: None—No command since startup. SavedToPrimary—The upload succeeded to the primary file server. SavedToSecondary—The upload succeeded to the secondary file server. Failed—The last file transfer failed between the system and the file servers. For BulkCPEDownload, the software download failed for one or more candidate CPEs. InProgress—The command is currently running. Succeeded—The last command succeeded. Aborted—For BulkCPEDownload, the command was aborted before all candidate CPEs were downloaded.
StatusText	The text string describing the last CommandStatus value.

(*) indicates object used as an index.

cmMemberTable

Use this group to assign specific interfaces to a particular user-defined group. The table has a dual index of the Group Name and the PII of the interface. The Group Name you use must exist in the cmGroup table before it will be recognized as a valid object by cmMember.

Syntax: *command* cmmember *[grpname, pii] object_1*

Example 1: To create a new row in the group TomCo:

```
set cmme [tomco, 1.6.3] row=create
```

Example 2: To display the new row in the TomCo group:

```
get cmme [tomco, 1.6.3]
Group: cmMemberTable
Instance: [tomco, 1.6.3.0]
GrpName              =  tomco
PII                  =  1.6.3.0
RowStatus            =  Active
```

Note: DSL ports are configurable as group members; DSL VCs are not.

Objects:

*GrpName	The group name from the cmGroup table.
*PII	The Permanent Interface Identifier to be associated with the specified group name.

RowStatus	The operational status of a row (for the interface) in the cmMember table: Active—In a Get command, the row is actively being used by the CE200. In a Set command, overwrites an existing row. CreateAndGo—Creates a new row that will be immediately used by the CE200. This value prevents an existing row from being overwritten. Destroy (alias: delete)—Deletes a row. Use this object to establish and remove membership of CopperEdge PIIs in established group names.

(*) indicates object used as an index.

cmNameTable

This dual-index group contains the list of any user-defined CE200 names assigned to physical ports or protocol interfaces.

While the cmName table contains the list of current interface names and their PIIs, the names are created and assigned through the cmSubIface command (to assign names to DLCIs) or the cmIface command (to assign names to physical ports or protocol interfaces). Specific interfaces are assigned to their groups using cmMember.

As noted elsewhere, if you will be referring to a T1/E1 IMA group, always use the "virtual" PII reference for the IMA group. Do *not* use individual link PIIs to configure an IMA group.

Syntax: *command* cmname *[ifname, pii]*

Example 1: To generate a list of all of the named interfaces and their PIIs:

```
geta cmn
IfName              PII

Instance: [Ascend Pipeline, 1.7.3.0]
Ascend Pipeline     1.7.3.0

Instance: [Cisco 776, 1.6.3.0]
Cisco 776           1.6.3.0

Instance: [Cisco 800, 1.7.1.0]
Cisco 800           1.7.1.0

Instance: [Copper Rocket 201, 1.6.4.0]
Copper Rocket 201   1.6.4.0

Instance: [Copper Rocket 201, 1.9.12.0]
Copper Rocket 201   1.9.12.0

... etc....
```

Example 2: To query VC 22 on port 3, slot 11 for its name and PII:

```
get cmn [jetstream, 1.11.3.0]
Group: cmNameTable
Instance: [jetstream, 1.11.3.0]
IfName               =  Jetstream
PII                  =  1.11.3.0
```

Example 3: If you do not know the name or full PII for a particular cmName entry, you can view them one at a time using getnext:

```
getn cmn
```

Objects:

*IfName	The name assigned to the interface through cmSubIface or cmIface.
*PII	The Permanent Interface Identifier of the named interface.

(*) indicates object used as an index.

cmOperatorTable

Use this group to configure or display authorized operators of your CE200 unit. The CE200 is shipped with cmOperator as the default management tool for configuration of operator user-names, passwords, and privileges. But for centralized management and control of a user base in which operators may require access to many geographically dispersed systems, Radius Authentication is a more secure and efficient choice. (See *cmRadius* on page 159.)

Once the Radius server is provisioned and cmRadius Auth is enabled, the CopperEdge user base and configuration of its security features are under the sole control of the Radius server. While you can still configure the cmOperator group, and its contents continue to exist, they are not normally serviceable as long as the cmRadius Authentication remains enabled, except as described below.

By enabling the *cmRadiusLocalFallback*, you can configure the CopperEdge to reclaim the authentication function in case none of the Radius authenticating servers can be reached.

Syntax: *command* cmoperator *[name] object_1...object_4*

Example 1: To create a new operator:

```
set cmoper [SarahGolden] password=4821 state=create
```

Example 2: To display information for the new operator:

```
get cmoper [sarahgolden]
Group: cmOperatorTable
Instance: [sarahgolden]
Name                =  SarahGolden
Password            =  **********
State               =  Active
Context             =  All
Privilege           =  View
```

Example 3: To set operator context and privilege:

```
set cmoper [SarahGolden] context=all privilege=monitor
```

Notes:

1. Operators can either be configured directly from the serial interface or through Telnet. The cmOper table is accessible only to operators with the Security privilege level.
2. The factory default user (ce200) cannot be deleted, but you can change its preset password. Also, to use CopperView EM or another SNMP-based manager, you must first configure two operators (any community strings, but typically, public and private). The first must have the View privilege, and the

second must have the Provision privilege. They are to be used in the Read Community and Write Community login prompts, respectively.

3. Do not use the following characters in the UserName or Password entries because they could be misinterpreted by the CopperEdge unit as other commands: backslash (\), double quotes (“), or ending bracket (]). The # character is not allowed.

4. Spaces are not supported in UserName strings. For first and last names, use uppercase and lowercase, or an underscore, to separate them (UserName or user_name). Due to a trait of the underlying software, passwords are not completely case sensitive; the first character of a password may be accepted regardless of case. After the first character, the system is case-sensitive to all succeeding password characters.

5. The default operator (ce200) is also the system Security Administrator, and is the only operator that can: configure new operators; change or assign operator privileges, context and passwords; delete any active or configured operator; or access/list configured and active operators. Configured operators may change their own passwords.

Objects:

*Name	The operator name.
Password	The operator password.
State	The current state of a row in the cmOperator table: Active—In a Get command, the row is actively being used by the CE200. In a Set command, overwrites an existing row. CreateAndGo—Creates a new row that will be immediately used by the CE200. This value prevents an existing row from being overwritten. Destroy (alias: delete)—Deletes a row.
Context	The method used by this operator to access the system: SNMP—An SNMP manager such as CopperView or H-P OpenView. nonSNMP—Telnet and the CopperCraft CLI. All—The default.
Privilege	The privilege level for this operator: View—The default Monitor Provision Security

(*) indicates object used as an index.

The following table presents a matrix of the Read and Write privileges associated with the various operator privilege levels.

	View	Monitor	Provision	Security
Read	All groups except cmOperator and cmActive	All groups except cmOperator and cmActive	All groups except cmOperator and cmActive	All
Write	Change own password only	Change own password Configure and conduct loopback and similar tests	Change own password Set configurable performance monitoring objects (thresholds, etc.) Set configurable service-affecting objects (cmIface entries, etc.) Configure and conduct loopback and similar tests	All, including creating, configuring, and deleting operators
Other	N/A	N/A	N/A	Establish and conduct FTP sessions

cmParamSummaryTable

This group has objects that focus on the values for Peak Cell Rate (PCR) and Sustainable Cell Rate (SCR) in both the rtVBR and nrtVBR classes of service for all VCs on a WAN port. The objects also focus on the percentage of over (or under) subscription for all VCs on a WAN port.

With the first four objects, you can determine the amount of bandwidth for the PCR and SCR you have assigned to all VCs on the port. With the second four objects, you can raise or lower the upper limit for the bandwidth of PCR and SCR assigned to all VCs on the port. You can allow extreme oversubscription (up to 2500 percent), and can change the subscription limits at any time simply by setting new values.

If you set the OsfRtSCR and the OsfNrtSCR objects at 100, you are restricting the value for SCRsumRtVBR to 110,000 cells/second and the value of SCRsumNrtVBR to 110,000 cells/second. Since 110,000 cells/second is slightly above the capacity of a DS3 port (107,000 cells/second), you are assuming that all of the VCs will never be active at once. Similarly, if you set the OsfRtSCR and the OsfNrtSCR objects at 200, you are restricting the values for SCRsumRtVBR and SCRsumNrtVBR to 220,000 cells/second each (more than twice the capacity of a DS3 port).

The CE200 will not allow you to assign any bandwidth to new VCs when you have reached your subscription limits.

Syntax: *command* cmparamsummary *[pii]*

You must specify a WAN port, not a DSL port, and do not include a VC identifier. Also, do not attempt to specify the constituent links of an IMA group; use the “virtual PII,” including the IMA port number, 41.

Example: To display the parameters for port 1, slot 3:

```
get cmpa [1.3.1]
Group: cmParamSummaryTable
Instance: [1.3.1.0]
PII                    =  1.3.1.0
PCRsumRtVBR            =  0
SCRsumRtVBR            =  0
PCRsumNrtVBR           =  0
SCRsumNrtVBR           =  0
OsfRtPCR               =  100
OsfRtSCR               =  100
OsfNrtPCR              =  100
OsfNrtSCR              =  100
```

Note: The service categories for nrtVBR and rtVBR are not supported in this release. You can view the configurations in the quality of service tables, but the nrtVBR and rtVBR configurations will not be in force during transmission of data.

Objects:

*PII	The Permanent Interface Identifier of the WAN port.
PCRsumRtVBR	The sum of all settings for PCRs in the rtVBR class of service on this port. Units are in cells/second. A DS3 port can transmit at 107000 cells/second, but the maximum setting for PCR is 110000 cells/second.
SCRsumRtVBR	The sum of all settings for SCRs in the rtVBR class of service on this port. Units are in cells/second.
PCRsumNrtVBR	The sum of all settings for PCRs in the nrtVBR class of service on this port. Units are in cells/second.
SCRsumNrtVBR	The sum of all settings for SCRs in the nrtVBR class of service on this port. Units are in cells/second.
OsfRtPCR	The percentage of oversubscription for all PCRs in the rtVBR class of service on this port: 0 to 2500. The default is 100.
OsfRtSCR	The percentage of over subscription for all SCRs in the rtVBR class of service on this port: 0 to 2500. The default is 100.
OsfNrtPCR	The percentage of over subscription for all PCRs in the nrtVBR class of service on this port: 0 to 2500. The default is 100.
OsfNrtSCR	The percentage of over subscription for all SCRs in the nrtVBR class of service on this port: 0 to 2500. The default is 100.

(*) indicates object used as an index.

cmProxyArpTable

This group controls the Proxy ARP function for the specified DSL port or IMUX bundle. Proxy ARP is valid only if the netmodel is set to IP and the network connection is through the 10/100Base-T Ethernet interface on the primary System Control module.

If you try to set cmProxyArp on a port not configured for IP, the CE200 will return an error message.

If a DSL port or IMUX bundle is configured with cmProxyArp (RowStatus displays Active), and you then change its netmodel to an incompatible value, that row is automatically deleted.

To disable cmProxyArp on a DSL port, you can either change its networking model to something other than IP, or set RowStatus to Destroy.

Syntax: *command* cmproxyarp *[pii] object_1*

Example: To configure port 2, slot 6 for Proxy ARP:

```
set cmpr [1.6.2] rowstat=active
Group: cmProxyArpTable
Instance: [1.6.2.0]
PII                 =  1.6.2.0
RowStatus           =  Active
```

Objects:

*PII	The Permanent Interface Identifier of the SDSL interface.
RowStatus	The operational state of a row in the cmProxyARP table: Active—In a Get command, the row is actively being used by the CE200. In a Set command, overwrites an existing row. CreateAndGo—Creates a new that will be immediately used by the CE200. This value prevents an existing row from being overwritten. Destroy (alias: delete)—Deletes a row.

(*) indicates object used as an index.

cmRadius

Use this group to configure the CE200 for operation with a remote Radius server for central control of user authentication and privileges, and to track the identity and address of users accessing the system. For security reasons, the cmRadius group is write-accessible only by operators with the Security privilege level; SNMP users cannot configure or change the table.

Once the Radius server is provisioned and *cmRadius* Authentication set to enabled, the CopperEdge user base and configuration of its security features are under the sole control of the Radius server and can no longer be controlled from the CopperEdge except as described below. The cmOperator table and its contents continue to exist, but are not serviceable as long as the cmRadius Authentication remains enabled.

Radius uses separate servers to perform its authentication and accounting (user audit trail) functions. Also, the cmRadius group allows two backup (secondary and tertiary) authenticating and accounting servers to be designated in case the primary server is inaccessible. You can also configure the CopperEdge to reclaim the authentication function in case none of the Radius authenticating servers can be reached.

For more information about access parameters (context and privilege), see *cmOperatorTable* on page 155.

Syntax: *command* cmradius

Example 1: To configure the CE200 for operation with a Radius server:

```
set cmr localfall=enabled authkey=4cr37ei
authprimaryip=10.122.4.4 authprimaryport=1645
acctprimaryip=10.122.4.6 acctprimaryport=1646
```

Example 2: To display the Radius server information:

```
get cmr
Group: cmRadius
Authentication       =  Disabled
LocalFallback        =  Enabled
AuthKey              =  ""
AuthPrimaryIpAddr    =  10.122.4.4
AuthPrimaryPort      =  1645
AuthSecondaryIpAddr  =  0.0.0.0
AuthSecondaryPort    =  1645
AuthTertiaryIpAddr   =  0.0.0.0
AuthTertiaryPort     =  1645
AcctPrimaryIpAddr    =  10.122.4.6
AcctPrimaryPort      =  1646
AcctSecondaryIpAddr  =  0.0.0.0
AcctSecondaryPort    =  1646
AcctTertiaryIpAddr   =  0.0.0.0
AcctTertiaryPort     =  1646
```

Notes:

1. When configuring the cmRadius table, first set all of its objects except authentication, then perform a get cmRadius and check the configuration. When you are sure the configuration is correct, set Authentication to Enabled.
2. Be sure to use whichever port numbers are recognized by your Radius servers. Although the officially assigned port numbers for Radius Authentication and Accounting are 1812 and 1813 respectively, many current radius servers still use the port numbers of the original RFC: 1645 for authentication and 1646 for accounting. Be sure that the numbers you assign in the CopperEdge cmRadius configuration match those used by your remote Radius servers.

Objects:

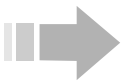
Authentication	The Radius authentication for this CE200: Enabled or Disabled (the default).
LocalFallback	With Authentication enabled, use this object to specify whether or not to fall back to local authentication if the Radius server is unreachable. VALUES: *Enabled* or *Disabled* (Default)
AuthKey	A character string of 4 to 16 characters that must match the authentication key for this Radius client, and which is currently contained on the Radius server(s). The actual auth key does not display.
AuthPrimaryIpAddr alias: authpripaddr	The IP address of the primary authenticating Radius server. It must be a valid address if Authentication is Enabled.

AuthPrimaryPort alias: authprport	The UDP port number of the primary authenticating Radius server (see Note 2 above).
AuthSecondaryIpAddr alias: authseipaddr	The IP address of the secondary authenticating Radius server, if available.
AuthSecondaryPort alias: authseport	The UDP port number of the secondary Radius authenticating server, if available.
AuthTertiaryIpAddr alias: authteipaddr	The tertiary IP address of the authenticating Radius server, if available.
AuthTertiaryPort alias: authteport	The UDP port number of the tertiary Radius authenticating server, if available.
AcctPrimaryIpAddr alias: acctpripaddr	The primary IP address of the accounting Radius server.
AcctPrimaryPort alias: acctprport	The UDP port number of the primary accounting Radius server (see Note 2 above).
AcctSecondaryIpAddr alias: acctseipaddr	The IP address of the secondary accounting Radius server, if available.
AcctSecondaryPort alias: acctseport	The UDP port number of the secondary accounting Radius server, if available.
AcctTertiaryIpAddr alias: acctteipaddr	The IP address of the tertiary accounting Radius server, if available.
AcctTertiaryPort alias: acctteport	The UDP port number of the tertiary accounting Radius server, if available.

cmSDSLTestTable

This group allows you to control the SDSL loop test function, which can be used to help isolate a defective or inoperative DSL link (Loop Profile), and to determine the highest data rate it can support (SeekMaxRate). The test can be used to verify loop integrity and continuity (shorts/opens) within 500 feet of the DSL physical port, and thus can localize problems as being within the CO or beyond it.

With the aid of an on-site assistant at the customer premise, the SDSL Loop Test can record data that can aid in assessing the relative quality of the loop, as well as verifying basic continuity from the CE200 to the CPE.



N O T E

Although you can enter CpeDownload, it is typically set automatically, when necessary, as part of the cmMaintCmd BulkCPEDownload sequence for each CPE included in the bulk download.

N O T E

When you set CPEDownload through the cmSDSLTest group, the CE200 reduces the data rate of the CPE connected to the specified PII to a level supported by its prom code, if necessary. When the download is complete (or the test times out), the data rate for that CPE is restored to its original setting. The CPEDownload command will preempt all other cmSDSLTest command types.

Syntax: *command* cmsdsltest [*index*]

Example 1: To configure a loop test on port 4, slot 6:

```
set cmsd [1.6.4] type=profile action=start
```

Example 2: To query port 4, slot 6 for its loop test configuration:

```
get cmsd [1.6.4]
Group: cmSDSLTestTable
Instance: [1.6.4.0]
Index                =  1.6.4.0
Type                 =  Profile
Action               =  Start
Status               =  InProgress
TestCompleteTime     =  2000/06/14-09:23:59
ContinuityTestResult =  0
NearEndLoopProfile   =  Pending
DebugResult          =  10
MaxRate              =  Unknown
```

Note: The cmSDSLTest group does not apply to IMUX bundles. Individual DSL ports must be unbundled (use cmBundle and set the pertinent MemberPII to 0.0.0) before testing. When testing is complete, reinstate the port as a constituent of the bundle.

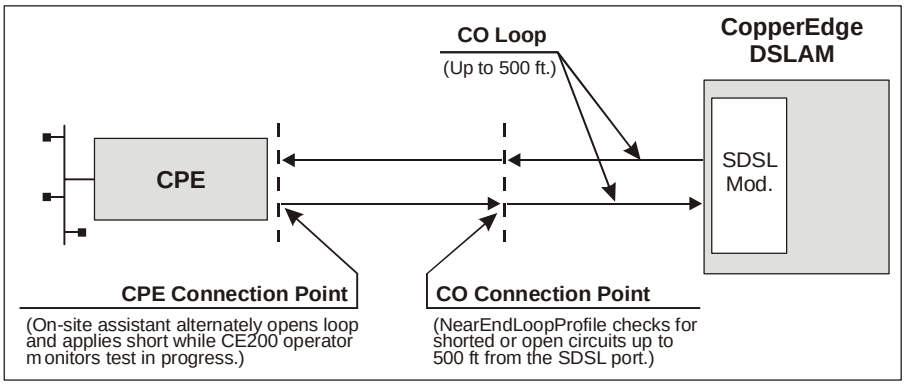
Objects:

*Index	The Permanent Interface Identifier (PII) of the interface.
Type	Type of test: None, Profile, MaxRate, CPEDownload
Action	Begin or interrupt test of the specified interface: None, Start, Stop When viewing test information (get results), this object displays None.
Status	The current status of the test function on the specified port: None—No test has been started since powerup. Complete—At least one test was run and stopped since powerup. InProgress—A test is currently running.
TestCompleteTime	The calendar time when the last test finished. If no test has been run, the value is a null string.
ContinuityTestResult	Displays an integer corresponding to the current result of the continuity test. This object is updated during the continuity test at least every second.It retains its value after the test is stopped.

NearEndLoopProfile	The current result of the Loop Profile test: Normal, PossibleShort, PossibleOpen, Pending CPETalking—Applicable within approximately 500 feet of the specified SDSL physical port.
DebugResult	A generic 32-bit value for factory use only.
MaxRate	Result of the Seek Max Rate test as the highest data rate (kbps) the loop can support: 1568, 1040, 784, 416, 320, 208, 160, None, Unknown

(*) indicates object used as an index.

The following example is a functional representation of the SDSL loop test. Instructions for performing the test are provided the *CopperEdge 200 Installation and Operating Guide*.



SDSL Loop Tests

cmServiceClass
(alias: sclass)

Use this group to configure the CE200 to support the DSL Class of Service function. You can establish up to four different classes (A, B, C, or D) and assign a relative weight to them. Depending on the class assigned to it (through the cmIface ServiceClass object), incoming traffic *from* a given DSL port is treated as higher or lower in priority by the CE200 CPU at times when the CPU packet queue is congested. The Class of Service applies only *within* the CE200, and only to packets *received at* the DSL ports; it does not apply to packets to or from the WAN interfaces. The Class of Service assigns priority to *packets*, not to individual bytes. Thus, a DSL port with a lower class of service might occasionally have higher throughput than a port with a higher class, just because the former is sending larger packets than the latter.

The Class of Service feature is provided primarily to facilitate processing of digital voice as well as data communication through the same CE200 by giving preference to voice traffic, which is more sensitive to delay than data. In fact, the feature only comes into play during periods when the CPU on the CE200 receives more packets than it can process in a period.

The default values for the four classes of service are A=4, B=3, C=2, D=1. Thus, if all classes were configured for their default settings and CPU traffic became congested, 4 out of every 10 Class A packets would be processed (40% probability). For class B, 3 out of every 10 would be processed, etc. The proportion of packets of a specific class among all packets forwarded during congested periods is determined by the ratio of the class weight divided by the sum of all class weights, as follows:

$$\frac{wtX}{wtA + wtB + wtC + wtD} \times 100 = pX$$

Where *wtX* is the numeric weight of the class of interest, and *P* is the percentage of packets in the composite data stream that will come from interfaces assigned this service class.

To properly configure cmServiceClass, the following rules apply:

- Every class must have a weight value from 1 to 8. No class can have a weight of zero.
- Higher classes, such as A or B, must have higher weight values than lower classes, such as C or D.
- Two or more classes can be set to the same value, but any distinction between those classes is lost. Setting all classes to the same value eliminates all classes.
- Every configured (operational) DSL port has *some* class of service. The default is Class D.

Syntax: *command* cmserviceclass *object_1...object_4*

Example 1: To increase the preference level for class A:

```
set cmse a=8
```

Example 2: To change four classes of service to three classes by elevating levels A and B well above the third or basic level:

```
set cmse a=8 b=5 c=1 d=1
```

Example 3: To see how the classes are weighed:

```
get cmse
Group: cmServiceClass
Aweight              =  8
Bweight              =  5
Cweight              =  1
Dweight              =  1
```

Objects:

Aweight	The weight of Class A: 8 to 1, where 8 is the highest. The default is 4.
Bweight	The weight of Class B: 8 to 1, where 8 is the highest. The default is 3.
Cweight	The weight of Class C: 8 to 1, where 8 is the highest. The default is 2.
Dweight	The weight of Class D: 8 to 1, where 8 is the highest. The default is 1.

cmShelfTable

This group displays information about the basic physical configuration (which slots are filled with which type of compatible modules) and current operational status of the CE200 shelf.

The object, Command, lets you restart the shelf without restarting an entire system.

Syntax: *command* cmshelf [*index*]

Example: **get cmsh [1.0.0]** or **get cmsh [1]**

```
Group: cmShelfTable
Instance: [1.0.0.0]
Index                =  1.0.0.0
ObjectClass          =  CE200Shelf
Operability          =  Enabled
HWType               =  CE200Shelf
UpTime               =  7 day 1 hour 53 min 10.0 sec
                        (2000/09/28-14:45:43)
Master               =  0.0.0.0
BoardMap             =  ff.d.0.0
ShelfNumber          =  1
Command              =  None
AlarmPanelType       =  DS3Switch
```

Objects:

*Index alias: idx	The number associated with the shelf.
ObjectClass	The Object Class to which this shelf belongs: Shelf CE200Shelf CE150Shelf

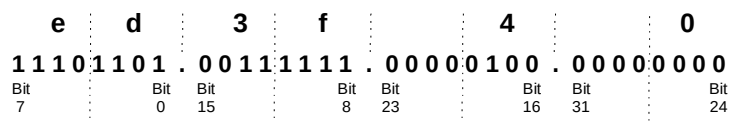
Operability alias: oper	The current operational state of this shelf: Unknown Disabled Enabled Testing For a complete definition of Operability as it relates to this group, see *Operating Status* on page 20.
HWType	The hardware type of this shelf: CE200Shelf or CE150Shelf.
UpTime	The elapsed time since the system was reset, and the calendar date/time when the operator logged in to the system. The elapsed time is expressed as *day hour min sec*. This is the value of the SysUpTime counter. The calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.
Master	The PII of the Control module operating as the master for this shelf.
BoardMap	A 32-bit (four-byte) bitmap in hexadecimal format of the CE200, indicating which slots in the shelf are populated with modules. Bit 0 indicates the fan assembly, bit 1 indicates slot 1, and so on through bit 18 which indicates slot 18. Bits 19 through 35 are not used, and are always zero. See the paragraph immediately following this table for an example and a graphic key to the bit arrangement.
ShelfNumber	The position of this shelf in a rack: 1 to 4.
Command alias: cmd	An operator command: None or Restart.
AlarmPanelType	The type of alarm panel installed in the chassis: None Standard DS3Switch

(*) indicates object used as an index.

CE200 Board Map — As described in the table above, this bitmap lets you see which slots are filled with modules (Bit 0 = Fan Tray Assembly, Bit 1 = slot 1 . . . Bit 18 = slot 18). *Example:*

```
BoardMap = ed.3f.4.0
```

translates to a CE200 with a fan tray, an SCM in Slot 2, a WAN board in Slot 3, a Buffer Module in Slot 5, DSL Modules in slots 6 through 13, and a Power Supply Module in Slot 18. The diagram below shows the arrangement and includes a key to the order of the data bits.



cmSonetSdhTable
(alias: cmsnt)

This group controls configuration and provides status information regarding the SONET/SDH interface.

Syntax: *command* cmsonetsdh [*pii*] *object_1*

Example:

```
set cmsnt [1.3.1] clock=local cellpayloadscr=enable

Group: cmSonetSDHTable
Instance: [1.3.2.0]
ClockSource      =  1.6.2.0
Wavelength       =  len1310nm
cellPayloadScrambling=  Disable
```

Objects:

*PII	The Permanent Interface Identifier of the SONET interface.
clockSource	Indicates whether the SONET/SDH interface will use its own transmit clock (LocalTiming) or use the recovered receive clock (LoopTiming).l
wavelength	The wavelength of the SONET transport network in nanometers: len1310nm (default), or len1550nm
cellPayloadScrambling	Controls cell payload scrambling on the ATM interface: Disable (default), or Enable

(*) indicates object used as an index.

cmStateTable

This group allows you to display information about the current state of the specified shelf, module, port, or link.

You can also set the administrative state of DSL and WAN ports by using the AdminState object in this group instead of the MIB-II ifAdminStatus. In fact, we recommend you only use cmState's AdminState object, although you can still use MIB-II's ifAdminStatus for backward compatibility.

Syntax: *command* cmstate [*genericclass*, *index*] *object_1*

Example 1: To query the module in slot 6 for its information:

```
get cmst [board, 1.6]
Group: cmStateTable
Instance: [Board, 1.6.0.0]
GenericClass         =  Board
Index                =  1.6.0.0
ObjClass             =  SDSL30xModule-24
OperState            =  Disabled
LastChange           =  0 day 0 hour 0 min 38.0 sec
                        (2000/06/05-15:08:24)
AdminState           =  Enabled
ProvisionState       =  notApplicable
AggregateAdminState  =  2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.
                        2.2.2.2.2
AggregateOperState   =  1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.
                        1.1.1.1.1
AggregateProvisionSt =  2.3.3.2.2.2.2.2.2.2.2.2.2.2.2.2.
                        2.2.2.2.2
```

Example 2: To disable port 1, slot 6:

```
set cmst [port, 1.6.1] admin=disabled
```

Example 3: To query port 1, slot 6 for its information:

```
get cmst [port, 1.6.1]
Group: cmStateTable
Instance: [Port, 1.6.1.0]
GenericClass         =  Port
Index                =  1.6.1.0
ObjClass             =  SDSLPort
OperState            =  Disabled
LastChange           =  0 day 0 hour 0 min 50.0 sec
                        (2000/06/27-12:39:39)
AdminState           =  Disabled
ProvisionState       =  unProvisioned
AggregateAdminState  =
AggregateOperState   =
AggregateProvisionSt =
```

Objects:

*GenericClass alias: gclass	The general class of the resource: System Shelf Board Port Link CPE Support
*Index alias: Idx	The Permanent Interface Identifier (PII) of the resource.

ObjClass alias: Class	The specific class of the resource: System class: System Operator Shelf class: Shelf CE200Shelf CE150Shelf Board Class: Board SystemControlModule SDSL10xModule-1 SDSL10xModule SDSL30xModule SDSL30xModule-24 IDSLModule-24 BufferControlModule BufferControlModule2 V.35-WAN DS3FR-WAN DS3ATM-WAN QuadT1-WAN G.liteModule-24 DualT1-WAN—Not supported in this release. SDSLModule-8—Not supported in this release. T1Module-12 SystemControlModule3 SONETSingleMode-WAN SONETMultiMode-WAN ADSLMultiModeModule-24A ADSLMultiModeModule-24B GSHDSLATMModule-24IMAWANModule GSHDSLPacketModule-24 Port class: Port SDSLPort EthernetPort V.35Port RS232Port DS3FRPort IDSLPort DS3ATMPort T1Port-WAN T1Port G.litePort G.dmtPort T1Port-LC SONETPort G.shdslPort IMAPort Link class: Link LCPortLink EthernetLink FrameRelayLink FrameRelayPVC ATMLink InterShelfTrunk CPE class: CR CR201-10x CR201-30x CR201-SDSL CR201IDSL CPE-SDSL CPE-IDSL Netopia-SDSL CPE-T1 Support class: Support PowerModule FanModule
OperState	The current operational state of the resource: Enabled Disabled Testing Unknown—The default Testing applies only to ports or links, and then only in those elements in which the feature has been implemented. This object will not be Enabled if the AdminState object is Disabled. For a complete definition of OperState as it relates to this group, see *Operating Status* on page 20.

LastChange	The elapsed time since the system was reset, and the calendar date/time when the operator logged in to the system. The elapsed time is expressed as *day hour min sec*. This is the value of the SysUpTime counter. The calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.
AdminState	The administrative state of the resource. Unknown Disabled Enabled—The default This object is an extension of MIB-II's ifAdminStatus. **Note:** Not all of the ifAdminStatus values are supported. If this object is set to Disabled, the OperState object will not be Enabled.
ProvisionState	The provision state of the resource. notApplicable—Used for non-DSL ports. unProvisioned—The interface is not configured. preProvisioned—The interface is configured, but a CPE is not trained on the line (OperState= Disabled). provisioned—The interface is configured and a CPE is trained on the line (OperState=Enabled). This object only applies to DSL ports and IMUX bundles. The default for DSL ports is unProvisioned. The default for non-DSL ports is notApplicable. For more information about provisioning, see *Notes on Provisioning: 1. When you configure a port using cmIface, the port's ProvisionState changes to either preProvisioned (a CPE is not trained on the line) or Provisioned (a CPE is trained on the line).* on page 171.
AggregateAdminState	The operational state of all ports on a module identified by the PII in the Index object, if GenericClass=Board. 1 = Disabled 2 = Enabled 4 = Unknown Up to 32 ports can be listed in the format *byte1.byte2. byte3*..., where byte1 represents the AdminState of port 1, byte 2 represents the AdminState of port 2, and so on. For example, 1.2.1 indicates that port 1 is disabled, port 2 is enabled, and port 3 is disabled. This object is only supported for the generic class, Board.

AggregateOperState	The operational state of all ports on a module identified by the PII in the Index object, if GenericClass=Board. 1 = Disabled 2 = Enabled 3 = Testing 4 = Unknown Up to 32 ports can be listed in the format *byte1.byte2. byte3*..., where byte1 represents the OperState of port 1, byte 2 represents the OperState of port 2, and so on. For example, 1.2.2.indicates that port 1 is disabled, port 2 is enabled, and port 3 is enabled. This object is only supported for the generic class, Board.
AggregateProvisionState	The operational state of all ports on a module identified by the PII in the Index object, if GenericClass=Board. 1 = notApplicale 2 = unProvisioned 3=preProvisioned 4 = Provisioned Up to 32 ports can be listed in the format *byte1.byte2. byte3*..., where byte1 represents the ProvisionState of port 1, byte 2 represents the ProvisionState of port 2, and so on. For example, 2.4.3 indicates that port 1 is unProvisioned, port 2 is Provisioned, and port 3 is preProvisioned. This object is only supported for the generic class, Board.

(*) indicates object used as an index.

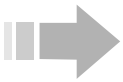
Notes on Provisioning:

1. When you configure a port using cmIface, the port's ProvisionState changes to either preProvisioned (a CPE is not trained on the line) or Provisioned (a CPE is trained on the line).
2. If a CPE trains on a port whose ProvisionState returns unProvisioned, the Events table records the message PortMisprovisioned. To resolve this event, configure the port in cmIface. The port's ProvisionState changes to Provisioned.
3. If you reconfigure a port (change the IP address or netmodel), the port's ProvisionState remains Provisioned.
4. If you change the AdminState of a Provisioned port to Disable, the port's Operstate changes to Disabled and the ProvisionState changes to preProvisioned.
5. If you delete the cmIface entry for a Provisioned port, the port's ProvisionState changes to unProvisioned.
6. To disable a port, set AdminState=Disabled. The port's OperState changes to Disabled, and the ProvisionState changes to preProvisioned.

cmSubIfaceTable

This group allows you to create and manage virtual circuits (VCs) on a DSL link. The table provides DLCI mapping information and DSL DLCI-specific configuration options.

Before a cmSubIface entry can be configured, however, there must first be an entry in the cmIface table for the corresponding DSL interface, or if configuring an IMUX multilink, the corresponding bundle must have an entry in the cmIface group. Multiple cmSubIface entries can be configured for any DSL port up to the value of frDlcmi MaxSupportedVCs (currently 64). Also, DLCIs 528 and 529 are not allowed on DSL links.



N O T E

G.SHDSL Modules currently support a maximum of 8 VCs per port.

Syntax: *command* cmsubiface [*pii*]

Example 1: To configure the interfaces for a DSL link and its VC:

```
set cmiface [1.6.1] netmodel=cross-connect encap=q922
set cmsubiface [1.6.1.100] rowstatus=create
```

Creating the cmSubIface entry for this DLCI automatically creates the frCircuit and cmFrCircuit entries for 1.6.1.100.

Example 2: You must still configure the entry for 1.3.1.200:

```
set cmsubiface [1.6.1.100] destpii=1.3.1.200
```

Note: When configuring DSL VCs, you can assign any number from 16 to 991 inclusive, except 528 and 529, which are reserved.

Objects:

*PII	The Permanent Interface Identifier of the interface, including the virtual circuit.
DestPII	The PII of the WAN interface, including the VC, to which the DSL interface is mapped.
Name	The user-assigned name for this DLCI (up to 31 characters of text, which must begin with an alpha character). The name you assign for the DLCI will be added to the cmNameTable.

RowStatus	The current state of a row (for the DLCI) in the cmSubIface table: Active—In a Get command, the row is actively being used by the CE200. In a Set command, overwrites an existing row or activates a row that displays NotInService. CreateAndGo—Creates a new row that will be immediately used by the CE200. This value prevents an existing row from being overwritten. Destroy (alias: delete)—Deletes a row. NotInService—In a Get command, the row exists but is not being used by the CE200. In a Set command, creates a new row or deactivates a row that is currently Active.
Priority	The priority queuing for packets transmitted to the subscriber: High or Low (the default). This object is only valid for DSL VCs with NetModel=Cross-Connect. For voice or other real time packets, use High. For netmodels other than Cross-Connect, preference is automatically given to voice or real time packets.

(*) indicates object used as an index.

cmSystem

This group allows you to display or configure information about the current operating state of the system.

Syntax: *command* cmsystem *object_1...object_5*

Example 1: To display the current operating status:

```
get cmsy
Group: cmSystem
ObjectClass          =  System
OperState            =  Enabled
Version              =  E 2.11
Master               =  0.0.0.0
ConfigFileName       =  config.tgz
CalendarTime         =  2000/06/14-09:34:51
MyPII                =  1.2.0.0
PrimaryPII           =  1.2.0.0
SecondaryPII         =  0.0.0.0
Redundancy           =  NotAvailable
ShelfCount           =  1
ExpIpSubNet          =  192.168.250.0
ConfigSynch          =  NotSaved
Command              =  None
CommandStatus        =  None
```

Example 2: To display the date/time stored in the system real time clock:

```
get cmsy cal
Group: cmSystem
CalendarTime         =  2000/06/14-09:35:47
```

Example 3: To save the current configuration to the Config file:

```
set cmsy command=save
```

Notes:

1. The readConfig command option has been removed. To reinstate a stored configuration, use the cmMaintCmd ConfigRestore option (page 148).
2. In a redundant system, configuration changes related to routing data (ipRoute, cmIface, frCircuit, cmAtmVcl, etc.) are not immediately propagated to the secondary System Control Module. Instead, when the operator issues a save, the file is saved locally, copied to the Secondary, and the Secondary is restarted by the Primary. When the Secondary comes up, the settings for these objects are read from the saved Config file and then take effect.
3. In a redundant system, do not try to save a Config file if the primary has lost contact with the secondary. Attempting to do so may result in a loss of configuration (especially settings related to the remote WAN ports).

Objects:

ObjectClass alias: Class	The object class assigned to this resource: System, Shelf, Board, etc.
OperState	The current operational state of the resource: Enabled Disabled Testing Unknown For a complete definition of OperState as it relates to this group, see *Operating Status* on page 20.
Version	The system version: *type major.minor*. For example, R 4.1 (release version 4.1).
Master	Not supported in this release.
ConfigFileName	The name of the saved configuration file (config.tgz or config.txt).
CalendarTime	The current date and time stored in the system real time clock in the format *yyyy/mm/dd-hh:mm:ss*.
MyPII	The Permanent Interface Identifier of the System Control Module to which this console session is attached.
PrimaryPII	The Permanent Interface Identifier of the System Control Module currently designated as Primary.
SecondaryPII	The Permanent Interface Identifier of the System Control Module currently designated as Secondary.
MgmtPII	The logical Permanent Interface Identifier of the WAN VC or Ethernet port. The IP address assigned to this PII is used as the source IP address, and is provided in unsolicited SNMP messages. In a redundant system, this PII is configured only for the preferred side.
Redundancy	The status of the redundancy functionality: Disabled Enabled NotAvailable When Enabled, the secondary control and WAN complex can take over as primary.

ShelfCount	The number of shelves reporting as members of this system.
ExpIpSubnet	The IP address that will be interpreted as a Class C subnet for internal system use. Addresses from the subnet are used to assign each expansion SCM with an IP address for use in loading. When assigning addresses, be sure that the address range does not conflict with any other assigned IP addresses, and provide security filtering where necessary to prevent external access. The format for the Class C subnetwork is A.B.C.0, where A, B, and C are each in the range of 0 to 255.
ConfigSynch	The current state of system configuration: Saved—The current configuration has been saved on both the Primary and Secondary modules for a redundant system; or on the local SCM for a non-redundant system. NotSaved—The configuration has changed and the changes have not been saved.
Command alias: cmd	When used in a Get statement (such as **get cmsy**), shows the last command issued. When used in a Set statement (such as **set cmsy command=restart**), issues a subsequent value as a command for the module to perform. Command values are: None ReadConfig (Not supported; see Note 1 above) TakeOver Relinquish Restart SaveConfig—Do not use if the CommandStatus object displays InProgress. The Command object is not stored as part of a saved configuration. **Caution:** The Restart command is service affecting; it initiates a warm restart of this entire CE200.
CommandStatus	The status of the last command issued by the operator: None Pending InProgress Aborted Succeeded Failed

cmTracePiiTable

This troubleshooting tool allows you to visually see some of the packet traffic traversing the CE200. In previous releases, this information was available via the DIAG port, but has now been moved to the *CopperCraft* interface. If you are a system administrator with *Security* level access, you can use cmTracePii to:

- Enable or disable tracing of a specific PII
- Turn off tracing of all PIIs
- Hide or show tracing display of the session issuing the command.

When the Security privilege operator logs out, the tracing display defaults to "show."

For more information on using cmTracePii, see the Troubleshooting chapter in your Installation and Operating Guide.

Syntax: *command* cmtracepii [*pii*]

Example 1: To start a trace on a specific PII:

```
set cmtracepii [1.6.1] raw=false
```

Objects:

*PII	The Permanent Interface Identifier of the interface, including the virtual circuit if applicable.
RowStatus	Used to enable or disable tracing on the specified interface: Active (Default) – Tracing enabled. Destroy – Tracing stopped
Raw	Specifies whether the traceroute output will be a raw hex dump or a verbose decode of the data: *True* (raw hex output) or *False* (Default)

(*) indicates object used as an index.

cmTrapAlarmTable
(alias: tralarm)

This group allows you to display information about current alarms (alarms that have not been cleared). You can also reset alarms for which the alarming condition has been corrected or is no longer present.

The cmTrapAlarm table is one of five cmTrap groups that deal with events and alarms.

From the Craft interface, you can also use the Alarms command to monitor alarms in real time as described in *Alarms and the Alarm Log* of the *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmtrapalarm [*seqnum*] *object_1*

You can view alarms individually by specifying the sequence number or using the Getnext command. To view the entire contents of the Alarm table, use the Getall command.

Example 1: To display information for alarm #223:

```
get tralarm [223]
Group: cmTrapAlarmTable
Instance: [223]
SeqNum                =  223
Type                  =  LinkDown
ObjectClass           =  FrameRelayLink
ClassId               =  1.4.1.120
ProbableCause         =  0
Severity              =  MinorAlarm
TimeTag               =  2000/06/12-15:45:54
Text                  =  1.4.1.120, "", 1.4.1.120, ""
```

Example 2: To reset the alarm status of alarm #223:

```
set tralarm [223] severity=info
```

Objects:

*SeqNum	The sequence number of this event (an integer that is automatically augmented as new events occur).
Type	The type of event: ColdStart, BoardDown, BoardUp, LinkDown, LinkUp, WANLinkUp, WANLinkDown, DLCIStateDisabled, DLCIStateEnabled, LoginFailed, LoginSucceeded, LoginSuspended, LoginAllowed, LoginsSaturated, LoginsAvailable, Logout, BootFileFailed, BootFileSucceeded, AttributeChanged, ConfigReadFailed, ConfigReadSucceeded, ConfigWriteFailed, ConfigWriteSucceeded, StateChange, Diagnostic, BoardRestart, FanFault, FanfaultClear, PowerSupplyFault, PowerSupplyFaultClear, AtmVccDown, AtmVccUp, TestStatus, TestError, LoopStatusChange, RisingThresholdAlert, FallingThresholdAlert, DS3LineStatusAlarm, DS3LineStatusClear, IDSLTimingLossAlarm, IDSLTimingLossClear, RedundancyChanged, RoleChanged, RedundancyConflict, MaintSucceed, MaintFailed, RateFallBackAlarm, RateFallbackClear, DS1LineStatusAlarm, DS1LineStatusClear, EndPointConflictAlarm, PortMisprovisioned, PortMisprovisionedClear, CpePlugAndPlayFailure, CpePlugAndPlayClear, SONETSectionStatusAlarm, SONETSectionStatusClear, SONETLineStatusAlarm, SONETLineStatusClear, SONETPathStatusAlarm, SONETPathStatusClear, SHDSLConfigInitFailure, SHDSLLoopbackFailure, SHDSLPowerBackOff, SHDSLDeviceFault, SHDSLNoNeighbor

ObjectClass	The object class of the resource reporting the event (not all listed classes are supported): System class: System, Operator Shelf class: Shelf, CE200Shelf, CE150Shelf Board Class: Board, SystemControlModule, SDSL10xModule-1, SDSL10xModule, SDSL30xModule, SDSL30xModule-24, IDSLModule-24, BufferControlModule, BufferControlModule2, V.35-WANDS3FR-WAN, DS3ATM-WAN, QuadT1-WAN, G.liteModule-24, DualT1-WAN., SDSLModule-8, T1Module-12, SystemControlModule3, IMAWANModule, ADSLMultiModeModule-24A, ADSLMultiModeModule-24B, SONETSingleMode-WAN, SONETMultiMode-WAN, G.shdslPacketModule-24, GSHDSLATMModule-24 Port class: Port, SDSLPort, EthernetPort, V.35Port, RS232Port, DS3FRPort, IDSLPort, DS3ATMPort, T1Port-WAN, T1Port, G.litePort, G.dmtPort, T1Port-LC, SONETPort, G.ShdslPort, IMAPort Link class: Link, LCPortLink, EthernetLink, FrameRelayLink, FrameRelayPVC, ATMLink, InterShelfTrunk CPE class: CR, CR201-10x, CR201-30x, CR201-SDSL, CR201IDSL, CPE-SDSL, CPE-IDSL, Netopia-SDSL, CPE-T1, CPE-GSHDSL Support class: Support, PowerModule, FanModule
ClassId	The 32-bit PII of the resource in the class.
ProbableCause	A 32-bit, event-specific cause code (not supported).
Severity	Severity level of the event: None, Information, MinorAlarm, MajorAlarm, CriticalAlarm, Warning
TimeTag	System calendar time when the event occurred.
Text	Additional information that describes the event.

(*) indicates object used as an index.

cmTrapDestinationTable
(alias: trdest)

Use this group to configure and display information about SNMP managers to which the trapped events will be sent. Trap destination records are in the form of a table, with information about each configured manager constituting a row in the table. You can configure up to 10 different trap destinations.

The Trap Destination table is one of five cmTrap groups that deal with events and alarms which are captured as SNMP traps. For more information about using and interpreting CE200 Events and Alarms, see the *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmtrapdestination [*ipaddr, port*] *object_1...object_3*

Example 1: To display trap destination information for a manager at IP address 194.168.101.2, port 162:

```
get trdest [194.168.101.2, 162]
Group: cmTrapDestinationTable
Instance: [194.168.101.2, 162]
IpAddr              =  194.168.101.2
Port                =  162
Community           =  trap
RowStatus           =  Active
OwnerString         =  r-phillips
```

Example 2: To configure a trap destination for a manager at IP address 202.001.222.255, port 162:

```
set trdest [202.001.222.255, 162] comm=trap rowstat=act
ownerstr=opsctrregion2
```

Note: The port number for SNMP trap managers is generally standardized as 162. In the case of two separate applications with a common IP address, however, separate port numbers must be assigned for each. Obtain the port numbers to use from the owner/operator.

Objects:

*IpAddr	The IP address of this manager.
*Port	The IP port number (162 is the standard port for SNMP traps) to which traps from this CE200 are sent.
Community	The SNMP Community string to use in traps sent to this manager. The standard community string for SNMP traps is *trap*.
RowStatus	Sets/displays the operational status of a row in the cmTrapDestination table: Active: Indicates the row exists and is active. IIn a *Set* command, creates a new row or activates a row that displays NotInService. Destroy (alias: delete): Deletes a row. NotInService: In a *Get* command, the row exists but is not being used by the CE200. In a Set command, creates a new row or deactivates a row that is currently Active.
OwnerString	Up to 40 characters of text to identify the manager.

Filters	Filter numbers established in *cmTrapTrapFilterTable* that are associated with this destination. For *Get* and *Getnext* commands, the response is a text string listing the filter number(s) and the numeric equivalent of the bit order for that Filter number in parenthesis. For *GetAll* commands, only the numeric values are returned. If no bits are set (0 length octet string), a null is returned, indicating that no filters are in effect for this destination, and all traps directed to this destination will be accepted. Values: filter1 (1) . . . filter32 (32) To assign multiple filters to this destination, enter the applicable filter numbers separated by a plus sign (e.g., Filters=4+6+14)

(*) indicates object used as an index.

cmTrapEventConfigTable
(alias: treconfig)

This group allows you to change the severity level of the various SNMP trap events from the factory default setting to suit the operational needs of your system, and to either generate, suppress or change the severity of the resulting alarms.

Should you wish to restore the factory default levels to all trap types, you can do so with a single command. See *cmTrapFilteredLogTable* on page 187.

Syntax: *command* cmtrapeventconfig [*type*] *object_1 object_2*

Example: To upgrade the severity of LoopStatusChange traps from *Information* (the default) to *Warning*:

```
set treconfig [loopstatuschange] severity=warning
Group: cmTrapEventConfigTable
Instance: [loopstatuschange]
Severity:        =  Warning
DefaultSeverity: =  Information
```

Objects:

*Type	Type of trap to be configured/displayed. **Event Type** — **Default Severity** ColdStart — Information BoardDown — Critical BoardUp — Information LinkDown — Minor LinkUp — Information DLCIStateDisabled — Minor DLCIStateEnabled — Information LoginFailed — Information LoginSucceeded — Information LoginSuspended — Warning LoginAllowed — Information LoginsSaturated — Warning LoginsAvailable — Information Logout — Information RadiusLocalFallback — Information BootFileFailed — Critical BootFileSucceeded — Information AttributeChanged — Information ConfigReadFailed — Major ConfigReadSucceeded — Information ConfigWriteFailed — Major ConfigWriteSucceeded — Information StateChange — Information Diagnostic — Information BoardRestart — Information FanFault — Critical FanfaultClear — Information PowerSupplyFault — Critical PowerSupplyFaultClear — Information TestStatus — Information TestError — Information LoopStatusChange — Information RisingThresholdAlert — Information FallingThresholdAlert — Information DS3LineStatusAlarm — Major DS3LineStatusClear — Information IDSLTimingLossAlarm — Major IDSLTimingLossClear — Information RedundancyChanged — Minor RoleChanged — Major RedundancyConflict — Minor MaintSucceed — Information MaintFailed — Minor RateFallBackAlarm — Minor RateFallbackClear — Information DS1LineStatusAlarm — Major DS1LineStatusClear — Information EndPointConflictAlarm — Minor PortMisprovisioned — Minor PortMisprovisionedClear — Information CpePlugAndPlayFailure — Minor CpePlugAndPlayClear — Information AtmVccDown — Minor AtmVccUp — Information SONETSectionStatusAlarm — Major SONETSectionStatusClear — Information SONETLineStatusAlarm — Major SONETLineStatusClear — Information SONETPathStatusAlarm — Major SONETPathStatusClear — Information WANLinkDown — Critical WANLinkUp — Information SHDSLConfigInitFailure — Information SHDSLLoopbackFailure — Information SHDSLPowerBackOff — Information SHDSLDeviceFault — Information SHDSLNoNeighbor — Information
Severity	The severity to be assigned to the specified SNMP trap type: *Information, Warning, Minor, Major, Critical, UseDefault* (the *None* option is not supported)
DefaultSeverity	Displays factory default setting for this trap type: *None, Information, Warning, Minor, Major, Critical*

(*) indicates object used as an index.

cmTrapEventFilterTable
(alias: trefilter)

Use this group to set up filters for various trapped events which then become collections of attributes such as event type, severity and time range. With these kinds of filters in place, you can then retrieve only those events which pass the filters you specify. By applying filters, you can then selectively list filtered collections of events using the *cmTrapFilteredLogTable*. Filters can be configured for any combination of the following attributes:

- Event Type
- Class ID (Board, Port, Link, etc.)
- Events that occur between specified start and stop times
- Events of a specific severity or range of severities (i.e. events that occur and that are less severe than specified will not be listed in the events for that filter.

The TrapEventFilter table is one of several cmTrap groups that deal with events and alarms which are captured as SNMP traps. For more information about using and interpreting CE200 Events and Alarms, see the *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmtrapeventfilter [*filternumber*] *object_1...object_3*

Example 1: To display the filter specifications for TrapEventFilter number 2:

```
get trefilter [2]
Group: cmTrapEventFilterTable
FilterNumber     =  2
Type             =  SONETLineStatusAlarm (55)
StartTime        =  2002/06/30-13:05:00
EndTime          =  2002/06/30-13:20:00
LowestSeverity   =  Minor
ClassId          =  0.0.0.0
OwnerString      =  Bosco
RowStatus        =  Active
```

Example 2: To configure a trapeventfilter for board restarts of an OC-3C/STM-1 Single Mode Module:

```
set trefilter [14] type=55 classid=1.3.0 rowstat=act
ownerstr=opsctrregion2
```

Objects:

*Number	The sequential number of this event filter (0-32).
Type	A bitmap of the event type that will pass the filter. For *Get* and *Getnext* commands, the response is a text string with the name of the Type and the numeric equivalent of the bit order for that type in parenthesis. For *GetAll* commands, only the numeric values are returned. If no bits are set (0 length octet string), a null is returned, indicating that events will not be filtered based on type. Values: ColdStart (1) BoardDown (2) BoardUp (3) LinkDown (4) LinkUp (5) DLCIStateDisabled (21) DLCIStateEnabled (22) LoginFailed (6) LoginSucceeded (7) LoginSuspended (10) LoginAllowed (11) LoginsSaturated (8) LoginsAvailable (9) Logout (14) RadiusLocalFallback (62) BootFileFailed (12) BootFileSucceeded (13) AttributeChanged (15) ConfigReadFailed (16) ConfigReadSucceeded (17) ConfigWriteFailed (18) ConfigWriteSucceede (19) StateChange (20) Diagnostic (23) BoardRestart (24) FanFault (25) FanFaultClear (26) PowerSupplyFault (27) PowerSupplyFaultCle (28) TestStatus (29) TestError (30) LoopStatusChange (31) RisingThresholdAler (32) FallingThresholdAle (42) DS3LineStatusAlarm (33) DS3LineStatusClear (34) IDSLTimingLossAlarm (35) IDSLTimingLossClear (36) RedundancyChanged (37) RoleChanged (38) RedundancyConflict (39) MaintSucceed (40) MaintFailed (41) RateFallbackAlarm (43) RateFallbackClear (44) DS1LineStatusAlarm (45) DS1LineStatusClear (46) EndPointConflictAla (47) PortMisprovisioned (48) CpePlugAndPlayFailu (49) CpePlugAndPlayClear (50) AtmVccDown (51) AtmVccUp (52) SONETSectionStatusA (53) SONETSectionStatusC (54) SONETLineStatusAlar (55) SONETLineStatusClea (56) SONETPathStatusAlar (57) SONETPathStatusClea (58) PortMisprovisionedC (59) WANLinkDown (63) WANLinkUp (64) SHDSLLoopbackFailur (89) SHDSLPowerBackOff (90) SHDSLDeviceFault (91) SHDSLNoNeighbor (92), SHDSLLocalPowerLoss (93)
StartTime	This field specifies the earliest time stamp that can appear on events to be displayed. Events occurring before the specified time will not be displayed. If no value is entered, events will not be filtered by start time. Enter the time in CalendarTime format: *yyyy/mm/dd-hh:mm:ss.*
EndTime	This field specifies the latest time stamp that can appear on events to be displayed. Events occurring after the specified time will not be displayed. If no value is entered, events will not be filtered by end time. Enter the time in CalendarTime format: *yyyy/mm/dd-hh:mm:ss*

LowestSeverity	Specifies the range of event severity levels to be filtered. Severities milder than the specified lowest will be filtered. Values: *Information* (default)*, Warning, Minor, Major, Critical*. Specifying Information means that events will not be filtered based on severity.
ClassID	Specifies the class ID (PII) which must be named in any event which passes the filter. An entry of 0.0.0.0 indicates that events will not be filtered based on ClassID.
OwnerString	Up to 40 characters of text to identify the owner of the filter, and/or the person defining the filter.
RowStatus	Used to activate or delete a trap event filter. Options are: *Active* (creates a new row in the table) *CreateAndGo* (also creates a new row, but only if an identical entry did no already exist, *destroy* (deletes the row/filter)

(*) indicates object used as an index.

cmTrapEventTable
(alias: trevent)

This group allows you to display information about events from the Event log. You can view events individually by specifying the SequenceNumber, or you can use the Getall command to review all of the events in the log. Note, however, that there can be up to 1,000 events in the Event log, including alarms or simple events such as an operator login.

The cmTrapEvent table is one of five cmTrap groups that deal with events and alarms, all of which are captured as SNMP traps. Its structure is identical to the cmTrapAlarm group, except that all of the objects are read-only.

You can also use the Elog command to monitor events in real time as described in "Event Log" in Chapter 9 of the *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmtrapevent [*seqnum*]

Example: To display information for event #2 from the log:

```
get trevent [2]
Group: cmTrapEventTable
Instance: [2]
SeqNum              =  2
Type                =  ConfigReadSucceeded
ObjectClass         =  Board
ClassId             =  1.2.0.0
ProbableCause       =  0
Severity            =  Information
TimeTag             =  2000/06/05-15:13:28
Text                =  config.tgz
```

Objects:

*SeqNum	The sequence number of this event (an integer that is automatically augmented as new events occur).

Type	The type of event:	
	ColdStart	BoardDown
	BoardUp	LinkDown
	LinkUp	DLCIStateDisabled
	DLCIStateEnabled	LoginFailed
	LoginSucceeded	LoginSuspended
	LoginAllowed	LoginsSaturated
	LoginsAvailable	Logout
	RadiusLocalFallback	
	BootFileFailed	BootFileSucceeded
	AttributeChanged	ConfigReadFailed
	ConfigReadSucceeded	ConfigWriteFailed
	ConfigWriteSucceeded	StateChange
	Diagnostic	BoardRestart
	FanFault	FanfaultClear
	PowerSupplyFault	PowerSupplyFaultClear
	TestStatus	TestError
	LoopStatusChange	RisingThresholdAlert
	FallingThresholdAlert	DS3LineStatusAlarm
	DS3LineStatusClear	IDSLTimingLossAlarm
	IDSLTimingLossClear	RedundancyChanged
	RoleChanged	RedundancyConflict
	MaintSucceed	MaintFailed
	RateFallBackAlarm	RateFallbackClear
	DS1LineStatusAlarm	DS1LineStatusClear
	EndPointConflictAlarm	
	PortMisprovisioned	PortMisprovisionedClear
	CpePlugAndPlayFailure	CpePlugAndPlayClear
	AtmVccDown	AtmVccUp
	SONETSectionStatusAlarm	SONETSectionStatusClear
	SONETLineStatusAlarm	SONETLineStatusClear
	SONETPathStatusAlarm	SONETPathStatusClear
	WANLinkDown	WANLinkUp
	SHDSLConfigInitFailure	SHDSLLoopbackFailure
	SHDSLPowerBackOff	SHDSLDeviceFault
	SHDSLNoNeighbor	

ObjectClass	The object class of the resource reporting the event (not all listed classes are supported): System class: System Operator Shelf class: Shelf CE200Shelf CE150Shelf Board class: Board SystemControlModule SDSL10xModule-1 SDSL10xModule SDSL30xModule SDSL30xModule-24 IDSLModule-24 BufferControlModule BufferControlModule2 V.35-WAN DS3FR-WAN DS3ATM-WAN QuadT1-WAN G.liteModule-24 DualT1-WAN SDSLModule-8 T1Module-12 SystemControlModule3 ADSLMultiModeModule-24A ADSLMultiModeModule-24B SONETSingleMode-WAN SONETMultiMode-WAN. GSHDSLATMModule-24 GSHDSLPacketModule-24 Port class: Port SDSLPort EthernetPort V.35Port RS232Port DS3FRPort IDSLPort DS3ATMPort T1Port-WAN T1Port G.litePort G.dmtPort T1Port-LC SONETPort G.shdslPort IMAPort Link class: Link LCPortLink EthernetLink FrameRelayLink FrameRelayPVC ATMLink InterShelfTrunk CPE class: CR CR201-10x CR201-30x CR201-SDSL CR201IDSL CPE-SDSL CPE-IDSL Netopia-SDSL CPE-T1 CPE-GSHDSL Support class: Support PowerModule FanModule
ClassId	The 32-bit Permanent Interface Identifier (PII) associated with the event, either a VC, physical interface, module or shelf as appropriate.
ProbableCause	A 32-bit event-specific code (not supported).
Severity	The severity level of the event: None Information MinorAlarm MajorAlarm CriticalAlarm Warning
TimeTag	System calendar time when the event occurred.
Text	Additional information that describes the event.

(*) indicates object used as an index.

cmTrapFilteredLogTable
(alias: trfiltered)

This read-only group allows you to display lists of events that match the parameters of event filters configured on this system (see *cmTrapEventFilterTable* on page 182). The objects are the same as for the cmTrapAlarmTable, with the addition of the FilterNumber object. The table is double indexed to both the FilterNumber and Sequence Number (SeqNum) object.

For more information about using and interpreting CE200 events and alarms, see your *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmtrapfilteredlog [filternumber, seqnum]

Example: To query the most recent event:

```
find trfiltered filternumber=1
Group: cmTrapfilteredlog
SeqNum           =  1255
Type             =  0
ObjectClass      =  0
ClassId          =  0
ProbableCause    =  0
Severity         =  4
TimeTag          =  None
Text             =  SeverityDefault
```

Objects:

*FilterNumber	The unique numeric identifier of a row in the cmTrapEventFilterTable which specifies the filter you wish to apply to this event log search.
*SeqNum	The trap/event sequence number of the reported event
Type	The value of cmTrapTypeTableType for the specified event (see the preceding MIB group for a list of event types).
ObjectClass	The Object Class (Board, Port, etc.) for the specified event.
ClassId	The ClassId (PII) associated with the specified event
ProbableCause	Probable cause description, if available, for the event
Severity	Severity level (Critical (6), Major (5), Minor (4), Warning(3), Info (2)) of the event
TimeTag	Calendar time at which the event occurred.
Text	Text string associated with the event, if any

(*) indicates object used as an index.

cmTrapSummary
(alias: trsum)

This group allows you to display the identifying sequence number of the most recent event (events are numbered sequentially as they occur), and list the number of outstanding alarms and warnings by severity (alarms that have not been cleared). You can also use the cmTrapSummary group's *Command* object to reset the severity levels of all traps to their factory defaults.

The cmTrapSummary is one of several cmTrap groups that deal with events and alarms, all of which are captured as SNMP traps.

For more information about using and interpreting CE200 events and alarms, see the *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmtrapsummary

Example 1: To query the most recent alarm event:

```
get trsum
Group: cmTrapSummary
EventSeqNum           =  1255
CriticalAlarmCount    =  0
MajorAlarmCount       =  0
MinorAlarmCount       =  0
WarningCount          =  0
NextTrapFilterIndex   =  14
Command               =  None
```

Example 2: To reset all trap alarm severities to their factory defaults:

```
set trsum command=severitydefault
Group: cmTrapSummary
EventSeqNum           =  1332
CriticalAlarmCount    =  0
MajorAlarmCount       =  2
MinorAlarmCount       =  2
WarningCount          =  0
NextTrapFilterIndex   =  37
Command               =  SeverityDefault
```

Objects:

EventSeqNum	The highest current event sequence number.
CriticalAlarmCount	The number of current uncleared critical alarms.
MajorAlarmCount	The number of current uncleared major alarms.
MinorAlarmCount	The number of current uncleared minor alarms.
WarningCount	The number of current uncleared warnings.
NextTrapFilterIndex	Next available unused trap filter index: 0 - 32; a value of 0 means that all trap filters are in use and none is available
Command	Provides a command for simultaneous reset of all trap severities to their factory default settings: *None, SeverityDefault*

cmTrapTrapFilterTable
(alias: trtfilter)

Use this group to configure trap filters, which specify the parameters for traps that will be blocked from delivery to destinations that use the filters. The Index object of *cmTrapTrapFilter* becomes the filternumber in the *cmTrapDestinationTable* where filters are designated for each of the various trap receivers.

For more information about using and interpreting CE200 events and alarms, see your *CopperEdge 200 Installation and Operating Guide*.

Syntax: *command* cmtraptrapfilter [Index No.] *Object1, ... Object n*

Example 1: To create a filter to block Link Down and State Change traps:

```
set trtfilter [9] type=4+20
Set successful
```

Example 2: To list the specifications of a specific filter:

```
get trtfilter [7]
Group: cmTrapTrapFilter
Index            =  7
Type             =  " "
TypeCompare      =  EQ
MinClassID       =  0.0.0.0
MaxClassID       =  0.0.0.0
ObjectClass      =  brdSDSL30xModule-24 + brdIDSLModule-24
ObjectClassCompar=  EQ
Severity         =  none
SeverityCompare  =  EQ
OwnerString      =  " "
InUseFlag        =  True
RowStatus        =  Active
```

Objects:

*Index	The index of the filter. When creating a new filter, find the value of the next available filter by getting *cmTrapSummaryNextTrapFilterIndex*. Range: 1 - 32

Type	A bitmap of the event types included in the filter. For *Get* and *Getnext* commands, the response is a text string with the name of the Type and the numeric equivalent of the bit order for that type in parenthesis. For *GetAll* commands, only the numeric values are returned. If no bits are set (0 length octet string), a null is returned, indicating that events will not be filtered based on type. For Set commands, use the numeric value; to include multiple types, separate the numerics with a + as shown in the example above. Values: ColdStart (1) BoardDown (2) BoardUp (3) LinkDown (4) LinkUp (5) DLCIStateDisabled (21) DLCIStateEnabled (22) LoginFailed (6) LoginSucceeded (7) LoginSuspended (10) LoginAllowed (11) LoginsSaturated (8) LoginsAvailable (9) Logout (14) RadiusLocalFallback (62) BootFileFailed (12) BootFileSucceeded (13) AttributeChanged (15) ConfigReadFailed (16) ConfigReadSucceeded (17) ConfigWriteFailed (18) ConfigWriteSucceede (19) StateChange (20) Diagnostic (23) BoardRestart (24) FanFault (25) FanFaultClear (26) PowerSupplyFault (27) PowerSupplyFaultCle (28) TestStatus (29) TestError (30) LoopStatusChange (31) RisingThresholdAler (32) FallingThresholdAle (42) DS3LineStatusAlarm (33) DS3LineStatusClear (34) IDSLTimingLossAlarm (35) IDSLTimingLossClear (36) RedundancyChanged (37) RoleChanged (38) RedundancyConflict (39) MaintSucceed (40) MaintFailed (41) RateFallbackAlarm (43) RateFallbackClear (44) DS1LineStatusAlarm (45) DS1LineStatusClear (46) EndPointConflictAla (47) PortMisprovisioned (48) CpePlugAndPlayFailu (49) CpePlugAndPlayClear (50) AtmVccDown (51) AtmVccUp (52) SONETSectionStatusA (53) SONETSectionStatusC (54) SONETLineStatusAlar (55) SONETLineStatusClea (56) SONETPathStatusAlar (57) SONETPathStatusClea (58) PortMisprovisionedC (59) WANLinkDown (63) WANLinkUp (64) SHDSLConfigInitFailure (69) SHDSLLoopbackFailur (89) SHDSLPowerBackOff (90) SHDSLDeviceFault (91) SHDSLNoNeighbor (92)
TypeCompare	The comparison operator for Type: EQ (Default) or NE. If set to EQ, the filter blocks the traps you specified in the Type object from being sent to the Trap Destinations that use the filter; if set to NE, the filter blocks *all* traps *except* those you specified in the Type object. In effect, this object sets whether the filter will be positive or negative in its effect.
MinClassID	Used to configure a filter that will apply to (or exclude) a range of interfaces in PII format. This object specifies the lowest-numbered ClassId in the range, and must be equal to or lower than the value specified in MaxClassId. If both MinClassId and MaxClassId are set to their default values (0.0.0.0), events will not be filtered based on ClassId.

MaxClassID	Used to configure a filter that will apply to (or exclude) a range of interfaces in PII format. This object specifies the highest-numbered ClassId in the range, and must be equal to or greater than the value specified in MinClassId. If both MinClassId and Max-ClassId are set to their default values (0.0.0.0), events will not be filtered based on ClassId.
ObjectClass	The Object Class (Board, Port, etc.) for the specified event. A bitmap of the object classes included in the filter. For *Get* and *Getnext* commands, the response is a text string with the name and numeric equivalent of the bit order for that ObjectCLass in parenthesis. For *GetAll* commands, only the numeric values are returned. If no bits are set (0 length octet string), a null is returned, indicating that events will not be filtered based on ObjectClass. For Set commands, use the numeric value; to include multiple object classes, separate the numerics with a + as shown in the example above. Values: Shelf (1) CE200Shelf (2) CE150Shelf (3) Board (4) SystemControlModule (5) SDSL10xModule(1) (6) SDSL10xModule (7) SDSL30xModule (10) SDSL30xModule-24 (12) IDSLModule-24 (13) BufferControlModule (8) BufferControlModule2 (22) V.35-WAN (9) DS3FR-WAN (11) DS3ATM-WAN (14) QuadT1-WAN (15) G.liteModule-24 (16) SystemControlModule (17) ADSLMultiModeModule (18) ADSLMultiModeModule (21) DualT1-WAN (19) SDSLModule-8 (20) T1Module-12 (23) SONETSingleMode-WAN (57) SONETMultiMode-WAN (58) G.shdslModule-24 (55) G.shdslATMModule-24 (71) IMAWANModule (72) Port (24) SDSLPort (25) EthernetPort (26) V.35Port (27) RS232Port (28) DS3FRPort (29) IDSLPort (30) DS3ATMPort (31) T1Port-WAN (32) G.litePort (33) G.dmtPort (34) T1Port-LC (35) SONETPort (59) G.shdslPort (56) IMAPort (73) Link (36) LCPortLink (37) EthernetLink (38) FrameRelayLink (39) FrameRelayPVC (40) ATMLink (41) CR (42) CR201-10x (43) CR201-30x (44) CR201-SDSL (45) CR201-IDSL (47) CPE-SDSL (48) CPE-IDSL (49) Netopia-SDSL (46) CPE-T1 (50) CPE-GSHDSL (60) Operator (51) Support (52) PowerModule (53) FanModule (54)
ObjectClassCompare	The comparison operator for ObjectClass: EQ, NE
Severity	Severity level of traps to be filtered: Critical (6), Major (5), Minor (4), Warning(3), Info (2)

SeveritryCompare	The comparison operator for the Severity: eq - block traps that match filter severity ne - block traps that don't match filter severity gt - block traps with severity > filter severity lt - block traps with severity < filter severity ge - block traps with severity ≥ filter severity le - block traps with severity ≤ filter severity
OwnerString	Text string (up to 32 characters) to describe the filter or list the name of the person who created it.
InUseFlag	If this filter is specified for use by any row in the *cmTrapDestinationTable*, the value of the InUseFlag is True; otherwise False.
RowStatus	Used to activate or delete a trap filter. Options are: *Active* (creates a new row in the table) *CreateAndGo* (also creates a new row, but only if an identical entry did not already exist, *destroy* (deletes the row/filter)

(*) indicates object used as an index.

cmTrapTypeTable
(alias: trtype)

This group allows you to use the same information as the cmTrapEvent group to provide a display of the last 1,000 trapped events. In cmTrapType, however, the results are indexed first by internal event type and then by sequence number.

The cmTrapType table is one of five cmTrap groups that deal with events and alarms, all of which are captured as SNMP traps.

Syntax: *command* cmtraptype [*type, seqnum*]

Example 1: To display all information in the table for all linkdown events starting with #50:

```
geta trtype [linkdown, 50]
Type            SeqNum          ObjectClass          ClassId
ProbableCause   Severity        TimeTag              Text

Instance: [LinkDown, 94]
LinkDown        94              DSLLink              1.10.4.0
0             MinorAlarm    2000/08/04-15:28:41  1.10.4.0, "", 1.10.0

Instance: [LinkDown, 120]
LinkDown        120             DSLLink              1.11.3.0
0             MinorAlarm    2000/08/04-15:29:53  1.11.3.0, "", 1.11.0

Instance: [LinkDown, 227]
LinkDown        227             DSLLink              1.6.2.0
0             MinorAlarm    2000/08/08-10:33:54  1.6.2.0, "", 1.6.2.0

Instance: [LinkDown, 240]
LinkDown        240             DSLLink              1.11.3.0
0             MinorAlarm     2000/08/08-14:50:55  1.11.3.0, "",
1.11.3.0
-- ESC to quit, any other key to continue --
```

This Getall command displays all linkdown events in the table, if they exist, starting from the sequence number.

Example 2: To display information for a specific event, such as linkdown 94:

```
get trtype [linkdown, 94]
Group: cmTrapTypeTable
Instance: [LinkDown, 94]
Type                =  LinkDown
SeqNum              =  94
ObjectClass         =  DSLLink
ClassId             =  1.10.4.0
ProbableCause       =  0
Severity            =  MinorAlarm
TimeTag             =  2000/08/04-15:28:41
Text                =  1.10.4.0, "", 1.10.4.0, ""
```

Objects:

*Type	The type of event: ColdStart, BoardDown, BoardUp, LinkDown, LinkUp, DLCIStateDisabled, DLCIStateEnabled, LoginFailed, LoginSucceeded, LoginSuspended, LoginAllowed, LoginsSaturated, LoginsAvailable, Logout, BootFileFailed, BootFileSucceeded, AttributeChanged, ConfigReadFailed, ConfigReadSucceeded, ConfigWriteFailed, ConfigWriteSucceeded, StateChange, Diagnostic, BoardRestart, FanFault, FanfaultClear, PowerSupplyFault, PowerSupplyFaultClear, TestStatus, TestError, LoopStatusChange, RisingThresholdAlert, FallingThresholdAlert, DS3LineStatusAlarm, DS3LineStatusClear, IDSLTimingLossAlarm, IDSLTimingLossClear, RedundancyChanged, RoleChanged, RedundancyConflict, MaintSucceed, MaintFailed, RateFallBackAlarm, RateFallbackClear, DS1LineStatusAlarm, DS1LineStatusClear, EndPointConflictAlarm, PortMisprovisioned, CpePlugAndPlayFailure, CpePlugAndPlayClear, AtmVccDown, AtmVccUp, SONETSectionStatusAlarm, SONETSectionStatusClear, SONETLineStatusAlarm, SONETLineStatusClear, SONETPathStatusAlarm, SONETPathStatusClear, WANLinkDown, WANLinkUp, PortMisprovisionedClear, RadiusLocalFallback, SHDSLConfigInitFailure (69), SHDSLLoopbackFailur (89), SHDSLPowerBackOff (90), SHDSLDeviceFault (91), SHDSLNoNeighbor (92)
*SeqNum	The sequence number of this event (an integer that is automatically augmented as new events occur).

ObjectClass	The object class of the resource reporting the event: System class: System, Operator Shelf class: Shelf, CE200Shelf, CE150Shelf Board class: Board, SystemControlModule, SDSL10xModule-1, SDSL10xModule, SDSL30xModule, SDSL30xModule-24, IDSLModule-24, BufferControlModule, BufferControlModule2, V.35-WAN, DS3FR-WAN, DS3ATM-WAN, QuadT1-WAN, G.liteModule-24, DualT1-WAN, SDSLModule-8, T1Module-12, SystemControlModule3, ADSLMultiModeModule-24A, ADSLMultiModeModule-24B, SONETSingleMode-WAN, SONETMultiMode-WAN, GSHDSLATMModule-24, GSHDSLPacketModule-24, IMAModule Port class: Port, SDSLPort, EthernetPort, V.35Port, RS232Port, DS3FRPort, IDSLPort, DS3ATMPort, T1Port-WAN, T1Port
ObjectClass (continued)	Port class: (continued) G.litePort, G.dmtPort, T1Port-LC, SONETPort, G.shdslPort, IMAPort Link class: Link, LCPortLink, EthernetLink, FrameRelayLink, FrameRelayPVC, ATMLink, InterShelfTrunk CPE class: CR, CR201-10x, CR201-30x, CR201-SDSL, CR201IDSL, CPE-SDSL, CPE-IDSL, Netopia-SDSL, CPE-T1, CPE-GSHDSL Support class: Support, PowerModule, FanModule
ClassId	The 32-bit Permanent Interface Identifier (PII) of the resource in the class.
ProbableCause	A 32-bit, event-specific reason code (not supported).
Severity	The severity level of the event: None, Information, MinorAlarm, MajorAlarm, CriticalAlarm, Warning
TimeTag	The system calendar time when the event occurred.
Text	Additional information that describes the event.

(*) indicates object used as an index.

cmV35Table

This group displays information about a specified port on a V.35-WAN module.

Syntax: *command* cmv35 *[pii]*

Example: To query port 1 of the V.35-WAN module in slot 4:

```
get cmv3 [1.4.1]
Group: cmV35Table
Instance: [1.4.1.0]
Index                =  1.4.1.0
IfIndex              =  1.4.1.0
DTR                  =  On
RTS                  =  On
DSR                  =  On
DCD                  =  On
CTS                  =  On
Sync                 =  On
TxClockSource        =  External
DTEDCE               =  DTE
CRC                  =  CRC-CCITT
ClockRate            =  1544000
AdminState           =  Enabled
OperState            =  Enabled
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the port.
ifIndex	The ifIndex value assigned by the system during powerup; used as the index to the ifTable group.
DTR	The current state of Data Terminal Ready (output signal): On or Off.
RTS	The current state of Request to Send (output signal): On or Off.
DSR	The current state of Data Set Ready (input signal): On or Off.
DCD	The current state of Carrier Detect (input signal): On or Off.
CTS	The current state of Clear to Send (input signal): On or Off.
Sync	The status of HDLC flags on the serial line: On (flags are detected) or Off.
TxClockSource	Indicates the V.35 module clock source: External or Internal.
DTEDCE	Indicates how this port is operating: Dte or Dce.
CRC	The CRC format used on packets: always CRC-CCITT.
ClockRate	The clock rate of the port in bps: always 1544000. This value is used in bandwidth control calculations.
AdminState	Current administrative state of the port: Enabled or Disabled. This port must be Enabled to be operable.
OperState	Current operational state of the port: Enabled or Disabled. For a more on OperState as it relates to this group, see *Operating Status* on page 20.

(*) indicates object used as an index.

cmVbridgeTable

This group displays loading and performance information about a specified VWAN Virtual Bridge. Configuring the Option object as IP-Special prevents passage of any packets except IP packets and Ethernet ARP packets over the WAN link to an upstream service router, effectively blocking any broadcast packets such as IPX.

Syntax: *command* cmvbridge [*pii*] *object_1*

The *pii* identifies the frame relay DLCI or ATM Virtual Link of the WAN interface associated with this Virtual Bridge. As noted elsewhere, if you are referring to a T1/E1 IMA group, use the "virtual" PII reference for the IMA group. Do *not* use individual link PIIs to configure an IMA group.

Example 1: To query port 1, slot 3 for information about the Virtual Bridge associated with VC 17:

```
get cmvb [1.3.1.17]
Group: cmVbridgeTable
Instance: [1.3.1.17]
PII                  = 1.3.1.17
NMAC                 =  0
NMulticast           =  0
NUnicast             =  0
NDSL                 =  0
NUnknownDest         =  0
NOverflow            =  0
Option               =  None
```

Example 2: To set the Virtual Bridge to pass only IP and ARP packets:

```
set cmvb [1.3.1.17] option=ip-special
```

Objects:

*PII	The Permanent Interface Identifier of the WAN VC on which this bridge is aggregated.
NMAC	The number of MAC entries in this bridge.
NMulticast alias: nmul	The number of multicast frames received over the associated WAN VC or Ethernet.
NUnicast alias: nuni	The number of unicast frames received over the associated WAN VC or Ethernet.
NDsl alias: ndsl	The number of frames received from DSL interfaces in this bridge group.
NUnknownDest alias: nunk	The number of frames received with unknown destination (and thus discarded).
NOverflow alias: nover	The number of MAC entries unlearned due to a "table full" condition.
Option	The optional bridge function: None—Passes all packet formats. IP-special—Passes only IP and Ethernet ARP packets.

(*) indicates object used as an index.

Notes:

1. The Option object is configurable in both VWAN modes (point-to-point and Bridged). Setting Option to IP-Special provides an easy means of reducing unnecessary wide-area traffic over a virtual bridge, and preventing broadcast

messages meant for a subscriber LAN from being distributed to another subscriber LAN (for example, on another CE200 served by a common router).

2. In IP-Special mode, the CE200 can only learn the service router's MAC address from a unicast frame and will not learn it from a multicast frame. IP-Special thus insures that the CE200 will not be supplied with an incorrect MAC address, which might otherwise occur, if for example, the service router cannot be configured to prevent broadcasts arriving on one PVC from being forwarded to any other PVC.
3. The IP-Special mode is not intended as a security feature, as ARP broadcasts can still be passed from one subscriber LAN to another if the router is incorrectly configured or incapable of broadcast filtering.

cmVpnGroupTable

Use this table to configure CopperVPN groups, including upstream WAN interface membership and group attributes. Each row in the table represents a separate CopperVPN group.

A CopperVPN Group is automatically created when a cmIface entry is created for a WAN interface and assigned a CopperVPN netmodel. A subscriber member is created and added to the group when configured with the GroupPII as its DestPII. A group is deleted when the cmIface entry for the GroupPII is deleted. A GroupPII can only be deleted from the cmIfaceTable after all of its subscriber and uplink members are removed from the group. Note that deleting a group also removes its *cmVpnSubnetTable* and *cmVpnRouteTable* entries. Enabling the peer-to-peer function for a group allows all hosts that are members of the group, *and* that are connected to the same CopperEdge, to communicate with each other through the CopperEdge.

Syntax: *command* cmvpngroup *[vpngrouppii] object_1*

Example:

```
get cmvpng [1.3.1.30]
Group: cmVpnGroupTable
Instance: [1.3.1.30]
VPNGroupPii      =  1.3.1.30
Name             =  PTCC
DefaultUplink    =  1.3.1.30
WANUplink1       =  1.3.1.30
WANUplink2       =  0.0.0.0
WANUplink3       =  0.0.0.0
WANUplink4       =  0.0.0.0
WANUplink5       =  0.0.0.0
PeerToPeer       =  enabled
IpValidation     =  enabled
GatewayByInARP   =  disabled
DHCPSnooping     =  disabled
WanArpPeerHosts  =  enabled
DefaultTTL       =  30
Command          =  none
PolicyBlocked    =  9
UnresolvedAddr   =  0
InvalidPackets   =  0
InternalLimits   =  0
DIPFailures      =  0
```

Note: This table can only be configured after the corresponding *cmIface* entry has been created for the VPN Group PII.

Object:

*VPNGroupPii	Permanent Interface Identifier of the CopperVPN group, automatically created by cmIface when a WAN VC is configured with netmodel=CopperVPN.
Name	Operator assigned name for this group.

DefaultUplink	Permanent Interface identifier (PII) of the upstream WAN VC interface used in the CopperVPN group.
WANUplink1	Displays the WAN interface selected when the CVPN group was created. Same as *VPNGroupPii*
WANUplink2	The second upstream WAN VC used in a CVPN group beyond the WAN VC originally configured to create the CVPN group. cmIface entry for the Uplink2 PII with NetModel of CVPN must exist before it can be added as an Uplink2 PII.
WANUplink3	Specifies/lists the third upstream WAN VC used in a CopperVPN group beyond the WAN VC originally configured to create the CopperVPN group. cmIface entry for the Uplink3 PII with NetModel of CVPN must exist before it can be added as an Uplink3 PII.
WANUplink4	Specifies/lists the fourth upstream WAN VC used in a CopperVPN group beyond the WAN VC originally configured to create the CopperVPN group. cmIface entry for the Uplink4 PII with NetModel of CVPN must exist before it can be added as an Uplink4 PII.
WANUplink5	Specifies/lists the fifth upstream WAN VC used in a CopperVPN group beyond the WAN VC originally configured to create the CopperVPN group. cmIface entry for the Uplink5 PII with NetModel of CVPN must exist before it can be added as an Uplink5 PII.
PeerToPeer	Allow or forbid DSL-to-DSL peer communication in a CopperVPN group: enabled (default), or disabled. When disabled, upstream packets are always forwarded to the upstream WAN VC for the group.
IPValidation	Specifies whether to perform source IP address validation for incoming traffic and the restricted MAC address resolution for outdoing traffic on all subscriber interfaces in the CopperVPN group: enabled (default), or disabled.
GatewayByInARP	Specifies whether to allow the DSLAM to learn the IP address of the upstream WAN interface by Inverse ARP and store it as the IP gateway: disabled (default) or enabled.
DHCPSnooping	Specifies whether to use DHCP snooping to learn the assignment of host IP address and gateway/subnet information. disable (default) or enable
WanArpPeerHosts	Specifies whether to ARP for peer hosts over the MAC-encapsulated WAN interfaces. (Peer hosts can be on other DSLAMs that belong to the same bridge group). If disabled, all packets are forwarded to the default gateway. Enable (default), or Disable
DefaultTTL	Numeric specifies Time To Live, in minutes, to be used by the CVPN group in building its dynamic route entry in *cmVPNRouteTable*. Value must be at least 1. Default: 30

Command	A command executed on the entire CVPN group: *None* (default) *FlushArpCache* clears the MacAddr field in cmVpnRouteTable for all entries except those with Type=DynamicDHCP *DeleteDynamicArp* clears all cmVPNRouteTable entries learned via ARP *DeleteDynamicDhcp* clears all cmVPNRouteTable entries learned via DHCP snooping. *DeleteStatic* clears all operator-originated entries in the cmVPNRouteTable *DeleteAll* (clears all entries in cmVPNRouteTable)
PolicyBlocked	Displays the number of packets discarded due to policy restrictions such as filters or IP Validation
UnresolvedAddress	Displays the number of packets discarded due to unknown or unresolvable address
InvalidPackets	Displays the number of packets discarded due to packet-header errors
InternalLimits	Displays the number of packets discarded due to internal limits of the system
DIPFailures	Displays the number of times the Dynamic Information Preservation feature has failed. DIP failures include flash-save failures and redundancy-sync failures.

(*) indicates object used as an index.

cmVpnRouteTable

This table displays the specifications used for IP-to-MAC address translation and IP forwarding for CopperVPN groups. It is a double-indexed table, keyed to both the VPNGroup PII and the Destination IP address.

Syntax: *command* cmvpnroute *[vpngrouppii, destination] object_1*

Example:

```
get cmvpnr [1.3.1.30, 10.26.2.120]
Group: cmVpnRouteTable
Instance: [1.3.1.30, 10.26.2.120]
VPNGroupPii    =  1.3.1.30
Destination    =  10.26.2.120
RowStatus      =  active
Interface      =  1.3.1.30
MacAddr        =  0.72.26.18.36.ba
Type           =  Static
TimeToLive     =  30
Command        =  none
DefaultGateway =  10.24.8.112
SubnetMask     =  255.255.255.128
```

Note: This table can only be configured after the corresponding *cmIface* entry has been created for the VPNGroup PII, and the appropriate subnet table has been created.

Objects:

*VPNGroupPii	The Permanent Interface Identifier of the CVPN group, automatically created when the group is configured. This PII is identical to that of the first WAN VC configured in the CopperVPN group. (the net-model for the WAN VC is set to CopperVPN).
*Destination	An IP address entry in the atTable (address translation/forwarding table) of the CVPN group.
RowStatus	Used to create or delete entries in this table: active (default) or destroy.
Interface	The PII of the interface to which the Destination IP address belongs.
MacAddr	Displays MAC address corresponding to the destination IP address. Applies only to Ethernet encapsulation types.
Type	Lists how the gateway IP address is obtained: Static (1) -- Operator input. This is the only type supported in the current release. DynamicARP (2) -- Learned through InARP DynamicDhcp (3) -- Learned via DHCP snooping
TimeToLive	Remaining time (in minutes) that the entry will stay in the CopperVPN group. Does not apply if *Type* is *Static*.
Command	*None* (default) *FlushArpCache* clears the MacAddr Field for this route
DefaultGateway	IP address of the default gateway for the route entry
SubnetMask	Subnet mask for the route entry

(*) indicates object used as an index.

cmVpnSubnetTable

This table specifies a list of subnets and subnet-related information for a CVPN group. Entries can be configured statically, or automatically populated if you have enabled *cmVPNGroupGatewayByInArp*. For dynamically created entries, the GatewayIpAddress and NetMask objects cannot be changed.

To delete an entry in this table, use rowstatus=destroy. Deleting an entry from this table will not delete the route table entries. Entries in this table learned using DHCP are automatically deleted when the lease time expires.

Syntax: *command* cmvpnSubnetTable *[vpngrouppii, number] object_1*

Example:

```
get cmvpnsub [1.3.1.30, 1]
Group: cmVpnSubnetTable
Instance: [1.3.1.30, 1]
VPNGroupPii        =  1.3.1.30
Number             =  1
RowStatus          =  active
GatewayIpAddress   =  10.12.122.0
NetMask            =  255.255.255.128
DiagnosticIpAddr   =  10.14.22.2
Type               =  static
```

Notes:

1. CPE hosts connected to a DSL port must use one of the specified gateway IP addresses. If *cmVPNGroup GatewayByInARP* is enabled, the Inverse ARP function will be used to learn the upstream router's IP address, and will use it as the gateway IP address. Subnets will be populated automatically with net-mask 0.0.0.0. In this case the functionality is equivalent to the old CopperVPN Auto forwarding mode. If IPValidation is enabled, the IP address of any CPE host has to be in the range of one of subnets constrained by GatewayIpAddr and NetMask. The CE200 enforces this restriction during the configuration.
2. The *DiagnosticIpAddr* object is intended for use in trouble isolation only. Take care not to assign this address to any operational interface.
3. This table can only be configured after the corresponding cmIface entry has been created for the VPN Group PII.

Objects:

*VPNGroupPii	The Permanent Interface Identifier of the CopperVPN group, automatically created when the group is configured. This PII is identical to that of the first WAN VC configured in the CopperVPN group. (the netmodel for the WAN VC is set to CopperVPN).
*Number	An integer in the range 1 - 8 designating the gate-way/subnet entry
RowStatus	Used to create or delete entries in this table: active (default) or destroy.
GatewayIpAddress	The IP address of the upstream gateway serving the CVPN group. This value cannot be changed if the *Type* is *dynamicArp* or *dynamicDhcp*.
NetMask	The netmask associated with the IP Gateway for this CVPN group. Together with the GatewayIpAddress, it specifies the subnet covered by this CVPN group. Must be a legal netmask value, and cannot be changed if the *Type* is *dynamicArp* or *dynamicDhcp*.

DiagnosticIpAddr	The IP address used by the group to support Ping and Traceroute diagnostic functions. Receives and replies to ICMP messages only (i.e., you cannot Telnet or FTP to it).
Type	Configure/show how the gateway IP address is obtained: Static (1) -- Operator input. DynamicARP (2) -- Learned through InARP DynamicDhcp (3) -- Learned via DHCP snooping

(*) indicates object used as an index.

cmWanMemberTable

This table provides the mapping from an upstream WAN interface PII to the PII of the group (GroupPii) of which it is a member.

Syntax: *command* cmWANMemberTable *[pii] Object_1*

Example:

```
getall cmwanm
Group: cmWanMemberTable
Instance: [1.3.1.30]
Pii             =  1.3.1.30
GroupPii        =  1.3.1.30
```

Objects:

*Pii	WAN Interface PII
GroupPii	The PII of the group to which this WAN interface belongs.

(*) indicates object used as an index.

dsx1ConfigTable
(alias: ds1cfg)

This group allows you to set and display configuration information about Quad T1 WAN and T1 DSL interfaces and their connected links. It also allows you to control loopback testing, set a transmit clock source, and enable traps for all of the line status changes.

You can perform five types of loopback testing of the interfaces and their associated links: an inward loop, a near-end line loop, a far-end line loop, a near-end payload loop, and a far-end payload loop. (See the diagram following the Objects table.)

Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx1Config table is one of seven DSX1groups that support the Quad T1 WAN and T1 DSL modules. For more information about these groups, see *Dsx1 Groups* on page 17.

Syntax: *command* dsx1config *[lineindex]*

Quad T1 WAN modules can only be installed in slots 1.3, 1.4, 1.15, and 1.16. T1 line cards can be installed in slots 1.6 through 1.13.

Example: To query the interface at port 1, slot 3 about various parameters, DS1 line length, and transmission attentuation:

```
get ds1cfg [1.3.1]
Group: dsx1ConfigTable
Instance: [1.3.1.0]
LineIndex             =  1.3.1.0
TimeElapsed           =  183476
ValidIntervals        =  96
LineType              =  ESF
LineCoding            =  B8ZS
SendCode              =  NoCode
CircuitIdentifier     =  ""
LoopbackConfig        =  NoLoop
LineStatus            =  8192
SignalMode            =  None
TransmitClockSource   =  LocalTiming
Fdl                   =  None
InvalidIntervals      =  0
LineStatusLastChange  =  0 day 0 hour 0 min 19.0 sec
                         (2000/06/27-12:40:14)
LineStatusChangeTrap  =  TrapDisabled
LoopbackStatus        =  1
Ds1ChannelNumber      =  0
Channelization        =  Disabled
```

Note: Note the recent addition of *OtherPattern* to the SendCode object. You must use this value for setting up and tearing down (resetting) custom FDL loops between a T1 port on a CE and a device at the other end of the loop, such as a SmartJack. For complete details on configuring custom FDL loops, see the *Troubleshooting* chapter in your *CopperEdge Installation and Operating Guide*.

Objects:

*LineIndex alias: pii	The Permanent Interface Identifier (PII) of the Quad T1 WAN or T1 DSL port. This object is used as an index to the configuration table.
TimeElapsed	Elapsed time, in seconds, since the beginning of the near-end current interval: 0 - 899.

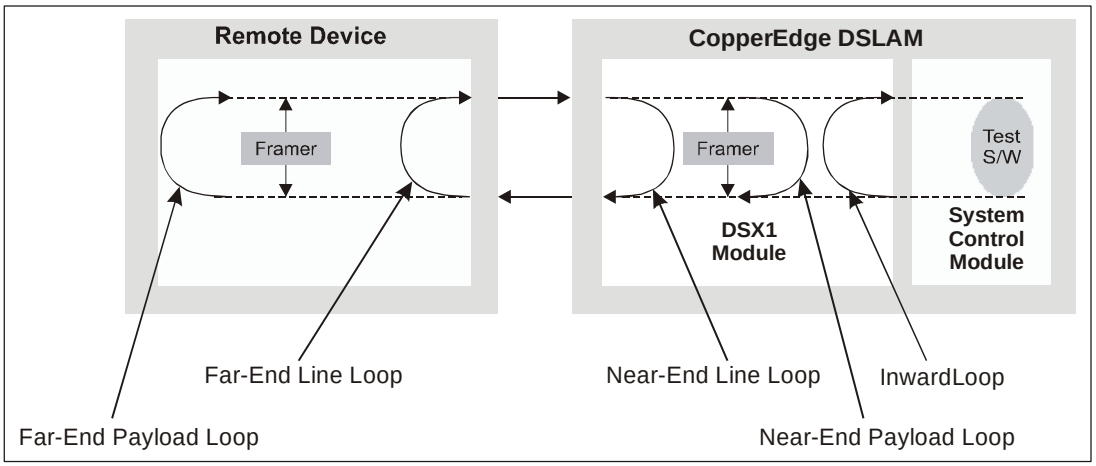
ValidIntervals	Number of previous near-end intervals during which data was collected: 0 - 96
LineType alias: lt	The type of line implementing the T1 circuit and thus, the framing format for this port. This release supports ESF (extended superframe), E1, and E1CRC
LineCoding alias: lc	Zero Code Suppression. Supported values are: B8ZS (Bipolar 8 Zero Substitution, the default) HDB3 (High Density Bipolar 3) The codes refers to the use of a specified pattern of normal bits and bipolar violations, which are used to replace a sequence of eight zero bits.
SendCode alias: sc	Type of code sent through the interface by the near-end device: NoCode--Send normal data (the default) LineCode--Request for line loopback PayloadCode--Request for payload loopback ResetCode--Loopback Termination request OtherPattern--Send a pattern not described here This object is used to initiate far-end loopbacks **Note:** If *Fdl*, on the *cmDsx1ConfigTable*, is set to *None*, the only valid option for *SendCode* is *NoCode*.
CircuitIdentifier	A unique numeric string identifying this circuit; used primarily for troubleshooting: 0 - 255
LoopbackConfig alias: lb	Near-end loopback configuration for the interface: NoLoop—(default) The interface is not in loop-back state. LineLoop (near-end loopback) — Before the framer on a WAN or DSL link, the receiver reflects the signal back to the transmitter. InwardLoop—On a WAN-SCM loop, the WAN module reflects the signal back to the SCM. PayloadLoop (near-end loopback) — After the framer on a WAN or DSL link, the receiver strips off old framing, inserts new framing, and sends the message back to the transmitter.

LineStatus	A numeric code that indicates the status of any alarms, loopbacks, or failures on the Quad T1 WAN or T1 DSL line. The status is a bitmap sum, which can simultaneously indicate multiple failures. For example, number 72 means that bits 64 and 8 are set. Number 1 (bit 1) can be set only if no other bit is set: 1 dsx1NoAlarm—No alarm present 2 dsx1RcvFarEndLOF—Receivingremote/yellow alarm msg. 4 dsx1XmtFarEndLOF—Transmitting remote/yellow alarm msg. 8 dsx1RcvAIS—Receiving Alarm Indication Signal 16 dsx1XmtAIS—Transmitting Alarm Indication Signal 32 dsx1LossOfFrame—Receiving Loss Of Frame message 64 dsx1LossOfSignal—Receiving Loss Of Signal message 128 dsx1LoopbackState—Looping back the received signal 8192 dsx1UnavailableSigState
SignalMode alias: mode	The only valid response is *None*, indicating that no bits have been reserved for signaling on this channel.
TransmitClockSource alias: txclk	Timing source to be used by the interface: LoopTiming—(default) The recovered receive clock used on the circuit. LocalTiming—The interface's own clock. Always use *LoopTiming* if available to obtain the accuracy of the Stratum 1 clock used by the carriers.
Fdl	The type of Facilities Data Link (FDL): FdlNone or AnsiT1403 (default).
InvalidIntervals	The number of intervals in the past 24 hours for which valid data could not be collected: 0 to 96.
LineStatusLastChange	The elapsed time since the system was reset, and the calendar date/time when the operator logged in to the system. Elapsed time is displayed as *day hour min sec*. This is the value of the SysUpTime counter. Calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.
LineStatusChangeTrapEnable	The Status of Change trap for this interface: TrapEnabled or TrapDisabled. When *TrapEnabled* is selected, SNMP traps are issued based on changes in the LineStatus object. If disabled (the default), SNMP traps are not sent and all alarm processing for the selected interface is suspended.

LoopbackStatus	Current state of the loopback on the Quad T1 WAN or T1 DSL interface. The status is a bitmap sum, which can simultaneously indicate multiple states. For example, number 68 means that bits 64 and 4 are set. 1 dsx1NoLoopback 2 dsx1NearEndPayloadLoopback 4 dsx1NearEndLineLoopback 16 dsx1NearEndInwardLoopback 32 dsx1FarEndPayloadLoopback 64 dsx1FarEndLineLoopback
Ds1ChannelNumber	The channel number of the DS1/E1 on its parent DS2/E2 or DS3/E3. A value of 0 indicates that the DS1/E1 does not have a parent. Currently, the only valid value is 0.
Channelization	Indicates whether the interface is channelized or unchannelized. The only valid display is *Disabled*.

(*) indicates object used as an index.

The following diagram is a pictorial representation of the five loopback modes supported under dsx1Config.



DS1 Loopback Modes

dsx1CurrentTable
(alias: ds1ncur)

This group allows you to display near-end performance statistics about a Quad T1 WAN, DS1/E1 IMA or T1 DSL interface for the current time interval. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx1Current table is one of seven DSX1groups that support the DS1/E1 modules. For more information about these groups, see *Dsx1 Groups* on page 17.

Syntax: *command* dsx1current (*index)*

Example: To query the current interval:

```
get ds1ncur [1.3.1]
Group: dsx1CurrentTable
Instance: [1.3.1.0]
Index                =  1.3.1.0
ESs                  =  0
SESs                 =  0
SEFSs                =  0
UASs                 =  612
CSSs                 =  0
PCVs                 =  0
LESs                 =  0
BESs                 =  0
LCVs                 =  0
```

Objects:

*Index alias: pii	The Permanent Interface Identifier (PII) of the DS1 or E1 port.
ESs	The number of Errored Seconds recorded on the near-end interface during the period.
SESs	The number of Severely Errored Seconds recorded on the near-end interface during the period.
SEFSs	The number of Severely Errored Framing Seconds recorded on the near-end interface in this period. (This object not supported for E1 interfaces)
UASs	The number of Unavailable Seconds recorded on the near-end interface during the period.
CSSs	The number of Controlled Slip Seconds recorded on the near-end interface during the period.
PCVs	The number of Path Coding Violations recorded on the near-end interface during the period. (This object not supported for E1 interfaces)
LESs	The number of Line Errored Seconds recorded on the near-end interface during the period.
BESs	The number of Bursty Errored Seconds recorded on the near-end interface during the period. (This object not supported for E1 interfaces)
LCVs	The number of Line Code Violations recorded on the near-end interface during the period.

(*) indicates object used as an index.

dsx1FarEndCurrentTable
[alias: ds1fcur]

This group allows you to display far-end performance statistics from a DS1 or E1 interface for the current time interval. For network-side interfaces, statistics are sent by a switch or router. For DS1 DSL interfaces, the statistics are sent by the associated CPE.

Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx1FarEndCurrent table is one of seven DSX1groups that support the various DS1/E1 modules. For more information about these groups, see *Dsx1 Groups* on page 17.

Syntax: *command* dsx1farendcurrent (*currentindex]*

Example: To query the current interval and look at errored seconds, severely errored seconds, and other similar problems on the interface:

```
get ds1fcur [1.3.1]
Group: dsx1FarEndCurrentTable
Instance: [1.3.1.0]
CurrentIndex          =  1.3.1.0
TimeElapsed           =  787
ValidIntervals        =  0
CurrentESs            =  0
CurrentSESs           =  0
CurrentSEFSs          =  0
CurrentUASs           =  0
CurrentCSSs           =  0
CurrentLESs           =  0
CurrentPCVs           =  0
CurrentBESs           =  0
InvalidIntervals      =  0
```

Objects:

*CurrentIndex alias: pii	The Permanent Interface Identifier (PII) of the DS1 or E1 interface.
TimeElapsed	Elapsed time, in seconds, since the beginning of the far-end current error measurement period. If the current interval exceeds the maximum value, the agent will return the maximum value.
ValidIntervals	The number of previous intervals during which data was collected, as reported from the far-end interface. The value is 96 unless the interface was brought online within the last 24 hours. In that case, the value will be the number of complete 15 minute intervals since the interface was brought online.
CurrentESs	The number of Errored Seconds reported from the far-end interface during the period.
CurrentSESs	The number of Severely Errored Seconds reported from the far-end interface during the period.
CurrentSEFSs	The number of Severely Errored Framing Seconds reported from the far-end interface during the period.

CurrentUASs	The number of Unavailable Seconds reported from the far-end interface during the period.
CurrentCSSs	Controlled Slip Seconds (not supported for this group; value is always 0)
CurrentLESs	The number of Line Errored Seconds reported from the far-end interface during the period.
CurrentPCVs	The number of Path Coding Violations reported from the far-end interface during the period.
CurrentBESs	The number of Bursty Errored Seconds reported from the far-end interface during the period.
InvalidIntervals	The number of intervals for which no data is available, as reported from the far-end interface. This is typically 0, except in cases where data for some intervals is not available (such as in proxy situations).

(*) indicates object used as an index.

dsx1FarEndIntervalTable
(alias: ds1fint)

This group allows you to display far-end performance statistics from a DS1 or E1 interface for any specified time interval in the past 24 hours. For the network-side interfaces, the statistics are sent by a switch or router. For the DS1 DSL interfaces, the statistics are sent by the related CPE.

Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx1FarEndInterval table is one of seven DSX1groups that support the DS1/E1 modules. For more information about these groups, see *Dsx1 Groups* on page 17.

Syntax: *command* dsx1farendinterval (*index, number*)

The *number* tells the unit how far back in time to look. The most recently completed interval has a value of 1. To see statistics from 30 minutes ago when the interval is set to 15 minutes, enter 2; to see statistics from five hours ago when the interval is set to 60 minutes, enter 5; and so on.

Example: To query port 1, slot 3 on its performance during the most recently completed 15-minute interval:

```
get ds1fint [1.3.1, 1]
Group: dsx1FarEndIntervalTable
Instance: [1.3.1.0, 1]
Index                =  1.3.1.0
Number               =  1
ESs                  =  0
SESs                 =  0
SEFSs                =  0
UASs                 =  0
CSSs                 =  0
LESs                 =  0
PCVs                 =  0
BESs                 =  0
```

```
ValidData            =  Valid
```

Note: The interval number does not include the current, in-progress interval. To see statistics for the current interval, use the dsx1FarEndCurrent group.

Objects:

*Index	The Permanent Interface Identifier (PII) of the DS1 or E1 interface.
*Number	The number of the interval: 1 to 96, where 1 is the most recently completed interval, and 96 is the last interval.
ESs	The number of Errored Seconds reported from the far-end interface during the period.
SESs	The number of Severely Errored Seconds reported from the far-end interface during the period.
SEFSs	The number of Severely Errored Framing Seconds reported from the far-end interface during the period.
UASs	The number of Unavailable Seconds reported from the far-end interface during the period.
CSSs	Controlled Slip Seconds (not supported in this group; value is always 0)
LESs	The number of line Errored Seconds reported from the far-end interface during the period.
PCVs	The number of Path Coding Violations reported from the far-end interface during the period.
BESs	The number of Bursty Errored Seconds reported from the far-end interface during the period.
ValidData	The validity of the data for this interval: Valid or Invalid.

(*) indicates object used as an index.

dsx1FarEndTotalTable
(alias: ds1ftot)

This group allows you to display the cumulative far-end totals from a DS1 or E1 interface for the full 24 hours preceding the current interval. For network-side interfaces, the statistics are sent by a switch or router. For DS1 DSL interfaces, the statistics are sent by the related CPE.

Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx1FarEndTotal table is one of seven DSX1groups that support the Quad T1 WAN and T1 DSL modules. For more information about these groups, see *Dsx1 Groups* on page 17.

Syntax: *command* dsx1farendtotal (*index*)

Example: To query the past 24 hours:

```
get ds1ftot [1.3.1]
Group: dsx1FarEndTotalTable
```

```
Instance: [1.3.1.0]
Index                =  1.3.1.0
ESs                  =  0
SESs                 =  0
SEFSs                =  0
UASs                 =  0
CSSs                 =  0
LESs                 =  0
PCVs                 =  0
BESs                 =  0
ValidData            =  Valid
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DS1 or E1 interface.
ESs	The number of Errored Seconds reported from the far-end interface during the period.
SESs	The number of Severely Errored Seconds reported from the far-end interface during the period.
SEFSs	The number of Severely Errored Framing Seconds reported from the far-end interface in this period.
UASs	The number of Unavailable Seconds reported from the far-end interface during the period.
CSSs	Controlled Slip Seconds (not supported for this group; value is always 0)
LESs	The number of Line Errored Seconds reported from the far-end interface during the period.
PCVs	The number of Path Coding Violations reported from the far-end interface during the period.
BESs	The number of Bursty Errored Seconds reported from the far-end interface during the period.
ValidData	The validity of the data for this interval: Valid or Invalid.

(*) indicates object used as an index.

dsx1IntervalTable
(alias: ds1nint)

This group allows you to display near-end performance statistics about a DS1 or E1 interface for any specified time interval in the past 24 hours. Statistics are always stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx1Interval table is one of seven DSX1groups that support the various DS1/E1 modules. For more information about these groups, see *Dsx1 Groups* on page 17.

Syntax: *command* dsx1interval (*index, number)*

The *number* tells the unit how far back in time to look. The most recently completed interval has a value of 1. To see statistics from 30 minutes ago when the interval is set to 15 minutes, enter 2; to see statistics from five hours ago when the interval is set to 60 minutes, enter 5; and so on.

Example: To query port 1, slot 3 on its performance from the second 15-minute interval (30 minutes ago):

```
get ds1nint [1.3.1, 2]
Group: dsx1IntervalTable
Instance: [1.3.1.0, 2]
Index                =  1.3.1.0
Number               =  2
ESs                  =  0
SESs                 =  0
SEFSs                =  0
UASs                 =  0
CSSs                 =  0
PCVs                 =  0
LESs                 =  0
BESs                 =  0
LCVs                 =  0
ValidData            =  Valid
```

Note: The interval number does not include the always current, in-progress interval. To see statistics for the current interval, use the dsx1Current group.

Objects:

*Index	The Permanent Interface Identifier (PII) of the DS1 or E1 port
*Number	The number of the interval: 1 to 96, where 1 is the most recently completed interval, and 96 is the last interval.
ESs	The number of Errored Seconds recorded on the near-end interface during the period.
SESs	The number of Severely Errored Seconds recorded on the near-end interface during the period.
SEFSs	The number of Severely Errored Framing Seconds recorded on the near-end interface during the period. (This object not supported for E1 interfaces)
UASs	The number of Unavailable Seconds recorded on the near-end interface during the period.
CSSs	Controlled Slip Seconds (not supported in this group, so the value is always 0).
PCVs	The number of Path Coding Violations recorded on the near-end interface during the period. (This object not supported for E1 interfaces)
LESs	The number of Line Errored Seconds recorded on the near-end interface during the period.
BESs	The number of Bursty Errored Seconds recorded on the near-end interface during the period. (This object not supported for E1 interfaces)
DMs	Number of Degraded Minutes recorded on the near-end interface during the period.
LCVs	The number of Line Code Violations recorded on the near-end interface during the period.
ValidData	The validity of the data for this interval: Valid or Invalid.

(*) indicates object used as an index.

dsx1TotalTable
(alias: ds1ntot)

This group allows you to display the cumulative near-end totals about a DS1 or E1 interface for the full 24 hours preceding the current time interval. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx1Total table is one of seven DSX1groups that support the Quad T1 WAN and T1 DSL modules. For more information about these groups, see *Dsx1 Groups* on page 17.

Syntax: *command* dsx1total (*index)*

Example: To query the past 24 hours:

```
get ds1ntot [1.3.1]
Group: dsx1TotalTable
Instance: [1.3.1.0]
Index                =  1.3.1.0
ESs                  =  0
SESs                 =  0
SEFSs                =  0
UASs                 =  86400
CSSs                 =  0
PCVs                 =  0
LESs                 =  0
BESs                 =  0
LCVs                 =  0
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DS1 or E1 port.
ESs	The number of Errored Seconds recorded on the near-end interface during the period.
SESs	The number of Severely Errored Seconds recorded on the near-end interface during the period.
SEFSs	The number of Severely Errored Framing Seconds recorded on the near-end interface in this period. (This object not supported for E1 interfaces)
UASs	The number of Unavailable Seconds recorded on the near-end interface during the period.
CSSs	Number of Controlled Slip Seconds (not supported in this group; value is always 0)
PCVs	The number of Path Coding Violations recorded on the near-end interface during the period. (This object not supported for E1 interfaces)
LESs	The number of Line Errored Seconds recorded on the near-end interface during the period.
BESs	The number of Bursty Errored Seconds recorded on the near-end interface during the period. (This object not supported for E1 interfaces)
LCVs	The number of Line Code Violations recorded on the near-end interface during the period.

(*) indicates object used as an index.

dsx3ConfigTable
(alias: ds3cfg)

This group allows you to set and display configuration information about DS3 WAN interfaces (Frame Relay or ATM) and their connected links. It also allows you to control loopback testing, set a transmit clock source, and enable traps for all of the line status changes.

You can set three types of loopback testing of the DS3 modules and their associated WAN links: an inward loop, a near-end line loop, and a far-end line loop. (See the diagram following the Objects table.) To generate packets over an inward loop and a far-end line loop, use the object, Type, in the cmLoop table. A remote device must generate packets for a near-end line loop. If you have a DS3 ATM or Frame Relay module, all three loopback types are available. For more information, see *cmLoop* on page 142.

Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx3Config table is one of seven dsx3 groups that are available for configuring and monitoring the status of DS3 WAN links to and from the CE200. For more information about these groups, see *Dsx3 Groups* on page 18.

Syntax: *command* dsx3config [*lineindex*]

DS3 modules can only be installed in slots 1.3, 1.4, 1.15, and 1.16.

Example 1: To display information about the DS3 interface at port 1, slot 4:

```
get ds3cfg [1.4.1]
Group: dsx3ConfigTable
Instance: [1.4.1.0]
LineIndex             =  1.4.1.0
TimeElapsed           =  857465
ValidIntervals        =  96
LineType              =  ClearChannel
LineCoding            =  B3ZS
SendCode              =  NoCode
CircuitIdentifier     =  ""
LoopbackConfig        =  NoLoop
LineStatus            =  1
TransmitClockSource   =  LocalTiming
InvalidIntervals      =  0
LineLength            =  1
LineStatusLastChange  =  0 day 0 hour 0 min 25.0 sec
                         (2000/06/27-12:34:59)
LineStatusChangeTrap  =  TrapDisabled
LoopbackStatus        =  1
```

Example 2: To set the source of the transmit clock to LoopTiming for the DS3 interface at port 1, slot 3:

```
set ds3cfg [1.3.1] txclk=loop
```

In any loopback test, be sure to specify only one timing source for the entire loop.

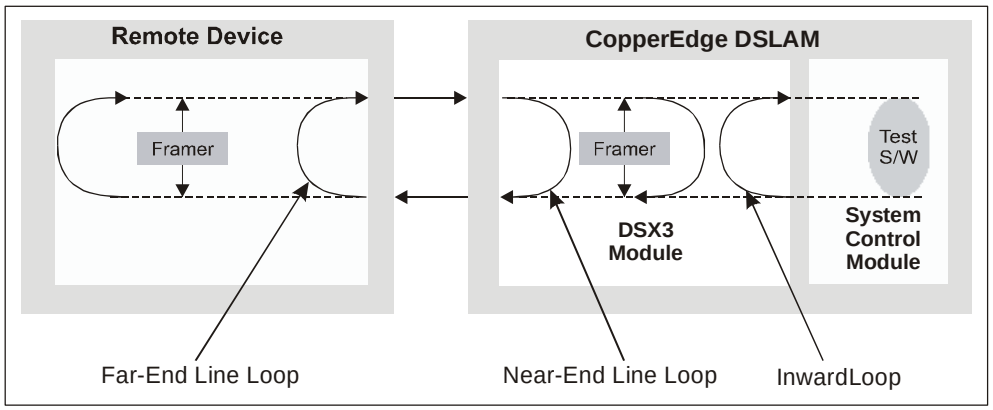
Objects:

*LineIndex alias: pii	The Permanent Interface Identifier (PII) of the index to the configuration table.
TimeElapsed	The elapsed time, in seconds, since the beginning of the near-end current interval.
ValidIntervals	Number of previous intervals during which valid data was collected in the past 24 hours: 0 to 96.
LineType alias: lt	The type of DS3 line: ClearChannel or M23Multiplex (ATM modules do not support M23).
LineCoding alias: lc	The type of Zero Code Suppression used on the interface. The only supported value is B3ZS.
SendCode alias: sc	The type of code that is to be sent through the DS3 interface by the near-end device: NoCode — Send normal data. LineLoopCode — Request a line loopback. ResetLoopCode — Deactivate a loopback.
CircuitIdentifier	A unique string identifying this DS3 circuit. It is used primarily for troubleshooting.
LoopbackConfig alias: lb	The loopback configuration for the interface: NoLoop— Interface is not in loopback state. LineLoop—On a WAN link, the receiver of the message reflects the signal back to the transmitter. Either the near-end (WAN module) or far-end (router) can do the reflecting. InwardLoop—On a WAN-SCM link, the WAN module reflects the signal back to the SCM.
LineStatus	A numeric code indicating the status of any alarms, loopbacks, or failures on the DS3 line. The status is a bitmap represented as a sum, which can simultaneously indicate multiple states. For example, number 72 means that bits 64 and 8 are set. Number 1 (bit 1) can be set only if no other bit is set: 1 dsx3NoAlarm 2 dsx3RcvRAIFailure 4 dsx3XmitRAIAlarm 8 dsx3RcvAIS 16 dsx3XmitAIS 32 dsx3LOF
LineStatus (continued)	64 dsx3LOS 128 dsx3LoopbackState 256 dsx3RcvTestCode 512 dsx3OtherFailure 1024 dsx3UnavailSigState 2048 dsx3NetEquipOOS
TransmitClockSource alias: txclk	The timing source to be used on the DS3 interface: LocalTiming (internal clock) or LoopTiming (recovered receive clock) If available, LoopTiming is recommended for carrier-class (Stratum 1) accuracy.

InvalidIntervals	The number of intervals in the past 24 hours for which valid data could not be collected: 0 to 96.
LineLength	Approximate length, in meters, of the line connected to this interface: 0 to 137 meters.
LineStatusLastChange	Elapsed time since the system was reset, and the calendar date/time when the operator logged in to the system. The elapsed time is expressed as *day hour min sec*. This is the value of the SysUpTime counter. Date/time expressed as *(yyyy/mm/dd-hh:mm:ss)*.
LineStatusChangeTrap alias: trap	The Status of Change trap for this interface: TrapEnabled or TrapDisabled.
LoopbackStatus	Current state of the loopback on the DS3 interface. The status is a bitmap represented as a sum, which can simultaneously indicate multiple states. For example, the displayed number 68 indicates that bits 64 and 4 are set. 1 No Loopback 2 NearEndPayloadLoopback—Not supported in this release. 4 NearEndLineLoopback 8 NearEndOtherLoopback—Not supported in this release. 16 NearEndInwardLoopback 32 FarEndPayloadLoopback—Not supported in this release. 64 FarEndLineLoopback

(*) indicates object used as an index.

The diagram below shows the available dsx3 loopbacks.



DS3 Loopback Modes

dsx3CurrentTable
(alias: ds3ncur)

This group displays near-end performance statistics for a DS3 WAN port (Frame or ATM) for the current time interval. Statistics are stored for user-configured intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx3Current table is one of seven dsx3 groups that are available for configuring and monitoring the status of DS3 WAN links to and from the CE200. For more information about these groups, see *Dsx3 Groups* on page 18.

Syntax: *command* dsx3current [*index*]

Example: To display information measured at DS3 WAN port 1, slot 4 from the current 15-minute interval:

```
get ds3ncur [1.4.1]
Group: dsx3CurrentTable
Instance: [1.4.1.0]
Index                =  1.4.1.0
PESs                 =  0
PSESs                =  0
SEFSs                =  0
UASs                 =  0
LCVs                 =  0
PCVs                 =  0
LESs                 =  0
CCVs                 =  0
CESs                 =  0
CSESs                =  0
```

Objects:

*Index alias: pii	The Permanent Interface Identifier (PII) for the DS3 WAN port. This object is used as an index
PESs	The number of P-bit Errored Seconds recorded on the near-end interface during the period. For ATM modules, this object will always be zero, as it only applies when the LineType is M23. Not supported by DS3 ATM modules.
PSESs	The number of P-bit Severely Errored Seconds recorded on the near-end interface during the period. For ATM modules, this object will always be zero, as it only applies when the LineType is M23. Not supported by DS3 ATM modules.
SEFSs	The number of Severely Errored Framing Seconds recorded on the near-end interface during the period.
UASs	The number of unavailable seconds recorded on the near-end interface during the period.
LCVs	The number of Line Code Violations recorded on the near-end interface during the period.
PCVs	The number of P-bit Coding Violations (P-bit parity errors) recorded on the near-end interface during the period. For ATM modules, this object will always be zero, as it only applies when the LineType is M23. Not supported by DS3 ATM modules.

LESs	The number of Line Errored Seconds recorded on the near-end interface during the period.
CCVs	The number of C-bit Coding Violation events recorded on the near-end interface during the period.
CESs	The number of C-bit Errored Seconds recorded on the near-end interface during the period.
CSESs	The number of C-bit Severely Errored Seconds recorded on the near-end interface during the period.

(*) indicates object used as an index.

dsx3FarEndCurrentTable
(alias: ds3fcur)

This group displays far-end performance statistics from a DS3 WAN port (Frame Relay or ATM) for the current time interval. The statistics are sent by a switch or router, and collected from the far-end block error code within the C-bits.

Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

Far-end statistics are available only for the ClearChannel line type (as set in the dsx3Config group); for M23 Multiplex lines, the displayed response (all zeros) does not represent actual data.

The dsx3FarEndCurrent table is one of seven dsx3 groups that are available for configuring and monitoring the status of DS3 WAN links to and from the CE200. For more information about these groups, see *Dsx3 Groups* on page 18.

Syntax: *command* dsx3farendcurrent [*currentindex*]

Example: To display statistics from the current interval, as measured at the far end of the DS3 link:

```
get ds3fcur [1.4.1]
Group: dsx3FarEndCurrentTable
Instance: [1.4.1.0]
CurrentIndex          =  1.4.1.0
TimeElapsed           =  863857
ValidIntervals        =  96
CurrentCESs           =  0
CurrentCSESs          =  0
CurrentCCVs           =  0
CurrentUASs           =  0
InvalidIntervals      =  0
```

Objects:

*CurrentIndex	The Permanent Interface Identifier (PII) of this DS3 WAN port. This object is used as an index
TimeElapsed	The elapsed time, in seconds, since the beginning of the current 15-minute interval.
ValidIntervals	The number of completed intervals for which valid data was collected during the last 24 hours
CurrentCESs	The number of C-bit Errored Seconds reported from the far-end interface during the period.
CurrentCSESs	The number of C-bit Severely Errored Seconds reported from the far-end interface in this period.
CurrentCCVs	The number of C-bit Coding Violations reported from the far-end interface during the period.
CurrentUASs	The number of Unavailable Seconds reported from the far-end interface during the period.
InvalidIntervals	The number of invalid intervals during the last 24 hours, as reported from the far-end interface.

(*) indicates object used as an index.

dsx3FarEndIntervalTable
(alias: ds3fint)

This group displays far-end performance statistics from a DS3 WAN port for any specified time interval in the past 24 hours. The statistics are sent by a switch or router. They are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

Far-end statistics are available only for the ClearChannel line type; for M23 Multiplex lines, the displayed response is all zeros and does not represent actual data.

Syntax: *command* dsx3farendinterval [*index*, *number*]

The *number* tells the unit how far back in time to look. The most recently completed interval has a value of 1. To see statistics from 30 minutes ago when the interval is set to 15 minutes, enter 2; to see statistics from five hours ago when the interval is set to 60 minutes, enter 5; and so on.

Example: To query port 1, slot 4 on its performance during the third interval:

```
get ds3fint [1.4.1, 3]
Group: dsx3FarEndIntervalTable
Instance: [1.4.1.0, 3]
Index                =  1.4.1.0
Number               =  3
CESs                 =  0
CSESs                =  0
CCVs                 =  0
UASs                 =  0
ValidData            =  Valid
```

Note: The dsx3FarEndInterval group does not include the current, in-progress interval. To see statistics for the current interval, use the dsx3FarEndCurrent group.

Objects:

*Index	The Permanent Interface Identifier (PII) of the DS3 WAN port. This object is used as an index
*Number	The number of the interval: 1 to 96, where 1 is the most recently completed interval, and 96 is the last interval.
CESs	The number of C-bit Errored Seconds reported from the far-end interface during the period.
CSESs	The number of C-bit Severely Errored Seconds reported from the far-end interface during the period.
CCVs	The number of C-bit Coding Violations reported from the far-end interface during the period.
UASs	The number of Unavailable Seconds reported from the far-end interface during the period.
ValidData	The validity of the data for this interval: Valid or Invalid.

(*) indicates object used as an index.

dsx3FarEndTotalTable
(alias: ds3ftot)

This group displays far-end cumulative performance statistics from a DS3 WAN port (Frame Relay or ATM) for all 96 intervals preceding the current time interval. The statistics are sent by a switch or router. They are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

Far-end statistics are available only for the ClearChannel line type (as set in the dsx3Config group); for M23 Multiplex lines, the displayed response (all zeros) does not represent actual data.

The dsx3FarEndTotal table is one of seven dsx3 groups that are available for configuring and monitoring the status of DS3 WAN links to and from the CE200. For more information about these groups, see *Dsx3 Groups* on page 18.

Syntax: *command* dsx3farendtotal [*index*]

Example: To display cumulative totals for all 96 intervals, as measured at the far end of the DS3 link:

```
get ds3ftot [1.4.1]
Group: dsx3FarEndTotalTable
Instance: [1.4.1.0]
Index                =  1.4.1.0
CESs                 =  0
CSESs                =  0
CCVs                 =  0
UASs                 =  0
ValidData            =  Valid
```

Objects:

*Index	The Permanent Interface Identifier (PII) of this DS3 WAN port. This object is used as an index
CESs	The number of C-bit Errored Seconds reported from the far-end interface during the period.
CSESs	The number of C-bit Severely Errored Seconds reported from the far-end interface during the period.
CCVs	The number of C-bit Coding Violations reported from the far-end interface during the period.
UASs	The number of Unavailable Seconds reported from the far-end interface during the period.
ValidData	The validity of the data for this interval: Valid or Invalid.

(*) indicates object used as an index.

dsx3IntervalTable
(alias: ds3nint)

This group displays near-end performance statistics about a DS3 WAN port (Frame Relay or ATM) for any specified time interval in the past 96 intervals. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx3Interval table is one of seven dsx3 groups that are available for configuring and monitoring the status of DS3 WAN links to and from the CE200. For more information about these groups, see *Dsx3 Groups* on page 18.

Syntax: *command* dsx3interval [*index, number*]

The *number* tells the unit how far back in time to look. The most recently completed interval has a value of 1. To see statistics from 30 minutes ago when the interval is set to 15 minutes, enter 2; to see statistics from five hours ago when the interval is set to 60 minutes, enter 5; and so on.

Example: To query port 1, slot 4 on its performance during the third interval:

```
get ds3nint [1.4.1, 3]
Group: dsx3IntervalTable
Instance: [1.4.1.0, 3]
Index                =  1.4.1.0
Number               =  3
PESs                 =  0
PSESs                =  0
SEFSs                =  0
UASs                 =  0
LCVs                 =  0
PCVs                 =  0
LESs                 =  0
CCVs                 =  0
CESs                 =  0
CSESs                =  0
ValidData            =  Valid
```

Note: The interval number does not include the current, in-progress interval. To see statistics for the current interval, use the dsx3Current group.

Objects:

*Index	The Permanent Interface Identifier (PII) of the DS3 WAN port. This object is used as an index
*Number	The number of the interval: 1 to 96, where 1 is the most recently completed interval, and 96 is the last interval.
PESs	The number of P-bit Errored Seconds recorded on the near-end interface during the period. For ATM modules, this object will always be zero, as it only applies when the LineType is M23. Not supported by DS3 ATM modules.
PSESs	The number of P-bit Severely Errored Seconds recorded on the near-end interface during the period. For ATM modules, this object will always be zero, as it only applies when the LineType is M23. Not supported by DS3 ATM modules.

SEFSs	The number of Severely Errored Framing Seconds recorded on the near-end interface during the period.
UASs	The number of Unavailable Seconds recorded on the near-end interface during the period.
LCVs	The number of Line Code Violations recorded on the near-end interface during the period.
PCVs	The number of P-bit Coding Violations (P-bit parity errors) recorded on the near-end interface during the period. For ATM modules, this object will always be zero, as it only applies when the LineType is M23. Not supported by DS3 ATM modules.
LESs	The number of Line Errored Seconds recorded on the near-end interface during the period.
CCVs	The number of C-bit Coding Violation events recorded on the near-end interface during the period.
CESs	The number of C-bit Errored Seconds recorded on the near-end interface during the period.
CSESs	The number of C-bit Severely Errored Seconds recorded on the near-end interface during the period.
ValidData	The validity of the data for this interval: Valid or Invalid.

(*) indicates object used as an index.

dsx3TotalTable
(alias: ds3ntot)

This group displays cumulative near-end statistics about a DS3 WAN port (Frame Relay or ATM) for all of the intervals preceding the current interval. Statistics are stored for user-configured time intervals of 5, 15, or 60 minutes. For information about setting the intervals, see *cmAlarm* on page 48.

The dsx3Total table is one of seven dsx3 groups that are available for configuring and monitoring the status of DS3 WAN links to and from the CE200. For more information about these groups, see *Dsx3 Groups* on page 18.

Syntax: *command* dsx3total [*index*]

Example: To display cumulative totals for all of the intervals preceding the current interval, as measured at port 1, slot 4:

```
get ds3ntot [1.4.1]
Group: dsx3TotalTable
Instance: [1.4.1.0]
Index                =  1.4.1.0
PESs                 =  0
PSESs                =  0
SEFSs                =  0
UASs                 =  0
LCVs                 =  0
PCVs                 =  0
LESs                 =  0
CCVs                 =  0
CESs                 =  0
CSESs                =  0
ValidData            =  Valid
```

Objects:

*Index	The Permanent Interface Identifier (PII) of the DS3 WAN port. This object is used as an index
PESs	The number of P-bit Errored Seconds recorded on the near-end interface during the period. For ATM modules, this object will always be zero, as it only applies when the LineType is M23. Not supported by DS3 ATM modules.
PSESs	The number of P-bit Severely Errored Seconds recorded on the near-end interface during the period. For ATM modules, this object will always be zero, as it only applies when the LineType is M23. Not supported by DS3 ATM modules.
SEFSs	The number of Severely Errored Framing Seconds recorded on the near-end interface during the period.
UASs	The number of Unavailable Seconds recorded on the near-end interface during the period.
LCVs	The number of Line Code Violations recorded on the near-end interface during the period.
PCVs	The number of P-bit Coding Violations (P-bit parity errors) recorded on the near-end interface during the period. For ATM modules, this object will always be zero, as it only applies when the LineType is M23. Not supported by DS3 ATM modules.
LESs	The number of Line Errored Seconds recorded on the near-end interface during the period.
CCVs	The number of C-bit Coding Violation events recorded on the near-end interface during the period.
CESs	The number of C-bit Errored Seconds recorded on the near-end interface during the period.
CSESs	The number of C-bit Severely Errored Seconds recorded on the near-end interface during the period.
ValidData	The validity of the data for this interval: Valid or Invalid.

(*) indicates object used as an index.

frCircuitTable
(alias: frpvc)

This group displays information about Data Link Connection Identifiers (DLCIs) and their corresponding virtual circuits (PVCs) on both DSL and WAN links. It also allows you to configure performance parameters for WAN VCs, such as Throughput, Committed-Burst, and ExcessBurst. But it does not allow you to configure performance parameters for DSL VCs or for (DSL) IMUX bundles.

If you attempt to set performances parameters on a DSL VC or an IMUX bundle, the CopperCraft CLI may return a `Set Successful` message, but the command will have no effect. Its values will be discarded.

The frCircuit group does allow you to display information about traffic on DSL VCs. For example, you are able to look at traffic on voice and data VCs serving IADs, such as CR408s, CR508s, and CR508Ts. The table allows you to see the number of frames and octets received and transmitted since the VCs were created.

The frCircuit group for both received and transmitted frames will not increment for voice circuits after call setup when a PathStar gateway is involved. Call packets go from peer-to-peer over the CE200 without using the WAN interface. When a TollBridge, CopperCom, or JetStream gateway is involved, the situation is different. Call packets going from peer-to-peer must use the WAN interface and travel to and from the voice gateway.

Syntax: *command* frcircuit [*ifindex*]

The *ifindex* points to the full table entries for this interface in the ifTable, and in the frCircuit group. A DLCI may be any number from 16 to 991.

Example 1: To query VC 23 on port 1, slot 8:

```
get frc [1.8.1.23]
Group: frCircuitTable
Instance: [1.8.1.23]
IfIndex              =  1.8.1.0
Dlci                 =  23
State                =  active
ReceivedFECNs        =  0
ReceivedBECNs        =  0
SentFrames           =  531117
SentOctets           =  212945712
ReceivedFrames       =  608855
ReceivedOctets       =  151851006
CreationTime         =  0 day 0 hour 0 min 0.0 sec
                        (2000/06/19-13:41:16)
LastTimeChange       =  0 day 0 hour 0 min 0.0 sec
                        (2000/06/19-13:41:16)
CommittedBurst       =  0
ExcessBurst          =  0
Throughput           =  0
RuntimeState         =  up
```

Example 2: To set the traffic parameters for VC 101 on port 3, slot 6:

```
set frc [1.6.3.101] cir=764 bcmax=256 be=256
```

The general rules for configuring CIR, Bc, and Be are as follows:

- If B_c=0 and CIR=0, $B_e \leq$ the configured port speed

- If Bc>0 and CIR>0, CIR $(B_e / B_c + 1) \leq$ the configured port speed

The maximum port speeds are:

- An SDSL port has a maximum speed of 1.568 Mbps
- An IDSL port has a maximum speed of 144 Kbps
- DS3 ATM and Frame Relay WAN port has a maximum speed of 45 Mbps
- A V.35 port has a maximum speed of 1.568 Mbps
- A Quad T1 Frame Relay WAN port has a maximum speed of 1.568 Mbps

Example 3: To query data VC 528, port 7, slot 11 on a CR408:

```
get frc [1.11.7.528]
Group: frCircuitTable
Instance: [1.11.7.528]
IfIndex              =  1.11.7.0
Dlci                 =  528
State                =  active
ReceivedFECNs        =  0
ReceivedBECNs        =  0
SentFrames           =  122668
SentOctets           =  37523630
ReceivedFrames       =  0
ReceivedOctets       =  0
CreationTime         =  0 day 0 hour 0 min 0.0 sec
                        (2000/06/19-13:41:16)
LastTimeChange       =  0 day 0 hour 0 min 0.0 sec
                        (2000/06/19-13:41:16)
CommittedBurst       =  0
ExcessBurst          =  0
Throughput           =  0
RuntimeState         =  down
```

The only encapsulation type for data currently allowed by the CR408 is rfc1483. Although you do not specify VC 528 when configuring it in cmIface, you must specify VC 528 when querying it in frCircuit group.

Example 4: To query voice VC 22, port 2, slot 11 on a CR408:

```
get frc [1.11.2.22]
Group: frCircuitTable
Instance: [1.11.2.22]
IfIndex              =  1.11.2.0
Dlci                 =  22
State                =  active
ReceivedFECNs        =  0
ReceivedBECNs        =  0
SentFrames           =  65853
SentOctets           =  8170131
ReceivedFrames       =  65902
ReceivedOctets       =  7667352
CreationTime         =  0 day 0 hour 0 min 0.0 sec
                        (2000/06/19-13:41:16)
LastTimeChange       =  0 day 0 hour 0 min 0.0 sec
                        (2000/06/19-13:41:16)
CommittedBurst       =  0
ExcessBurst          =  0
Throughput           =  0
RuntimeState         =  up
```

The only encapsulation type for voice currently allowed by the CR408, CR508, and CR508T is IP-1490. You must specify VC 22 when configuring the VC in cmIface, and when querying it in frCircuit group.

Notes:

1. To ensure simultaneous application and correct implementation, all three of the configurable congestion-management objects must be entered in a single command as shown in Example 2.
2. The values of Throughput and Committed Burst must both be set to 0 (no commitment), or both must be set to values other than 0. If one parameter is configured for a commitment, both must be configured for a commitment.
3. Any FR WAN circuits configured with Throughput=0, CommittedBurst=0, and ExcessBurst=Line Rate will have all packets marked with DE=1. During times of excess burst, packets that are part of the burst will be marked Discard Eligible (DE). If the WAN transmit queue becomes congested, the DE packets will be dropped. This feature makes the function Telecordia-compliant.

Objects:

*IfIndex alias: Ifx	The value of the ifIndex object from the IfTable. All virtual circuits are layered onto existing interface entries in the IfTable.
Dlci	The Data Link Connection ID for this virtual circuit.
State	The operational state of a row in the frCircuit table (the virtual circuit): Invalid—Deletes a row (virtual circuit entry). Active—In a Get command, the row is actively being used by the CE200. In a Set command, creates a new row. Inactive—Temporarily disables a row. For a complete definition of State as it relates to this group, see the “Links” discussion in *Operating Status* on page 20.
ReceivedFECNs alias: fecn	The number of frames received from the network reporting forward congestion since the virtual circuit was created.
ReceivedBECNs alias: becn	The number of frames received from the network reporting backward congestion since the virtual circuit was created.
SentFrames alias: txframes	The number of frames sent from this virtual circuit since it was created.
SentOctets alias: txoctets	The number of octets sent from this virtual circuit since it was created.
ReceivedFrames alias: rxframes	The number of frames received over this virtual circuit since it was created.
ReceivedOctets alias: rxoctets	The number of octets received over this virtual circuit since it was created.
CreationTime alias: createtime	The value of System UpTime when the PVC was created, whether by DLCMI or a Set command.
LastTimeChange alias: lastchange	The value of System UpTime at the instant of the most recent change to the virtual circuit state.

CommittedBurst alias: bcmax	The maximum amount of data, in bits, that the network will transfer under normal conditions during the measurement interval. The default value is 0 (that is, no commitment).
ExcessBurst alias: be	The maximum amount of uncommitted data bits that the network will attempt to deliver over the measurement interval. If not configured when the entry was created, the default is the value of ifTableSpeed.
Throughput alias: cir	The committed throughput value, commonly referred to as the Committed Information Rate (CIR). On the CE200, CIR is expressed in bits per second. CIR is the average number of Frame Relay Information Field bits transferred per second across a user network interface in one direction, within the period of a measurement interval as defined below. If CommittedBurst (bcMax) and Throughput (CIR) are both configured with a value other than zero, the committed measurement interval Tc is defined as: Tc=bcMax/CIR If CommittedBurst (bcMax) and Throughput (CIR) are both configured as zero, then Tc is defined as: Tc=be/if Speed The default value is 0 (no commitment).
RunTimeState alias: obstate	The current functional state of the virtual circuit: Up or Down.

(*) indicates object used as an index.

frDlcmiTable
(alias: frlink)

This group displays information relating to the Frame Relay Data Link Connection Management Interface for the Frame Relay service on a specified interface.

When you configure an interface (WAN port, DSL port, or IMUX DSL bundle) for frame relay through cmIface or cmSubIface, the corresponding row in the frDlcmi table is automatically created for the physical interface or the IMUX DSL bundle using the default settings for all relevant objects.

Syntax: *command* frdlcmi [*ifindex*]

Example: To query port 1, slot 3:

```
get frd [1.3.1]
Group: frDlcmiTable
Instance: [1.3.1.0]
IfIndex              =  1.3.1.0
State                =  noLmiConfigured
Address              =  q922-FinalStandard
AddressLen           =  two-octets
PollingInterval      =  10
FullEnquiryInterval  =  6
ErrorThreshold       =  3
MonitoredEvents      =  4
MaxSupportedVCs      =  976
Multicast            =  nonBroadcast
RuntimeState         =  down
```

Notes:

1. Always disable the link using cmFrDlcmi (Admin=dis) before changing the LMI state or any other link attributes. Modifying link attributes is service-affecting.
2. State=lmiRev1 is not supported on DSL interfaces.

Objects:

*IfIndex alias: Ifx	The Permanent Interface Identifier (PII) of the interface or IMUX bundle on the specified module (DSL or WAN); it comes from the ifTable. All VCs are layered onto existing interface entries in the IfTable.
State alias: lmi	The Data Link Connection Management (LMI) scheme that is active on the Frame Relay interface: noLmiConfigured lmiRev1 ansiT1-617-D Q9.33-Annex-A For a complete definition of State as it relates to this group, see the "Links" discussion in *Operating Status* on page 20.
Address alias: addrfmt	The address format used on the frame relay interface: q921-13bitDLCI q922-11bitDLCI q922-10bitDLCI q922-FinalStandard In the current CE200 implementation, only q922-FinalStandard is used. The other options are not currently supported by the CE200.

AddressLen alias: addrlen	The address length in octets: two-octets three-octets four-octets In the current CE200, this value is always two-octets.
PollingInterval alias: pollintv	The time, in seconds, between successive status inquiry messages.
FullEnquiryInterval alias: enquiryinterval	The number of status-inquiry intervals before a full status inquiry message is issued (N/A for DCE).
ErrorThreshold alias: errthreshold	The number of unanswered Status Inquiries before an interface down message is sent.
MonitoredEventsf alias: monitoredev	The length of the time window (the number of status polling intervals) within which the error threshold lies. For example, if the station receives *ErrorThreshold* number of errors within the specified *MonitoredEvents* number of events, the interface is marked as Down.
MaxSupportedVCs alias: maxpvcs	The maximum number of VCs allowed on this interface. This is usually a function of the Frame Relay network. With a 128 megabyte SCM and a DS3 module, the maximum number of VCs is 976. With a 128 megabyte SCM and a DS1 module, the maximum number of VCs is about half the number supported by a DS3 module.
Multicast	Indicates if the interface uses a Multicast service: nonBroadcast or Broadcast.
RuntimeState alias: obstate	The current status of the interface and LMI: Up—LMI configured and functioning on the interface. Down—LMI not configured or not functioning on the interface.

(*) indicates object used as an index.

frErrTable

This group displays information about the current (most recently detected) error reported on a specified Frame Relay interface.

Syntax: *command* frerr [*ifindex*]

Example: To query port 1, slot 4:

```
get fre [1.4.1]
Group: frErrTable
Instance: [1.4.1.0]
IfIndex              =  1.4.1.0
Type                 =  DLCMIUnknownRpt
Data                 =  ""
Time                 =  0 day 0 hour 0 min 0.0 sec
                        (2000/06/05-15:07:45)
```

Objects:

*IfIndex alias: Ifx	Permanent Interface Identifier (PII) from the IfTable. All virtual circuits are layered onto existing interface entries in the IfTable.
Type	The type of error for the last error detected on the interface: UnknownError ReceiveShort ReceiveLong IllegalDLCI UnknownDLCI DLCIProtoErr DLCIUnknownIE DLCISequenceErr DLCMIUnknownRpt NoErrorSinceReset
Data	An octet string containing as much of the error packet as possible, including at least the Q.922 address, or delivered fragment thereof.
Time	The elapsed time since the system was reset, and the calendar date/time when the operator logged in to the system. The elapsed time is expressed as *day hour min sec*. This is the value of the SysUpTime counter. The calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.

(*) indicates object used as an index.

frGlobals

This group tells you if the global Frame Relay trap state is enabled (that is, the Frame Relay trap state is applicable to all CE200 Frame Relay circuits). It controls whether RFC-1315 frDLCI StatusChange traps are allowed to be generated.

Syntax: *command* frglobals

Example:

```
get frg

Group: frGlobals
TrapState            =  enabled
```

Objects:

TrapState alias: state	The state of the frame relay DLCI trap: enabled or disabled. When enabled, this trap detects and reports to the SNMP manager when any connected (WAN) virtual circuit changes state. A change in state is indicated if a new VC is created, or if an existing VC is switched between active and inactive states.

frSystem

This group displays information about the connected Frame Relay system.

Syntax: *command* frsystem

Example:

```
get frs

Group: frSystem
SwRelease            =  4.52
MaxLinks             =  8
MaxPVC               =  2048
```

Objects:

SWRelease alias: rel	The identifier of the frame relay software release.
MaxLinks alias: links	The maximum number of interfaces allowed on this frame relay system.
MaxPVC alias: pvcs	The maximum number of virtual circuits allowed per Frame Relay system: up to 2048. **Note:** This value may be reduced by the number of FRF.5 data links (physical ATM links) in the system.

hdsl2Shdsl15MinIntTable
(alias: shint)

Use this group to retrieve performance information via the embedded operations channel (EOC) from a specified HDSL2/SHDSL endpoint/interface. Each entry in the table represents the information for one collection interval (15 minutes) at the specified endpoint.

Objects:

*ifIndex	PII of the interface
*Invindex	An integer specifying the unique identification for all of the units in an HDSL2/SHDSL span. Values are: xtuC (1) xtuR (2) xru1 (3) xru2 (4) xru3 (5) xru4 (6) xru5 (7) xru6 (8) xru7 (9) xru8 (10) The *xru* designation applies to regenerator units (as many as eight units supported), which are numbered from the CO side.
*EPSide	Identifies the referenced side of an HDSL2/SHDSL unit as either the side facing the Network (CO) or the side facing the Customer (CPE): *networkSide* (1) or *customerSIde* (2)
*EPWirePair	Referenced copper pair in an HDSL2/SHDSL segment. HDSL2 only supports a single pair (WirePair1), while SHDSL can support4 an operational second pair (WirePair2): wirePair 1 (1) or wirePair2 (2)
*Number	The number of the interval with 1 being the most recently completed 15-minute interval, and 96 the most distant in the past 24 hours.
ES	Number of Errored Seconds during the reported interval
SES	Number of Severely Errored Seconds during the reported interval
CRCanomalies	Number of CRC anomalies during the reported interval
LOSWS	Number of Loss Of Sync Word seconds during the reported interval
UAS	Number of Unavailable Seconds during the reported interval

(*) indicates object used as an index.

hdsl2Shdsl1DayTable
(alias: sh1dint)

Use this group to retrieve performance information via the embedded operations channel (EOC) from a specified HDSL2/SHDSL endpoint/interface. Each entry in the table represents the information for one 24-hour (1-day) collection interval at the specified endpoint.

Objects:

*ifIndex	PII of the interface
*Invindex	An integer specifying the unique identification for all of the units in an HDSL2/SHDSL span. Values are: xtuC (1) xtuR (2) xru1 (3) xru2 (4) xru3 (5) xru4 (6) xru5 (7) xru6 (8) xru7 (9) xru8 (10) The *xru* designation applies to regenerator units (as many as eight units supported), which are numbered from the CO side.
*EPSide	Identifies the referenced side of an HDSL2/SHDSL unit as either the side facing the Network (CO) or the side facing the Customer (CPE): *networkSide* (1) or *customerSIde* (2)
*EPWirePair	Referenced copper pair in an HDSL2/SHDSL segment. HDSL2 only supports a single pair (WirePair1), while SHDSL can support4 an operational second pair (WirePair2): wirePair 1 (1) or wirePair2 (2)
*Number	The History Data Interval Number. Interval 1 is the most-recently completed 24-hour period and 30 being the most distantly removed in time.
MoniSecs	Amount of time in the 1-day interval over which the performance information is actually counted. This value will be the same as the interval duration except in any case where performance monitoring data could not be collected for any reason.
ES	Number of Errored Seconds during the reported interval
SES	Number of Severely Errored Seconds during the reported 1-day interval
CRCanomalies	Number of CRC anomalies during the reported 1-day interval
LOSWS	Number of Loss Of Sync Word seconds during the reported 1-day interval
UAS	Number of Unavailable Seconds during the reported 1-day interval

(*) indicates object used as an index.

hdsl2ShdslEndPointCurrTable
(alias: shcur)

This group provides status and performance information for segment endpoints in HDSL2 / SHDSL lines. Each entry in the table relates to the status and performance of a single segment endpoint.

Objects:

*ifIndex	PII of the interface
*Invindex	An integer specifying the unique identification for all of the units in an HDSL2/SHDSL span. Values are: xtuC (1) xtuR (2) xru1 (3) xru2 (4) xru3 (5) xru4 (6) xru5 (7) xru6 (8) xru7 (9) xru8 (10) The *xru* designation applies to regenerator units (as many as eight units supported), which are numbered from the CO side.
*EPSide	Identifies the referenced side of an HDSL2/SHDSL unit as either the side facing the Network (CO) or the side facing the Customer (CPE): *networkSide* (1) or *customerSIde* (2)
*EPWirePair	Referenced copper pair in an HDSL2/SHDSL segment. HDSL2 only supports a single pair (WirePair1), while SHDSL can support4 an operational second pair (WirePair2): wirePair 1 (1) or wirePair2 (2)
CurrAtn	The current loop attenuation for this endpoint, in dB
CurrSnrMgn	Current signal-to-noise ratio margin for this endpoint, in dB as reported in a StatusResponse/SNR message
CurrStatus	An integer supplementing ifOperStatus to indicate current status of the endpoint as follows: noDefect (1) powerBackoff (2) deviceFault (4) snrMarginAlarm (16) loopAttenAlarm (32) loswFailureAlarm (64) noNeighborPresent (512) loopbackActive (1024)
ES	Total number of Errored Seconds at this endpoint since the last restart
SES	Total number of Severely Errored Seconds at this endpoint since the last restart
CRCanomalies	Total number of CRC anomalies at this endpoint since the last restart
LOSWS	Total Loss of Sync Word Seconds at this endpoint since the last restart
UAS	Total unavailable seconds at this endpoint since the last restart.
Curr15MinTimeElapsed	Elapsed time in seconds since the beginning of the current 15-minute interval

Curr15MinES	Errored Seconds during the current 15-minute interval
Curr15MinSES	Severely Errored Seconds during the current 15-minute interval
Curr15MinCRCAnomalies	CRC Anomalies during the current 15-minute interval
Curr15MinLOSWS	Loss of Sync Word Seconds during the current 15-minute interval
Curr15MinUAS	Unavailable seconds during the current 15-minute interval
Curr1DayTimeElapsed	Number of seconds that have elapsed in the current 1-day interval
Curr1DayES	Errored Seconds during the current 1-day interval
Curr1DaySES	Severely Errored Seconds during the current 1-day interval
Curr1DayCRCAnomalies	CRC Anomalies during the current 1-day interval
Curr1DayLOSWS	Loss of Sync Word Seconds during the current 1-day interval
Curr1DayUAS	Unavailable seconds during the current 1-day interval

(*) indicates object used as an index.

hdsl2ShdslEndPointMaintTable
(alias: shmaint)

Use this group to configure and observe maintenance operations such as loopbacks to be performed on HDSL2/SHDSL segment endpoints. Each entry corresponds to a single segment endpoint, and is indexed according to three different parameters: the if Index of the HDSL2/SHDSL line, the UnitID of the associated unit, and the side of the unit.

Objects:

*ifIndex	PII of the interface
*Inventoryindex	An integer specifying the unique identification for all of the units in an HDSL2/SHDSL span. Values are: xtuC (1) xtuR (2) xru1 (3) xru2 (4) xru3 (5) xru4 (6) xru5 (7) xru6 (8) xru7 (9) xru8 (10) The *xru* designation applies to regenerator units (as many as eight units supported), which are numbered from the CO side.
*EPSide	Identifies the referenced side of an HDSL2/SHDSL unit as either the side facing the Network (CO) or the side facing the Customer (CPE): *networkSide* (1) or *customerSIde* (2)
LoopbackConfig	Configures loopbacks for the specified segment endpoint: *noLoopback* or *normalLoopback*

TipRingReversal	Indicates the status of the tip/ring pair at the segment endpoint as: normal (1) or reversed (2)
PowerBackOff	Selects the power back-off mode in which the specified segment endpoint operates: *default* or *enhanced*
SoftRestart	Object used to initiate soft restart of the modem at the associated segment endpoint. If you specify *Softrestart=restart*, the agent will perform a restart after approximately five seconds, after which the object is restored to its default (*ready*) state.

(*) indicates object used as an index.

hdsl2ShdslInventoryTable
(alias: shinv)

Use this group to retrieve inventory information via the embedded operations channel (EOC) from units connected to a G.SHDSL interface. Each entry in the table represents the information for a separate unit in the line.

Objects:

*ifIndex	PII of the interface
*Inventoryindex	An integer specifying the unique identification for all of the units in an HDSL2/SHDSL span. Values are: xtuC (1) xtuR (2) xru1 (3) xru2 (4) xru3 (5) xru4 (6) xru5 (7) xru6 (8) xru7 (9) xru8 (10) The *xru* designation applies to regenerator units (as many as eight units supported), which are numbered from the CO side.
vendorID	A bitmap representation of the Vendor ID for the inventory item. This object is supported in *cmCpeBoardVendorDescription* for STU-R.
VendorModelNumber	Vendor model number of the item as listed in cmCPEBoard
VendorSerialNumber	Vendor serial number of the item as listed in cmCPEBoardSerialNumber (STU-R) or cmBoardSerialNumber (STU-C)
VendorEOCSoftwareVer	The version of the EOC software used to query the SHDSL interfaces. This object is only supported if the CPE device also supports EOC messaging.
StandardVersion	The implemented version of the standard
VendorListNumber	Character string
VendorIssueNumber	Character string
VendorSoftwareVersion	The version number of the vendor software
EquipmentCode	Equipment Code per ANSI T1.213. This object is only supported if the CPE device also supports EOC messaging.

VendorOther	Other Vendor information in the table
TransmissionModeCapa	Transmission mode capability of the inventory item

(*) indicates object used as an index.

hdsl2ShdslUnitMaintTable
(alias: shumaint)

This group consists of maintenance operations for units in the HDSL2/SHDSL line. It is indexed by the interface PII and the particular unit within the span.

Objects:

*ifIndex	PII of the interface
*Invindex	An integer specifying the unique identification for all of the units in an HDSL2/SHDSL span. Values are: xtuC (1) xtuR (2) xru1 (3) xru2 (4) xru3 (5) xru4 (6) xru5 (7) xru6 (8) xru7 (9) xru8 (10)
LoopbackTimeout	Integer indicating the timeout value in seconds for loopbacks initiated at segment endpoints contained in the associated unit. A value of 0 disables the timeout function.
UnitPowerSource	Identifies the DC power source being used by the associated unit: Only *Local* is supported.

(*) indicates object used as an index.

icmp

This group displays statistics relating to Internet Control Message Protocol (ICMP) messages received and sent by the CE200.

Syntax: *command* icmp

Example: **get icmp**

```
Group: icmp
InMsgs                =  887
InErrors              =  0
InDestUnreachs        =  887
InTimeExcds           =  0
InParmProbs           =  0
InSrcQuenchs          =  0
InRedirects           =  0
InEchos               =  0
InEchoReps            =  0
InTimestamps          =  0
InTimestampReps       =  0
InAddrMasks           =  0
InAddrMaskReps        =  0
OutMsgs               =  441
OutErrors             =  440
OutDestUnreachs       =  440
OutTimeExcds          =  0
OutParmProbs          =  0
OutSrcQuenchs         =  0
OutRedirects          =  0
OutEchos              =  0
OutEchoReps           =  0
OutTimestamps         =  0
OutTimestampReps      =  0
OutAddrMasks          =  0
OutAddrMaskReps       =  0
```

Objects:

InMsgs	The total number of ICMP messages received, including errored messages
InErrors	The number of ICMP messages received that had ICMP-specific errors (such as checksum or length).
InDestUnreachs	The number of ICMP Destination Unreachable messages received.
InTimeExcds	The number of ICMP Time Exceeded messages received.
InParmProbs	The number of ICMP Parameter Problem messages received.
InSrcQuenchs	The number of ICMP Source Quench messages received.
InRedirects	The number of ICMP Redirect messages received.
InEchos	The number of ICMP Echo requests received.
InEchoReps	The number of ICMP Echo replies received.
InTimestamps	The number of ICMP Timestamp Request messages received.
InTimestampReps	The number of ICMP Timestamp Reply messages received.
InAddrMasks	The number of ICMP Address Mask Request messages received.
InAddrMaskReps	The number of ICMP Address Mask Reply messages received.
OutMsgs	The total number of ICMP messages sent, including those counted as errored.
OutErrors	The number of ICMP messages that were generated, but failed to transmit because of errors within the ICMP. This count does *not* include errors discovered outside the ICMP layer, such as the inability of IP to route the datagram.
OutDestUnreachs	The number of ICMP Destination Unreachable messages sent.
OutTimeExcds	The number of ICMP Time Exceeded messages sent.
OutParmProbs	The number of ICMP Parameter Problem messages sent.
OutSrcQuenchs	The number of ICMP Source Quench messages sent.
OutRedirects	The number of ICMP Redirect messages sent (if the entity is a host, this number remains at zero because hosts do not send redirects.
OutEchos	The number of ICMP Echo requests sent.
OutEchoReps	The number of ICMP Echo replies sent.

OutTimestamps	The number of ICMP Timestamp Request messages sent.
OutTimestampReps	The number of ICMP Timestamp Reply messages sent.
OutAddrMasks	The number of ICMP Address Mask Request messages sent.
OutAddrMaskReps	The number of ICMP Address Mask Reply messages sent.

if

This group displays the number of interfaces in the interface table.

Syntax: *command* if

Example:

```
get if

Group: if
Number                =  85
```

Objects:

Number	The number of entries in the ifTable (page 242).

ifStackTable

This group displays the relationship between a specified CE200 port and its configured PVCs. It is useful as a quick path to review configured PVC and DLCI assignments.

Syntax: *command* ifstack [*higherlayer, lowerlayer*]

Example: To query VCs 0 to 16, port 1, slot 4:

```
getn ifs [1.4.1.16, 1.4.1.0]
Group: ifStackTable
Instance: [1.4.1.16, 1.4.1.0]
HigherLayer           =  1.4.1.16
LowerLayer            =  1.4.1.0
Status                =  Active
```

Objects:

*HigherLayer	Permanent Interface Identifier (PII) of the “higher layer” of the interface pair; that is, the virtual circuit that overlays the CE200 physical interface.
*LowerLayer	The PII of the “lower layer” of the interface pair; that is, the CE200 physical interface (port).
Status	The status of a row in the ifStack table: Active—The row is actively being used by the CE200. Destroy (alias: delete)—The row is deleted. NotInService—The row exists but is not being used by the CE200.

(*) indicates object used as an index.

ifTable

This group displays packet statistics for the specified interface, either a physical interface, a VC, or an IMUX bundle.

You can selectively enable or disable any DSL or WAN physical port on the CE200. This is an administrative action and results in the generation of a trap, but does not trigger an alarm.

Syntax: *command* iftable [*index*]

Example: To display packet statistics for VC 120 on port 1, slot 4:

```
get ift [1.4.1.120]
Group: ifTable
Instance: [1.4.1.120]
Index                =  1.4.1.120
Descr                =  vmi6
Type                 =  propVirtual
Mtu                  =  1500
Speed                =  0
PhysAddress          =
AdminStatus          =  up
OperStatus           =  up
LastChange           =  0 day 0 hour 0 min 0.0 sec
                        (2000/06/27-12:34:26)
InOctets             =  62738
InUcastPkts          =  4826
InNUcastPkts         =  0
InDiscards           =  4826
InErrors             =  0
InUnknownProtos      =  0
OutOctets            =  62738
OutUcastPkts         =  4826
OutNUcastPkts        =  0
OutDiscards          =  0
OutErrors            =  0
OutQLen              =  0
Specific             =  0.0
```

Notes:

1. The CopperVPN, HDIA, and VWAN netmodels have multicast pass through on the downstream as well as the upstream. Traffic over the DSL ports, WAN ports, and WAN VCs in multicast packets appears in the non-unicast packet objects: InNUcastPkts and OutNUcastPkts.

   However, the **HDIA netmodel does not have peer-to-peer multicast.**

2. The IP and Cross-Connect netmodels do not have multicast pass through.

3. When ATM VCs and Frame Relay PVCs are specified, the entries in the ifTable actually refer to the virtual interface between the ATM/FR driver and the CE200's IP Stack. Thus, these interfaces will always display as Up, regardless of the actual status of the VC/PVC.

Objects:

*Index	The Permanent Interface Identifier (PII) of the interface.
Descr	A string describing the interface.
Type	The type of interface: atm, ds1, ds3, ethernet-csmacd, frame-relay, other, propPointToPointSerial, propVirtual, Sonet, SWLoopback, v35

Mtu	The Maximum Transmission Unit (the size of the largest datagram supported).
Speed	The estimated speed of the interface in bps. For Ethernet interfaces only, if the value of Speed ends in 2, the interface is operating in full-duplex mode. If it ends in 0, the interface is operating in half-duplex mode. For G.SHDSL interfaces, this value is the same as *cmGshdslModemCurrDataRate*
PhysAddress	The physical address. The format is *byte1.byte2*..., where bytes are displayed as hex numbers.
AdminStatus	The status of a specified DSL or WAN port (see the Note above this table): up down Testing Operational packets cannot be passed in the Testing state. This object does not apply to the Ethernet port (whose PII is 1.2.1) since you cannot set that port's AdminStatus=down.
OperStatus	The actual operational status of the instance: up down Testing If you enter the PII in short format, Shelf.Slot.Port (such as 1.3.1), this object reports the status of the port at that location. If you also specify a DLCI (such as 1.3.1.900), the status of the corresponding link is reported. For a complete definition of OperStatus as it relates to this group, see *Operating Status* on page 20.
LastChange	The elapsed time since the system was reset, and the calendar date/time when the operator logged in to the system. The elapsed time is expressed as *day hour min sec*. This is the value of the SysUpTime counter. The calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.
InOctets	The total number of octets received on this interface.
InUcastPkts	The number of unicast packets delivered.
InNucastPkts	The number of non-unicast packets delivered.
InDiscards	The number of discarded inbound error free packets.
InErrors	The number of inbound packets with errors.
InUnknownProtos	The number of inbound packets with unknown Protocol ID numbers. For more information, see the IPProtocol object in *cmFilterTable* on page 114.
OutOctets	The number of octets transmitted on this interface.

OutUcastPkts	The number of unicast packets transmitted by the interface.
OutNUcastPkts	The number of non-unicast packets transmitted by the interface.
OutDiscards	The number of error-free outbound packets discarded.
OutErrors	The number of outbound packets with errors.
OutQLen	The length of the outbound packet queue.
Specific	The object identifier for the interface specific information.

(*) indicates object used as an index.

ifXTable

This group displays additional information about CE200 interfaces.

Syntax: *command* ifx [*index*]

Example: To retrieve the ifXTable entry for Frame Relay PVCs or ATM VCs, you must enter the number as part of the Index entry:

```
get ifx [1.3.1.16]
Group: ifXTable
Instance: [1.3.1.16]
Index                =  1.3.1.16
Name                 =  Link 1.3.1.16
LinkUpDownTrapEnable =  Enable
HighSpeed            =  0
ConnectorPresent     =  False
```

Objects:

*Index	The Permanent Interface Identifier (PII) of this interface.
Name	A string describing the type of interface (its ObjectClass from cmState) and its PII. For example, EthernetPort 1.2.1.0.
LinkUpDownTrapEnable	The capability of the interface to generate LinkUp and LinkDown traps: Enable or Disable. Default is *Disabled*
HighSpeed	The nominal value of the bandwidth of this interface in Mbps (the value is rounded up or down). A zero is displayed for G.SHDSL ports, interfaces with a BW of less than 500 Kbps, and sublayers that have no concept of bandwidth.
ConnectorPresent	Indicates whether the interface sublayer has a physical connector: True or False.

(*) indicates object used as an index.

imaGroupMappingTable

This is one of four groups dealing with Inverse Multiplexing over ATM (IMA). The imaGroupMappingTable is simple, read-only table that cross-references the ifIndex values of configured IMA groups to their imaGroupIndex values. This pre-mapped table makes look-ups by the SNMP managers more efficient by avoiding the need for sorts and searches.

You can view the contents of the table from the CopperCraft interface as detailed below.

Syntax: *command* imagroupMappingTable [PII of ifIndex]

Example: To view the IMA group PIIs for configured groups with their ifIndex

```
getall imagroupmap
```

Objects:

*GIndex	The IfIndex of the IMA group
Index	PII of the IMA Group

(*) indicates object used as an index.

imaGroupNumber

This scalar group is one of four groups dealing with Inverse Multiplexing over ATM (IMA). The imaGroupNumber is a simple, read-only table that reports the total number of configured IMA groups on this system.

You can view the contents of the table from the CopperCraft interface as detailed below.

Syntax: *command* imagroupnumber

Example:

```
get imagroupnumber
```

Objects:

Number	The number of configured IMA groups on this system

imaGroupTable

The imaGroupTable is the largest of the four groups dealing with Inverse Multiplexing over ATM (IMA). The imaGroupTable is used to configure groups and monitor and control their status and performance.

You can view the contents of the table from the CopperCraft interface as detailed below.

Syntax: *command* imagroupTable [Index] *Object 1 . . . Object n*

Example: To view the IMA group table for configured groups:

```
getall imagrouptable
```

Objects:	*Index	PII for this IMA group. The Index object provides the linkage mechanism between constituent IMA links (the physical DS1 links) and the group that comprises them.
	ifIndex	PII of the IMA Group, which becomes the destPII for VCLs routed through this WAN interface. The PII "Port" number, is the IMA "Group" designator, currently fixed at 41.
	RowStatus	Create, change, and delete conceptual rows (instances) on *imaGroupTable* entries. NOTE: Before you can make a new row Active, you must first configure the following objects: *MinNumTxLinks*, *MinNumRxLinks* and *TxImaId*. Setting RowStatus to 'active' has the effect of activating the Group Startup Procedure. The Group Startup Procedure uses provisioned links that have *imaLinkRowStatus* set to *active* and *imaLinkGroupIndex* set to the *imaGroupIndex* in this conceptual row. When RowStatus is not *active*, the Group State machine is in its 'Not Configured' state. VALUES: Active Create CreateAndWait Destroy NotInService NotReady
	NeState	The current operational state of the near-end IMA Group State Machine for this group. VALUES: notConfigured, startUp, startUpAck, configAbortUnsupportedM, configAbortIncompatibleSymmetry, configAbortOther, insufficientLinks, blocked, operational, configAbortUnsupportedImaVersion
	FeState	The current operational state of the far-end IMA Group State Machine for this group VALUES: notConfigured, startUp, startUpAck, configAbortUnsupportedM, configAbortIncompatibleSymmetry, configAbortOther, insufficientLinks, blocked, operational, configAbortUnsupportedImaVersion
	FailureStatus	The current failure status of the IMA Group (the reason why the GTSM is in the down state) VALUES: noFailure, startUpNe, startUpFe, invalidMValueNe, invalidMValueFe, failedAssymetricNe, failedAssymetricFe, insufficientLinksNe, insufficientLinksFe, blockedNe, blockedFe, otherFailure

Symmetry	The IMA Group Symmetry mode. This object can be set or its value changed only at the time the group is established. Also, if the rowstatus is *active*, it should be changed to *NotInService* before attempting a change in the Symmetry. VALUES: symmetricOperation asymmetricOperation symmetricConfiguration
MinNumTxLinks	The Minimum number of Active transmit links required for the IMA Group to be in the Operational State
MinNumRxLinks	The Minimum number of Active receive links required for the IMA Group to be in the Operational State
NeTxClkMode	The transmit clocking mode used by the near-end IMA Group. Values: Ctc (Default), itc (not supported)
FeTxClkMode	The transmit clocking mode used at the far end of the IMA group. Values: Ctc, Itc
TxTimingRefLink	The PII of the transmit timing reference link to be used by the near-end for IMA data cell clock recovery from the ATM layer. The value of Zero may be used if no link has been configured in this IMA group, or if the transmit timing reference link has not yet been selected.
RxTimingRefLink	The PII of the receive timing reference link to be used by the near-end for IMA data cell clock recovery from the ATM layer. The value of Zero may be used if no link has been configured in this IMA group, or if the receive timing reference link has not yet been detected.
LastChange	The time (calendar time) that the IMA group last changed operational state (i.e., value of NeState changed).
TxImaId	The IMA ID (PII) currently in use by the near-end IMA transmit function. This object can be set or its value changed only at the time the group is established.
RxImaId	The IMA ID currently in use by the near-end IMA receive function
TxFrameLength	The frame length to be used by the IMA group in the transmit direction. This object can be set or its value changed only at the time the group is established. Also, if the rowstatus is already *active*, it should be changed to *NotInService* before attempting a change in the TxFrameLength. In this release, only the default value, m128, is supported.
RxFrameLength	The IMA frame length as received from the remote IMA function. VALUES: m32, m64, m128, m256

DiffDelayMax	The maximum differential delay (in milliseconds) that will be tolerated among the links on this interface
LeastDelayLink	The ifIndex (PII) of the constituent link of this IMA group that has the smallest link propagation delay. The value of Zero may be used if no link has been configured in this IMA group, or if the link with the smallest link propagation delay has not yet been determined.
DiffDelayMaxObs	The observed differential delay (in milliseconds) between the link having the greatest and least amounts of propagation delay among the receive links configured in this IMA group.
AlphaValue	The 'alpha' value specifies the number of consecutive invalid IMA Control Protocol (ICP) cells to be detected before moving to the IMA Hunt state from the IMA Sync state
BetaValue	The 'beta' value specifies the number of consecutive errored IMA Control Protocol (ICP) cells to be detected before moving to the IMA Hunt state from the IMA Sync state
GammaValue	The 'gamma' value specifies the number of consecutive valid IMA Control Protocol (ICP) cells to be detected before moving to the IMASync state from the IMA PreSync state
RunningSeconds	Length of time (in seconds) this IMA group has been in the Operational state.
UnavailSeconds	Number of seconds that the IMA Group Traffic State Machine has been down (unavailable).
NeNumFailures	The number of times a near-end group failure (Config-Aborted, Insufficient-Links) has been reported since power-up or reboot
FeNumFailures	The number of far-end group failures (Config-Aborted-FE, Insufficient-Links-FE, Blocked-FE) reported since power-up or reboot.
TxAvailCellRate	The current cell rate (truncated value in cells per second) provided by this IMA group in the transmit direction, considering all transmit links in the Active state.
RxAvailCellRate	The current cell rate (truncated value in cells per second) provided by this IMA group in the receive direction, considering all active transmit links
NumTxCfgLinks	Number of links configured for transmit function in this IMA group. This attribute overwrites the value of the imaGroupNumRxActLinks when the IMA group is configured in the Symmetrical Configuration group symmetry mode.
NumRxCfgLinks	Number of links configured for receive function in this IMA group. This attribute is overwritten by the value of the imaGroupNumTxActLinks attribute when the IMA group is configured in the Symmetrical Configuration group symmetry mode.

NumTxActLinks	Number of configured Active-state transmit links in this IMA group.
NumRxActLinks	Number of configured Active-state receive links in this IMA group.
TestLinkIfIndex	PII of the interface designated as test link for the Test Pattern Procedure. (not supported in this release)
TestPattern	Numeric indication of which Tx Test Pattern will be used in an IMA group loopback operation (Test Patterns are not supported in this release).
TestProcStatus	Status of Test Pattern Procedure (Test Patterns are currently not supported) in this release
TxOamLabelValue	Integer designating the IMA OAM Label value transmitted by the near-end IMA Unit
RxOamLabelValue	Integer designating the IMA OAM Label value transmitted by the far-end IMA Unit. A value of 0 likely means that the IMA unit has not received an OAM Label from the far-end Unit

(*) indicates object used as an index.

imaLinkTable

Use the imaLinkTable to configure constituent links of IMA groups, to relate individual links to their groups, and to monitor status and performance of individual constituent links. This is one of four groups dealing with Inverse Multiplexing over ATM (IMA).

You can view the contents of the table from the CopperCraft interface as detailed below.

Syntax: *command* imaLinkTable [PII of the link]

Example: To successively view the configured T1/E1 IMA links, their group membership, and related performance parameters. enter:

```
getnext imalink
```

Objects:

*GIndex	PII of this port
RowStatus	Create, change, and delete operations; VALUES: Active, Create, CreateAndWait, Destroy, NotInService, NotReady
GroupIndex	PII of the IMA Group that this link is a member of
NeTxState	The current operational state of the far-end transmit link as reported via IMA Control Protocol (ICP) cells. VALUES: notInGroup, unusableNoGivenReason, unusableFault, unusableMisconnected, unusableInhibited, unusableFailed, usable, active
NeRxState	The current state of the near-end receive link. VALUES: notInGroup, unusableNoGivenReason, unusableFault, unusableMisconnected, unusableInhibited, unusableFailed, usable, active

FeTxState	The current link failure status of the far-end receive link as reported via IMA Control Protocol (ICP) cells. VALUES: notInGroup, unusableNoGivenReason, unusable-Fault, unusableMisconnected, unusableInhibited, unusable-Failed, usable, active
FeRxState	The current link failure status of the far-end receive link as reported via IMA Control Protocol (ICP) cells. VALUES: notInGroup, unusableNoGivenReason, unusable-Fault, unusableMisconnected, unusableInhibited, unusable-Failed, usable, active
NeRxFailureStatus	The current link failure status of the near-end receive link. VALUES: noFailure, ImaLinkFailure, LifFailure, LodsFailure, Misconnect, Blocked, Fault, FarEndTxLinkUnusable, FarEndRxLinkUnusable
FeRxFailureStatus	The current link failure status of the far-end receive link. VALUES: noFailure, ImaLinkFailure, LifFailure, LodsFailure, Misconnect, Blocked, Fault, FarEndTxLinkUnusable, FarEndRxLinkUnusable
TxLid	Outgoing Link Identification (LID) used on the link by the local end. This value has meaning only if the link is a constituent of an IMA group.
RxLid	Incoming LID used currently on the link by the remote end as reported by IMA Control Protocol (ICP) cells. This value has meaning only if the link is a constituent of an IMA group.
RelDelay	The latest measured delay (in milliseconds) on this link relative to the link in the same IMA group with the least delay
ImaViolations	IMA Control Protocol (ICP) violations: count of errored, invalid or missing ICP cells, except during SES-IMA or UAS-IMA conditions
OifAnomalies	The number of OIF (Out of IMA Frame) anomalies, except during SES-IMA and UAS-IMA conditions, at the near-end.
NeSevErroredSecs	Count of one second intervals containing ≥30% of the ICP cells counted as IV-IMAs, or one or more link defects (e.g., LOS, OOF/LOF, AIS, or LCD), LIF defects, or LODS defects, except during UAS-IMA
FeSevErroredSecs	Count of one second intervals containing one or more RDI-IMA defects, except during UAS-IMA-FE condition
NeUnavailSecs	Count of unavailable seconds at near-end. Unavail-ability begins at the onset of 10 contiguous SES-IMA, and ends at the onset of 10 contiguous seconds with no SES-IMA

FeUnavailSecs	Count of unavailable seconds at far-end. Unavailability begins at the onset of 10 contiguous SES-IMA-FE and ends at the onset of 10 contiguous seconds with no SES-IMA-FE
NeTxUnusableSecs	Count of Tx Unusable seconds at the near-end Tx Link State Machine (LSM).
NeRxUnusableSecs	Count of Rx Unusable seconds at the near-end Rx Link State Machine (LSM).
FeTxUnusableSecs	Tx Unusable seconds at far-end: count of seconds with Tx Unusable indications from the far-end Tx LSM.
FeRxUnusableSecs	Rx Unusable seconds at far-end: count of seconds with Rx Unusable indications from the far-end Rx LSM.
NeTxNumFailures	The number of times a near-end transmit failure alarm condition has been entered on this link (i.e., some form of implementation-specific transmit fault).
NeRxNumFailures	The number of times a near-end receive failure alarm condition has been entered on this link (i.e., LIF, LODS, RFI-IMA, Mis-Connected or some form of implementation-specific receive fault).
FeTxNumFailures	The number of times a far-end transmit failure alarm condition has been entered on this link (i.e., Tx-Unusable-FE).
FeRxNumFailures	The number of times a far-end receive failure alarm condition has been entered on this link (i.e., Rx-Unusable-FE).
TxStuffs	Count of stuff events inserted in the transmit direction.
RxStuffs	Count of stuff events detected in the receive direction.

(*) indicates object used as an index.

ip

This group displays datagram statistics for all IP packets transmitted or received over this system.

Syntax: *command* ip

Example:

```
get ip

Group: ip
Forwarding           =  forwarding
DefaultTTL           =  64
InReceives           =  212241
InHdrErrors          =  0
InAddrErrors         =  0
ForwDatagrams        =  119
InUnknownProtos      =  837
InDiscards           =  12832
InDelivers           =  198441
OutRequests          =  18709
OutDiscards          =  0
OutNoRoutes          =  0
ReasmTimeout         =  60
ReasmReqds           =  18
ReasmOKs             =  6
ReasmFails           =  0
FragOKs              =  6
FragFails            =  0
FragCreates          =  12
RoutingDiscards      =  0
```

Objects:

Forwarding	The status of this entity an IP gateway: forwarding—It is a gateway not_forwarding
DefaultTTL	The value of the “time to live” field (the number of hops allowed before the packet is discarded as undeliverable) for all packets initiated by the CE200.
InReceives	The total number of datagrams received, including errors.
InHdrErrors	The number of inbound datagrams discarded with header errors.
InAddrErrors	The number of inbound datagrams discarded due to invalid IP address.
ForwDatagrams	The number of datagrams for which this is not the final destination.
InUnknownProtos	The number of datagrams discarded for unknown protocol.
InDiscards	The number of good datagrams discarded for other reasons.
InDelivers	The number of good datagrams delivered.
OutRequests	The number datagrams requested for transmission.
OutDiscards	The number good outbound datagrams discarded.
OutNoRoutes	The number of datagrams discarded because no route was found.
ReasmTimeout	The number of maximum time fragments that are held for reassembly.

ReasmReqds	The number of fragments received.
ReasmOKs	The number of datagrams successfully reassembled.
ReasmFails	The number of failures detected by the IP reassembly algorithm.
FragOKs	The number of datagrams successfully fragmented.
FragFails	The number of datagrams for which fragmentation failed.
FragCreates	The number of datagram fragments generated.
RoutingDiscards	The number of error-free routing entries discarded.

ipAddrTable

This group displays the address table for a specified IP address of the interface associated with the entry.

Syntax: *command* ipaddr [*addr*]

Example: To query the interface with IP address 10.254.8.120:

```
get ipa [10.254.8.120]
Group: ipAddrTable
Instance: [10.254.8.120]
Addr                =  10.254.8.120
IfIndex             =  1.2.1.0
NetMask             =  255.255.255.0
BcastAddr           =  1
ReasmMaxSize        =  65535
```

Objects:

*Addr	The IP address for this entity.
IfIndex	The index of the interface entry associated with this IP address (same as the Index from the ifTable).
NetMask	The subnet mask associated with this entity.
BcastAddr	The IP broadcast address. A value of 1 indicates the Internet standard broadcast address of 255.255.255.255 (all bits are set to 1).
ReasmMaxSize	The maximum IP datagram size that can be reassembled from fragments.

(*) indicates object used as an index.

IPNetToMediaTable

Use this group to set/display the address translation table for the specified interface.

This group is virtually synonymous with the atTable (Address Translation) group. Configuration changes made in one group will be reflected in the other, and information displayed in one table applies to the other.

Syntax: *command* ipnettomedia [*ifindex, physaddress*]

Example: To query the interface at port 1, slot 2:

```
get ipn [1.2.1.0, 10.254.8.2]
Group: IPNetToMediaTable
Instance: [1.2.1.0, 10.254.8.2]
IfIndex              =  1.2.1.0
PhysAddress          =  0.a0.cc.53.1c.cb
NetAddress           =  10.254.8.2
Type                 =  Dynamic
```

Objects:

*IfIndex	The Permanent Interface Identifier (PII) of the interface associated with this entry.
PhysAddress	The physical address associated with IP address. The format is *byte1.byte2*..., where bytes are displayed as hex numbers.
*NetAddress	The IP address associated with this entity.
Type	The type of mapping: Other Invalid Dynamic Static

(*) indicates object used as an index.

ipRouteTable

Use this group to set and display routing instructions for a specified destination address. Only those route entries that are *directly* configured by the operator are stored in the Config file. Transient entries, such as those created from cmIface entries, are created automatically when needed, but are not stored in the Config file.

Up to 600 different routes can be defined in a single CE200 (Version 2.1 and later).

The CE200's Routes with Masks feature provides support for subnet masks to accommodate different networks. The IPRoute group will accept Mask objects to specify classless IP prefixes as destinations. But multiple routes to the same destination are still not allowed, so this feature does not actually provide full Classless Routing as that term is usually understood.

Syntax: *command* iproute [*dest*] *object_1...object_3*

Example 1: To set the next hop to IP address 10.254.8.120:

```
set ipr [10.254.8.0] next=10.254.8.120 type=direct
mask=255.255.255.0
```

Example 2: To query IP address 10.254.8.0:

```
get ipr [10.254.8.0]
Group: ipRouteTable
Instance: [10.254.8.0]
Dest                =  10.254.8.0
IfIndex             =  1.2.1.0
Metric1             =  0
Metric2             =  -1
Metric3             =  -1
Metric4             =  -1
NextHop             =  10.254.8.120
Type                =  Direct
Proto               =  local
Age                 =  761495
Mask                =  255.255.255.0
Metric5             =  -1
Info                =  0.0
```

Objects:

*Dest	The destination IP address for this route. It must be a valid, operational IP address.
IfIndex	The identifying index for the next hop of this route.
Metric1	The primary routing metric used for this route.
Metric2	An alternate routing metric.
Metric3	An alternate routing metric.
Metric4	An alternate routing metric.
NextHop	The IP address of the next hop of this route.
Type	The type of route: Other—None of the following Invalid—Invalidate (delete) this route. Direct—The route to a directly connected net/subnet. Indirect—The route to a non-local host/net/subnet. The only value that you can set is Invalid. The Direct and Indirect values are read only.
Proto	The routing mechanism by which the route was learned: other, local, netmgmt, icmp, egp, ggp, hello, rip, is-is, es-is, ciscoIgrp, bbnSpfIgp, ospf, bgp
Age	The elapsed time since this route was updated.
Mask	The route mask.
Metric5	An alternate routing metric.
Info	The MIB entry associated with this routing protocol.

(*) indicates object used as an index.

snmp

This group displays statistics (counts) relating to the SNMP agent (SNMP messages received or sent by the CE200). You can retrieve all statistics or statistics for one or more objects.

Syntax: *command* snmp *object_1*

Example 1: To display the number of incoming SNMP Set-Requests processed:

```
get sn insetreq
Group: snmp
InSetRequests        =  79
```

Example 2: To display all statistics for SNMP messages:

```
get sn
Group: snmp
InPkts               =  5080
OutPkts              =  5080
InBadVersions        =  0
InBadCommunityNames  =  0
InBadCommunityUses   =  0
InASNParseErrs       =  0
InTooBigs            =  0
InNoSuchNames        =  0
InBadValues          =  0
InReadOnlys          =  0
InGenErrs            =  0
InTotalReqVars       =  69090
InTotalSetVars       =  214
InGetRequests        =  4946
InGetNexts           =  30
InSetRequests        =  79
InGetResponses       =  0
InTraps              =  0
OutTooBigs           =  0
OutNoSuchNames       =  0
OutBadValues         =  0
OutGenErrs           =  4
OutGetRequests       =  0
OutGetNexts          =  0
OutSetRequests       =  0
OutGetResponses      =  5080
OutTraps             =  1930
EnableAuthenTraps    =  disable
```

Objects:

InPkts	The number of inbound SNMP messages delivered from the network.
OutPkts	The number of outbound SNMP messages delivered to the network.
InBadVersions	The number of inbound SNMP messages intended for an SNMP version not supported by the CE200.
InBadCommunityNames	The number of inbound SNMP messages using an unknown community name.
InBadCommunityUses	The number of inbound SNMP messages specifying an operation not allowed for the specified community.
InASNParseErrs	The number of ASN.1 or BER errors that occurred when decoding received SNMP messages.
InTooBigs	The number of SNMP Protocol Data Units (PDUs) received with “too big” error status.

InNoSuchNames	The number of SNMP PDUs received with “noSuchName” error.
InBadValues	The number of SNMP PDUs received with “BadValue” error.
InReadOnlys	The number of SNMP PDUs received with “ReadOnly” error status.
InGenErrs	The number of SNMP PDUs received with generic “genErr” error status.
InTotalReqVars	The number of SNMP MIB objects successfully changed in response to valid Set-Request PDUs.
InTotalSetVars	The number of SNMP MIB objects successfully retrieved in response to valid Get-Request and Get-Next PDUs.
InGetRequests	The number of SNMP Get-Requests accepted and processed by the agent.
InGetNexts	The number of SNMP GetNext-Requests processed.
InSetRequests	The number of SNMP Set-Requests processed.
InGetResponses	The number of SNMP Get-Response PDUs processed.
InTraps	The number of SNMP Trap PDUs processed.
OutTooBigs	The number of SNMP Protocol Data Units (PDUs) generated with “too big” errors.
OutNoSuchNames	The number of SNMP PDUs generated with “noSuchName” errors.
OutBadValues	The number of SNMP PDUs generated with “BadValue” errors.
OutGenErrs	The number of SNMP PDUs generated with generic errors.
OutGetRequests	The number of SNMP Get-Requests generated.
OutGetNexts	The number of SNMP GetNext-Requests generated.
OutSetRequests	The number of SNMP Set-Requests generated.
OutGetResponses	The number of SNMP Get-Response PDUs generated.
OutTraps	The number of SNMP Trap PDUs generated.
EnableAuthTraps	Indicates whether the SNMP agent is allowed to generate an authentication failure trap: enabled or disabled.

SonetFarEndLineCurrentTable
(alias: sntflcur)

This group allows you to display far-end performance statistics relating to a SONET line for the current time interval. The SonetFarEndLineCurrentTable is one of 12 supported groups from the SONET (RFC2558) MIB.

Syntax: *command* SonetFarEndLineCur [*pii*]

Example:

```
get sntflcur [1.3.1]

Group: SonetFarEndLineCurrentTable
Instance: [1.3.1.0]
ESs             =  0
SESs            =  0
CVs             =  0
UASs            =  1
```

Objects:

*IfIndex	The Permanent Interface Identifier of the SONET interface.
ESs	Number of Errored Seconds in the current 15-minute interval
SESs	Number of Severely Errored Seconds in the current 15-minute interval
CVs	Number of Coding Violations in the current 15-minute interval
UASs	Number of Unavailable Seconds in the current 15-minute interval

(*) indicates object used as an index.

SonetFarEndLineIntervalTable
(alias: sntflint)

This group allows you to display far-end performance statistics relating to a SONET line for a specified 15-minute interval. You can specify interval numbers up to 96, covering the previous 24 hours of operation.

Syntax: *command* SonetFarEndLineInt [*pii, number*]

The *number* object specifies how far back in time to look. The most recently completed interval has a value of 1. To see statistics from 30 minutes ago, enter 2; to see statistics from five hours ago enter 20; and so on.

Example:

```
get sntflint [1.3.1, 22]

Group: SonetFarEndLineIntervalTable
Instance: [1.3.1.0]
Number          =  22
ESs             =  0
SESs            =  0
CVs             =  0
UASs            =  0
ValidData       =  Valid
```

Objects:

*IfIndex	The Permanent Interface Identifier of the SONET interface.
*Number	The number of the interval: 1 to 96, where 1 is the most recently completed 15-minute interval, and 96 is the last interval.
ESs	Number of Errored Seconds in the specified 15-minute interval
SESs	Number of Severely Errored Seconds in the specified 15-minute interval
CVs	Number of Coding Violations in the specified 15-minute interval
UASs	Number of Unavailable Seconds in the specified 15-minute interval
ValidData	Indicates validity of the data for this interval: valid, invalid

(*) indicates object used as an index.

SonetFarEndPathCurrentTable
(alias: sntfpcur)

This group provides status information regarding the SONET/SDH path associated with the specified interface.

Syntax: *command* SonetFarEndLineCur [*pii*]

Example:

```
get sntfpcur [1.3.1]

Group: SonetFarEndLineCurrentTable
Instance: [1.3.1.0]
ESs               =  0
SESs              =  0
CVs               =  0
UASs              =  1
```

Objects:

*IfIndex	The Permanent Interface Identifier of the SONET interface.
ESs	Number of Errored Seconds in the current 15-minute interval
SESs	Number of Severely Errored Seconds in the current 15-minute interval
CVs	Number of Coding Violations in the current 15-minute interval
UASs	Number of Unavailable Seconds in the current 15-minute interval

(*) indicates object used as an index.

SonetFarEndPathIntervalTable
(alias: sntfpint)

This group allows you to display far-end performance statistics relating to a SONET/SDH path associated with the specified interface, for a specified 15-minute interval. You can specify interval numbers up to 96, covering the previous 24 hours of operation.

Syntax: *command* SonetFarEndPathInt [*pii, number*]

The *number* object specifies how far back in time to look. The most recently completed interval has a value of 1. To see statistics from 30 minutes ago, enter 2; to see statistics from five hours ago enter 20; and so on.

Example:

```
get sntfpint [1.3.1, 22]

Group: SonetFarEndPathIntervalTable
Instance: [1.3.1.0]
Number           =  22
ESs              =  0
SESs             =  0
CVs              =  0
UASs             =  0
ValidData        =  Valid
```

Objects:

*IfIndex	The Permanent Interface Identifier of the SONET interface.
*Number	The number of the interval: 1 to 96, where 1 is the most recently completed 15-minute interval, and 96 is the last interval.
ESs	Number of Errored Seconds in the specified 15-minute interval
SESs	Number of Severely Errored Seconds in the specified 15-minute interval
CVs	Number of Coding Violations in the specified 15-minute interval
UASs	Number of Unavailable Seconds in the specified 15-minute interval
ValidData	Indicates validity of the data for this interval: valid, invalid

(*) indicates object used as an index.

SonetLineCurrentTable
(alias: sntnlcur)

Controls configuration and provides status information regarding the SONET/SDH line associated with the specified interface.

Syntax: *command* SonetLineCur *[pii]*

Example:

```
get sntnlcur [1.3.1]

Group: SonetLineCurrentTable
Instance: [1.3.1.0]
ESs             =  0
SESs            =  0
CVs             =  0
UASs            =  1
```

Objects:

*IfIndex	The PII of the SONET interface.
Status	Current status of the SONET line Interface; the listed numeric value is a bitmap sum of three bit positions, thus allowing multiple defects to be reported. Note that bit position 1 (no defect) can only be reported if no other flag is present: 1: sonet LineNoDefect 2: sonetLineAIS 4: sonetLineRDI
ESs	Number of Errored Seconds in the current 15-minute interval
SESs	Number of Severely Errored Seconds in the current 15-minute interval
CVs	Number of Coding Violations in the current 15-minute interval
UASs	Number of Unavailable Seconds in the current 15-minute interval

(*) indicates object used as an index.

SonetLineIntervalTable
(alias: sntnlint)

This group displays near-end performance statistics relating to a SONET line for a specified 15-minute interval. Interval numbers are 1 to 96, covering the previous 24 hours of operation.

Syntax: *command* SonetLineInt [*pii*]

The *number* object specifies how far back to look. The most recently completed interval is 1. To see statistics from 30 minutes ago, enter 2; to see statistics from five hours ago enter 20; and so on.

Example:

```
get sntnlint [1.3.1, 22]

Group: SonetLineIntervalTable
Instance: [1.3.1.0]
Number          =  22
ESs             =  0
SESs            =  0
CVs             =  0
UASs            =  0
ValidData       =  Valid
```

Objects:

*IfIndex	The Permanent Interface Identifier of the SONET interface.
*Number	The number of the interval: 1 to 96, where 1 is the most recently completed 15-minute interval, and 96 is the last interval.
ESs	Number of Errored Seconds in the specified 15-minute interval
SESs	Number of Severely Errored Seconds in the specified 15-minute interval
CVs	Number of Coding Violations in the specified 15-minute interval
UASs	Number of Unavailable Seconds in the specified 15-minute interval
ValidData	Indicates validity of the data for this interval: valid, invalid

(*) indicates object used as an index.

SonetMediumTable
(alias: sntmed)

This group allows you to set and display configuration information about both the optical and electrical characteristics of the SONET/SDH interface and its status.

Syntax: *command* SonetMedium [*pii*]

Example:

```
get sntmed [1.3.1.48]

Group: SonetMediumTable
Instance: [1.3.1.48]
Type             =  SDH
TimeElapsed      =  550
ValidIntervals   =  96
LineCoding       =  SONETMediumNRZ
LineType         =  SONETMultiMode
CircuitIdentifier=  ....
InvalidIntervals =  0
LoopbackConfiguration=  SONETNoLoop
```

Objects:

*IfIndex	The Permanent Interface Identifier of the SONET interface.
Type	The type of signal to be supported on this interface: Sonet or SDH (default)
TimeElapsed	Amount of time in seconds since the beginning of the current 15-minute measurement interval: 0...900
ValidIntervals	Number of previous 15-minute intervals for which data has been collected and is available (0-96)
LineCoding	Lists the line coding used on this interface for the SONET/SDH optical signal. In this release, only sonetMediumNRZ is supported.

LineType	Lists the configured SONET/SDH line type for this interface. This release supports two types: sonetShortSingleMode and sonetMultiMode
CircuitIdentifier	A unique, user-defined string identifying this SONET/SDH circuit.
InvalidIntervals	Number of intervals for which no data is available
LoopbackConfiguration	Sets/displays the loopback state of the Interface: sonetNoLoop (not in loopbackstate), sonetFacility-Loop (received signal looped back through the corresponding xmtr in the return direction), sonetTerminalLoop (signal to be transmitted is directly connected to the associated incoming receiver)

(*) indicates object used as an index.

SonetPathCurrentTable
(alias: sntnpcur)

This group provides current status information, and performance/error statistics relative to the SONET/SDH path associated with the specified interface for the current 15-minute interval.

Syntax: *command* SonetPathCur [*pii*]

Example:

```
get sntnpcur [1.3.1]

Group: SonetLineCurrentTable
Instance: [1.3.1.0]
Width            =  2
Status           =  1
ESs              =  0
SESs             =  0
CVs              =  0
UASs             =  0
```

Objects:

*IfIndex	The Permanent Interface Identifier of the SONET interface.
Width	A numeric value which translates to the type of SONET/SDH path associated with this interface. In this release, the supported value is 2 (sts3cSTM1).
Status	Current status of the SONET/SDH path associated with this Interface; the listed numeric value is a bit-map sum of six bit positions, thus allowing multiple defects to be reported. Note that bit position 1 (no defect) can only be set if no other flag is present. 1: sonetPathNoDefect 2: sonetPathSTSLOP 4: sonetPathSTSAIS 8; sonetPathSTSRDI 16: sonetPathUnequipped 32: sonetPathSignalLabelMismatch
ESs	Number of Errored Seconds in the current 15-minute interval

SESs	Number of Severely Errored Seconds in the current 15-minute interval
CVs	Number of Coding Violations in the current 15-minute interval
UASs	Number of Unavailable Seconds in the current 15-minute interval

(*) indicates object used as an index.

SonetPathIntervalTable
(alias: sntnpint)

This group allows you to display far-end performance statistics relative to the SONET/SDH path associated with the specified interface for a 15-minute interval. You can specify interval numbers up to 96, covering the previous 24 hours of operation.

Syntax: *command* SonetPathInt [*pii, number*]

The *number* object specifies how far back in time to look. The most recently completed interval has a value of 1. To see statistics from 30 minutes ago, enter 2; to see statistics from five hours ago enter 20; and so on.

Example:

```
get sntnpint [1.3.1, 22]

Group: SonetPathIntervalTable
Instance: [1.3.1.0]
Number           =  22
ESs              =  0
SESs             =  0
CVs              =  0
UASs             =  0
ValidData        =  Valid
```

Objects:

*IfIndex	The Permanent Interface Identifier of the SONET interface.
*Number	The number of the interval: 1 to 96, where 1 is the most recently completed 15-minute interval, and 96 is the last interval.
ESs	Number of Errored Seconds in the specified 15-minute interval
SESs	Number of Severely Errored Seconds in the specified 15-minute interval
CVs	Number of Coding Violations in the specified 15-minute interval
UASs	Number of Unavailable Seconds in the specified 15-minute interval
ValidData	Indicates validity of the data for this interval: valid, invalid

(*) indicates object used as an index.

SonetSectionCurrentTable
(alias: sntnscur)

This group provides current status information, and performance/error statistics relative to SONET sections/frames for the current 15-minute interval on the specified SONET/SDH interface.

Syntax: *command* SonetSectionCur *[pii]*

Example:

```
get sntnscur [1.3.1]

Group: SonetSectionCurrentTable
Instance: [1.3.1.0]
Status           =  1
ESs              =  0
SESs             =  0
SEFSs            =  0
CVs              =  0
```

Objects:

*IfIndex	The Permanent Interface Identifier of the SONET interface.
Status	Current status of the SONET/SDH Section Interface; the listed numeric value is a bitmap sum of three bit positions, thus allowing multiple defects to be reported. Valid values are: 1, 2, 4 and 6. 1: sonet Section NoDefect 2: sonetSectionLOS 4: sonetSectionLOF
ESs	Number of Errored Seconds in the current 15-minute interval
SESs	Number of Severely Errored Seconds in the current 15-minute interval
SEFSs	Number of Severely Errored Framing Seconds in the current i5-minute interval
CVs	Number of Coding Violations in the current 15-minute interval

(*) indicates object used as an index.

SonetSectionIntervalTable
(alias: sntnsint)

Use this group to display performance statistics for SONET sections/frames for a specified 15-minute interval out of the previous 96 intervals, that is, the previous 24 hours of operation.

Syntax: *command* SonetSectionInt *[pii, number]*

where *number* specifies how far back in time to look. The most recently completed interval is 1. To view statistics from 30 minutes ago, enter 2; review data from five hours ago enter 20; and so on.

Example:

```
get sntnsint [1.3.1, 22]

Group: SonetSectionIntervalTable
Instance: [1.3.1.0]
Number          =  22
ESs             =  0
SESs            =  0
SEFSs           =  0
CVs             =  0
ValidData       =  Valid
```

Objects:

*IfIndex	PII of the Sonet interface.
*Number	The number of the interval: 1 to 96, where 1 is the most recently completed 15-minute interval, and 96 is the last interval.
ESs	Number of Errored Seconds in the specified 15-minute interval
SESs	Number of Severely Errored Seconds in the specified 15-minute interval
SEFSs	Number of Severely Errored Framing Seconds in the specified interval
CVs	Number of Coding Violations in the specified 15-minute interval
ValidData	Indicates validity of the data for this interval: valid, invalid

(*) indicates object used as an index.

SonetSESthreshSetTable
(alias: sntthresh)

This group displays the set of error thresholds that apply to the SONET interface for event/alarm generation and performance monitoring.

Syntax: *command* SonetSESthreshSet *[pii]*

Example:

```
get sntthresh

group: sonetsesthreshsettable
sonetSESthrssholdset=  ITU1995
```

Objects:

SonetSESthreshSet	The set of thresholds used for error monitoring. In this release, the supported set is: ITU1995

System

This group displays or configure information about the system.

Syntax: *command* system *object_1...object_3*

Example 1: To assign a name to the CE200:

```
set system name=cmtn
```

Example 2: To display system information about this CE200:

```
get sys
Group: System
Descr                =  CopperEdge200
ObjectID             =  1.3.6.1.4.1.1996.1.3
UpTime               =  8 day 19 hour 35 min 32.0 sec
                        (2000/06/05-15:07:45)
Contact              =  Copper Mountain Networks, Inc.
Name                 =  CMTN
Location             =  San Diego, CA 92121
Services             =  4
```

Objects:

Descr	A text description of the system.
ObjectID	The object identifier for the system.
UpTime	The elapsed time since the system was reset, and the calendar date/time when the operator logged in to the system. The elapsed time is expressed as *day hour min sec*. This is the value of the SysUpTime counter. The calendar date/time is expressed as *(yyyy/mm/dd-hh:mm:ss)*.
Contact	The person or organization to contact for information about this system.
Name	A locally assigned name for the system. The default is ce200. This name is displayed as part of the Craft prompt and is also picked up by other objects, such as CircuitID in cmDHCPTable. For this reason, we recommend you change the name from the default to something more meaningful.
Location	The physical location of this system.
Services	The set of services this system provides. The value is counted in the following way: $2^{(L-1)} + 2^{(L-1)}$..., where L is a functional layer: 1 Physical (such as repeaters) 2 Datalink/Subnetwork (such as bridges) 3 Internet (such as IP gateways) 4 End-to-end (such as IP hosts) 7 Applications (such as mail relays) For systems including OSI protocols, layers 5 and 6 may also be counted. For example, a system that primarily performs routing functions has the value 4, which is $2^{(3-1)}$. A system that is a host offering application services has a value of 72, which is $2^{(4-1)} + 2^{(7-1)}$.

tcp

This group displays information about TCP connections and connection criteria. Because information about actual TCP connections (such as the number of currently established connections) is transient, it remains valid only as long as all connections are in place.

Syntax: *command* tcp

Example:

```
get tcp

Group: tcp
RtoAlgorithm          =  vanj
RtoMin                =  1000
RtoMax                =  64000
MaxConn               =  -1
ActiveOpens           =  0
PassiveOpens          =  26
AttemptFails          =  0
EstabResets           =  5
CurrEstab             =  1
InSegs                =  11958
OutSegs               =  11945
RetransSegs           =  0
InErrs                =  3
OutRsts               =  10
```

Objects:

RtoAlgorithm	The algorithm used to determine the retransmit timeout (rto) value for retransmitting unacknowledged octets: constant—A constant rto rsre—MIL-STD-1778, Appendix B vanj—Van Jacobson's algorithm other—None of the above
RtoMin	The minimum value (expressed in milliseconds) for retransmit timeout.
RtoMax	The maximum value (expressed in milliseconds) for retransmit timeout.
MaxConn	The maximum number of TCP connections that can be supported by this CE200. A value of -1 indicates a dynamic maximum.
ActiveOpens	The number of times TCP connections have transitioned from CLOSED to SYN-SENT state.
PassiveOpens	The number of times TCP connections have transitioned from LISTEN to SYN-RCVD state.
AttemptFails	The number of times TCP connections have transitioned from either SYN-SENT or SYN-RCVD state to CLOSED state, plus the number of times connections have transitioned from SYN-RCVD state to LISTEN state.
EstabResets	The number of times TCP connections have transitioned from either ESTABLISHED or CLOSE-WAIT state to CLOSED state.
CurrEstab	The number of TCP connections with a current state of either ESTABLISHED or CLOSE-WAIT.

InSegs	The number of segments received, including any received in error.
OutSegs	The number of segments sent, including those on current connections but excluding those containing only retransmitted octets.
RetransSegs	The number of segments retransmitted; TCP segments with one or more previously transmitted octets.
InErrs	The number of TCP segments received in error (such as a bad checksum).
OutRsts	The number of TCP segments sent containing the Reset (RST) flag.

tcpConnTable

Use this group to configure or display the state of a specified current TCP connection. The configured state and the values of all objects in the group are transient; they remain valid only as long as the connection continues.

Syntax: *command* tcpconn *[localaddress, localport, remaddr, remport] object_1*

This group is unique in having four index objects. Note that the LocalPort and RemotePort do not refer to physical interfaces, but rather to IP port numbers (value: 0 to 65535).

Example 1: To see the configuration of a connection:

```
get tcpc [10.254.8.120, 23, 10.64.20.242, 2638]
Group: tcpConnTable
Instance: [10.254.8.120, 23, 10.64.20.242, 2638]
State                =  Established
LocalAddress         =  10.254.8.120
LocalPort            =  23
RemAddress           =  10.64.20.242
RemPort              =  2638
```

Example 2: To delete this connection:

```
set tcpc [10.254.8.120, 23, 10.64.20.242, 2638] state=delete
```

Objects:

State	The current state of this connection: Closed, Listen, SynSent, SynReceived, Established, FinWait1, FinWait2, CloseWait, LastAck, Closing, TimeWait, DeleteTCB The only value that a management workstation can set is DeleteTCB. This object will return BadValue if you try to set it to any other value.
*LocalAddress alias: laddr	The IP address of the local side of the connection. If the local side of the connection is in the listen state (that is, it will accept connections for any IP interface associated with the node), LocalAddress will display as 0.0.0.0.
*LocalPort alias: lport	The TCP Port number of the local side of the connection. **Note:** This is a logical assignment; not a physical address.
*RemAddress alias: raddr	The IP address of the remote side of the connection.
*RemPort alias: rport	The TCP Port number of the remote side of the connection.

(*) indicates object used as an index.

udp

This group displays statistics for UDP datagrams received by and sent from this facility.

Syntax: *command* udp

Example:

```
get udp

Group: udp
InDatagrams          =  5080
NoPorts              =  181786
InErrors             =  0
OutDatagrams         =  7007
```

Objects:

InDatagrams	The total number of UDP datagrams delivered to UDP users.
NoPorts	The number of UDP datagrams received for an unsupported UDP port.
InErrors	The number of UDP datagrams received with errors.
OutDatagrams	The number of UDP datagrams sent.

udpTable

This group displays information about UDP end points (UDP Listeners) that are currently accepting datagrams. The values of all objects in the group are transient; they remain valid only as long as the connection continues.

The LocalPort does *not* refer to a physical interface, but rather to an IP port number (0 to 65535).

Syntax: *command* udptable *[local_addr, local_port]*

Example: To query port 1024 at local address 127.0.0.1:

```
get udpt [127.0.0.1, 1024]
Group: udpTable
Instance: [127.0.0.1, 1024]
LocalAddress         =  127.0.0.1
LocalPort            =  1024
```

Objects:

*LocalAddress alias: laddr	The IP address of the local UDP Listener. If the listener will accept datagrams for any IP interface associated with the node, LocalAddress will display as 0.0.0.0.
*LocalPort alias: lport	The UDP Port number of the local UDP Listener.

(*) indicates object used as an index.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W