



HiPer Network Management Card

Network Application Card
SNMP and MIB Reference

Part No. 1.024.1661-00
Version Numbers
6.0, 6.1, 6.2



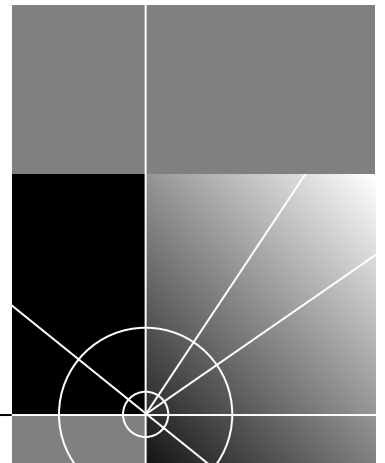


HiPer Network Management Card

Network Application Card
SNMP and MIB Reference
Versions 6.0, 6.1, 6.2

<http://www.3com.com/>

Part No. 1.024.1661-00



3Com Corporation
5400 Bayfront Plaza
Santa Clara, California
95052-8145

Copyright © 1999, 3Com Corporation. All rights reserved. No part of this documentation may be reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without written permission from 3Com Corporation.

3Com Corporation reserves the right to revise this documentation and to make changes in content from time to time without obligation on the part of 3Com Corporation to provide notification of such revision or change.

3Com Corporation provides this documentation without warranty of any kind, either implied or expressed, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. 3Com may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time.

UNITED STATES GOVERNMENT LEGENDS:

If you are a United States government agency, then this documentation and the software described herein are provided to you subject to the following:

United States Government Legend: All technical data and computer software is commercial in nature and developed solely at private expense. Software is delivered as Commercial Computer Software as defined in DFARS 252.227-7014 (June 1995) or as a commercial item as defined in FAR 2.101(a) and as such is provided with only such rights as are provided in 3Com's standard commercial license for the Software. Technical data is provided with limited rights only as provided in DFAR 252.227-7015 (Nov 1995) or FAR 52.227-14 (June 1987), whichever is applicable. You agree not to remove or deface any portion of any legend provided on any licensed program or documentation contained in, or delivered to you in conjunction with, this User Guide.

Unless otherwise indicated, 3Com registered trademarks are registered in the United States and may or may not be registered in other countries.

3Com, the 3Com logo, Total Control, and Total Control Manager are registered trademarks of 3Com Corporation.

Other brand and product names may be registered trademarks or trademarks of their respective holders.

YEAR 2000 INFORMATION:

For information on Year 2000 compliance and 3Com products, visit the 3Com Year 2000 web page:

<http://www.3Com.com/products/yr2000.html>

CONTENTS

ABOUT THIS REFERENCE

Conventions	ii
Notice Icons	ii
Text Conventions.....	ii
Compatibility.....	iii
Related Documentation	iv
The Total Control Documentation Library CD-ROM	iv
Contacting 3Com	v

PART I OVERVIEW OF SNMP AND MIBS

1 SNMP OVERVIEW

SNMP-based Network Management	1-1
Overview of Managed Networks	1-1
Network Monitoring with SNMP	1-2
Communicating with SNMP	1-2
SNMP Network Managers.....	1-2
SNMP Agents	1-4
Proxy Agents	1-4
Management Information Base.....	1-5
Managed Objects	1-5
Management Information Base	1-5
Industry Standards.....	1-6
SNMP Commands	1-6
Security	1-7
SNMP Community.....	1-8
Authentication Scheme	1-8

2 MANAGEMENT INFORMATION BASE OVERVIEW

Introduction.....	2-1
Overview	2-1
Managed Objects	2-2
MIB Trees.....	2-2
Groups within a MIB.....	2-4
Example of a Group within a MIB	2-4
Tables within a Group	2-4
Example: mdmIDTable from the MDM MIB	2-5
The mdmIDTable as viewed from a text editor (the MDM.txt file):	2-5
The mdmIDTable as viewed from a SNMP browser:	2-6
The mdmIDTable viewed from an SNMP browser	2-7
Description of the mdmIDTable from an SNMP browser:.....	2-7
Identifying Each Object in a MIB	2-8
Formats.....	2-8
Portions of the OID.....	2-9
The Basic Structure of a MIB Object	2-10
Example of the concise MIB format	2-12
Understanding the Syntax of MIB Objects	2-12
Abstract Syntax Notation One (ASN.1).....	2-12
Types.....	2-13
Values	2-14
A value assignment as a node	2-14
A value assignment as a variable	2-14
Macros.....	2-14
Types.....	2-15
Primitive Data Types.....	2-15
INTEGER.....	2-15
OCTET STRING	2-15
OBJECT IDENTIFIER.....	2-16
NULL	2-16
Defined Data Types.....	2-16
IpAddress	2-17
Counter	2-17

Gauge.....	2-17
Opaque.....	2-18
TimeTicks.....	2-18
SubTypes	2-18
Constructor Types.....	2-19
SEQUENCE	2-19
SEQUENCE OF.....	2-19
Sample Object Definitions	2-19
Integer	2-19
Octet String	2-20
Object Identifier	2-20
IpAddress.....	2-20
Counter	2-20
Gauge.....	2-21
TimeTicks.....	2-21
SEQUENCE OF.....	2-21
SEQUENCE	2-21
Identifying an Instance of an Object.....	2-22
SNMP Variables	2-22
Table Objects.....	2-23
Objects that are Not in Tables (Scalar Objects).....	2-24

PART II **USR (3COM) MIBs**

3 TOTAL CONTROL MIBs

Overview of the USR Enterprise	3-1
Standard MIBs	3-1
Proprietary MIBs	3-2
Experimental MIBs	3-4
Slots and Entities in the Chassis	3-4
Command Tables	3-5
A Typical Use of the Command Table	3-5
Additional Features of the Command Table	3-5
Supported MIB2 Groups	3-6

4 CHS MIB

Purpose of the MIB.....	4-1
Products using this MIB.....	4-1
Registration ID.....	4-1
Configuration Data.....	4-1
The Configuration Group	4-2
Chassis status.....	4-2
uchasFrontPanelLedStates	4-2
uchasFrontPanelLedColor	4-5
uchasNicStates	4-7
uchasFrontPanelLedStates2	4-7
uchasFrontPanelLedColor2	4-7
Tables	4-8
Chassis slots (uchasSlotTable).....	4-8
Chassis NACs	4-8
Chassis NICs.....	4-8
Additional information	4-9
Chassis Entities (uchasEntityTable).....	4-9
Chassis entities.....	4-9
Example	4-9
Computing the entity index number.....	4-10
uchasEntityTable.....	4-10
Additional information	4-10
Monitoring the Power Supply	4-10
uchasPowerSupplyTable.....	4-10
uchasPowerSupplyOutTable.....	4-11
Monitoring the Chassis Environment	4-11
uchasEnvironTable	4-11
Chassis Commands	4-11
uchasCmdTable	4-11
Chassis command function	4-11
Trap Enables	4-13
Traps	4-13
Other MIB-specific Information	4-14
Software Download to Chassis NACs.....	4-14

Overview of the Software Download 1 process to chassis NACs.....	4-14
Software Download to 386- and 486-NMCs.....	4-16
Software Download to the HiPer NMC	4-17
Known Types	4-17
Auto Response	4-17
Auto response group.....	4-17
Auto response slot table.....	4-18
Auto response timer table	4-18

5 CHS_TRAP MIB

Purpose of the MIB.....	5-1
Products using this MIB.....	5-1
Registration ID	5-1
Configuration Data.....	5-1
Tables	5-1
Traps	5-1
Other MIB-specific Information	5-2

6 DUAL TRUNK CARD MIBs

Overview	6-3
DS1-related MIBs.....	6-3
Checking Span Status with Dual Trunk Card MIBs.....	6-4
DT1 MIB	6-4
Purpose of the MIB.....	6-4
Products using this MIB	6-4
Registration ID.....	6-5
Configuration Data	6-5
MIB Tables.....	6-5
Dual T1 identification (dt1IdTable)	6-5
Status (dt1StatTable).....	6-5
NAC-level commands (dt1CmdTable).....	6-5
Trap enables (dt1TrapEnaTable).....	6-7
Traps	6-7
IDT1 MIB	6-8

Purpose of the MIB.....	6-8
Products using this MIB	6-8
Registration ID.....	6-8
Configuration Data	6-8
DNIS-based resource access (multiple objects).....	6-8
MIB Tables	6-8
Configuration (idt1CfgTable)	6-9
Call Routing (idt1CrTable).....	6-9
Reserved resource pool (idt1PITable)	6-9
Resource pool mapping assignment (idt1MdmRPATable)	6-9
Gateway to reserved resource pool mapping assignment (idt1GwyRPA)	6-9
Traps	6-9
DS1 MIB	6-9
Purpose of the MIB.....	6-9
Products using this MIB	6-9
Registration ID.....	6-10
Configuration Data	6-10
MIB Tables	6-10
Configuration (ds1ConfigTable).....	6-10
Current interval (ds1CurrentTable)	6-10
Total intervals (ds1TotalTable).....	6-10
Intervals (ds1IntervalTable).....	6-10
Traps	6-11
Other MIB-specific Information	6-11
UDS1 MIB.....	6-11
Purpose of the MIB.....	6-11
Products using this MIB	6-11
Registration ID.....	6-11
Configuration Data	6-11
MIB Tables	6-11
Configuration (uds1ConfigTable).....	6-12
Interval (uds1IntervalTable)	6-12
Current interval (uds1CurrentTable)	6-12
Total intervals (uds1TotalTable)	6-12

Status (uds1StatTable)	6-13
Commands (uds1CmdTable)	6-13
Trap enables (uds1TrapEnaTable)	6-14
Traps	6-14
DSO MIB	6-15
Purpose of the MIB	6-15
Products using this MIB	6-15
Registration ID	6-16
Configuration Data	6-16
MIB Tables	6-16
Configuration (dsOCfgTable)	6-16
Status (dsOStatTable)	6-16
Commands (dsOCmdTable)	6-16
Bulk access (dsOBulkAccessTable)	6-17
Traps	6-17
IDS0 MIB	6-17
Purpose of the MIB	6-17
Products using this MIB	6-17
Registration ID	6-17
Configuration Data	6-18
MIB Tables	6-18
Configuration (idsOCfgTable)	6-18
Status (idsOStatTable)	6-18
Commands (idsOCmdTable)	6-18
Bulk access (idsOBulkAccessTable)	6-19
Traps	6-19

7 FILE MIB

Purpose of the MIB	7-1
Products using this MIB	7-1
Registration ID	7-2
Configuration Data	7-2
Tables	7-2
Traps	7-2
Other MIB-specific Information	7-2

8 GW MIB

Purpose of the MIB.....	8-1
Products using this MIB.....	8-1
Registration ID.....	8-1
Configuration Data.....	8-1
Tables	8-1
Trap enables.....	8-1
Traps	8-2

9 HiPer DSP TRUNK MIBs

Overview	9-1
Checking Span Status with HiPer DSP	
Trunk MIBs	9-1
HDR2 MIB	9-2
Purpose of the MIB.....	9-2
Products using this MIB	9-2
Registration ID.....	9-2
Configuration Data	9-2
MIB Tables	9-2
Configuration (hdr2CfgTable).....	9-2
Call categories (hdr2CatMapTable)	9-3
Trap enables (hdr2TeTable).....	9-3
Traps	9-4
RDS0 MIB	9-5
Purpose of the MIB.....	9-5
Products using this MIB	9-5
Registration ID.....	9-5
Configuration Data	9-5
MIB Tables	9-5
Configuration (usrds0ConfigTable)	9-5
Status (usrds0StatTable).....	9-5
Commands (usrds0CmdTable)	9-5
RDS1 MIB	9-6
Purpose of the MIB.....	9-6
Products using this MIB	9-6

Registration ID	9-6
Configuration Data	9-6
MIB Tables	9-7
Configuration (usrds1ConfigTable)	9-7
Status (usrds1StatTable)	9-7
Commands (usrds1CmdTable)	9-8
Event configuration (usrds1EventCfgTable)	9-9
Traps	9-9
T1H MIB	9-10
Purpose of the MIB	9-10
Products using this MIB	9-10
Registration ID	9-11
Configuration Data	9-11
MIB Tables	9-11
Configuration (t1hCfgTable)	9-11
Call routing (t1hCrTable)	9-15
Commands (t1hCmdTable)	9-15
Trap enables (t1hTeTable)	9-19
Traps	9-19

10 HIST MIB

Purpose of the MIB	10-1
Products using this MIB	10-1
Registration ID	10-1
Configuration Data	10-1
MIB Tables	10-2
General Information about the Tables	10-2
Current tables	10-2
Interval tables	10-2
Traps	10-3

11 HDM MIB

Purpose of the MIB	11-1
Comparing the HDM MIB to the MDM MIB	11-1
Products using this MIB	11-2

Registration ID	11-2
Configuration Data	11-2
Configuring a Template	11-2
Changing a Channel's Configuration	11-3
Enabling traps	11-4
MIB Tables	11-4
Indexing	11-4
HDM MIB Tables	11-5
DTE Interface (hdmDiTable)	11-5
hdmDteNvramLock	11-5
AutoResponse (hdmArTable)	11-6
Traps	11-6

12 IMDM MIB

Purpose of the MIB	12-1
Products using this MIB	12-1
Registration ID	12-1
Configuration Data	12-1
MIB Tables	12-1
Call Control (imdmCcTable)	12-1
Line Interface (imdmLiTable)	12-2
Traps	12-2

13 IPGW MIB

Purpose of the MIB	13-1
Products using this MIB	13-1
Registration ID	13-1
Configuration Data	13-2
Tables	13-2
IPGW configuration (ipgwCfgTable)	13-2
IPGW commands (ipgwCmdTable)	13-2
Command function	13-2
Trap enables (ipgwTrapEnaTable)	13-3
Traps	13-3

14 MDM MIB

Purpose of the MIB.....	14-1
Products using this MIB.....	14-1
Registration ID.....	14-1
Configuration Data.....	14-1
MIB Tables.....	14-2
Modem Identification (mdmIDTable).....	14-2
Line Interface (mdmLiTable).....	14-3
Data Compression (mdmDcData Compression).....	14-3
ModemTests (mdmTf).....	14-3
mdmTfTable.....	14-3
mdmTfToneTable.....	14-4
mdmTfRspndrTable.....	14-5
DTE Interface (mdmDiTable).....	14-7
Signal Converter (mdmScTable).....	14-8
Call Control (mdmCcTable).....	14-8
Error Correction (mdmCcTable).....	14-8
mdmCcDtmTerminationTone.....	14-8
mdmCcDataOverVoice.....	14-9
Call Statistics (mdmCsTable).....	14-9
mdmCsCollectedDtmfDigits.....	14-9
Event counter (mdmEvTable).....	14-9
Event thresholds (mdmEtTable).....	14-10
Modem commands (mdmCdTable).....	14-11
Command function.....	14-11
Trap Enables (mdmTeTable).....	14-14
Link Security (mdmLsTable).....	14-15
Hub Security (mdmHsTable).....	14-15
Auto Response.....	14-15
Cellular Support (mdmCeTable).....	14-16
Modem Status (mdmStsTable).....	14-16
Modem Mapping (mdmMaTable).....	14-16
mdmMaChannelConfig.....	14-16
mdmMaChangeIndicator.....	14-16
Traps.....	14-16

15

NMC MIB

Purpose of the MIB.....	15-1
Products using this MIB.....	15-1
Registration ID	15-1
Configuration Data.....	15-1
NMC Configuration Group.....	15-2
The NMC's real time clock	15-2
Configuring the WAN (SLIP) port for dial access.....	15-2
TFTP timeout	15-4
Session ID/Call reference number for RADIUS logging	15-4
Configuring security and accounting servers.....	15-4
NMC Status Group	15-5
Test result bitmaps.....	15-5
MIB compatibility	15-6
Hub status LED.....	15-6
Auxiliary I/O objects.....	15-6
NMC Trap Group	15-6
Trap destination table	15-6
NMC Command Group	15-7
Configuring and saving settings on the NMC	15-7
NMC configuration commands.....	15-8
AUX I/O commands.....	15-11
Viewing the results of an NMC command	15-11
NMC Hub Security Group	15-11
NMC Trap Enable Group.....	15-11
NMC User Interface Configuration Group.....	15-12
NMC Authorized Access Group	15-12
IP address validation	15-13
NMC Network Time Protocol group.....	15-13
MIB Tables	15-14
Traps	15-14
Other MIB-specific Information	15-15
Configuring Security and Accounting, Servers.....	15-15
Log records	15-16
1-8 RADIUS Accounting	15-16

1-8 RADIUS Security	15-16
MD5 Calculation	15-17
Call statistics groups.....	15-17
DNS RADIUS Security/Accounting Host Name	15-17
User Server Selection Disabled	15-17
Recovery Mechanism uses Status-Server Request ...	15-18
Server trouble clearing.....	15-18
Status-Server Poll Interval	15-18
DNS Resolver Feature.....	15-18
RADIUS Security & Accounting Host Name	15-18
Basic DNS Configuration Suggestions	15-19
NMC Host Names	15-19
Refresh DNS Cache Suggestion	15-19
SNMP forwarding	15-19

16 **ADDITIONAL USR MIBS**

Packet Bus (PB) MIB	16-1
Products using this MIB	16-1
Registration ID.....	16-1
Tables	16-1
Traps	16-2
Packet Bus Datagram (PBDG) MIB	16-2
Products using this MIB	16-2
Registration ID.....	16-2
Tables	16-2
Traps	16-2
ULPB MIB	16-3
Products using this MIB	16-3
Registration ID.....	16-3
Tables	16-3
Traps	16-3
X.25 Interface MIB (UX25 MIB)	16-3
Products using this MIB	16-3
Registration ID.....	16-4
Tables	16-4

Traps	16-5
X.25 Gateway MIB	16-5
Products using this MIB	16-5
Registration ID	16-5
Tables	16-5
Traps	16-5
X.25 WAN MIB	16-6
Products using this MIB	16-6
Registration ID	16-6
Tables	16-6
Traps	16-6

PART III CONFIGURATION

17 CONFIGURING BASIC NMC CARD PARAMETERS

NMC Command Table Overview	17-1
What you Must Configure through the NMC	
User Interface	17-3
Setting Basic NMC System Configuration	17-3
Configuration notes	17-5
nmcPowerUpAutoCfgEnable	17-5
Setting User Interface Configuration	17-6
Configuring the Serial Line Internet Protocol Port	17-7
Configuring added cost features	17-8
Setting basic NMC security	17-9
Setting NMC community strings and the	
UI password	17-9
Setting the authorized access list	17-10

18 CONFIGURING NMC ACCOUNTING AND EVENT LOGGING

Overview	18-1
Configuring NMC Accounting	18-1
Configuring the NMC Logging Group	18-2

Configuring Logging Servers.....	18-3
Configuring RADIUS DNS Servers.....	18-4
Setting Server Traps	18-4

19 CONFIGURING NMC HUB SECURITY

Overview	19-1
Configuring NMC Hub Security	19-1
Configuring Hub Security Traps.....	19-4

20 CONFIGURING NMC NTP SERVERS

Overview	20-1
Configuring NTP servers	20-1
Setting server traps.....	20-2

21 CONFIGURING NMC DIAL-OUT

Overview	21-1
Configuring NMC Dial-out.....	21-1

PART IV MONITORING THE SYSTEM

22 TRAPS OVERVIEW

Understanding Events and Traps	22-1
Events	22-1
Traps	22-2
Why Use Traps?	22-2
Example of a USR (3Com) Trap	22-3
Understanding Alarms	22-3
Alarm Servers	22-3
Configuring Traps	22-4
Trap Enable and Disable.....	22-4
Configuration Options.....	22-4
Trap Destination Table.....	22-5

Transient Events.....	22-5
Chassis Trap.....	22-6
Types of Chassis Traps.....	22-6
NMC traps	22-6
T1, E1, and PRI-ISDN traps.....	22-6
Modem traps	22-6
Multiple Traps.....	22-6
SNMP Generic Traps	22-7
Chassis Trap MIB	22-7
Understanding the Chassis Trap MIB.....	22-7
Understanding the Structure of a SNMP Trap PDU	22-9
Event Messages in an SNMP Browser	22-10
Numbering the Events	22-10
Sequence number	22-10
Event number	22-11
Trouble Clearing Trap Packets.....	22-11
Example 1	22-11
Example 2	22-12

23 TRAP REFERENCE

Trap Table	23-1
------------------	------

24 MODEM DISCONNECT AND FAIL TO CONNECT REASON REFERENCE

Overview of Disconnect and Fail to Connect	24-1
Outgoing Calls	24-1
Incoming Calls.....	24-1
On Every Call.....	24-1
Modem Disconnect and Fail to Connect Reasons	24-3

25 AUTORESPONSE REFERENCE

Overview	25-1
Basic Operation	25-1

Events and Responses	25-2
Chassis Events	25-2
Generic Chassis-level Events	25-3
Chassis Slot-level Events	25-4
Chassis-level Responses	25-5
Possible Chassis-level Responses	25-5
Modem Events	25-10
Entity-level responses.....	25-12
Example of a entity-level response	25-12
Possible entity-level responses.....	25-12
Actions on DS1 Spans	25-14
Actions on DS0s.....	25-16
Off-Hook Response	25-17
Gateway Events.....	25-17
Configuring AutoResponse	25-18
Trouble Clearing AutoResponse	25-18
Software Download Errors	25-18
Simultaneous Event Occurrences	25-18
Script Errors.....	25-19
Invalid hex number in the script.....	25-19
Errors During AutoResponse Execution	25-19

INDEX

ABOUT THIS REFERENCE

This section provides an overview of this reference, describes reference conventions, tells you where to look for specific information, and lists other useful publications.

This reference is intended for network administrators with some training or experience working in a data center using Total Control equipment. Prior experience with SNMP is recommended. This reference is most useful if you are already familiar with using an SNMP browser.

This reference mentions several different network devices and software applications. However, it does not provide an extensive discussion of each piece of software and each device mentioned. Please refer to the documentation provided for a particular device or piece of software for a complete description.







3Com® ships release notes with some products. If the information in the release notes differs from the information in this reference, follow the instructions in the release notes.

This document was written with the assumption that the user has some knowledge of data processing, telecommunications, and networking.

Conventions

These tables list conventions that are used throughout this guide.

Notice Icons

Icon	Notice Type	Description
	Information note	Information that contains important features or instructions.
	Caution	Information to alert you to potential damage to a program, system, or device.
	Warning	Information to alert you to potential personal injury or fatality. May also alert you to potential electrical hazard.
	ESD	Information to alert you to take proper grounding precautions before handling a product.

Text Conventions

Convention	Description
Text represented as a screen display	This typeface represents displays that appear on your terminal screen, for example: Netlogin: This typeface also represents objects written in a MIB text (.txt) file.
Text represented as commands	This typeface represents commands that you enter, for example: setenv TCMHOME directory
Text represented as menu or sub-menu names.	This typeface represents all menu and sub-menu names within procedures, for example: On the File menu, click New.



Compatibility

You must use the 10/100 Ethernet Aux I/O Network Interface Card (NIC) with the HiPer NMC Network Access Card (NAC).

The NMC software is compatible with this NMC hardware:

NMC software version	NMC hardware
6.0	4MB 486-based NMC
6.1	16MB 486-based NMC
6.2	HiPer NMC

The NMC version 6.x software is compatible with these Total Control products:

Product	Version
HiPer DSP	2.0 (North America)
	2.0 (International)
	2.1
Quad modem	6.0 (Double Sided)
	6.1 (Single Sided)
Total Control Manager	Windows 6.0
	UNIX 6.0
Security and Accounting	Windows 6.0
	UNIX 6.0
T1-386	4.3
T1 / PRI	3.2
T1-186	3.5
E1 / PRI	3.1
E1-R2	1.3
NETServer	3.8 (Ethernet)
NETServer Manager	3.4
HiPer ARC	4.1 (Ethernet)
HiPer ARC Manager (HARM)	Windows 1.1
	UNIX 1.1
	HP 1.1
EdgeServer	1.6
EdgeServerPRO	1.6

Related Documentation

Complete HiPer NMC documentation is available on the Total Control Documentation Library CD-ROM. The documentation set includes:

- **HiPer NMC NAC Getting Started Guide** — this document contains installation and trouble clearing information for the HiPer NMC NAC
- **NMC Parameter Reference** — this document contains a complete tabular listing of NMC MIBs and their related data, plus a cross reference to Total Control Manager commands
- **NMC Product Reference** — this document provides information about new features, user interface configuration, and trouble clearing. A software download (SDL-2) explanation and instructions are included.
- **Software Download-2 (SDL-2) Instructions** — this document provides instructions for downloading software to the HiPer NMC NAC
- **10/100 Ethernet Aux I/O NIC Getting Started Guide** — this document contains installation and troubleshooting information for the 10/100 Ethernet Aux I/O NIC that is used with the HiPer NMC NAC

Additional documentation for 486-based NMC releases is available on the Total Control Documentation Library CD-ROM. Documentation for other previous NMC releases is available at <http://totalservice.3Com.com>.

The Total Control Documentation Library CD-ROM

The Total Control Documentation Library CD-ROM contains documentation for:

- Chassis and Fan Tray
- Network Management Card (NMC)
- Quad Modem Card
- NETServer
- Security and Accounting
- HiPer DSP Card
- HiPer ARC
- E1 Card
- T1 Card
- EdgeServer
- X.25 Card

Contacting 3Com

Call the appropriate toll free number listed below for technical support.



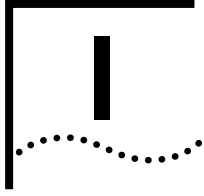
For European countries that do not have a toll free number listed, call +31 30 602 9900.

Country	Toll Free Number	Country	Toll Free Number
Austria	06 607468	Netherlands	0800 0227788
Belgium	0800 71429	Norway	800 11376
Canada	1800 2318770	Poland	00800 3111206
Denmark	800 17309	Portugal	0800 831416
Finland	0800 113153	South Africa	0800 995014
France	0800 917959	Spain	900 983125
Germany	0800 1821502	Sweden	020 795482
Hungary	00800 12813	Switzerland	0800 553072
Ireland	1800 553117	UK	0800 966197
Israel	0800 9453794	United States	1800 2318770
Italy	1678 79489	All Other Locations (Outside Europe)	1847 7976600

Refer to the Total Control Hub Documentation CD-ROM for more information regarding product warranty.



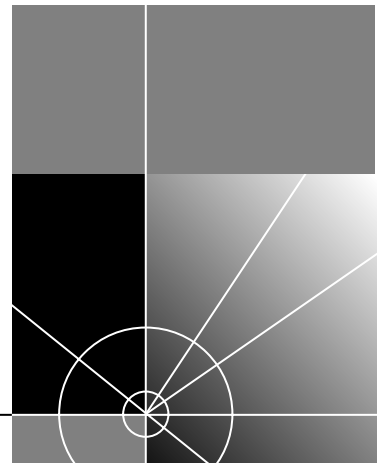
For information about Customer Service, including support, training, contracts, and documentation, visit our website at <http://totalservice.3com.com>

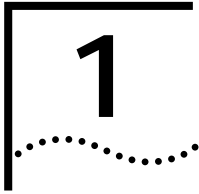


OVERVIEW OF SNMP AND MIBs

Chapter 1 SNMP Overview

Chapter 2 Management Information Base Overview





SNMP OVERVIEW

This chapter reviews basic SNMP concepts. It is not intended to be an exhaustive explanation of SNMP and MIBs. Refer to third-party books, websites, and public courses for additional information.

SNMP-based Network Management

To make sure all devices on a network will work together, systems must adhere to a common framework and language, also known as a protocol. Within the Internet network management framework (TCP/IP), that protocol is the Simple Network Management Protocol (SNMP).

In its simplest form, SNMP is a protocol designed to collect and exchange data between two network locations. SNMP uses the UDP as its transport layer and IP addresses as the network layer.

SNMP allows you to remotely manage a network by setting device values, polling devices, and monitoring network events. SNMP is comprised of three elements: the network manager, the agent(s), and the MIB(s). These topics will be discussed throughout this chapter.

Overview of Managed Networks

A managed network contains two main components: a network manager (software) and managed network elements (or resources). Network managers are responsible for running SNMP and the management application software that monitors and controls network elements. Network managers typically are installed on Management Stations (MS).

Managed network elements are devices such as hosts, hubs, bridges, and routers that contain agents that perform network management functions requested by the network manager. Agents are specialized software entities that are contained within the various managed network devices.

These network components are described in greater detail throughout this chapter.

**Network Monitoring
with SNMP**

Under SNMP, the network manager monitors the network's state. The network manager either polls a network element to get desired information or to set a controlling variable. The network element cannot poll or set any variables on the network manager.

A network element initiates communication with a network manager only when it needs to report an event or condition. If enabled when you configure the network, unsolicited messages called traps will be sent from a network element to inform the network manager of events and conditions.

**Communicating with
SNMP**

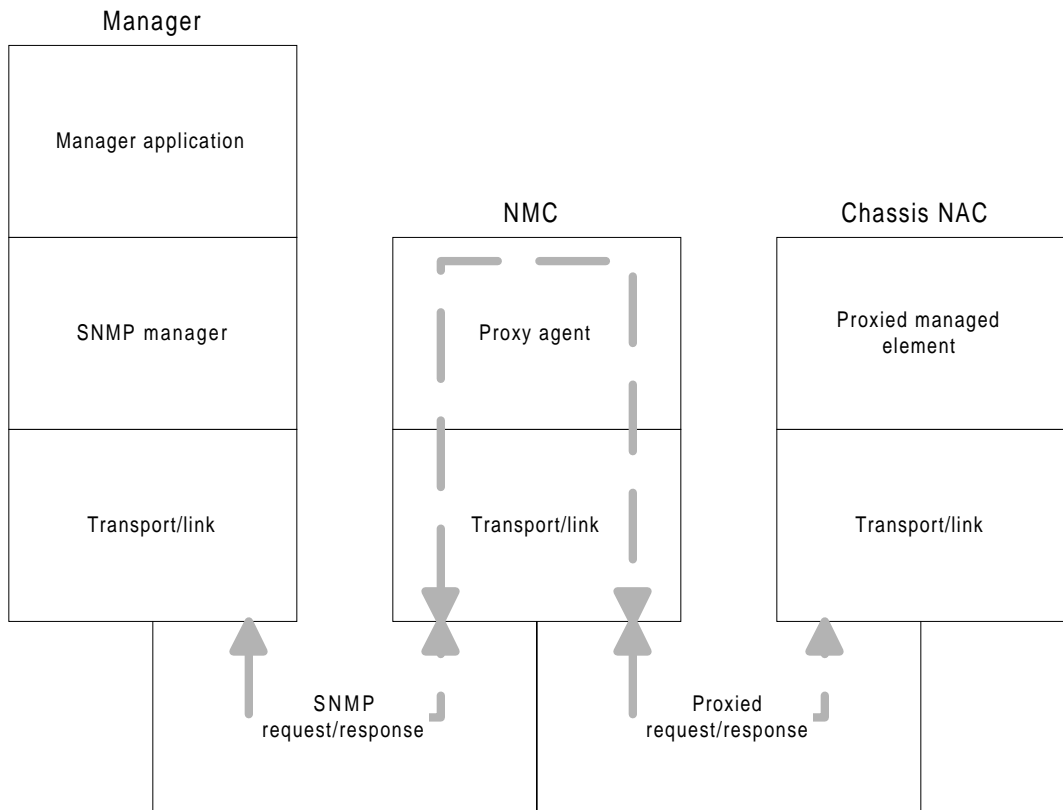
SNMP is the means by which the network management station and the agents communicate. Implementation of SNMP is relatively easy and requires little overhead on a network element. Within the Total Control chassis, Total Control Manager or another third-party network manager communicates with the Network Management Card (NMC) using SNMP version 1.

In summary, SNMP:

- Is an application-layer protocol designed to assist in the exchange of information between network devices
- Is the standard management protocol for multi-vendor IP networks
- Supports transaction-based queries that allow the protocol to format messages and to transmit information between agents and network managers
- Runs on top of the User Datagram Protocol (UDP) layer, offering a connectionless-mode service (this means that no previous connections exist prior to data transmission)

**SNMP Network
Managers**

A SNMP network manager is a collection of software applications and databases that control a group of network elements. The network manager is located on the network's host computer, typically called the Management Station (MS). Network managers range in complexity from simple shareware to highly complex and expensive software programs.



A network manager typically includes these five components:

- **User interface** — Enables you to enter management commands and receive either solicited or unsolicited agent responses. Total Control Manager is a Total Control user interface.
- **Management application software** — Assists in the analysis of network management information obtained from the agent processes. This software usually includes applications for data analysis and fault recovery.
- **Databases** — Contain all names, configuration, performance, topology, and audit data.

There is a distinction between MIBs and the other network manager databases. The MIB defines the variables of interest contained within all of the managed systems comprising the network. The other

databases are physical storage for instances of the MIB objects and other data collected and used for management.

- **SNMP engine** — The process that implements SNMP, exchanges SNMP messages, and allows the managing system to remotely access management information in the network elements.
- **Transport/Link** — Provides access to the underlying data communication paths.

SNMP Agents

To monitor and control network elements, SNMP uses specialized software modules called agents. These modules reside within a network device and serve as the interface between the device and the MS. In response to a network manager's request, agents access information about that device and return it to the network manager. The agent handles all MS requests and responses for that device.

This information is stored in the device within "managed objects". These managed objects include information about the device's hardware and software, configuration parameters, performance, traps, etc. Refer to "Management Information Base" in this chapter for additional information about managed objects.

Agents can also be programmed to generate trap messages to report events and unusual conditions to the network manager.

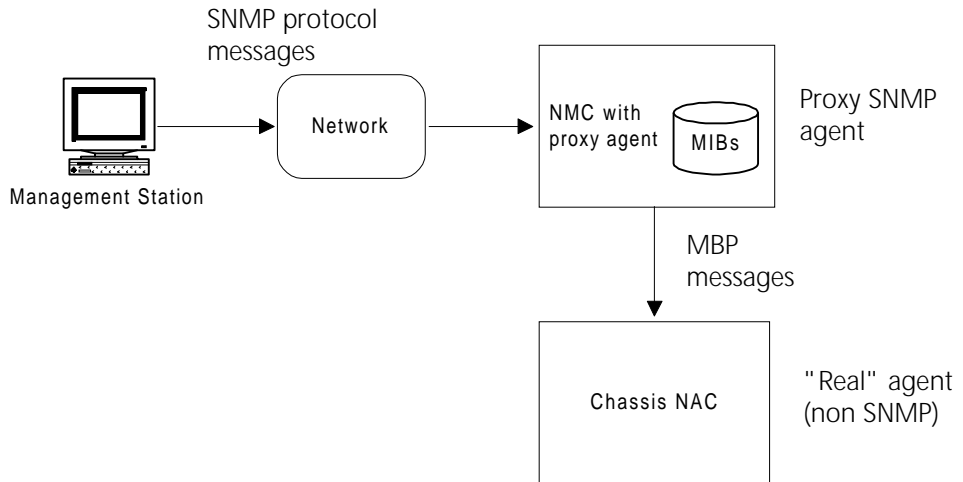
Proxy Agents

The SNMP architecture also uses proxy agents. A proxy agent acts as a management gateway for devices that do not run SNMP. The proxy agent translates between SNMP and the network elements' management protocol(s). It converts requests and event reports from one protocol and object format to another protocol and object format.

The Total Control chassis uses a proprietary protocol called the Management Bus Protocol (MBP) for internal communications between the NMC and the other Network Application cards (NACs). MBP allows for greater internal network efficiency than SNMP. Since the NACs are not running SNMP, they are unable to communicate directly with the network manager.

The NMC acts as a proxy agent for chassis NACs, providing the network manager with chassis management information. The network manager sends SNMP query requests to the NMC. As a proxy agent, the NMC then

uses MBP to query a card in the chassis. The card returns the report using MBP. The NMC then translates the report to SNMP and returns the data to the network manager.



Management Information Base

SNMP device management requires each device on the network to maintain a set of values that provides information about the device. To manage network elements, network managers can write to, read from, or read and write to these values. These values are defined as specific variables.

Managed Objects

Each variable is called a managed object. A managed object contains useful information about a managed element, such as the IP address, MAC layer address, and configuration setting. These objects define the different characteristics of the network.

Management Information Base

SNMP also requires that managed objects be logically accessible to the network manager. Logical accessibility means that management information must be stored somewhere, and that the information must be retrievable and modifiable using SNMP. This collection of managed objects resides in a virtual storage area, or database, referred to as the Management Information Base (MIB). This section provides an introduction to MIBs. Additional information is provided throughout the rest of this document.

The MIB is the critical element of an SNMP-based network. The MIB contains information about all of the objects under the control of a particular agent or network manager, specifying what requests and responses the agent can process and answer. Some MIBs simply list all of the objects they contain (including MIB-2 and the NMC MIB). Other MIBs that are used to manage more than one entity in a network device are organized into tables.

All managed objects for a device are contained in the MIB. These objects can be retrieved by a **get** command to provide the network manager with information about the device.

The network manager can control a managed device by sending a **set** command to an agent, requiring the device to change the value of one or more of its variables.

Industry Standards

Requests for Comment (RFCs) were created to control and publish changes to the protocols, standards, and rules of the Internet. Members of the Internet community who have a suggestion on how to change standards or wish to publish information of interest to the entire community can write an RFC. RFCs are approved and published by Network Working Group.

SNMP refers to a set of RFC standards:

- The SNMP protocol (RFC 1157)
- A set of data object definitions known as MIB-2 (RFC 1213)
- The SMI database architecture (RFC 1155)

The Structure of Management Information (SMI) is an industry standard that is defined in RFC 1155. SMI standards require that all managed objects must have a name that uniquely identifies the object, a syntax that defines the data type, and an encoding that specifies how the information in the object is transmitted.

SNMP Commands

SNMP uses simple request-response commands. Rather than defining a large set of commands, SNMP has five basic Protocol Data Units (PDUs) that it uses to manage network devices:

Command	Function
Get-Request	<p>This PDU requests the agent to return current MIB attribute values for a specific list of managed objects. The get allows you to read data.</p> <p>The MS issues one get request at a time to an agent, specifying a list containing the name of each object for which it wants information. In response, the agent sends a reply indicating the success or failure of the request. If the request was successful, the resulting message also contains the values of all requested objects.</p>
Get-Next-Request	<p>Get-Next is an efficient process to collect data from MIB tables. Use this PDU for traversing a MIB by moving sequentially through the MIB tree (and any tables the MIB may contain).</p> <p>Since object attributes are stored in lexicographical order (sequential order of integers), the result of the previous get-next request can be used as an argument in a subsequent get-next request. A network manager can then traverse a variable length table until it has extracted all information for each row in the table.</p> <p>This PDU is used both to read all rows in a given table, and to retrieve all management information available through the agent.</p>
Set-Request	<p>Use this PDU to request an agent to set the attribute values of selected objects. The network manager sends a list of object names and values to the agent.</p>
Get-Response	<p>The agent uses this PDU to respond to any of the three "Request" commands with a response or error message. The three commands are get-request, get-next-request, and set-request.</p>
Trap	<p>An agent sends a trap to report events and changes of state to the MS.</p>

All operations are conducted in this **get-set** format.

<div>Security</div>	<p>SNMP provides minimal security. The agent validates requests before responding. This prevents unauthorized network managers from viewing or changing the configuration of a device. The exact security process is dependent on your network needs and applications. All implementations use the value of the community string for authentication.</p>
---------------------	--

SNMP Community

An SNMP community is a relationship between an SNMP agent and one or more SNMP network managers. All members of a given community receive the same access privileges. The agent can be configured so that only network managers that are members of a known community can send requests and receive responses.

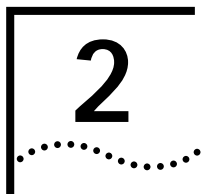
A community is identified by a community string. This string, which is typically human-readable, consists of octets (eight-bit bytes of binary or textual information).

Authentication Scheme

SNMP uses a trivial authentication scheme that places the community name in every SNMP request PDU. The community name is not encoded or encrypted.

If the agent recognizes the community name, the network manager is considered authenticated, and is given the SNMP access allowed to members of that community.

Two access modes are available to the network manager: read-only and read-write.



MANAGEMENT INFORMATION BASE OVERVIEW

This chapter reviews basic Management Information Base (MIB) concepts. It is not intended to be an exhaustive explanation of SNMP and MIBs. Refer to third-party books, websites, and public courses for additional information.

Introduction

The basic structure of any managed network includes a database containing information about all of the managed elements in the network. Within SNMP, this database is called a MIB.

A MIB is an integrated collection of information about all of the objects under the control of a particular device agent or manager. Examples of agents and managers include the Network Management Card (NMC) and Total Control Manager. MIB objects are abstract representations of device resources that can be managed. Examples of these devices include Quad modems and HiPer DSP modems. Objects are frequently arranged into related groups within each MIB.

The Total Control chassis is managed by more than 30 different MIBs. An agent may implement one or more groups from one or more MIBs. Each managed object is described by a standard specification that includes the object name, syntax, definition, access, status, and grouping.

Overview

A MIB defines:

- What variables or parameters are accessible to management
- How each value is identified
- How each value is encoded
- How each value is interpreted

The MIB database is split into many management groups, or areas. These areas are divided into a hierarchical, tree-like structure. Each area contains

objects for managing the network. The branches of the tree are used to divide the objects into related groups. Managed objects are represented as nodes in the tree structure. Each node is labeled with an integer and a brief text definition.

Managed Objects

A managed object is any piece of data about a device that an agent can access and report back to the Network Manager. Managed objects are also known as variables. These variables are referred to as *objects* within this document.

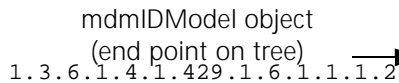
Related objects are contained within a single MIB. Objects can be set to change a configuration or read to provide information about chassis devices and interfaces. A Network Manager controls a managed chassis device (such as a modem) by sending a message to the NMC, requiring it to change the value of one or more of that device's objects.

MIB Trees

Each MIB object has a unique name. A hierarchical naming scheme uniquely identifies the managed objects contained within the MIB. The name structure is implemented through the use of a global naming tree. Each related group of objects is a branch of that tree. Each individual object is a leaf on the branch. Traveling from the root of the tree to an object is called "traversing the tree".

The name tree consists of a root (parent) node that is connected by branches to any number of child nodes. Each child node is provided with a label, which consists of a non-negative integer value and a brief descriptive name. Any child may have its own children (subordinates), which are also labeled. This process is extensible, meaning that it may continue to any depth and may contain any number of branches.

The path from the root to a particular object uniquely identifies that object. An object identifier (OID) is the sequence of non-negative integers assigned to each object that describes the path of traversal through the tree. Refer to "Table Description" in this chapter for additional information.



Groups within a MIB

A MIB organizes its many objects into related groups to facilitate and simplify management functions.

Example of a Group within a MIB

For example, groups within the MDM MIB include:

- Modem Identification — mdmID OBJECT IDENTIFIER ::= { mdm 1 }
- Line Interface — mdmLi OBJECT IDENTIFIER ::= { mdm 2 }
- Data Compression — mdmDc OBJECT IDENTIFIER ::= { mdm 3 }
- Test Function — mdmTf OBJECT IDENTIFIER ::= { mdm 4 }
- DTE Interface — mdmDi OBJECT IDENTIFIER ::= { mdm 5 }
- Signal Converter — mdmSc OBJECT IDENTIFIER ::= { mdm 6 }
- Call Control — mdmCc OBJECT IDENTIFIER ::= { mdm 7 }
- Error Correction — mdmEc OBJECT IDENTIFIER ::= { mdm 8 }
- Call Statistics — mdmCs OBJECT IDENTIFIER ::= { mdm 9 }
- Event — mdmEv OBJECT IDENTIFIER ::= { mdm 10 }
- Event Threshold — mdmEt OBJECT IDENTIFIER ::= { mdm 11 }
- Command — mdmCd OBJECT IDENTIFIER ::= { mdm 12 }
- Link Security — mdmLs OBJECT IDENTIFIER ::= { mdm 14 }
- Hub Security — mdmHs OBJECT IDENTIFIER ::= { mdm 15 }
- Cellular Support — mdmCe OBJECT IDENTIFIER ::= { mdm 17 }

Tables within a Group

Many MIBs contain related objects that are organized into tables. A MIB table is a collection of related data about a network device. Entries contain objects that define certain variables for that device.

Tables are only required when there is more than one occurrence of a logical group supported by a single agent (for example, a trap destination table). For example, within the MDM MIB, the mdmID group contains the mdmIDTable. This table contains a list of the various objects that identify a modem and describe its capabilities. The chassis also supports several tables that are not associated with any physical devices. These include the authorized access table in the NMC MIB and the interface table in MIB-2.

Example:
mdmIDTable from the
MDM MIB

This section uses the mdmIDTable as an example of how to read a MIB table text file and compares it to a MIB table generated through a SNMP browser.

The mdmIDTable as viewed from a text editor (the MDM.txt file):

Table name This section of the MIB identifies this portion of the MIB as a table definition and establishes an order for the objects within the table. The identifier ::= { mdmID 1 } states that this is the first table within the mdmID group.

```
mdmIDTable OBJECT-TYPE
SYNTAX SEQUENCE OF MdmIDEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A list of modem ID entries."
::= { mdmID 1 }
```

Table entry definition This section of the MIB text defines the table entries from the mdmID table.

```
mdmIDEntry OBJECT-TYPE
SYNTAX MdmIDEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"An Identification table entry, containing objects which
define the identity and capabilities of a modem."
INDEX { mdmIDIndex }
::= { mdmIDTable 1 }
```

Table index This section of the MIB defines the index for the table. The index represents each occurrence of a device. Each occurrence of the device is represented in the table as a separate row. The objects listed in the table index represent each column within an index row.

```
MdmIDEntry ::=
SEQUENCE {
mdmIDIndex
    INTEGER,
mdmIDModel
    INTEGER,
mdmIDCountry
    INTEGER,
```

```

mdmIDHardwareSerNum
    DisplayString,
mdmIDHardwareRev
    DisplayString,
mdmIDSupervisorSwRev
    DisplayString,
mdmIDDataPumpSwRev
    DisplayString,
mdmIDIoProcessorSwRev
    DisplayString,
mdmIDSupervisorDate
    DisplayString}

```

The mdmIDTable as viewed from a SNMP browser:

The following illustration shows the same mdmID table as captured from an SNMP browser. Note that each index number starts a new table row. Each table column represents the MIB object that is stated in the MIB as an index entry. Refer to the legend following this illustration for a description of the rows and columns.

In this case, modems are indexed by slot number and modem (creating a double-digit index number). Tables that are not related to modems usually contain a single-digit index number.

The table is a SEQUENCE of MdmIdEntry. Each row of the table is an entry of MdmIdEntry that contains eight objects. A subset of these eight objects, known as the INDEX element, is used to uniquely distinguish the table rows. Refer to later sections of this chapter for more information about SEQUENCE and INDEX.

The mdmIdTable viewed from an SNMP browser

	2	3	4	5	6	7	8	9	10
	↓	↓	↓	↓	↓	↓	↓	↓	↓
1 →	Index	Model	Country	HardwareSerNum	HardwareRev	SupervisorSwRev	DataPumpSwRev	IoProcessorSwRev	SupervisorDate
	1001	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1002	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1003	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1004	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1005	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1006	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1007	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1008	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1009	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1010	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1011	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1012	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1013	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1014	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1015	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1016	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98
	1017	hdm24Channel	northamerica	B18809V4	0.49.0	1.2.90	1.2.90		06/05/98

Description of the mdmIdTable from an SNMP browser:

Number	Description
1	Each row represents an occurrence of a device. In this table, each row represents a different HiPer DSP modem — 1007 represents HiPer DSP modem #7 in this chassis.
2	<i>mdmIDIndex</i>
3	<i>mdmIDModel</i>
4	<i>mdmIDCountry</i>
5	<i>mdmIDHardwareSerNum</i>
6	<i>mdmIDHardwareRev</i>
7	<i>mdmIDSupervisorSwRev</i>
8	<i>mdmIDDatPumpSwRev</i>
9	<i>mdmIDIoProcessorSwRev</i>
10	<i>mdmIDSupervisorDate</i>

Identifying Each Object in a MIB

Each MIB object is identified by an OID, which serves as an address to its location within the MIB tree. When OIDs are created from the tree, each branch is separated with a period. OIDs are organized into a hierarchy, and no two MIBs have the same OID. Each OID describes the path from the root of the MIB tree to the specific object.

Formats

OIDs precisely identify an object's name because they are globally unique. Each object is assigned to precisely one node, and no other object can be assigned to that same node. In theory, no two OIDs anywhere in existence will have the same value.

Several formats are used to express an OID. The simplest format lists the numeric integer values (node labels) found by traversing the tree, starting at the root and proceeding to the desired object. The integer values are listed as a string of integers, each separated by a period.

This is the OID for the MDM MIB object *mdmIDModel* :

```
1.3.6.1.4.1.429.1.6.1.1.1.2
```

An OID can also be expressed in full text, human-readable form. In this example, this is the OID for *mdmIDModel*:

```
iso.org.dod.internet.private.enterprises.usr.nas.mdm.mdmIDgroup.mdmIDTable.mdmIDModel
```

Note that the last value is the MIB object *mdmIDModel* .

Combining both the integer value and the text format, this is the OID for the same MDM MIB object (shown within the tree hierarchy from the enterprise level on):

```
usr OBJECT IDENTIFIER ::= { enterprises 429 }
  nas OBJECT IDENTIFIER ::= { usr 1 }
    mdm OBJECT IDENTIFIER ::= { nas 6 }
      mdmID OBJECT IDENTIFIER ::= { mdm 1 }
        mdmIDTable OBJECT IDENTIFIER ::= { mdmID 1 }
          mdmIDEntry OBJECT IDENTIFIER ::= { mdmIDTable 1 }
            mdmIDModel OBJECT IDENTIFIER ::= { mdmIDEntry 2 }
```



The phrase “::=” is an assignment statement, meaning that the given name is assigned to this particular number. In other words, mdmIDTable is assigned as mdmIDEntry #2. Again, note that the last value is the number of the MIB object mdmIDModel .

Portions of the OID

Different levels of the OID are assigned by different organizations. Top-level portions of the identifier are assigned by the Internet Engineering Task Force (IETF):

```
iso.org.dod.internet.management
```

which is concisely written as:

```
1.3.6.1.2
```

Vendors must define their own private branches to include objects to manage their products. Standardized vendor MIBs are placed in the “private” branch:

```
iso.org.dod.internet.private
```

which is concisely written as:

```
1.3.6.1.4
```

USR (3Com)-specific naming begins below the enterprises level. The Assigned Numbers Authority has reserved enterprise number 429 for USR (USR-heritage products now owned by 3Com). All of the enterprise-specific proprietary Total Control MIBs reside under this node in the enterprise subtree.



3Com-heritage products appear in the 3Com node, which is assigned to node 43.

For the MDM MIB object *mdmIDModel*, the USR (3Com)-specific portion of the object is:

```
usr.nas.mib.group.table.entry.object
```

which is concisely written for *mdmIDModel* as:

```
429.1.6.1.1.1.2
```

To further explain this point,

- 429 = the USR enterprise node (all nodes below this are specific to USR products only)
- 1 = the nas node (defining USR network access products)
- 6 = a specific MIB within the USR MIBs
- 1 = a specific group within the specified MIB

- 1 = a specific table within the specified group
- 1 = a specific entry within the specified table
- 2 = a specific object (endpoint of the tree)

The Basic Structure of a MIB Object

The Concise MIB Definitions document contained in RFC 1212 for SNMPv1 defines how to write MIBs. This results in conciseness and reduces the chances of errors.

A managed object in a MIB can include up to eight clauses:

- **SYNTAX** — The abstract syntax notation for the object type. This defines that data type that models the object. Syntax examples include OBJECT-TYPE, INTEGER, OBJECT IDENTIFIER, and OCTET STRING.

For example:

```
mdmIDModel OBJECT-TYPE
```

- **ACCESS** — Represents the way in which an instance of an object can be accessed. The options are read-only, read-write, or non-accessible (table and row objects must use “non-accessible”, since only scalar objects are accessible).

For example:

```
ACCESS read-only
```

- **STATUS** — Defines implementation support required for the object: mandatory, optional, obsolete, or deprecated (currently out of favor).

For example:

```
STATUS mandatory
```

- **DESCRIPTION** — A textual semantic description of the object type. It is of type DisplayString and must be enclosed in double quotes.

For example:

```
DESCRIPTION
  "Defines the model (ie. V.23, HST, etc.) of the modem
  represented by this row in the table."
```

- **REFERENCE** — (Optional) Contains a textual cross-reference to an object defined in another MIB module. It is useful when interpreting a MIB produced by another organization. It is of type DisplayString.

- **INDEX** — Used only if the object type corresponds to a conceptual row in a MIB table. It defines instance identification information for the particular object type. The index specifies the column of object(s) in the table that will unambiguously define and distinguish a conceptual row in a table.

For example:

```
MdmIDEntry ::=
SEQUENCE {
mdmIDIndex
    INTEGER,
mdmIDModel
    INTEGER,
mdmIDCountry
    INTEGER,
mdmIDHardwareSerNum
    DisplayString,
mdmIDHardwareRev
    DisplayString,
mdmIDSupervisorSwRev
    DisplayString,
mdmIDDataPumpSwRev
    DisplayString,
mdmIDIoProcessorSwRev
    DisplayString,
mdmIDSupervisorDate
    DisplayString
}
```

- **DEFVAL** — (Optional) Defines a default value that may be assigned to the object when a new instance is created by an agent. The default value must correspond to the SYNTAX clause for the object.
- **VALUE NOTATION (object name)** — The name used to access the object via SNMP. It is of type ObjectName.

For example:

```
::= { mdmIDEntry 2 }
```

Example of the concise MIB format

Using the concise MIB format, the OBJECT-TYPE macro for the object *mdmIDModel* looks like this:



Descriptions were added in the left margin of this document to aid in understanding.

Object `mdmIDModel OBJECT-TYPE`

Syntax `SYNTAX INTEGER{`

Access `ACCESS read-only`

Status `STATUS mandatory`

Description `DESCRIPTION`
 `"Defines the model (ie. V.23, HST, etc.) of the modem`
 `represented by this row in the table."`

Identifier `::= { mdmIDEntry 2 }`

In this example, *mdmIDModel* is the object descriptor corresponding to the object type being defined. The keyword OBJECT-TYPE specifies the name of the macro being invoked. The keywords SYNTAX, ACCESS, STATUS, and DESCRIPTION are clauses. The VALUE NOTATION "`::= { mdmIDEntry 2 }`" specifies the object identifier of the object type.

Understanding the Syntax of MIB Objects

Each object's name is represented by an OID. The format of each object has a standard syntax. Because operating platforms vary greatly, ISO has issued a standard for defining a MIB called Abstract Syntax Notation (ASN.1) that defines the type of information that must be included in each MIB object.

Abstract Syntax Notation One (ASN.1)

SNMP standardizes the minimal amount of information provided by each network device via MIB objects. ASN.1 (pronounced ASN-dot-one) is an Application Programming Language (API) used to interchange data between network devices. ASN.1 defines an abstract syntax that provides a common data format for the network. The abstract syntax defines precisely the data structure for each object, independent of any platform-specific data structures.

Three kinds of objects, which are described later in this section, are defined using ASN.1:

- Types
- Values
- Macros

The information that type, value, and macro objects represent are carried within the SNMP message headers.

Each of these kinds of objects is named using an ASN.1-defined word. ASN.1 uses an alphabetic case format to indicate to which kind of object the word refers:

- Types begin with an uppercase letter (for example, Gauge)
- Values begin with a lowercase letter (for example, internet)
- Macros consist entirely of uppercase letters (for example, OBJECT-TYPE)

To reduce repetition, types, values, and macros can be exported from one MIB and imported into another. Objects that are imported from another MIB are listed at the beginning of the MIB and indicated by the word **IMPORTS**.

Types

A type is the basic unit of data structure definitions within ASN.1.

Types are used within each MIB to outline the structure for data to be used in this module of code. Each new type begins with an uppercase letter.

The ASN.1 syntax used to define a type consists of:

```
Type ::= TYPE
```

where Type is the name of a newly-defined type that makes use of one of the simple ASN.1 types: INTEGER, BIT STRING, OCTET STRING, OBJECT IDENTIFIER, and NULL

Types are described in more detail later in this chapter.

Values

A value is an instance of a given type. In SNMP all values begin with a lowercase letter.

The ASN.1 syntax used to define a value consists of:

```
variable Type ::= VALUE
```

where Type specifies the type associated with the variable and VALUE is the value assigned to the variable.

A value assignment can take two different forms: as a node, or as a variable. In either case, "value" only represents the OID used to retrieve information about the object.

A value assignment as a node

In the example:

```
ds1 OBJECT IDENTIFIER ::= {experimental 2}
```

The object (`experimental 2`) is simply a node and carries no information about the object. In this case, the value is not a variable.

A value assignment as a variable

In the example:

```
ds1TimeElapsed OBJECT-TYPE ::= { ds1ConfigEntry 3 }
```

the object name (`ds1TimeElapsed`) is a variable that has a single OID associated with it. The value is that of the OID node that specifically defines this variable. In this case, "value" refers to the unique OID needed to retrieve information from this variable. The value of the variable is (`ds1ConfigEntry 3`).

Macros

Macros are routines that change the grammar of ASN.1. Macros are used to define a group of related data types that are used to define objects.

The names of macros are in all uppercase letters. For example, SNMP defines the OBJECT-TYPE macro. The OBJECT-TYPE macro specifies the syntax of an object type.

The ASN.1 syntax that specifies a macro is described later in this chapter.

Types THE SMI categorizes the ASN.1 types into three groups. These data types define the management information that is contained within the MIB. The three categories of data types are:

- Primitive types (also called simple types)
- Defined types
- Constructor types

Primitive Data Types SMI permits the use of four designated ASN.1 primitive (also called simple) types:

- INTEGER (enumerated, integer-bitstring, Integer32)
- OCTET STRING (DisplayString, octet-bitstring)
- OBJECT IDENTIFIER (ObjectName)
- NULL (applies to SNMPv1 only)

INTEGER

An INTEGER type is used to specify a value whose range may include both positive and negative numbers.

Although ASN.1 does not specify any bit-level precision, by convention, a value of type INTEGER must fit into 32 bits. Integers are indicated by the notation INTEGER.

Integers usually contain associated human-readable names for the values that might be available to instances of that data type. For example:

```
mdmDcDataCompression OBJECT-TYPE
SYNTAX INTEGER{
    none(1) ,
    autoEnable(2) ,
    enable(3) ,
    mnpWoCompression(4)
```

where values 1–4 provide human-readable phrases. In other cases, they may be noted simply as:

```
mdmScTxPwrLv1 OBJECT-TYPE
SYNTAX INTEGER (0..31)
```

OCTET STRING

An OCTET STRING data type is used to specify octets (eight-bit bytes) of binary or textual information.

The OCTET STRING data type values are an ordered sequence of zero or more octets. Each octet may be assigned a value from 0 to 255. Octet strings are indicated by the notation OCTET STRING.

Octet strings typically are noted like this:

```
mdmCsFreqProbeData OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..60))
```

OBJECT IDENTIFIER

The OBJECT IDENTIFIER (OID) is a data type that names or identifies an object in the OBJECT IDENTIFIER tree, regardless of the object's meaning. It uniquely identifies an item that has assigned object identifier values.

For example:

```
usr          usr OBJECT IDENTIFIER ::= { enterprises 429 }
nas          nas OBJECT IDENTIFIER ::= { usr 1 }
mdm          mdm OBJECT IDENTIFIER ::= { nas 6 }
mdmID        mdmID OBJECT IDENTIFIER ::= { mdm 1 }
mdmIDTable   mdmIDTable OBJECT IDENTIFIER ::= { mdmID 1 }
mdmIDEntry   mdmIDEntry OBJECT IDENTIFIER ::= { mdmIDTable 1 }
mdmIDModel   mdmIDModel OBJECT IDENTIFIER ::= { mdmIDEntry 2 }
```

Refer to the section in this chapter titled *"Identifying each object in a MIB"* for additional information about OIDs.

NULL

The NULL data type serves as a placeholder. It might be used for an object that communicates information without the need to equate the object to any particular value.

Defined Data Types

Defined data types refer to the special data types defined in the SMI document. They are also called application-wide types. The currently-defined types for SNMP v1 are:

- APPLICATION 0 (IpAddress)
- APPLICATION 1 (Counter)
- APPLICATION 2 (Gauge)
- APPLICATION 4 (TimeTicks)

■ APPLICATION 5 (Opaque)

IpAddress

This data type identifies a 32-bit IP address. It is represented by a 4-byte OCTET STRING in network order. Each octet (byte) in the string identifies eight bits of an IP address.

```
IpAddress ::=
    [APPLICATION 0] - - in network-byte order
    IMPLICIT OCTET STRING (SIZE (4))
```

Counter

This defined type represents a non-negative integer that increases until it reaches a maximum value, at which it returns to zero (wraps) and begins increasing again. The maximum value allowed for a counter is $(2^{32})-1$, or 4,294,967,295 decimal.

The counter is one of the most common types used in objects. Typical applications count the number of packets or octets sent and received. These objects can be used to determine how many events have occurred over time. Counters are also useful for tracking things like slow response times and numerous errors.

```
Counter ::=
    [APPLICATION 1]
    IMPLICIT INTEGER (0..4294967295)
```

Gauge

This data type represents a non-negative integer, which may increase or decrease, but which stops and holds at its maximum value. A gauge value does not wrap. The maximum value allowed for a gauge is $(2^{32})-1$, or 4,294,967,295 decimal.

Gauges are most commonly attached to values that represent a time interval. Typically, a gauge measures the current value of an entity (such as the number of stored packets). These objects can also be used to report the difference in a value from the beginning to the end of a time interval. Objects that report their values as gauges typically provide information about network resource use.

```
Gauge ::=
    [APPLICATION 2]
    IMPLICIT INTEGER (0..4294967295)
```


Opaque

Opaque is not currently used in any Internet standard MIB. This data type may circumvent the limited types allowed in the SMI by providing the capability to pass any ASN.1 syntax.

TimeTicks

This defined type represents a non-negative integer that counts the time in hundredths of a second since some specified event. When object types are defined in the MIB using this data type, the actual description of the object type identifies the referenced event.

TimeTicks represent a relative timer; time is measured relative to a network event (such as a reboot). While these values make sense within the network, they usually cannot be compared to an outside timer value.

TimeTicks are useful for tracking the amount of time since a trap was generated. They can also be used for checking system up time or if a host was reset.

```
TimeTicks ::=
    [APPLICATION 3]
    IMPLICIT INTEGER (0..4294967295)
```

SubTypes A SubType further limits a type. This is the syntax for specifying a subtype:

```
SubType ::=
    Type (<subtype information>)
```

Six kinds of subtypes exist in ASN.1, not all of which are used. The most common subtypes are the value-range subtypes, as in this example:

```
Counter ::=
    [APPLICATION 1]
    IMPLICIT INTEGER (0..4294967295)
```

and the size-range subtype, as in this example:

```
IpAddress ::=
    [APPLICATION 0] - - in network-byte order
    IMPLICIT OCTET STRING (SIZE (4))
```

In these examples, Counter and IpAddress are the subtypes, while INTEGER and OCTET STRING are the types. The information within parenthesis limits the range of the type specified.

Constructor Types

Constructor types (also called aggregate types) use primitive types or other constructor types recursively to create more complex data types, typically either lists or tables of information objects. Constructor types are SEQUENCE and SEQUENCE OF.

SEQUENCE

The SEQUENCE type is used to construct lists of information objects. It commonly defines an object that corresponds to a row of one or more primitive (simple) objects in a conceptual MIB table.

In this example, each data item of the sequence is a different ASN.1 primitive type:

```
rowObject ::=
SEQUENCE { <primitive type 1>, ..., <primitive type N> }
```

SEQUENCE OF

The SEQUENCE OF type is used to construct tables of information objects where each <rowObject> is of the same ASN.1 type. For example:

```
SEQUENCE OF <rowObject>
```

Similar to the dynamic array in a programming language, there may be zero or more of the same type of entries. It commonly defines an object that corresponds to a conceptual table containing one or more rows, which, in turn, are defined using the SEQUENCE type.

Sample Object Definitions

These are examples of object types found within the USR (3Com) MIBs.

Integer

```
mdmLiDialPause OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Duration in seconds for the pause(') option in the dial
command and the pause between command re-executions(> and A>)
Default = 2. Equates to the modem's S8 register."
::= { mdmLiEntry 2 }
```

Octet String

```
mdmCsFreqProbeData OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..60))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Probe frequency."
::= { mdmCsEntry 75 }
```

Object Identifier

```
uchasSlotModuleType OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The type of physical module contained in this slot of the
chassis. A value of uchasSlotEmpty indicates an empty
slot. A value of uchasSlotUnknown indicates that the type
of module is unknown."
::= { uchasSlotEntry 2 }
```

IpAddress

```
ipgwCfgLocalIpAdrs OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Specifies the local IP address used by the IP Gateway Card
for management related access."
::= { ipgwCfgEntry 2 }
```

Counter

```
uchasPhysicalChanges OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The number of physical changes that have occurred in the
chassis since the agent was warm/cold started. This
includes additions and removal of physical modules."
::= { uchasConfig 4 }
```

Gauge

```
dsx1TotalESs
OBJECT-TYPE
SYNTAX Gauge
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The number of Errored Seconds encountered by a
DS1 interface in the previous 24 hour interval"
::= { dsx1TotalEntry 2 }
```

TimeTicks

```
uchasSlotLastChange OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The value of MIB-II's sysUpTime (in the agent supporting
this chassis MIB) at which a module was last inserted or
removed from this slot. If no module has been inserted or
removed from this slot since the last time the network
management card was re-initialized, then this object has a
zero value."
::= { uchasSlotEntry 3 }
```

SEQUENCE OF

```
mdmIDTable OBJECT-TYPE
SYNTAX SEQUENCE OF MdmIDEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A list of modem ID entries."
::= { mdmID 1 }
```

SEQUENCE

```
mdmIDEntry OBJECT-TYPE
SYNTAX MdmIDEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"An Identification table entry, containing objects which
define the identity and capabilities of a modem."
INDEX { mdmIDIndex }
::= { mdmIDTable 1 }
```

```

MdmIDEntry ::=
SEQUENCE {
mdmIDIndex
    INTEGER,
mdmIDModel
    INTEGER,
mdmIDCountry
    INTEGER,
mdmIDHardwareSerNum
    DisplayString,
mdmIDHardwareRev
    DisplayString,
mdmIDSupervisorSwRev
    DisplayString,
mdmIDDataPumpSwRev
    DisplayString,
mdmIDIoProcessorSwRev
    DisplayString,
mdmIDSupervisorDate
    DisplayString}

```

Identifying an Instance of an Object

From a theoretical perspective, objects are only templates for the actual instances of the managed objects that reside at the network entity. When a MS accesses an object through SNMP, it **gets** or **sets** data for a specific occurrence of that object. This specific occurrence is called an instance. To locate a specific instance, each object name contains an appendix suffix called an instance identifier.

SNMP Variables

The name of a managed object and its associated instance information is called an SNMP variable. Variables are the arguments to SNMP operations and designate to which instance of management information the operation should be applied.

The names of variables are encoded to look like OIDs. The prefix of the variable is the managed object's OID. The suffix of the variable is an encoding of the managed object's instance. The actual format of the suffix depend upon the object type. Typically, the instance identifier is a unique number or a "0".

Table Objects The instance identifier for an object in a table is determined by appending the value of the index element for a particular table row to the objects OID.

For example, in this table, the highlighted row is indexed as 1007, representing the seventh modem. An instance of any object in this row would contain the OID with ".7" appended to it. Again using the example of *mdmIdModel*, the instance of this object for the seventh modem is *mdmIdTable.7*, or 1.3.6.1.4.1.429.1.6.1.1.1.2.7.

Index	Model	Country	HardwareSerNum	HardwareRev	SupervisorSwRev	DataPumpSwRev	IoProcessorSwRev	SupervisorDate
1001	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1002	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1003	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1004	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1005	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1006	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1007	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1008	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1009	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1010	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1011	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1012	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1013	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1014	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1015	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1016	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98
1017	hdm24Channel	northamerica	B1880SV4	0.49.0	1.2.98	1.2.98		06/05/98

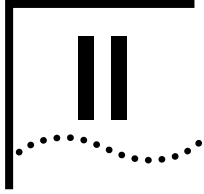
**Objects that are Not
in Tables (Scalar
Objects)**

If the object is not part of a table, then there is exactly one instance of the object type within a particular device. There is no ambiguity between an object type and the actual instance of the object.

For example, objects for the NMC card are not contained within a table. Instances of these objects are identified by appending ".0" to the object name. To identify the only instance of an object, the object instance would be expressed as *object.0*

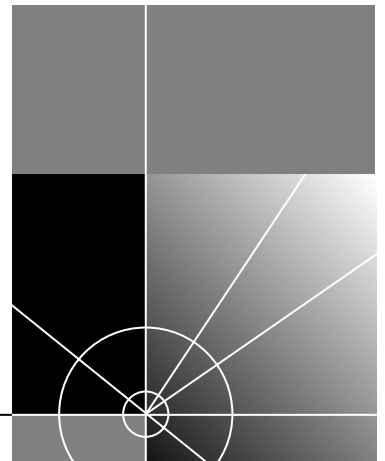
For example:

An instance of *nmcUiCfgLanIpAddr* will always be *nmcUiCfgLanIpAddr.0*, or 1.3.6.1.4.429.1.2.8.1.0.



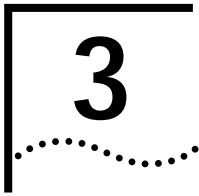
USR (3CoM) MIBs

Chapter 3	Total Control MIBs
Chapter 4	CHS MIB
Chapter 5	CHS_TRAP MIB
Chapter 6	Dual Trunk Card MIBs
Chapter 7	File MIB
Chapter 8	GW MIB
Chapter 9	HiPer DSP Trunk MIBs
Chapter 10	HIST MIB
Chapter 11	HDM MIB
Chapter 12	IMDM MIB
Chapter 13	IPGW MIB
Chapter 14	MDM MIB (continued)



Chapter 15 NMC MIB

Chapter 16 Additional USR MIBS



TOTAL CONTROL MIBs

This chapter contains an introduction to USR (3Com) MIBs and introduces the purpose of each MIB. Refer to the following chapters in this section for additional information.

Overview of the USR Enterprise

Three types of MIBs are supported by the Total Control chassis:

- Standard MIBs
- Proprietary enterprise-specific MIBs
- Experimental MIBs

Standard MIBs

For the purposes of this document, standard MIBs are those that were assigned an RFC number by the Internet Engineering Task Force (IETF).

Total Control supports these standard MIBs:

MIB filename	Function
MIB2.MIB (MIB-II)	Defines various protocol and administrative information (full standard classification)
RFC1406.MIB	<p>This is a proposed IETF standard that provides management of the DS1 on the HiPer DSP NAC. It is also known as the DSX1 MIB. The IETF named it as DSX1.MIB to avoid confusion with DS1 (RFC 1232). Not all objects in DSX1 are compatible with DS1.MIB.</p> <p>General information:</p> <ul style="list-style-type: none">■ The object <i>dsx1LineStatus</i> is good for a general status query for the HiPer DSP■ Use <i>dsx1IntervalTable</i>, <i>dsx1CurrentTable</i>, and <i>dsx1TotalTable</i> to query HiPer DSP interval statistics and error counters <p>This MIB is not a standard MIB.</p>

Proprietary MIBs

Proprietary enterprise-specific MIBs were developed by USR/3Com to define the managed objects for devices installed in a Total Control chassis. For example, since there are no approved standards for modems, USR/3Com developed a proprietary modem MIB (MDM MIB) prior to the creation of the IETF standard (RFC 1696).

The Assigned Numbers Authority has reserved enterprise number 429 for the USR branch of the MIB tree. All of the enterprise-specific proprietary Total Control MIBs reside under this node in the enterprise subtree.



The USR MIBs in enterprise 429 are limited to the Total Control chassis MIBs. Enterprise 429 does not include MIBs for other 3Com products.

The USR branch is further divided into branches that support USR (3Com) products. The Total Control chassis is supported by the NAS and SysOID branches. NAS includes all Total Control-specific MIBs. SysOID includes the MIB-2 sysobject IDs for any USR (3Com) product supporting MIB-2.

These USR/3Com proprietary MIBs are supported by NMC version 6.0/6.1/6.2. Refer to the following chapters for additional information about these MIBs.

MIB filename	Function
ANIC.MIB	Contains parameters used for detailed management of the analog NIC. Basic management is done via the CHS MIB. ANIC provides the ability to set "ring no answer" events and busy out analog phone lines.
CHS.MIB	Provides tables that describe the devices in the chassis. Allows the network manager to take generic actions on cards in the chassis (such as resetting a card or downloading software to the card). Indexes from the entity table are used to access tables in all other supported MIBs. You do not need to compile or integrate this MIB in order to use the AutoResponse feature.
CHS_TRAP.MIB	Defines SNMP traps for fault management. This MIB must be compiled and integrated into a trap manager in order to process traps.
DSO.MIB	Contains parameters that affect the operation of each timeslot in a T1 card.

MIB filename	Function (continued)
DT1.MIB	Contains parameters that affect the hardware operation of a T1 card.
GW.MIB	Provides access to IP address and Auto Response definitions for the NETServer.
EDGE.MIB	Contains parameters for managing a CDMA card. This MIB does not apply to the TCS 3.5 system release.
HDM.MIB	Contains the HDM templates used to set up mapped configurations of the HiPer DSP.
HDR2.MIB	Contains objects for R2 signalling events on the HiPer DSP card.
HDVI.MIB	This MIB does not apply to the TCS 3.5 system release.
HIST.MIB	Contains objects that support chassis-wide information logging and reporting for Quad and HiPer DSP modems.
IDS0.MIB	Contains parameters that affect the operation of each timeslot in a Primary Rate ISDN card.
IDT1.MIB	Contains parameters for basic management of a Primary Rate ISDN card on a Quad modem.
IMDM.MIB	Contains the parameters used to manage the ISDN capability of a Quad and a HiPer DSP modem.
IPGW.MIB	Provides the management information for the HiPer ARC.
IWFG.MIB	This MIB provides definitions for management Frame Relay parameters for the Analog Wireless Interworking Function Gateway (also known as WAS).
MDM.MIB	Defines all objects for a 3Com modem.
NMC.MIB	Contains NMC-specific information.
PB.MIB	Contains parameters used for configuring and monitoring the status of packet bus sessions.
PBDG.MIB	Contains parameters used for monitoring packet bus datagram traffic performance (non-session-related traffic used to set up PRI calls).
RDS0.MIB	Contains parameters that affect the operation of each timeslot in a HiPer DSP NAC. This MIB is similar to DS0.
RDS1.MIB	Provides management of a DS1 on a HiPer DSP NAC. This MIB is similar to UDS1.
RMDM.MIB	Not supported.
T1H.MIB	Contains parameters used to manage the DS1 on a HiPer DSP NAC. This MIB is similar to IDT1.
UDS1.MIB	Contains DS1- specific parameters not supported in the standard DS1 MIB, including span-level management objects for Dual trunk and E1/PRI applications.

MIB filename	Function (continued)
ULPB.MIB	Contains parameters used for LAPB management on the X.25 PAD.
UX25.MIB	Contains parameters used for Packet Level Protocol management on the X.25 PAD.
X25G.MIB	Contains parameters used for basic management of the X.25 PAD gateway.
X25W.MIB	Contains parameters for management of the two serial interfaces supported on the original X.25 NAC.

Experimental MIBs

While a working group is developing a new MIB module, it is assigned a temporary object identifier under the experimental subtree. If the MIB module is ever standardized, it is assigned to the standards subtree and given a new prefix. MIB modules under the experimental subtree do not ship with products. The NMC supports these MIBs for internal development only.

RFC 1232, also known as DS1.MIB, is an experimental MIB. The IETF no longer tends to publish RFCs on the experimental branch, so the use of experimental MIBs will be extremely limited in the future.

This experimental MIB was supported before RFC 1406 was published:

DS1.MIB	This is a very early proposed IETF standard that provides management of Dual T1/PRI (based on RFC 1232). It is declared in the IETF experimental branch.
---------	--

Slots and Entities in the Chassis

The Total Control chassis MIB (CHS MIB) is an enterprise-specific MIB that includes two main tables: the first defines the slots in the chassis and the second defines the manageable entities in the chassis.

The indexes to the entity table are used as the basis for the indexes to other MIB tables for a given entity. For example, if a modem entity has an entity index of 2001, then that same entity number is used to index the modem MIB (MDM MIB) tables to **get** and **set** objects in the modem MIB for that modem. Refer to the chapter in this document titled “CHS MIB” for additional information.

Command Tables

Understanding the concept of the command table is necessary to fully use the management capabilities within the Total Control chassis. The concept of this table is borrowed from RFC 1229 (*Generic Extensions to the Interfaces Table*), but was enhanced to fit USR's (3Com's) needs.

The command table provides the ability to use an SNMP **set** request to activate an action or command. The command table also provides for parameter passing and result polling. A mechanism for handling multiple management stations (MS) is implemented to help in situations when more than one Network Manager is trying to issue a command at the same time.

A Typical Use of the Command Table

This is an example of a typical use of the command table.

- 1 An SNMP **set** command is issued to set the *Function* object. Issuing this **set** command also internally sets the read-only *ReqId* object to the value of the request-id field in the SNMP **set** PDU.
- 2 If the NMC can handle the command, the read-only *Result* object is set to *InProgress*(3). This activates the command.
- 3 If the command is not supported by the particular device for which the command was sent, the *Result* object is set to *notSupported*(4), and the command is ignored.
- 4 If the device is in a state such that it is unable to execute the command at the particular time that it was issued, the *Result* object is set to *unableToRun*(5), and the command is ignored.
- 5 The MS issues SNMP **get** requests to repeatedly poll the Result and Code of the requested command.
 - If the Result changes to success, the requested command is complete.
 - If the Result changes to *aborted*(6) or *failed*(7), the *Code* object reveals a further description of the reason for which the requested command could not be completed.

Additional Features of the Command Table

Use the *Param* object to pass parameters to the command table. Most commands do not require additional parameters. If used, this parameter must accompany the *Function* object in a single **set** PDU.

Use the *Force* object to force the command to be issued in cases where the proxy agent would otherwise warn the Network Manager that it is *unAbleToRun(5)*. An example of this feature is if the requested function is to reset a modem that is currently in a connected state. If used, this parameter must accompany the *Function* object in a single **set** PDU.

The command table provides for an multiple management stations. Two objects, *MgtStationId* and *ReqId*, are used for this purpose. *MgtStationId* is a writable object that is provided for optional use by the MS. If desired, it can set this object to a unique value in the same PDU as is used to set *Function*, *Param*, and *Force*. The value of the request-id field of the SNMP **set** (used to set the *Function*) is stored internally and returned upon **getting** the read-only *ReqId*. While **getting** the command *Result* and *Code*, it makes sense to **get** *MgtStationId* and *ReqId* in the same PDU to guarantee that the results received by the MS are related to the command it had requested (and not the command requested by some other MS).

The proxy agent typically will not allow a second command of the same type as one already in progress for a given device. After completion of one command, it is possible that a second command could be issued and received before a MS was able to **get** the results of the first command. Using *MgtStationId* and *ReqId* eliminates this situation.

Supported MIB-2 Groups

The 3Com MIBs are not fully compliant with the latest version of MIB-2 (RFC 1213). The NMC is compliant with RFC 1158.

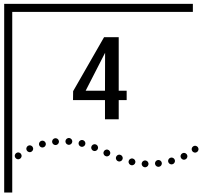
RFC 1213 (MIB-2) defines an enhanced version of MIB1. MIB1 (RFC 1156) was the first definition of the core Internet standard MIB. It specified the minimal number of objects thought to be necessary for management of TCP/IP-based networks. MIB-2 contains all of the technical content of MIB1 and expands it while retaining backward compatibility.

MIB-2 organizes its objects into related groups to facilitate and simplify management functions. Every object contained within a group is mandatory, meaning that an implementation must support either every object in a particular group or none at all. It was planned that additional MIBs groups and variables will be assigned to MIB-2 as management needs grow and evolve. Not all MIB-2 groups are mandatory.

These are the MIB-2 groups:

MIB-2 Group	Description
at	<p>Although this group was deprecated (and will be obsolete in the next MIB version), the NMC implements it for MIB1 compatibility as defined in the MIB-2. This group monitors services supplied by the Address Resolution Protocol (ARP). This group contains a single, indexed table composed of the mappings of Internet addresses to physical addresses.</p>
egp	<p>Objects operate on the Exterior Gateway Protocol (EGP).</p> <p>The NMC does not support the <code>egp</code> group.</p>
icmp	<p>Objects operate on the Internet Control Message Protocol (ICMP). This group defines configuration parameters for the ICMP. It contains the ICMP input and output statistics for each type of ICMP message.</p> <p>The NMC supports the <code>icmp</code> group.</p>
interfaces	<p>Objects operate on the network interface which attaches the device to the network. Each system interface to a communication facility is a member of this group. Some of the attributes in the interfaces group include the total number of interfaces and an Interfaces Table. This table includes the interface type, transmission speed, maximum transmission unit size, physical address, interface state, and various statistical counters.</p> <p>The NMC supports up to four rows in the interfaces table:</p> <ul style="list-style-type: none">■ One for each WAN port interface<p><i>The NMC has two serial ports, one of which can be configured for either UI or SLIP/WAN. When this port is configured to SLIP/WAN, it is included in the interface table.</i></p>■ One for the LAN port interface■ One for the loopback interface
ip	<p>Objects operate on the Internet Protocol (IP). The <code>ip</code> group defines the configuration parameters for the Internet Protocol on the managed system.</p> <p>This group includes three tables:</p> <ul style="list-style-type: none">■ The IP Address Table contains the device's IP addressing information■ The IP Route Table contains an entry for each route presently known to the device■ The <i>ipNetToMediaTable</i> object functions as the Address Translation Table, providing the identical functionality of the deprecated <code>atTable</code> in the Address Translation group <p>The NMC supports the mandatory <code>ip</code> group.</p>

MIB-2 Group	Description (continued)
snmp	<p>Implemented in MIB-2 only, the objects operate on the Simple Network Management Protocol (SNMP). A group of objects is defined to allow the management of SNMP applications.</p> <p>The NMC supports the mandatory snmp group.</p>
system	<p>Objects operate on the managed node. The system group describes each network element in the environment, including machine type, serial number, operating system, available resources, and other attributes.</p> <p>The NMC supports the mandatory system group.</p>
tcp	<p>Objects operate on the Transmission Control Protocol (TCP).</p> <p>The NMC does not support the tcp group because it does not implement the TCP protocol. All NMC functionality is implemented through UDP.</p>
transmission	<p>Implemented in MIB-2 only, the objects apply to media-specific types of information.</p> <p>The NMC does not implement the transmission group for its own Ethernet interfaces. It does support access to RFC 1406 for the HiPer DSP through this group.</p>
udp	<p>Objects operate on the User Datagram Protocol (UDP). Objects include the total number of transmitted and received UDP datagrams and error counters.</p> <p>The NMC supports the mandatory udp group.</p>



CHS MIB

This chapter contains detailed information about the Chassis MIB (CHS MIB).

Purpose of the MIB	The CHS MIB is an enterprise-specific MIB that was designed to provide the management information of all slots in the chassis. It includes two tables which define the slots of the chassis and the manageable entities in the chassis.
Products using this MIB	This MIB is used by the NMC. It contains configuration information for all chassis NICs and NACs. This MIB is also used by the MS software or TCM to identify and display the entities in the chassis slots and LED statistics.
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.chs (1.3.6.1.4.1.429.1.1)</p>
Configuration Data	<p>This MIB is comprised almost entirely of objects used to control the chassis. The implementation provides “logical” networking devices (entities) in the chassis and the configuration of the individual slots. The commands are provided in the command table of the chassis at card level which associated with special device MIBs to manage the device.</p> <p>Most of the MIB objects in the CHS MIB are self explanatory, except for the <i>chasFrontPanelLedStates</i> and <i>uchasFrontPanelLedColor</i> objects.</p>

The Configuration Group

Use the objects in the `uchasConfig` group to configure the chassis description and to query chassis status. Most of the objects in this group are self-explanatory from the MIB text, with the exception of the chassis status objects explained in the next section.

Chassis status

The NMC uses these MIB objects to communicate information about the presence of NACs and their Light Emitting Diode (LED) states to management stations:

- `uchasFrontPanelLedStates` (1.3.6.1.4.1.429.1.1.3.5)
- `uchasFrontPanelLedColor` (1.3.6.1.4.1.429.1.1.3.6)
- `uchasNicStates` (1.3.6.1.4.1.429.1.1.3.7)
- `ichasFrontPanelLedStates2` (1.3.6.1.4.1.429.1.1.3.8)
- `uchasFrontPanelLedColor2` (1.3.6.1.4.1.429.1.1.3.9)

These objects are bitmaps used to describe up to 12 LEDs per card. Each LED's information is encoded in a nibble (four bits). Each card uses six bytes out of the octet string; 17 chassis slots require 102 bytes in the octet string. The MIB does not identify how many nibbles are used to represent LEDs; it is up to the management station to identify the number of LEDs on each NAC. Power supply LEDs are also included in this object.

In addition to LED state and color, a few of the bits in these objects are used for additional information. For example, in the first byte of *uchasFrontPanelLedStates*, if the 8th bit is "0", it means slot #1 is empty; if the 8th bit is "1", it means the card is present. The 5th through 7th bits provide LED #1 status — off, on, flashing slow, or flashing fast. If the 4th bit is "0" it means the card type unknown; if it is "1", it means the card is fully discovered. The last three bits provide the LED #2 status — off, on, flashing slow, or flashing fast.

`uchasFrontPanelLedStates`

This object is used for the LEDs physical on/off/blinking states. Each NAC is represented by a 6-byte-long string.

Each LED uses a nibble to indicate its current state. A maximum of 12 LEDs are present in this string. If there are more than 12 LEDs, they are continued in the *uchasFrontPanelLedStates2* string.

LED byte strings This is a summary of the LEDs as represented in the byte strings:

- Information for the 16 NACs is in a continuous string (bytes 1–96)
- NMC LED information is contained in bytes 97–102
- Byte 103 is used for power supplies. One power supply is represented in each nibble of the byte
- Byte 104 is used as a null terminator
- Bytes 105–110 are not used

Bits in each nibble The most significant nibble of the of the first byte represents the first LED on the first NAC. The most significant nibble of the seventh byte represents the LED on the second NAC (and so forth for the remaining NACs). This table describes the least significant bits in each nibble (b3 = most significant bit, b0 = least significant bit):

b2	b1	b0	Description
0	0	0	LED off or not present
0	0	1	LED on solid
0	1	0	LED flashing slowly
0	1	1	LED flashing quickly
1	x	x	reserved

High bit information Additional information about the presence and status of NACs are embedded in the high-bit (b3)of each nibble (where b0 = most significant bit and b7 = least significant bit):



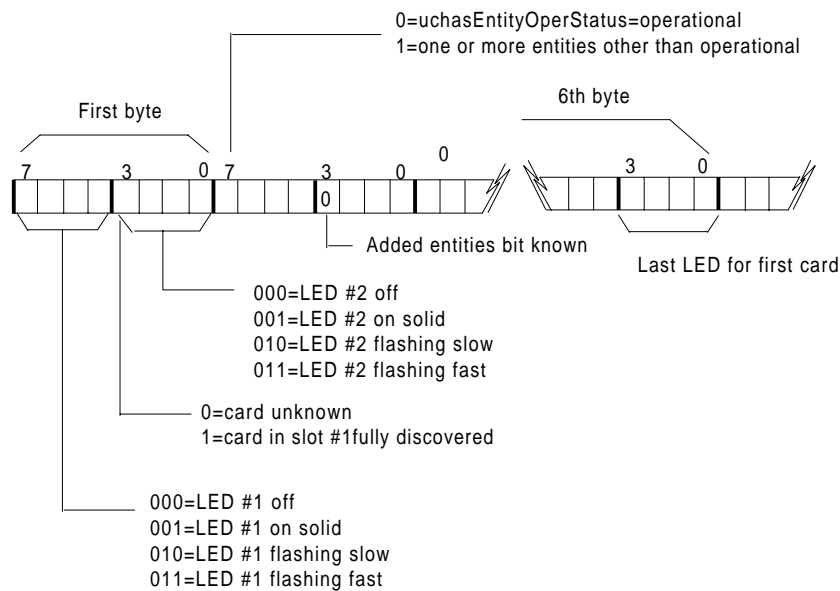
The most significant bits of the first three nibbles for each card are shown in this table. The most significant bits of the remaining nine nibbles for each card are reserved and always set to “0”.

LED	nibble b3
1	0 = slot empty
1	1 = card present
2	0 = card unknown
2	1 = card fully discovered
3	0 = <i>uchasEntityOperStatus</i> =operational (4)
3	1 = one or more entities other than operational (4)
4	x = reserved
5	x = reserved
6	x = reserved
7	x = reserved
8	x = reserved
9	x = reserved
10	x = reserved
11	x = reserved
12	x = reserved

The LED state The LED state is represented by these values:

CSN_LED_OFF	0x00	//LED is not lit
CSN_LED_ON	0x10	//LED is lit solid
CSN_LED_SF	0x20	//LED is flashing slowly
CSN_LED_FF	0x30	//LED is flashing rapidly

This illustration shows the bit definitions of the octet string returned when you query *uchasFrontPanelLedStates*.



uchasFrontPanelLedColor

This object is used for the LED's color information. Each NAC is represented by a 6-byte-long string.

Each LED's information is represented by a nibble. A maximum of 12 LEDs are present in this string. If there are more than 12 LEDs, they are continued in the *uchasFrontPanelLedColor2* string.

Bits in each nibble The most significant nibble of the of the first byte represents the first LED on the first NAC. The most significant nibble of

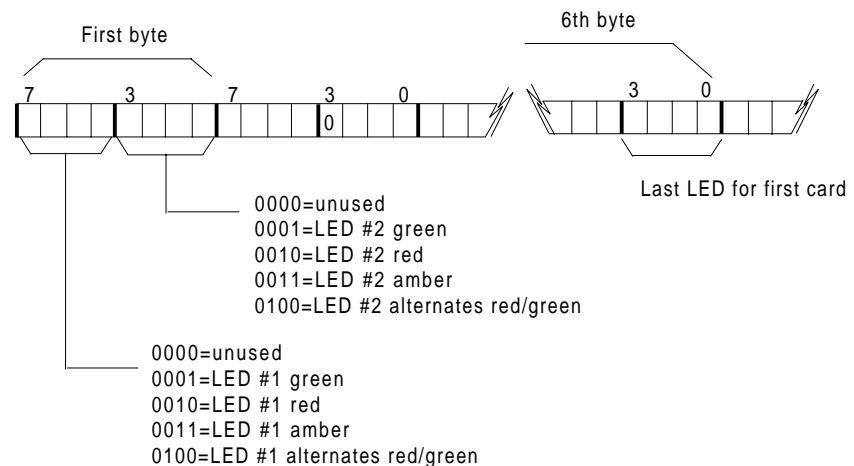
the seventh byte represents the LED on the second NAC (and so forth for the remaining NACs). This table describes the bits in each nibble:

b3	b2	b1	b0	Description
0	0	0	0	LED off or not present
0	0	1	1	LED green
0	0	1	0	LED red
0	0	1	1	LED amber (red and green)
0	1	0	0	LED alternates between green and red
0	1	0	1	reserved
:	:	:	:	:
1	1	1	1	reserved

LED color The LED color is represented by these values:

CSN_LED_GRN	0x01	//LED is green
CSN_LED_RED	0x02	//LED is red
CSN_LED_AMBER	0x03	//LED is amber
CSN_LED_REDGRN	0x04	//LED is alternating red and green

This illustration shows the bit definitions for the octet string returned when you query *uchasFrontPanelLedColor*.



LED byte strings This is a summary of the LEDs as represented in the byte strings:

- Information for the 16 NACs is in a continuous string (bytes 1–96)
- NMC LED information is contained in bytes 97–102
- Byte 103 is used for power supplies. One power supply is represented in each nibble of the byte
- Byte 104 is used as a null terminator
- Bytes 105–110 are not used

uchasNicStates

Use this object to query the status of each NIC in the chassis. Each NIC is represented by a 6-byte-long string.

NIC byte strings This is a summary of the NICs as represented in the byte strings:

- Information for the 16 NICs is in a continuous string (bytes 1–96).
- NMC information is contained in bytes 97–102
- Byte 103 is used as a null terminator
- Bytes 104–110 are not used
- The most significant nibble of the of the first byte represents the first LED on the first NAC. The most significant nibble of the seventh byte represents the LED on the second NAC (and so forth for the remaining NACs).

Since there are currently no NICs that have LEDs on them, the only bit in use is the most significant bit of the first three nibbles for each card. This indicates presence of the NIC and whether or not the proxy agent knows what kind of NIC is in the slot.

uchasFrontPanelLedStates2

This object and byte string are the continuation of *uchasFrontPanelLedStates*. If the NAC has more than 12 LEDs, state information for LEDs 13–24 is placed in this object. The format is the same.

uchasFrontPanelLedColor2

This object and byte string are the continuation of *uchasFrontPanelLedColor*. If the NAC has more than 12 LEDs, state

information for LEDs 13–24 is placed in this object. The format is the same.

Tables

The CHS MIB contains these tables:

- `uchasSlotTable` (chassis slots)
- `uchasEntityTable` (chassis entities)
- `uchasPowerSupplyTable` (power supply slots)
- `uchasPowerSupplyOutTable` (power supply performance)
- `uchasEnvironTable` (chassis environment)
- `uchasCmdTable` (chassis commands)
- `uchasArSlotTable` (chassis auto response scripts)
- `uchasArTimerTable` (chassis auto response timers)

Chassis slots
(`uchasSlotTable`)

The *uchasSlotTable* defines the physical aspects of the chassis for each installed card (module). It provides time of installation relative to when the NMC booted, each module's serial number, product code, and revision, as well as ROM and DIP Switch Settings.

The *uchasSlotTable* is a sparse table. This means there can be no “holes” in the table, or missing rows where cards may not be physically present in the chassis. Each card slot is represented by a different index number. Table rows are displayed sequentially; slots that do not contain cards are not displayed in the table.

Refer to the object IDs defined under *uchasKnownModules* for a complete list of cards that may show in the slot table.

Chassis NACs

Front slots in the chassis are numbered from 1–17 (for a 17-slot chassis), or 1–7 (for a 7-slot chassis). The rows of the table are indexed with the corresponding slot number. The NMC must be present physically to access the MIBs, so slot 17 (or 7 in the smaller chassis) will always show an NMC present.

Chassis NICs

The *uchasSlotTable* also includes the NICs. For numbering convenience, slots 18–20 are ignored. NIC slot numbering begins with 21 and

continues through 37 in a 17-slot chassis. (or through 27 in a 7-slot chassis). The slot 21 NIC resides behind the slot 1 NAC, and the slot 37 NIC resides behind the slot 17 NMC (or slot 7 in a 7-slot chassis).

Additional information

When the NMC discovers a card, the proxy agent adds a row to the slot table and fills in the read-only information about the newly-discovered module. When a card is removed, the NMC deletes the corresponding row from the MIB table. It is possible that a management station (MS) may get SNMP noSuchName errors when cards are removed from the chassis; when a card is removed, the object instances cease to exist.

Although the SNMP agent can provide information about the power supplies, for the purpose of the *uchasSlotTable*, they are not considered to be residing in slots. Refer to the section “Monitoring the Power Supply” in this chapter.

Chassis Entities
(uchasEntityTable)

Chassis entities

Entities are logical devices, processes, or interfaces that require an index to be provided in order to access the information associated with them.

These entities are currently defined in the chassis:

Card (Module)	Entities
Quad Modem	4 modem channel entities
Dual T1	1 card-level entity 2 span entities
HiPer DSP	1 card-level entity 24 modem channel entities (30 modems for E1) 1 span entity
Gateway cards	1 card-level entity
NMC	1 card-level entity

Example

The Quad Modem card is a good example of an application of entities. Since there are four modems on a card, the slot index alone does not provide enough information. Each modem channel is considered an entity and given its own index number. The resulting table is indexed, providing specific parameters for each entity, or modem, on the card.

Computing the entity index number

This formula is used to compute the entity index:

$$EI = (1000 * S) + E$$

Where:

EI is the entity index

S is the slot number of the card on which the entity resides

E is the entity number (E=0 for card-level entities)

Applying this formula to a Quad Modem, the third modem on the card in slot 5 has an entity index of 5003.

uchasEntityTable

The *uchasEntityTable* provides information about all chassis entities.

Like the *uchasSlotTable*, the *uchasEntityTable* is also a sparse table. The proxy agent adds rows to the entity table for the managed entities found on a card at the time it is discovered by the NMC. For numbering convenience, the indexes used on the *uchasEntityTable* maintain the slot number of the card on which they reside. The entities on a given card tend to be numbered sequentially, although there is no rule for this numbering. Refer to "Computing the entity index number".

Additional information

Refer to the object IDs defined under *uchasKnownEntities* for a complete list of the entities that can be on the entity table.

Failure of one or more entities in *uchasEntityOperStatus* will cause the NMC Status LED to turn red.

Monitoring the Power Supply

The *uchasPowerSupplyTable* and *uchasPowerSupplyOutTable* are used to provide power supply (PSU) monitoring. On a chassis with an integrated fan tray, the NMC also provides object data for the power source (AC/DC) and Max AMPs (110–130).

uchasPowerSupplyTable

The *uchasPowerSupplyTable* provides information about the PSU slots.

This is a dense table. For a 17-slot chassis, two rows are installed into this table by the proxy agent. Row 1 is for the left PSU as viewed from the front of the chassis. Row 2 is for the right PSU. The 7-slot chassis has only one permanent PSU, so only the first table row is installed.

The value of the *uchasPowerSupplyOperStatus* object indicates the status of each PSU LED. A value of bad(2) means the LED is red. A value or good(3) means the LED is green.

Other objects in this table are self-explanatory.

uchasPowerSupplyOutTable

The *uchasPowerSupplyOutTable* provides information about PSU performance and outputs.

This table is double-indexed, both by the PSU index (as with the *uchasPowerSupplyTable*) and by an output index. Each PSU has four rows in this table.

**Monitoring the
Chassis Environment**

uchasEnvironTable

The *uchasEnvironTable* provides information about the chassis environment.

Two rows are in this table. The first row is for the fan speed sensor as identified by the *uchasEnvironSensor* object ID. The second row is for the temperature sensor on the NMC card. Objects in each row are self-explanatory.

Failure in *uchasEnvironStatus* for any environmental sensor will cause the NMC Status LED to turn red.

Chassis Commands

uchasCmdTable

The *uchasCmdTable* contains objects used to issue commands to one or more devices in the chassis at a card level, force commands, check the results, and interpret the codes resulting from commands.

The table contains an entry for each of the NIC and NAC slots in the chassis. This table is indexed by slot number.

Chassis command function

The *uchasCmdFunction* object is used for basic system functions. This table describes its use.

Command	Description
noCommand(1)	This is the default value for <i>uchasCmdFunction</i> . Setting <i>uchasCmdFunction</i> to this value aborts any command currently in process. This statement does not apply to all commands.
removeFromService(2)	Use this command to hold a NIC or NAC in reset indefinitely until the <i>restoreToService</i> command is issued. Use <i>removeFromService</i> if a device has failed and is affecting the network negatively. Issuing this command holds all processors on the specified card in a reset state and puts all hardware in a fail-safe condition.
restoreToService(3)	Use this command to restore a NIC or NAC to service if it was removed previously by the <i>removeFromService</i> command.
hardwareReset(4)	Use this command to perform a hardware-level reset of the entire specified NIC or NAC. All channels on multi-channel card (Quad Modem, HiPer DSP) are affected by this command. Use this command to restore functionality to a device when the software-level reset command is not sufficient.
softwareDownload(5)	Use this command to perform a download of software to non-HiPer NMCs and other non-HiPer chassis NACs. A software download (SDL) allows a firmware upgrade for feature enhancements and bug fixes.
softwareDownload2(6)	Use this command to perform a download of software to HiPer NMCs and other HiPer chassis NACs. A software download (SDL) allows a firmware upgrade for feature enhancements and bug fixes.

Trap Enables

The uchasTrapEnable group provides objects for configuring chassis-level traps. You must configure traps in order for them to be sent to the MS alarm server and/or the RADIUS logging server.

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Traps for all cards in the chassis are handled by the NMC. If you use *nmcPowerUpAutoCfgEnable*, the NMC will automatically send trap enables to all NACs.

Traps

The CHS MIB provides access to a set of the events that can be controlled on a chassis level by the CHS MIB. Refer to “Trap enables” for an explanation of the uchasTrapEnable group.

The CHS MIB contains these trap enable objects:

Trap Enable	Generates this trap
uchasModuleInserted TrapEna	moduleInserted(1)
uchasModuleRemovedTrapEna	moduleRemoved(2)
uchasPSUWarningTrapEna	psuWarning(3)
uchasPSUFailureTrapEna	psuFailure(4)
uchasTempWarningTrapEna	tempWarning(5)
uchasFanFailureTrapEna	fanFailure(6)
uchasEntityWatchdogTrapEna	entityWatchdogTimeout(7)
uchasEntityMgtBusFailTrapEna	entityMgtBusFailure(8)
uchasPsuIncompatible	psuIncompatible(75)

Refer to the chapter titled *Trap Reference* for additional information about traps.

Other MIB-specific Information

This section contains information about:

- Software Download to Chassis NACs
- Software Download to 386- and 486-NMCs
- Software Download to the HiPer NMC
- Known Types
- Auto response

Software Download to Chassis NACs

Two mechanisms exist for downloading software to the chassis: Software Download 1 (also called “pcsd1”) and Software Download 2 (also called SDL-2). Use pcsd1 for all non-HiPer chassis NACs; use SDL-2 for all HiPer chassis NACs, including the HiPer NMC.

Use SDL to upgrade the firmware on the NMC/NACs for feature enhancements and bug fixes. SDL is non-trivial due to the coordinated series of SNMP operations together with on one or two TFTP file transfers. Implementation on the MS is straightforward.

Refer to the Appendix section of the *NMC Product Reference* for instructions on completing a Software Download.

Overview of the Software Download 1 process to chassis NACs

You may perform a software download (SDL) to multiple identical NACs simultaneously. Use the **softwareDownload(5)** command from the *uchasCmdTable* rows for the slots to which you want to download.

This is an overview of the SDL-1 process.

- 1 When these commands are executed by issuing a **set** command, each slot is put into a software download state, essentially rendering them active only for SDL. Querying the *uchasCmdTable* rows for the affected slots (by using the **get** command) should result in an **InProgress(3)** status in the *uchasCmdResult* column for the appropriate rows, indicating the command initiated successfully on these slots.
- 2 Once *uchasCmdFunction* is **set** to **softwareDownload(5)**, the MS must initiate the next step in the process — failure to do so will abort the software download.

- 3 The MS transfers the .SDL file to the NMC. This is an example of a Novell LAN Workplace TFTP syntax for downloading to a T1 card:



This syntax is transparent to the user.

```
tftp -b t1010200.sdl 192.77.203.65=t1010200.sdl
```

where the binary file is T1010200.SDL and the NMC's IP address is 192.77.203.65. The T1010200.SDL file represents the local file while the filename after the "=" refers to the destination file. The destination file must either have an extension of ".sdl" (case insensitive), or the filename must be SDL (compatible with early versions of NMC software).

- 4 If a TFTP write session is available on the NMC, it will send the trigger request to each slot involved in the SDL. This places the NAC in the SDL state and allows the NMC to receive the card ID of the NAC (which the NMC uses for .SDL file validation).
- 5 The MS begins transferring the .SDL packet in a series of 512 byte packets. The first packet contains the .SDL file 32-byte header used by the NMC to validate the SDL. If the validation succeeds, subsequent TFTP data packets transfer.



If a SDL fails or aborts, wait one minute before attempting another SDL. This wait allows the devices to time out before beginning a new SDL.

- 6 When the TFTP transfer is complete without error, the MS should poll *uchasCmdCode* for the slots involved in the SDL. All NACs involved in the SDL are currently erasing their Flash ROM; the MS must wait until all NACs complete this before beginning to transfer the .NAC files. The .NAC File transfer starts when all slots involved in the SDL have a *uchasCmdCode* value of *NACFile(82)*. The MS must initiate the next step in the process — failure to do so will abort the software download.

This is an example of a Novell LAN Workplace TFTP syntax for downloading the .NAC file to the NMC:

```
tftp -b sto10200.nac 192.77.203.65=st010200.nac
```

The first TFTP data packet carries the .NAC file's header, which the NMC uses to validate the .NAC file. If validation succeeds, subsequent data packets are sent until the entire .NAC file transfers. At this point, the MS polls *uchasCmdResult* for all slots involved in the SDL. The result should be *success(2)* for each slot.

The SDL is complete.

Software Download to 386- and 486-NMCs

NMCs with 386 and 486 processors use the SDL-1 format.

Since a larger portion of operational code must remain running to support the SDL transport, the NMC requires a slightly different SDL mechanism. When commanded to enter an SDL state, all other NACs use a "loader" (located in the device's Flash ROM boot block), which supports the .SDL file transfer. These NACs are not running operational code during an SDL. Because the NMC needs to support the TFTP protocol and, as a result, the entire IP stack, a large portion of its operational code remains running. SDL functionality is also built into the NMC's operational code; SNMP-initiated SDLs do not need to transfer the .SDL file to the NMC.

This is a summary of the SDL-1 process to the NMC:

- 1 Initiate an NMC SDL by **setting** *uchasCmdFunction* to **softwareDownload(5)** for slot 17. This places the NMC into an SDL state. As a result, all subsequent SNMP agent operation is suspended, and any incoming **get** or **set** requests are immediately sent back as **get** responses with a generic error.
- 2 Executing the SDL command forces the NMC to erase its Flash ROM. During this time, the SNMP error-status object is set to 125, meaning the NMC is in an SDL state and erasing its flash.
- 3 Once the Flash is erased, the NMC changes SDL state, forcing subsequent SNMP requests to be returned with the error-status object set to 124. This is the MS's trigger to begin the file transfer.
- 4 This TFTP command is issued to download the .NAC file to the NMC:
tftp -b nm020200.nac 192.77.203.65=nm020200.nac
- 5 The first TFTP data packet carries the .NAC's file header, which the NMC uses to validate the .NAC file. If the file validates, subsequent data packets are sent until the entire .NAC file transfers.
- 6 After a complete .NAC transfer, the MS should wait for the NMC to reboot and become operational with the new code. If the .NAC file transfer fails or aborts, the NMC remains in the "waiting for SDL" mode (blinking green Run/Fail LED). After waiting for the NMC to re-erase its Flash ROM, the MS may re-initiate the TFTP transfer.

There is no need to re-issue the **softwareDownload(5)** command. If the MS must verify that the NMC finished erasing Flash and is waiting for SDL, it may issue any SNMP request and match the 124 generic error code in the corresponding response.



Once the NMC has erased its operational software from the Flash devices, its network protocol stack exists only in RAM. Power cycling the NMC at this point means the NMC will only be able to power up in a “waiting for SDL” mode on its local User Interface port via the PCSDL.EXE mechanism.

Software Download to the HiPer NMC

Software download to the HiPer NMC uses the SDL-2 format.

SDL-2 is a protocol that allows you to perform a software download to the Total Control chassis. The HiPer NMC supports SDL-2 both from the UI port and across the IP network. You may use either a MIB browser or TCM to access SDL-2.

SDL-2 can be completed online across a LAN or WAN without affecting NMC chassis management. At boot-up only, you may also complete SDL-2 through the User Interface (UI) port using a terminal-emulation program that supports the Z-modem protocol.

SDL-2 supports a single .DMF file and uses the Z-Modem protocol. Refer to the “*HiPer Network Application Card Software Download (SDL-2)*” for additional information about SDL-2 installation and use.

Known Types

The *uchasKnownTypes* portion of the MIB contains OID definitions that are used as values within objects of the CHS MIB.

- OIDs within *uchasKnownChassis* are chassis type identifiers used as the values for *uchasType*
- OIDs within *uchasKnownModules* are NIC and NAC identifiers used as values for *uchasSlotModuleType* in the slot table
- OIDs within *uchasKnownEntities* are values for *uchasEntityObjectID* in the entity table
- OIDs within *uchasWellKnownSensors* define sensor types used as values for *uchasEnvironSensor* in the environment table

Auto Response

The CHS MIB supplies objects and tables for setting card-level and chassis-level auto response scripts.

Auto response group

The *uchasAutoResponse* group contains auto response scripts that will run when certain chassis-level events occur. Objects within this group contain a script that stores a predefined set of actions to take when a

chassis event occurs. If you do not write a script, the object will contain zero bytes.

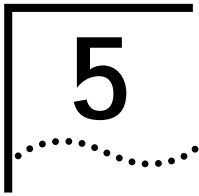
Auto response slot table

The `uchasArSlotTable` contains auto response scripts that will run when certain slot-level events occur.

Auto response timer table

The `uchasArTimers` table contains timers that may be activated to run `AutoResponse` scripts when certain events occur. The timers trigger a response when they expire. You may assign individual responses or entire scripts to these timers.

Refer to the chapter in this document titled “Auto Response Reference” for additional information.



CHS_TRAP MIB

This chapter contains detailed information about the Chassis Trap MIB (CHS_TRAP MIB).

Purpose of the MIB	<p>The CHS_TRAP MIB is an enterprise-specific MIB. It provides management information of all chassis-level traps.</p> <p>The CHS_TRAP MIB defines each of the enterprise-specific SNMP traps that can be generated for chassis devices.</p>
Products using this MIB	<p>This MIB is used by Network Management Card, NICs, and NACs.</p>
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node. It differs from all other USR MIBs in that it is formatted specifically to provide a reference for trap reporting, but does not define objects like other traditional MIBs. The CHS-TRAP MIB has no OID.</p>
Configuration Data	<p>This MIB contains no configurable parameters.</p>
Tables	<p>The CHS_TRAP MIB contains no tables.</p>
Traps	<p>The CHS_TRAP MIB provides a set of traps that can be controlled on a chassis level. The traps are enabled in separate MIBs.</p>

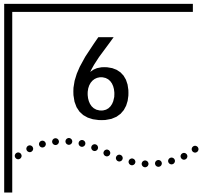
Refer to the chapter titled *Trap Reference* for additional information about traps.

Other MIB-specific Information

Each trap definition is annotated with comments following the description and preceding the trap number. These comments are used by SNMP trap handler software to properly display the trap information.

The CH_TRAP MIB contains many imports from other 3Com MIBs. These imports are needed for defining the var-bind list of traps defined in this MIB.

Traps generated by the chassis use the USR node (enterprises 429) for the enterprise OID. The enterprise OID does not use the *sysObjectId* reported by the NMC. All 3Com SNMP-manageable devices define traps within the same space under the USR enterprise.



DUAL TRUNK CARD MIBS

This chapter contains detailed information about the MIBs that apply to dual trunk card applications, including cahnnelized T1 , E1 , PRI, and R2.

Overview

This chapter includes explanations of these MIBs:

- DT1
- IDT1
- DS1
- DS0
- IDS0
- UDS1

The dual trunk cards use several different MIBs to configure settings, enable traps, provide statistics, and complete other management functions. The applicable MIBs depend upon what software is loaded on the card.

Application	Applicable MIBs
Channelized T1/Dual trunk card	DT1, DS1, UDS1, DS0
ISDN PRI/Dual trunk card	DT1, IDT1, DS1, UDS1, IDS0
Channelized E1/R2	DT1, DS1, UDS1, DS0

DT1 contains basic span line configuration and Channelized T1 information. IDT1 assumes the "basic" span objects from DT1 and ingores the Channelized T1 specific objects, using its own PRI- and R2-specific objects. The same is true of DS0 and IDS0; IDS0 assumes the "basic" parts of DS0, and adds its own PRI- and R2-specific objects.

DS1-related MIBs

RFC 1232, also known as DS1.MIB, is an older MIB in the experimental branch that applies to the dual span T1/PRI cards. The HiPer DSP uses the newer RFC 1406 MIB to manage its spans. Objects within RFC 1232 are prefixed by "ds1" . Objects within RFC 1406 are prefixed by "dsx1" .

Both RFC 1232 and RFC 1406 contain a span-level configuration table and additional tables for interval statistics.

To provide for enterprise-specific information, USR/3Com created several enterprise MIBs that are used along with the RFC-MIBs. Dual span T1/PRI cards use the UDS1 MIB (prefixed with "uds1"). The MIB contains tables for configuration, commands, and trap enables. The UDS1 MIB also contains additional interval tables to supplement RFC 1232.

Checking Span Status
with Dual Trunk Card
MIBs

Several of the tables and objects within these MIBs are useful for checking span status.

This list is a starting point for checking current span status. Locate other objects in the MIB trees to meet your specific needs.

Statistics	Recommended Table
General span status	uds1StatTable
	ds1ConfigTable
Interval status	ds1IntervalTable
	ds1CurrentTable
	ds1TotalTable
Trap enables	uds1TrapEnaTable

DT1 MIB

Purpose of the MIB The DT1 MIB is used to configure parameters, check status, and enable traps at the NAC level.

Products using this MIB This MIB is used in whole or in part by these software applications:

- Channelized T1
- T1/PRI
- E1/PRI
- E1/R2

Registration ID This MIB is an Internet Enterprise MIB registered under the USR enterprise node.

iso.org.dod.internet.private.enterprises.usr.nas.dt1
(1.3.6.1.4.1.429.1.3)

Configuration Data None.

MIB Tables The DT1 MIB includes these tables:

- dt1IdTable (dual T1 identification)
- dt1CfgTable (channelized T1 configuration)
- dt1StatTable (status)
- dt1CmdTable (NAC-level commands)
- dt1TrapEnaTable (trap enables)

Dual T1 identification (dt1IdTable)

Use the objects in the dt1IdTable to query about general NAC-level (hardware) identification. The objects in this table are self-explanatory from the MIB text.

Channelized T1 configuration (dt1CfgTable)

Use the objects in the dt1CfgTable only for configuring a channelized T1 trunk card. The dt1CfgTable is applicable to both spans.

Use the timing objects to synchronize the spans to a single timing source, and to specify a secondary timing source in case of primary source failure. The Telco is the recommended timing source. A value of "high" indicates primary source.

Status (dt1StatTable)

Use the objects in the dt1StatTable to query the trunk card to check NAC-level status.

NAC-level commands (dt1CmdTable)

The dt1CmdTable contains objects used to issue and force commands to the NAC, as well as to check the results and interpret the codes resulting from commands.

Command function The *dt1CmdFunction* object is used for basic commands. This table describes its use.

Command	Description
noCommand(1)	This is the default value for <i>dt1CmdFunction</i> .
saveToNVRAM(2)	Use this command to save all of the current settings to the dual trunk card's NVRAM.
restoreFromNVRAM(3)	Use this command to restore all configuration settings to the values that were last saved to the dual trunk card's NVRAM.
restoreFromDefault(4)	Use this command to restore all card settings to the factory default values.
nonDisruptSelfTest(5)	Use this command to start a nondisruptive self test while the trunk card is online.
disruptSelfTest(6)	This is a self test of the dual trunk card which, if executed, disrupts any calls in progress.
softwareReset(7)	Use this command to reset the dual trunk card.
resetToHiPrioTimingSrc(8)	Use this command to reset the timing source to high priority. This command avoids a card reboot. If the high priority source is not valid, the timing will revert to the secondary source. For example, you might use this command to reset the timing if a span went down and then came back up for a known reason.
forceTdmBusMastership(9)	Not used.
enterSpanToSpanLoopback(10)	Not used.
exitSpanToSpanLoopback(11)	Use this command to end a loopback test. This command will drop all current calls and reboot the NAC.
restoreDefaultUIPassword(12)	Use this command to restore the user interface console password to the default that is stored in the NMC's NVRAM.

Trap enables (dt1TrapEnaTable)

The dt1TrapEnaTable group provides objects for configuring NAC-level traps.

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Traps for all cards in the chassis are handled by the NMC. If you use *nmcPowerUpAutoCfgEnable*, the NMC automatically sends trap enables to all NACs.

Traps The DT1 MIB contains these trap enable objects:

Trap Enable	Generates this trap
dt1TrapEnaTxTmgSrcSwitch	transmitTimingSourceSwitch(26)
dt1TrapEnaCallEvent	callEvent(74) – obsolete
dt1TrapEnaCallArriveEvent	callArriveEvent(76)
dt1TrapEnaCallConnEvent	callConnectEvent(77) – obsolete
dt1TrapEnaCallTermEvent	callTermNormalEvent(78)
dt1TrapEnaCallFailEvent	callTermFailedEvent(79) – obsolete

Refer to the chapter titled *Trap Reference* for additional information about traps.

IDT1 MIB

Purpose of the MIB	The IDT1 MIB is used to configure NAC-level ISDN parameters and call routing.
Products using this MIB	<p>This MIB is used in whole or in part by these software applications:</p> <ul style="list-style-type: none">■ T1/PRI■ E1/PRI■ E1/R2
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.idt1 (1.3.6.1.4.1.429.1.15 15)</p>
Configuration Data	<p>Use this MIB to configure NAC-level ISDN parameters and call routing functionality for T1/PRI and E1/PRI.</p> <p>DNIS-based resource access (multiple objects)</p> <p>DNIS-based resource access allows inbound calls based on Dialed Number Identification Service (DNIS) to be routed to pre-partitioned resource pools. This feature is supported by objects in idt1PITable, idt1MdmRPATable, and idt1GwyRPA.</p>
MIB Tables	<p>The IDT1 MIB contains these tables:</p> <ul style="list-style-type: none">■ idt1CfgTable (configuration)■ idt1CrTable (call routing)■ idt1PITable (reserved resource pool)■ idt1MdmRPATable (resource pool mapping assignment)■ idt1GwyRPA (gateway to reserved resource pool mapping assignment)

Configuration (idt1CfgTable)

Use the objects in to the idt1CfgTable to configure NAC-level ISDN parameters. The objects in this table are self-explanatory from the MIB text.

Call Routing (idt1CrTable)

Use the objects in the idt1CrTable to configure ISDN call routing parameters. Configuration is based upon analog or digital incoming calls. Typically, this table is no longer used, but remains for backward compatibility.

Reserved resource pool (idt1PITable)

Use the objects in idt1PITable to set up pool routing to better control network traffic. This is a double-indexed table that defines phone numbers and the type of call. Twelve individual pools are available. Each pool represents a single phone number. The table also allows you to assign multiple pools and gateways to a specific number.

Resource pool mapping assignment (idt1MdmRPATable)

Use the objects in idt1MdmRPATable to assign modems to one of twelve modem pools.

Gateway to reserved resource pool mapping assignment (idt1GwyRPA)

Use the objects in idt1GwyRPA to assign gateway NACs to one of twelve modem pools.

Traps No trap enables are included in this MIB.

DS1 MIB

Purpose of the MIB The DS1 MIB is a subset of RFC 1406, including objects for the DS1 Near End Group. Use this MIB to configure the spans and check Telco error statistics.

 RFC 1406 is the MIB module for the DS1 span (trunk-level) objects.

Products using this MIB This MIB is used for channelized T1 applications only.

Registration ID This MIB is an experimental MIB registered under the experimental branch.

iso.org.dod.internet.experimental.ds1
(1.3.6.1.4.2)

Configuration Data No configurable parameters are included in this MIB.

MIB Tables The DS1 MIB includes these tables:

- ds1ConfigTable (configuration)
- ds1CurrentTable (current interval)
- ds1TotalTable (total intervals)
- ds1IntervalTable (interval)

Configuration (ds1ConfigTable)

Use the objects in the ds1ConfigTable for configuring the channelized T1 span. Some objects can also be used for checking status. Objects in this table are self-explanatory from the MIB text.

Current interval (ds1CurrentTable)

The objects in the ds1CurrentTable contain various statistics being collected about each span for the current 15 minute interval. All objects except *ds1LineType* are read-only.

Use *ds1LineType* to configure the line type for this DS1 circuit.

This is a single-indexed table.

Total intervals (ds1TotalTable)

The objects in the ds1TotalTable contain the cumulative sum of the various statistics about each span for the 24 hour period preceding the current interval. This table does not clear until the NAC reboots.

This is a double-indexed table.

Intervals (ds1IntervalTable)

The objects in the ds1IntervalTable contain various statistics and error counters collected about each span over the previous 24 hours of

operation. The past 24 hours are broken into 96 completed 15-minute intervals.

This is a double-indexed table.

Traps	No trap enables are included in this MIB.
Other MIB-specific Information	Two instances of each object are created, one for each span.

UDS1 MIB

Purpose of the MIB	The UDS1 MIB extends the DS1 MIB objects as defined by RFC 1232, providing additional parameters and statistics. Implementation of the USR DS1 Configuration group is mandatory for all systems that provide management of USR/3Com's Dual T1 Cards.
Products using this MIB	<div>This MIB is used in whole or in part by these software applications:</div> <ul style="list-style-type: none">■ Channelized T1■ T1/PRI■ E1/PRI■ E1/R2
Registration ID	<div>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</div> <div>iso.org.dod.internet.private.enterprises.usr.nas.uds1 (1.3.6.1.4.1.429.1.4)</div>
Configuration Data	The UDS1 MIB applies to all span applications.
MIB Tables	<div>The UDS1 MIB contains these tables:</div> <ul style="list-style-type: none">■ uds1ConfigTable (configuration)■ uds1IntervalTable (interval)■ uds1CurrentTable (current interval)

- uds1TotalTable (total intervals)
- uds1StatTable (status)
- uds1CmdTable (commands)
- uds1TrapEnaTable (trap enables)

Configuration (uds1ConfigTable)

The uds1ConfigTable contains additional configurable parameters for each span that are not supported in the DS1 MIB defined by RFC1232. Each entry in this table corresponds to an entry in the DS1 MIB. There is a one-to-one correspondence between the value of *uds1CSUIndex* and *ds1CSUIndex*.

The objects in the uds1ConfigTable are self-explanatory from the MIB text.

Interval (uds1IntervalTable)

The uds1IntervalTable extends the ds1IntervalTable, providing additional parameters for status counters over 15 minute intervals. Every entry in the ds1IntervalTable has a corresponding entry in this table.

Implementation of the USR DS1 Interval Group is mandatory for all systems that provide management of USR/3Com's Dual T1 Cards.

Current interval (uds1CurrentTable)

The uds1CurrentTable extends the ds1CurrentTable, providing additional parameters for status counters over the current 15 minute interval. Every entry in the ds1CurrentTable has a corresponding entry in this table.

Implementation of the USR DS1 Current Group is mandatory for all systems that provide management of USR/3Com's Dual T1 Cards.

Total intervals (uds1TotalTable)

The uds1TotalTable extends the ds1TotalTable, providing additional parameters for the cumulative sum of the various statistics for the 24 hour interval preceding the first valid interval in the uds1CurrentTable. Every entry in the ds1TotalTable has a corresponding entry in this table.

Implementation of the USR DS1 Current Group is mandatory for all systems that provide management of USR/3Com's Dual T1 Cards.

Status (uds1StatTable)

Use the objects in the uds1StatTable to obtain information about span line status. There is no corresponding table in the DS1 MIB.

Commands (uds1CmdTable)

The uds1CmdTable contains objects used to issue and force commands to the NAC, as well as to check the results and interpret the codes resulting from commands. This is a USR-specific version of the DS1 status table.

Use these commands when monitoring and performing trouble clearing on the span.

Command function The *uds1CmdFunction* object is used for basic commands. This table describes its use.

Command	Description
noCommand(1)	This is the default value for <i>uds1CmdFunction</i> .
forceReceiverReframe(2)	Use this command when frames are out-of-sync (Out of Frame "OOF" condition) to attempt to resynchronize with the DS1.
enterLoopback(3)	Use this command to signal the Telco that the chassis is performing a loopback test.
exitLoopback(4)	Use this command to exit local loopback testing.
inService(5)	Use this command to put the DS1 in service and restore all channels on the span to an in-service state.
localOutOfService(6)	Use this command to remove the DS1 from service and place all channels on the span into an out-of-service "OOS" state.
blockAnalogCalls(7)	Use this command to block all analog calls and only allow incoming BRI/PRI digital calls transmitted at 56/64 kbps. (PRI only)
blockDigitalCalls(8)	Use this command to block all digital BRI/PRI calls and allow only analog calls transmitted as voice or as 3.1 kHz audio.
blockAllCalls(9)	Use this command to block both analog and digital calls.
blockNoCalls(10)	Use this command to restore both analog and digital calls.

Command	Description (continued)
redAlarmOverride(11)	Use this command to ignore an existing red alarm condition. This command is typically used when performing maintenance to prevent reaction to an alarm state.
takeDownDChannel(12)	Use this command to remove a PRI D-channel from service. This will make the span (or NFAS spans) unavailable, and hunting resumes on the next D-channel in that hunt group.
bringUpDChannel(13)	Use this command to restore a PRI D-channel to service. This will make the span (or NFAS spans) available.

Trap enables (uds1TrapEnaTable)

The uds1TrapEnaTable provides objects for configuring span-level traps.

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Traps The UDS1 MIB contains these trap enable objects:

Trap Enable	Generates this trap
usrds1EventYellowAlarm	yellowAlarm(22)
usrds1EventRedAlarm	redAlarm(23)
usrds1EventLossOfSignal	lossOfSignal(24)
usrds1EventAlarmIndSignal	alarmIndicationSignal(25)
usrds1EventYellowAlarmClr	yellowAlarmClear(50)
usrds1EventRedAlarmClr	redAlarmClear(51)
usrds1EventLossOfSgnlClr	lossOfSignalClear(52)

Trap Enable	Generates this trap (continued)
usrds1EventAlrmIndSgnlClr	alarmIndicationSignalClear(53)
usrds1EventContCrcAlrm	contCrcAlarm(59) (E1/PRI, E1/CAS only)
usrds1EventContCrcAlrmClr	contCrcAlarmClear(60) (E1/PRI, E1/CAS only)
usrds1EventPhysStateChng	phyStateChng(61) (T1/PRI, E1/PRI, E1/CAS only)
usrds1EventDchanInSrvc	dChanInService(70) (PRI only)
usrds1DchanOutOfSrvc	dChanOutOfService(71) (PRI only)
uds1TrapEnaDs0InSrvc	ds0sInService(72) (PRI only)
uds1TrapEnaDs0OutOfSrvc	ds0sOutOfService(73) (PRI only)
uds1TrapEnaMultiFrame	uds1MultiFrame(104) (E1/CAS only)
uds1TrapEnaRemMultiFrame	uds1RemoteMultiFrame(105) (E1/CAS only)
uds1TrapEnaMultiFrmClr	uds1MultiFrameClr(106) (E1/CAS only)
uds1TrapEnaRemMultiFrmClr	uds1RemoteMultiFrameClr(107) (E1/CAS only)

Refer to the chapter titled *Trap Reference* for additional information about traps.

DSO MIB

- Purpose of the MIB

The DSO MIB contains objects for configuring and viewing status of channelized T1 DSOs (channels).
- Products using this MIB

This MIB is used for channelized T1 only.

Registration ID This MIB is an Internet Enterprise MIB registered under the USR enterprise node.

iso.org.dod.internet.private.enterprises.usr.nas.ds0
(1.3.6.1.4.1.429.1.5)

Configuration Data No configurable objects are included in this MIB.

MIB Tables The DS0 MIB contains these tables:

- dsOCfgTable (configuration)
- dsOStatTable (status)
- dsOCmdTable (commands)
- dsOBulkAccessTable (bulk access)

Configuration (dsOCfgTable)

Use the objects in dsOCfgTable to configure each DS0 timeslot. The objects in this table are self-explanatory from the MIB text.

Status (dsOStatTable)

Use the objects in dsOStatTable to obtain current information about each DS0 timeslot.

Use the object *dsOStatModem* to identify the modems available to a T1-E1 card.

Commands (dsOCmdTable)

The dsOCmdTable contains objects used to issue and force commands to the NAC, as well as to check the results and interpret the codes resulting from commands.

Command function The *dsOCmdFunction* object is used for basic commands. This table describes its use.

Command	Description
noCommand(1)	This is the default value for <i>dsOCmdFunction</i> .
hardBusyOut(2)	Use this command to drop any calls on the channel and set the channel to a busy state.
softBusyOut(3)	Use this command to wait for the channel to go idle, and then set the channel to a busy state.

Command	Description (continued)
restore(4)	Use this command to restore the channel to an in-service state.
disconnect(5)	Use this command to drop the call on this channel.
callIgnore(6)	Use this command to set a channelized T1 to ignore incoming calls.
transparentTest(7)	Use this command to place a channelized T1 into a maintenance state for responder testing.

Bulk access (ds0BulkAccessTable)

Use the ds0BulkAccessTable to obtain information about the DS0 configuration parameters in a bulk file format. This table provides information about all 24 DS0s on a DS1.

This table is most useful when using Total Control Manager or a custom application. A standard SNMP **get** on the objects in ds0BulkAccessTable will not provide much information about DS0 status.

Traps No trap enables are included in this MIB.

IDS0 MIB	
Purpose of the MIB	The IDS0 MIB contains objects for configuring and viewing status of PRI ISDN DS0s (channels).
Products using this MIB	<div>This MIB is used in whole or in part by these software applications:</div> <ul style="list-style-type: none"> ■ T1/PRI ■ E1/PRI ■ E1/R2
Registration ID	<div>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</div> <div>iso.org.dod.internet.private.enterprises.usr.nas.ids0 (1.3.6.1.4.1.429.1.16)</div>

Configuration Data No configurable objects are included in this MIB.

MIB Tables The IDS0 MIB contains these tables:

- idsOCfgTable (configuration)
- ids0StatTable (status)
- idsOCmdTable (commands)
- ids0BulkAccessTable (bulk access)

Configuration (idsOCfgTable)

Use the objects in idsOCfgTable to configure each DS0 timeslot. The objects in this table are self-explanatory from the MIB text.

Status (ids0StatTable)

Use the objects in ids0StatTable to obtain information about each DS0 timeslot. Many of the objects are nonaccessible and serve as placeholders for reporting trap values. the nonaccessible objects will not show when doing an SNMP **get**.

Commands (idsOCmdTable)

The idsOCmdTable contains objects used to issue and force commands to the NAC, as well as to check the results and interpret the codes resulting from commands.

Command function The *idsOCmdFunction* object is used for basic commands. This table describes its use.

Command	Description
noCommand(1)	This is the default value for <i>idsOCmdFunction</i> .
disconnect(2)	Use this command to drop the call on this channel.
inService(3)	Use this command to restore this channel to an in-service state.
localOutOfService(4)	Use this call to locally place this channel into an out-of-service "OOS" state.
blockAnalogCalls(5)	Use this command to block (reject) analog calls on this channel.
blockDigitalCalls(6)	Use this command to block (reject) digital calls on this channel.

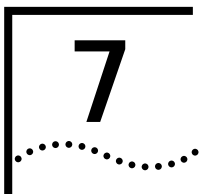
Command	Description (continued)
blockAllCalls(7)	Use this command to block (reject) all calls on this channel.
blockNoCalls(8)	Use this command to accept all call types on this channel.

Bulk access (ids0BulkAccessTable)

Use the ids0BulkAccessTable to obtain information about the DS0 configuration parameters in a bulk file format. This table provides information about all 24 DS0s on a DS1.

This table is most useful when using Total Control Manager or a custom application. A standard SNMP **get** on the objects in ids0BulkAccessTable will not provide much information about DS0 status.

Traps No trap enables are included in this MIB.



FILE MIB

This chapter contains detailed information about the FILE HEADER MIB (FILE MIB).

Purpose of the MIB

The FILE HEADER MIB was designed to be used as a template for the Single Configuration File Format (CFM).

The NMC creates and stores a single file in the 3Com-proprietary CFM format that contains configuration information for all NACs in the chassis (including HiPer ARC). This file allows you to use SNMP to:

- Configure an entire chassis with a single file (by downloading the file from a MS to the NMC)
- Compare different chassis configurations
- View the entire chassis configuration (by uploading the file from the NMC to a MS)

This feature is triggered through an SNMP browser and requires a TFTP script for file transfer.

The NMC owns this file format, and formats the configuration data on behalf of the chassis as defined in this MIB. Objects in the MIB are not accessible, but are be used by file parsers both within the NMC and elsewhere.

Products using this MIB

This MIB is used by the NMC and all other chassis NACs.

Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.file (1.3.6.1.4.1.429.1.31)</p>
Configuration Data	No configurable objects are included in this MIB.
Tables	No tables are included in this MIB.
Traps	No trap enables are included in this MIB.
Other MIB-specific Information	<p>This MIB is comprised of objects which represent the stored CFM data format which has been created by NMC. Contents of this MIB serve as a template for parsing a file already created.</p> <pre> FILE-MIB DEFINITIONS ::= BEGIN IMPORTS enterprises, experimental, IPAddress, TimeTicks, Gauge, Counter FROM RFC1155-SMI DisplayString FROM RFC1213-MIB OBJECT-TYPE FROM RFC-1212; usr OBJECT IDENTIFIER ::= { enterprises 429 } nas OBJECT IDENTIFIER ::= { usr 1 } file OBJECT IDENTIFIER ::= { nas 32 } fileHdr OBJECT IDENTIFIER ::= { file 1 } </pre>

```
cfmFileNmae OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..25))
ACCESS not-accessible
STATUS optional
DESCRIPTION
"This parameter is place holder for the file name of .CFM
file created on NMC."
::= { fileHdr 1 }

cfmFileVersion OBJECT-TYPE
SYNTAX INTEGER (0..4)
ACCESS not-accessible
STATUS optional
DESCRIPTION
"This parameter is place holder for NMC CFM file Version."
::= { fileHdr 2 }

cfmFileOwner OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..25))
ACCESS not-accessible
STATUS optional
DESCRIPTION
"This parameter is place holder for NMC CFM file Owner."
::= { fileHdr 3 }

cfmFileSize OBJECT-TYPE
SYNTAX INTEGER (0..4)
ACCESS not-accessible
STATUS optional
DESCRIPTION
"This parameter is place holder for NMC CFM file Size."
::= { fileHdr 4 }

cfmFileTimeStamp OBJECT-TYPE
SYNTAX INTEGER (0..4)
ACCESS not-accessible
STATUS optional
DESCRIPTION
"This parameter is place holder for NMC CFM file Creatin
time GMT."
::= { fileHdr 5 }

cfmFileCRC OBJECT-TYPE
SYNTAX INTEGER (0..4)
ACCESS not-accessible
```

STATUS optional

DESCRIPTION

"This parameter is place holder for NMC CFM file CRC."

::= { fileHdr 6 }

slotHdr OBJECT IDENTIFIER ::= { file 2 }

cfmSlotNum OBJECT-TYPE

SYNTAX INTEGER (0..4)

ACCESS not-accessible

STATUS optional

DESCRIPTION

"This parameter is place holder for NMC CFM file Slot Number in Slot Header."

::= { slotHdr 1 }

cfmSlotDataOwner OBJECT-TYPE

SYNTAX INTEGER{

snmp(1),

opequeue(2)

}

ACCESS not-accessible

STATUS optional

DESCRIPTION

"This parameter is place holder for Which tells the SLOT data belongs to NMC or A NAC."

::= { slotHdr 2 }

cfmSlotHWtype OBJECT-TYPE

SYNTAX INTEGER (0..4)

ACCESS not-accessible

STATUS optional

DESCRIPTION

"This parameter is place holder Which tells the Hardware Type of the slot ."

::= { slotHdr 3 }

cfmSlotSWtype OBJECT-TYPE

SYNTAX INTEGER (0..4)

ACCESS not-accessible

STATUS optional

DESCRIPTION

"This parameter is place holder Which tells the Software Type of the slot ."

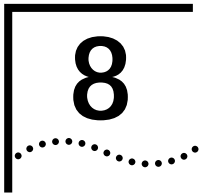
::= { slotHdr 4 }

```
cfmSlotDataSize OBJECT-TYPE
SYNTAX INTEGER (0..4)
ACCESS not-accessible
STATUS optional
DESCRIPTION
"This parameter is place holder for NMC CFM file Slot Data
Size."
::= { slotHdr 5 }

cfmSlotModuleVersion OBJECT-TYPE
SYNTAX INTEGER (0..4)
ACCESS not-accessible
STATUS optional
DESCRIPTION
"This parameter is place holder for NMC CFM file Slot's
Module Version."
::= { slotHdr 6 }

cfmSlotCRC OBJECT-TYPE
SYNTAX INTEGER (0..4)
ACCESS not-accessible
STATUS optional
DESCRIPTION
"This parameter is place holder for NMC CFM file SLOT's
CRC."
::= { slotHdr 7 }

END
```

GW MIB

This chapter contains detailed information about the Gateway MIB (GW MIB).

Purpose of the MIB	The GW MIB is an enterprise-specific MIB that was designed to provide trap enables for the NETServer gateway card.
Products using this MIB	This MIB is used by the NETServer gateway card.
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.gw (1.3.6.1.4.1.429.1.18)</p>
Configuration Data	This MIB contains objects to enable traps on the NETServer gateway cards.
Tables	These GW MIB is comprised of the gwTeTable.
Trap enables	The gwTeTable provides objects for configuring traps on the NETServer card.

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

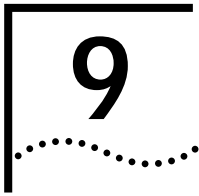
Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Traps

The GW MIB contains these trap enables:

Trap Enable	Generates this trap
gwTegwNetworkFailed	gwNetworkFailed(66)
gwTegwNetworkRestored	gwNetworkRestored(67)
gwTegwIpAddress	read only - does not enable a trap
gwTeArNetFailed	<p>This object stores the AutoResponse script that triggers when a NetFailed event occurs.</p> <p>This object does not enable a specific trap.</p>
gwTeArNetRestored	<p>This object stores the 'AutoResponse script that triggers when a NetRestored event occurs.</p> <p>This object does not enable a specific trap.</p>

Refer to the chapter titled *Trap Reference* for additional information about traps.



HiPer DSP TRUNK MIBs

This chapter contains detailed information about the MIBs that apply to HiPer DSP trunk applications, including T1, E1, PRI, ISDN, and R2.

Overview

This chapter includes explanations of these MIBs:

- HDR2
- RDS0
- RDS1
- T1H

The dual trunk cards use several different MIBs to configure settings, enable traps, provide statistics, and complete other management functions. The applicable MIBs depend upon what software is loaded on the card.

Application	Applicable MIBs
HiPer DSP Channelized T1/PRI	DSX1 (RFC 1406), RDS1, RDS0, T1H
HiPer DSP E1/R2	HDR2

Checking Span Status with HiPer DSP Trunk MIBs

Several of the tables and objects within these MIBs are useful for checking span status.

This list is a starting point for checking current span status. Locate other objects in the MIB trees to meet your specific needs.

Statistics	Recommended Table
General span status	usrds1StatTable dsx1ConfigTable (RFC 1406)

Statistics	Recommended Table (continued)
Interval status	dsx1IntervalTable
	dsx1CurrentTable
	dsx1TotalTable
Trap enables	usrds1EventCfgTable

HDR2 MIB

Purpose of the MIB The HDR2 MIB provides objects for managing E1/R2 features of the HiPer DSP NAC.

Products using this MIB This MIB is used by the HiPer DSP for E1/R2 applications.

Registration ID This MIB is an Internet Enterprise MIB registered under the USR enterprise node.

iso.org.dod.internet.private.enterprises.usr.common.hdr2
(1.3.6.1.4.1.429.4.48)

Configuration Data The HDR2 MIB contains three tables used to control the configuration of the E1/R2 HiPer DSP card. The configuration tables hdr2Cfg and hdr2InCatMap are used to configure the R2 signalling protocol at the span level.

- MIB Tables** The HDR2 MIB contains these tables:
- hdr2CfgTable (configuration)
 - hdr2CatMapTable (call categories)
 - hdr2TeTable (trap enables)

Configuration (hdr2CfgTable)

The hdr2CfgTable contains objects for configuring E1/R2 parameters, including signalling and signal timing.

Call categories (hdr2CatMapTable)

The `hdr2CatMapTable` contains objects for configuring the meaning of the fifteen available call categories for the E1/R2.

When using the R2 protocol, the only information that can be transferred about the type of call being made is the Calling Party Category (CPC). This is transmitted from the originating side as one of fifteen possible register signals, numbered II-1 to II-15 (or ii-1 to ii-15). There are four possible call types (analog, digital, test, maintenance). Because the mapping between call types and register signals can be different in different countries, objects are provided to set up mapping.

On the originating (outgoing) side, configure one register signal for each of the four call types using `hdr2CfgOutCatAnalog`, `hdr2CfgOutCatDigital`, `hdr2CfgOutCatTest`, and `hdr2CfgOutCatMaintenance`. For example, if `hdr2CfgOutCatAnalog` is set to **II-2**, the CPC signal sent will be **II-2** when an analog call is made.

On the answering (incoming) side, configure one call type for each of the fifteen register signals that could be received using the `hdr2InCatMapTable` objects. For example, if the CPC signal II-2 is received for an incoming call, the `hdr2InCatMapTable` entry for index 2 is checked to determine what type of call is being set up.

Trap enables (hdr2TeTable)

The `hdr2TeTable` provides objects for configuring E1/R2-related traps on the HiPer DSP.

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

Command	Description
<code>enableTrap (1)</code>	Enabling the trap allows the NMC to send the trap to the MS alarm server.
<code>disableAll (2)</code>	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
<code>enableLog (3)</code>	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
<code>enableAll (4)</code>	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Traps The HDR2 MIB contains these trap enable objects:

Trap Enable	Generates this trap
hdr2TeMultiFrame	uds1MultiFrame(104)
hdr2TeMultiFrameClr	uds1RemoteMultiFrame(105)
hdr2TeRemMultiFrame	uds1MultiFrameClr(106)
hdr2TeRemMultiFrameClr	uds1RemoteMultiFrameClr(107)

Refer to the chapter titled *Trap Reference* for additional information about traps.

RDSO MIB	
Purpose of the MIB	RDSO provides objects for managing features of a HiPer DSP span.
Products using this MIB	<p>This MIB is used in whole or in part by these software applications:</p> <ul style="list-style-type: none"> ■ T1/PRI ■ E1/PRI ■ E1/R2
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.rdsO (1.3.6.1.4.1.429.1.28)</p>
Configuration Data	The RDSO MIB contains tables and objects for configuring span-level parameters.
MIB Tables	<p>The RDSO MIB includes these tables:</p> <ul style="list-style-type: none"> ■ usrdsOConfigTable (configuration) ■ usrdsOStatTable (status) ■ usrdsOCmdTable (commands) <p>Configuration (usrdsOConfigTable)</p> <p>Use the objects in the usrdsOConfigTable to configure each DSO timeslot. The objects in this table are self-explanatory from the MIB text.</p> <p>Status (usrdsOStatTable)</p> <p>Use the objects in the usrdsOStatTable to obtain statistics about each DSO timeslot. The objects in this table are self-explanatory from the MIB text.</p> <p>Commands (usrdsOCmdTable)</p> <p>The usrdsOCmdTable contains objects used to issue and force commands to the NAC, as well as to check the results and interpret the codes resulting from commands.</p>

Command function The *usrdsOCmdFunction* object is used for basic commands. This table describes its use.

Command	Description
noCommand(1)	This is the default value for <i>usrdsOCmdFunction</i> .
disconnect(2)	Use this command to drop the call on this channel.
callIgnore(3)	Use this command to set the channel to ignore incoming calls.
dsOCmdInService(4)	Use this command to return the channel to an in-service state.
dsOCmdSoftBusyOut(5)	Use this command to wait for the channel to go idle, and then set the channel to a busy state.
dsOCmdHardBusyOut(6)	Use this command to drop any calls on the channel and set the channel to a busy state.

RDS1 MIB

Purpose of the MIB RDS1 is a span-level (trunk-level) MIB used to configure parameters and events and view statistics on a HiPer DSP span.

Products using this MIB This MIB is used in whole or in part by these software applications:

- T1/PRI
- E1/PRI
- E1/R2

Registration ID This MIB is an Internet Enterprise MIB registered under the U. S. Robotics enterprise node.

iso.org.dod.internet.private.enterprises.usr.nas.rds1
(1.3.6.1.4.1.429.1.27)

Configuration Data The RDS1 MIB contains tables and objects for configuring span-level parameters.

MIB Tables The RDS1 MIB includes these tables:

- *usrds1ConfigTable* (configuration)
- *usrds1StatTable* (status)
- *usrds1CmdTable* (commands)
- *usrds1EventCfgTable* (event configuration)

Configuration (*usrds1ConfigTable*)

The *usrds1ConfigTable* extends the DSX1 MIB configuration table as defined in RFC1406. The objects in this table are self-explanatory from the MIB text.

Receiver gain objects The *usrds1ConfigTable* contains two objects for configuring parameters related to receiver gain: *usrds1CfgRcvGain* and *usrds1StatReceiverGain*. The two parameters are entirely separate from each other.

The object *usrds1CfgRcvGain* establishes the limit values for the receiver's gain. For example, "43db" represents a greater amplification than "12db". The gain limit provides an increased noise margin for shorter distances, so set this object to the minimal required value. If you do not configure this object, the status object will only reflect the length of the cable.

The object *usrds1StatReceiverGain* represents the line attenuation value reported at the receiver.

The only relationship between these two objects is that *usrds1CfgRcvGain* should be set to a value larger than that reported by *usrds1StatReceiverGain*. At power on, set *usrds1CfgRcvGain* to maximum. Then, based on the status of *usrds1StatReceiverGain*, reduce the gain limit to the required value.

Status (*usrds1StatTable*)

Use the objects in the *usrds1StatTable* to obtain DS1 status information. There is no corresponding table in the DSX1 MIB defined by RFC1406. The objects in this table are self-explanatory from the MIB text.

Commands (usrds1CmdTable)

The *usrds1CmdTable* contains objects used to issue and force commands to the NAC, as well as to check the results and interpret the codes resulting from commands.

Command function The *usrds1CmdFunction* object is used for basic commands. This table describes its use.

Command	Description
noCommand(1)	This is the default value for <i>usrds1CmdFunction</i> .
forceReceiverReframe(2)	Use this command when frames are out-of-sync to reframe to the new frame type.
inService(3)	Use this command to put a DS1 in service and allow calls into the span.
localOutOfService(4)	If the DS0 is disabled locally on the NAC, use this command to send an off-hook message to the Telco to inform that a DS0 or span is busied out and request the following call be sent to the next channel or span. You can also use this command to prevent any calls from coming in on a particular span.
disconnect(5)	Use this command to disconnect a user on a DS0 or span. The DS0 or span remains in "service" mode after disconnecting. This command is useful for doing maintenance on a span.
enterDChaDisConnMaintMode(6)	Not used. Not included in Total Control Manager.
exitDChaDisConnMaintMode(7)	Not used. Not included in Total Control Manager.
enterBlueAlmMaintMode(8)	Not used. Not included in Total Control Manager.
exitBlueAlmMaintMode(9)	Not used. Not included in Total Control Manager.

Event configuration (usrds1EventCfgTable)

The usrds1EventCfgTable provides objects for configuring span-level traps.

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Traps The RDS1 MIB contains these trap enables:

Trap Enable	Generates this trap
usrds1EventYellowAlarm	yellowAlarm (22)
usrds1EventRedAlarm	redAlarm (23)
usrds1EventLossOfSignal	lossOfSignal (24)
usrds1EventAlarmIndSignal	alarmIndicationSignal (25)
usrds1EventYellowAlarmClr	yellowAlarmClear (50)
usrds1EventRedAlarmClr	redAlarmClear (51)
usrds1EventLossOfSgnlClr	lossOfSignalClear (52)
usrds1EventAlrmIndSgnlClr	alarmIndicationSignal (53)
usrds1EventContCrcAlrm	contCrcAlarm (59)
usrds1EventContCrcAlrmClr	contCrcAlarmClear (60)
usrds1EventPhysStateChng	phyStateChng (61)
usrds1EventDs0InSrv	ds0InService (72)
usrds1EventDs0OutOfSrv	ds0OutOfService (73)
usrds1EventDs0ServStateMt	changedtoMainSrvsStat (90)
usrds1EventloopBack	loopbackTrap (92)
usrds1EventloopBackCleard	loopbackcleared (91)
usrds1EvttelcoAbnormalRsp	telcoAbnormalResp (93)

Trap Enable	Generates this trap (continued)
usrds1EventDChanInSrv	dChanInSrv (70) (PRI only)
usrds1DchanOutOfSrv	dChanOutOfService (71) (PRI only)
usrds1EventDs0InConnFail	usrds1InCallFailedEvent (114) usrds1dspInCallFailedEvent (120)
usrds1EventDs0OutConnFail	usrds1OutCallFailedEvent (115) usrDs1dspOutCallFailedEvent (121)
usrds1EventCallArrive	callArriveEvent (76) callArriveEventHdsp (118)
usrds1EventCallTerm	callTermNormEvent (78) callTerminateEventHdsp (119)
usrds1EventNfasDchSwStart	rds1EvDchSwitchOverStart (122) (PRI only)
usrds1EventNfasDchSwEnd	rds1EvDchSwitchOverEnd (123) (PRI only)
usrds1EventNfasDchSwfail	rds1EvDchSwitchOverFailure (124) (PRI only)

Refer to the chapter titled *Trap Reference* for additional information about traps.

T1H MIB

Purpose of the MIB

T1H is a NAC-level MIB used to configure call routing on a HiPer DSP. This MIB is similar to IDT1 in its call routing parameters. T1H does not contain pool routing parameters. Use the commands within this MIB to:

- Save template settings to NVRAM
- Restore templates to factory defaults
- Restore templates to NVRAM settings
- Install template settings into modem channels

Products using this MIB

This MIB is used at the card-level by the HiPer DSP.

Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the U.S. Robotics enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.t1h (1.3.6.1.4.1.429.1.26)</p>
Configuration Data	<p>The T1H MIB contains tables and objects for configuring span-level parameters</p>
MIB Tables	<p>The T1H MIB contains these tables:</p> <ul style="list-style-type: none"> ■ t1hCfgTable (configuration) ■ t1hCrTable (call routing) ■ t1hCmdTable (commands) ■ t1hTeTable (trap enables) <p>Configuration (t1hCfgTable)</p> <p>Use the objects in to the t1hCfgTable to configure NAC-level ISDN parameters to control timeslot allocation on the HiPer DSP.</p> <p>Most objects in this table are self-explanatory from the MIB text.</p> <p><i>t1hCfgMdmRoutingMethod</i> Use <i>t1hCfgMdmRoutingMethod</i> to determine how all HiPer DSP modem channels are allocated to timeslots.</p> <p><i>t1hCfgLogCallStatGrpSel</i> To minimize network traffic from HiPer DSP modems, trap statistics (objects) are piggybacked to the trap PDU sent to the NMC when a trap generates. A minimum default of statistics is automatically reported to the NMC. Use <i>t1hCfgLogCallStatGrpSel</i> to specify additional statistics that will also be reported with call-related traps. This object is useful for gathering additional accounting usage and troubleshooting information.</p>

Options in *t1hCfgLogCallStatGrpSel* are listed in the MIB as groups. This table provides additional information about the statistics that are included in each group:

Event	Enumeration group	Piggybacked statistics
ctIncomingConnectionEstablished(54) Incoming Connection Established	group1Only(1) – standard statistics	Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriCardSlot</i>) Call start time/date
ctOutgoingConnectionEstablished(55) Outgoing Connection Established	group1Only(1) – standard statistics	Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriCardSlot</i>) Call start time/date
ctIncomingConnectionTerminated(56) Incoming Connection Terminated	group1Only(1) – standard statistics	Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriCardSlot</i>) Connection termination reason Last number dialed in Last calling party number Call start time/date Call end time/date
	group1And2(2) – standard and data path statistics	Characters sent Characters received Characters lost Octets sent Octets received Blocks sent Blocks received Blocks resent

Event	Enumeration group	Piggybacked statistics (continued)
	group1And3(3) – standard and DSP statistics	Initial TX link data rate Final TX link data rate Retrains requested Retrains granted BLER count Fallback count Link timeout count Link NAK count Initial RX link data rate Final RX link data rate Gain hit count (<i>mdmCsGainHitCount</i>)
	group1And4(4) – standard and call control statistics	Sync/async mode Modulation type Error control type Fallback mode Compression type Call type Orig/Ans mode (<i>mdmCsOriginateAnswer</i>) Modem status (<i>mdmCsStatus</i>) X2/V.90 status (<i>mdmX2Status</i>) Operational status (<i>uchasEntityOperStatus</i>)
ctOutgoingConnectionTerminated(57) Outgoing Connection Terminated	group1Only(1) – standard statistics	Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriCardSlot</i>) Call start time/date
inconnectAttemptFailure(86) Incoming Connection Attempt Failed	group1Only(1) – standard statistics	Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriCardSlot</i>) Connection termination reason Last number dialed in Last calling party number Call start time/date Call end time/date
	group1And2(2) – standard and data path statistics	Characters sent Characters received Characters lost Octets sent Octets received Blocks sent Blocks received Blocks resent

Event	Enumeration group	Piggybacked statistics (continued)
	group1And3(3) – standard and DSP statistics	Initial TX link data rate Final TX link data rate Retrans requested Retrans granted BLER count Fallback count Link timeout count Link NAK count Initial RX link data rate Final RX link data rate Gain hit count (<i>mdmCsGainHitCount</i>)
	group1And4(4) – standard and call control statistics	Sync/async mode Modulation type Error control type Fallback mode Compression type Call type Orig/Ans mode (<i>mdmCsOriginateAnswer</i>) Modem status (<i>mdmCsStatus</i>) X2/V.90 status (<i>mdmX2Status</i>) Operational status (<i>uchasEntityOperStatus</i>)
outconnectAttemptFailure(87) Outgoing Connection Attempt Failed	group1Only(1) – standard statistics	Failure to connect reason Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriSlot</i>) Call start time/date
connectTimerExpired(14) Connection Time Limit Expired	group1Only(1) – standard statistics	Connect time limit Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriSlot</i>) Call start time/date
dteTransmitIdleData(15) DTE TX Data Idle	group1Only(1) – standard statistics	DTE TX idle threshold Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriSlot</i>) Call start time/date

Event	Enumeration group	Piggybacked statistics (continued)
blerCountAtThreshold(18) BLER Threshold Exceeded	group1Only(1) – standard statistics	BLER threshold Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriSlot</i>) Call start time/date
fallbackCountAtThreshold(19) Fallback Threshold Exceeded	group1Only(1) – standard statistics	Fallback threshold Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriSlot</i>) Call start time/date
dteRingNoAnswer(29) DTE Ring No Answer	group1Only(1) – standard statistics	Time stamp Call reference number Q.931 call ID DS1 ID DS0 ID Card slot (<i>mdmCsPriSlot</i>) Call start time/date

Call routing (t1hCrTable)

Use the objects in the t1hCrTable to configure ISDN call routing parameters on the HiPer DSP.

Commands (t1hCmdTable)

The t1hCmdTable contains objects used to issue and force commands to the NAC, as well as to check the results and interpret the codes resulting from commands.

Command function The *t1hCmdTable* object is used for basic commands. This table describes its use.

Command	Description
noCommand(1),	This is the default value for <i>t1hCmdTable</i> .
restoreT1E1AndMdmDefaults(2)	<p>Use this command to override an applied template and restore all factory defaults to the entire card settings. This command will not change the template settings.</p> <p>Query <i>mdmMaChangeIndicator</i> to see the result of this command.</p> <p>If <i>nmcPowerUpAutoCfgEnable</i> is enabled, the NMC's NVRAM will overwrite the defaults upon NAC installation, NMC installation, or chassis power-up..</p>
restoreT1E1FromDefault(3)	<p>Use this command to override an applied template and restore all factory defaults to the span settings. This command will not change the template settings.</p> <p>Query <i>mdmMaChangeIndicator</i> to see the result of this command.</p> <p>If <i>nmcPowerUpAutoCfgEnable</i> is enabled, the NMC's NVRAM will overwrite the defaults upon NAC installation, NMC installation, or chassis power-up.</p>
restoreMdmFromDefault(4)	<p>Use this command to override an applied template and restore all factory defaults to the modem settings. This command will not change the template settings.</p> <p>Query <i>mdmMaChangeIndicator</i> to see the result of this command.</p> <p>If <i>nmcPowerUpAutoCfgEnable</i> is enabled, the NMC's NVRAM will overwrite the defaults upon NAC installation, NMC installation, or chassis power-up.</p>
saveT1E1AndMdmNvram(5)	<p>Use this command to save all card settings to the NAC's NVRAM. This command will not change the template settings.</p> <p>If <i>nmcPowerUpAutoCfgEnable</i> is enabled, the NMC's NVRAM will overwrite the NAC's NVRAM upon NAC installation, NMC installation, or chassis power-up.</p>

Command	Description (continued)
saveT1E1Nvram(6)	<p>Use this command to save all span settings to the NAC's NVRAM. This command will not change the template settings.</p> <p>If <i>nmcPowerUpAutoCfgEnable</i> is enabled, the NMC's NVRAM will overwrite the NAC's NVRAM upon NAC installation, NMC installation, or chassis power-up.</p>
saveMdmNvram(7)	<p>Use this command to save all modem settings to the NAC's NVRAM. This command will not change the template settings.</p> <p>If <i>nmcPowerUpAutoCfgEnable</i> is enabled, the NMC's NVRAM will overwrite the NAC's NVRAM upon NAC installation, NMC installation, or chassis power-up.</p>
restoreT1E1AndMdmNvram(8)	<p>Use this command to restore all card settings from the NAC's NVRAM. This command will not change the template settings.</p> <p>If <i>nmcPowerUpAutoCfgEnable</i> is enabled, the NMC's NVRAM will overwrite the NAC's NVRAM upon NAC installation, NMC installation, or chassis power-up.</p>
restoreT1E1Nvram(9)	<p>Use this command to restore all span settings from the NAC's NVRAM. This command will not change the template settings.</p> <p>If <i>nmcPowerUpAutoCfgEnable</i> is enabled, the NMC's NVRAM will overwrite the NAC's NVRAM upon NAC installation, NMC installation, or chassis power-up.</p>
restoreMdmNvram(10)	<p>Use this command to restore all modem settings from the NAC's NVRAM. This command will not change the template settings.</p> <p>If <i>nmcPowerUpAutoCfgEnable</i> is enabled, the NMC's NVRAM will overwrite the NAC's NVRAM upon NAC installation, NMC installation, or chassis power-up.</p>
softwareReset(11)	<p>Use this command to complete a software reset on the entire card. This command is more efficient than using the commands in <i>mdmCdFunction</i>.</p>
restoreDefaultUIPassword(12)	<p>Use this command to restore the console port password to the factory default (which is a carriage return).</p>
restorecfg1todef1t(13)	<p>Use this command to restore all settings stored in template 1 to the factory defaults. (template-level)</p>

Command	Description (continued)
restorecfg2todeflt(14)	Use this command to restore all settings stored in template 2 to the factory defaults. (template-level)
restorecfg3todeflt(15)	Use this command to restore all settings stored in template 3 to the factory defaults. (template-level)
restorecfg4todeflt(16)	Use this command to restore all settings stored in template 4 to the factory defaults. (template-level)
saveCfg1toNvram(17)	Use this command to save all settings stored in template 1 to the NAC's NVRAM. (template-level)
saveCfg2toNvram(18)	Use this command to save all settings stored in template 2 to the NAC's NVRAM. (template-level)
saveCfg3toNvram(19)	Use this command to save all settings stored in template 3 to the NAC's NVRAM. (template-level)
saveCfg4toNvram(20)	Use this command to save all settings stored in template 4 to the NAC's NVRAM. (template-level)
restoreCfg1FromNvram(21)	Use this command to restore all settings stored in template 1 to the template configuration saved in the NAC's NVRAM. (template-level)
restoreCfg2FromNvram(22)	Use this command to restore all settings stored in template 2 to the template configuration saved in the NAC's NVRAM. (template-level)
restoreCfg3FromNvram(23)	Use this command to restore all settings stored in template 3 to the template configuration saved in the NAC's NVRAM. (template-level)
restoreCfg4FromNvram(24)	Use this command to restore all settings stored in template 4 to the template configuration saved in the NAC's NVRAM. (template-level)
refreshCfg1Chans(25)	Use this command to install all template 1 settings into the modem channels mapped to that template. (template-level)
refreshCfg2Chans(26)	Use this command to install all template 2 settings into the modem channels mapped to that template. (template-level)

Command	Description (continued)
refreshCfg3Chans(27)	Use this command to install all template 3 settings into the modem channels mapped to that template. (template-level)
refreshCfg4Chans(28)	Use this command to install all template 4 settings into the modem channels mapped to that template. (template-level)

Trap enables (t1hTeTable)

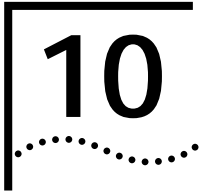
The t1hTrapEnaTable provides one object for configuring card-level traps on the HiPer DSP.

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Traps The T1H MIB contains the *t1hTeResetByDTE* object, which enables the modemResetByDte trap.

Refer to the chapter titled *Trap Reference* for additional information about traps.



HIST MIB

This chapter contains detailed information about the HISTORY MIB (HIST MIB).

Purpose of the MIB	The HIST MIB provides statistical information of modem activities in both the current time interval and in user-specified intervals. These statistics provide traffic-related information about your chassis, and can be used for network management reports and functions.
---------------------------	---

Products using this MIB	This MIB is used by Quad and HiPer DSP modem cards.
--------------------------------	---

Registration ID	This MIB is an Internet Enterprise MIB registered under the USR enterprise node.
------------------------	--

iso.org.dod.internet.private.enterprises.usr.nas.file
(1.3.6.1.4.1.429.1.33)



The text file for this MIB is named mdmHist.txt.

Configuration Data	This MIB provides statistics information for specific time intervals. The default recommended interval is 15 minutes. This interval provides 26 hours of modem statistics.
---------------------------	--

The History MIB is supported by SNMP browsers only. You cannot configure the time interval or read statistics from Total Control Manager.

MIB Tables

The History MIB contains these tables:

- `mdmNacHistCurrentTable` (current card-level statistics)
- `mdmNacHistIntervalTable` (interval card-level statistics)
- `mdmHistCurrentTable` (current modem-level statistics)
- `mdmHistIntervalTable` (interval modem-level statistics)

General Information about the Tables

The History MIB contains two table types: current and interval.

Current tables

Current tables provide “real-time” statistics for the current time interval. This information is useful for trouble clearing operation problems.

Single-indexed tables are provided for card-level and modem-level statistics. Each **get** is timestamped to indicate when the interval started. The timestamp is based on the NMC's GMT setting (*nmcGmTime*).

Interval tables

Interval tables provide history statistics about *each* chassis modem for 104 time intervals. These statistics are useful when tracking modem traffic and charting network trends.

Information is stored in a circular Last In First Out (LIFO) buffer in the NMC's RAM. The buffer clears when the chassis reboots. This means the interval table will only provide data since the last reboot. For example, if the table provides only 16 15-minute intervals of data, the chassis rebooted approximately four hours earlier.

Double-indexed tables are provided for card-level and modem-level statistics. The most recent completed interval is represented as “1”; the highest numbered interval is the oldest data.

The default recommended interval is 15 minutes, providing 26 hours of interval data.

Because of the large volume of data stored in the interval tables, it is highly recommended that you do not perform a **get** to obtain this data directly from the MIB tables. Use the **bulkFileUpload** NMC command to upload the .hst file to a management station. The .hst file contains interval data for both the card-level and the modem-level.

The format of the .hst file is based upon the file MIB template format.

Traps

No trap enables are included in this MIB.

HDM MIB

This chapter contains detailed information about the High Density Modem MIB (HDM MIB).

Purpose of the MIB

The HDM MIB provides scalability for configuring a HiPer DSP-based system. It can be thought of as a “configuration template MIB”.

The HDM MIB allows you to define four configuration templates that can be assigned to individual modem channels on a HiPer DSP NAC. Conceptually, the HDM MIB is a template-only MIB that can be used to quickly configure the HiPer DSP if you do not require a unique configuration on each channel. Since most customers use a single configuration for all modems on a card, this MIB provides a means of reducing the number of objects that must be configured to implement a change on all of the modems on the card.

The MIB contains four configuration templates that can be assigned to any combination of modem channels on a HiPer DSP NAC. The current implementation allows for four templates per card, which also provides a means of using combinations of templates within a single card or storage of special use templates (such as a test configuration).

Comparing the HDM MIB to the MDM MIB

The MDM MIB provides objects to configure each modem on each channel. You may configure each channel in a completely different way.

The HDM MIB provides objects to configure all of the modems contained (grouped) within one of four templates. All channel settings for each modem assigned to that template will be identical. The HDM MIB does not provide any objects for setting individual channels.

Products using this MIB

This MIB is used by the HiPer DSP.

Registration ID

This MIB is an Internet Enterprise MIB registered under the USR enterprise node.

iso.org.dod.internet.private.enterprises.usr.nas.hdm
(1.3.6.1.4.1.429.1.22)

Configuration Data

This MIB is comprised almost entirely of objects used to control the configuration of modem channels. The implementation in the HiPer DSP provides separate storage space for the templates controlled by the HDM MIB and the configuration of the individual channels. The channels can be controlled directly by the MDM MIB. In the current implementation, a command must be issued to the HiPer DSP in order for changes to the templates take effect on the channels. These commands are provided in the command table of the T1H MIB, which is a card-level MIB associated with the HiPer DSP NAC.

The NMC supports the four different configurations that can be mapped to the 24 (T1) or 30 (E1) channels in the HiPer DSP NAC. These four configurations are referred to as card-level configurations. They are also commonly called templates or profiles.

In most cases, a single configuration is applied to all 24 or 30 channels on the HiPer DSP NAC. You may also choose up to four configurations and assign any number of channels to the first configuration, any number of channels to the second configuration, and so on until all channels are assigned to one of the card-level configurations.

All channels on the HiPer DSP NAC must be mapped. That is, the NMC must have a card-level configuration for each channel. This card-level configuration is from the NMC point of view. It is possible to set alternate configurations on a channel-by-channel basis through the HiPer DSP card.

Configuring a Template

You must use objects from three different MIBs to configure a HiPer DSP template. This section explains the process.

- 1 Access the `mdmMaTable` in the MDM MIB.
- 2 Each group in this table represents a template. You may configure up to four templates. Assign the appropriate modems to a group.
- 3 Access the HDM MIB to assign values to the template. Enable objects as appropriate.



This step only changes the template configurations and does not change the actual modem configurations.

- 4 Access the T1H MIB. Use the command **saveCfg(1-4)toNvram** to save the template configuration to the HiPer DSP NAC's NVRAM.
- 5 Use the command **refreshCfg(1-4)Chans** to install the template settings into the modem channels that you mapped to the template in step 2.
- 6 Access the `mdmMaTable` in the MDM MIB. Query `mdmMaChangeIndicator` to ensure the channels match the template settings.

Changing a Channel's Configuration

The NMC only stores the four card-level configurations in NVRAM. These are the configurations that are affected when issuing SAVE, RESTORE, and DEFAULT commands to the NMC. The NMC also stores the configuration mapping object for the HiPer DSP.

In order for any configuration changes to operate on a channel(s), you must set the HDM mapping object after making these changes. If you previously mapped channels to a given configuration and then change a value in that channel's template, you must either issue a SET request for the mapping object or use the commands in the MDM MIB object `mdmChannelConfig`.

You may change a single channel's configuration through the HiPer DSP user interface, Total Control Manager or another SNMP manager. (Refer to the HiPer DSP documentation, the Total Control Manager online help, and the *NMC Parameter Reference*.) These changes will only be saved to the HiPer DSP's NVRAM and will not be recognized by the NMC.

When changing a single channel's configuration, the HiPer DSP modem must set a "change indicator" for that channel. This "change indicator" will tell the operator that the configuration currently being used by that

channel is different from the card-level configuration saved for that channel. This change indicator is contained in the `mdmMaTable` in the MDM MIB. The objects are *mdmMaChangeIndicator* and *mdmMaChannelConfig*. Refer to the chapter in this document titled "MDM MIB" for additional information.



Card-level configurations are saved in NMC NVRAM. The "change indicator" is not saved in NVRAM; it is a read-only information byte. When the NMC uses card-level configurations to do operations such as auto-configure, the "change indicator" will not be set, indicating no difference.

Enabling traps You must use objects from two MIBs to enable template-level traps.

Follow this procedure to enable HDM template-level traps:

- 1 Enable the appropriate traps from the `hdmTeTable`.
- 2 Access the T1H MIB.
- 3 Use the **refreshCfg(1-4)Chans** command to install the trap enables into the template.

MIB Tables

This MIB is a subset cloned from the current MDM MIB. It contains the same tables used for configuration of individual modem ports as the MDM MIB, but with two indices. The HDM MIB also maintains the same object groups as the MDM MIB, with different object names. While the prefix has changed (*mdmLiDialPause* becomes *hdmLiDialPause*), the associated values remain the same.

These are exceptions to the naming convention of MDM and HDM objects:

MDM MIB object name	HDM MIB object name
<code>mdmScLinkRateSelect</code>	<code>hdmScLinkRateMax</code>
<code>mdmScLinkRateAmpU</code>	<code>hdmScLinkRateMin</code>

Indexing The HDM MIB is a double-indexed table. It is indexed with a HiPer DSP card-level entity and a "template" index. The first index is the card-level entity addressed as 1000*slot. The second index is an INTEGER whose valid values are 1,2,3,4. Used in conjunction with a new object in the MDM.MIB a template number can be assigned to each modem channel.

For example: The third modem in slot 13 would be numbered as $1000 \times 13 + 3 = 13004$.

The MIB is stripped of objects that do not apply to a card-level entity, such as the call statistics objects, and those that can not be supported by the HiPer DSP, like many of the DTE port configuration parameters.

The HDM MIB includes support for the *imdmCcTable* from the IMDM MIB.

HDM MIB Tables These tables are included in the HDM MIB:

- hdmLiTable (line interface)
- hdmDcTable (data compression)
- hdmScTable (signal converter)
- hdmCcTable (call control)
- hdmEcTable (error correction)
- hdmEtTable (event threshold)
- hdmTeTable (trap enables)
- hdmArTable (AutoResponse)
- hdmDiTable (DTE interface)
- hdmIcTable (ISDN call control)

Use the objects within these tables to configure all parameters for every modems assigned to a specific template. In general, the table and object definitions are identical to similarly-named tables and objects within the MDM MIB and IMDM MIB (hdmIcTable only). The next section explains any differences from the MDM MIB.

**DTE Interface
(hdmDiTable)**

The hdmDiTable supports an analog DTE interface on the Quad modems. The HiPer DSP does not contain a DTE interface. Rather, the HiPer DSP contains a packetbus interface to the NAC. The hdmDiTable contains a subset of the mdmDiTable parameters.

Most of the objects in the hdmDiTable are self explanatory.

hdmDteNvramLock

Use *hdmDteNvramLock* to prevent configuration changes from being saved to the HiPer DSP's NVRAM. By default, this object is disabled.

For example, you may wish to enable this object if you do not want initialization strings from AT commands containing **@w** to be loaded to the NAC's NVRAM.

If you cannot save template changes to NVRAM, make sure *hdmDteNvramLock* is not enabled.

**AutoResponse
(hdmArTable)**

Not used in this release.

Traps

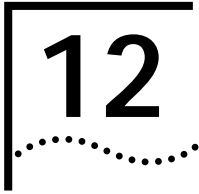
The HDM MIB provides access to a subset of the events that can be controlled on a channel level by the MDM MIB. You must use the templates to enable these traps.

The HDM MIB contains these trap enable objects:

Trap enable	Generates this trap
hdmTeInConnEstablished	incoming ConnectionEstablished(9)
hdmTeOutConnEstablished	outgoingConnectionEstablished(10)
hdmTeInConnTerminated	incomingConnectionTerminated(11)
hdmTeOutConnTerminated	outgoingConnectionTerminated(12)
hdmTeConnLimitExpired	connectTimerExpired(14)
hdmTeDteXmitDataIdle	dteTransmitDataIdle(15)
hdmTeBlerCountAtThresh	blerCountAtThreshold(18)
hdmTeFallbkCountAtThresh	fallbackCountAtThreshold(19)
hdmTePbActive	pktBusSessActive(30)
hdmTePbLost	pktBusSessLost(32)

Trap enable	Generates this trap (continued)
hdmTeDteRingNoAns	dteRingNoAnswer(29)
hdmTeInConnAttemptFail	inconnectAttemptFailure(86)
hdmTeOutConnAttemptFail	outconnectAttemptFailure(87)
hdmTePBClockLost	pktBusClockLost(68)
hdmTePBClockRestored	pktBusClockRestore(69)

Refer to the chapter titled *Trap Reference* for additional information about traps.



IMDM MIB

This chapter contains detailed information about the IMDM MIB.

Purpose of the MIB	The IMDM MIB provides management objects for features unique to ISDN modems.
Products using this MIB	This MIB is used by Quad and HiPer DSP modems configured for establishing ISDN connections.
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.imdm (1.3.6.1.4.1.429.1.19)</p>
Configuration Data	<p>The IMDM MIB contains two tables used to configure ISDN modem channels.</p> <p>The HiPer DSP modems only use the imdmCc Table for setting ISDN parameters at the channel level. HiPer DSP ISDN template-level parameters use objects in the hdmICcTable.</p>
MIB Tables	The IMDM MIB contains two tables: imdmCcTable (call control) and imdmLiTable (line interface).
Call Control (imdmCcTable)	The imdmCcTable provides objects for setting ISDN call control options. These parameters are related to establishing a connection, and enable or disable various protocols.

**Line Interface
(imdmLiTable)**

The imdmLiTable provides parameters for configuring line interface to ensure proper communications with the Telco central office.

Traps

No trap enable objects are included in this MIB.

13

IPGW MIB

This chapter contains detailed information about the IP Gateway MIB (IPGW MIB).

Purpose of the MIB

The IPGW MIB is an enterprise-specific MIB that provides management information for the IP gateway (IPGW) cards. It includes the IP gateway configuration tables, IP gateway command tables, and trap tables.

The NMC provides a proxy interface for NMC management support to the HiPer ARC similar to what is provided for earlier versions of NETServer. In addition, the IPGW MIB allows a minimum set of objects to be accessed that will provide a mechanism similar to the Auto Configure function for the HiPer ARC. The IPGW MIB also includes an object that can be used to disable SNMP access through the PNS's IP interfaces, restricting it to Telnet and MBP configuration access. Refer to the *HiPer NMC Parameter Reference* for additional information.

Products using this MIB

This MIB is used by IPGW cards.

Registration ID

This MIB is an Internet Enterprise MIB registered under the USR enterprise node.

iso.org.dod.internet.private.enterprises.usr.nas.ipgw
(1.3.6.1.4.1.429.1.13)

Configuration Data

Use objects in this MIB to configure IP addresses, trap destination, net mask, Ethernet framing, Ethernet interface name, Management Station IP, and default community string. This MIB also contains a trap enable object.

Tables

These tables are included in the IPGW MIB:

- `ipgwCfgTable` (configuration)
- `ipgwCmdTable` (commands)
- `ipgwTrapEnaTable` (trap enables)

IPGW configuration (`ipgwCfgTable`)

The `ipgwCfgTable` contains an entry for each of manageable IPGW cards in the chassis. It contains objects that reflect the current configuration of parameters that affect the operation of all the entities that reside on the given card. There is one IPGW card configuration table entry per IPGW card in the chassis.

IPGW commands (`ipgwCmdTable`)

The `ipgwCmdTable` contains an entry for each manageable IPGW card in the chassis. It provides objects used to issue and force commands, as well as to check the results and interpret the results. There is one IPGW card command entry per IPGW card in the chassis.



The ability to transfer the configuration file is not provided through the SNMP forwarding mechanism, but only through the bulk file transfer command as a FTP or TFTP transfer.

Command function

The `ipgwCmdFunction` object is used for basic IPGW functions. This table describes its use.

Command	Description
<code>noCommand(1)</code>	This is the default value for <code>ipgwCdFunction</code> .
<code>saveToNVRAM(2)</code>	Use this command to save all of the IPGW's current settings to the NVRAM on the IPGW card.
<code>restoreFromNVRAM(3)</code>	Use this command to restore all IPGW configuration settings to the values that were last saved to the NAC's NVRAM by the saveToNvram command.
<code>restoreFromDefault(4)</code>	Use this command to restore all IPGW settings to factory default values.

Command	Description (continued)
nonDisruptSelfTest(5)	Use this command to initiate a nondisruptive self test while the IPGW card is online.
disruptSelfTest(6)	Use this command to initiate a ndisruptive self test and take the IPGW card offline.
softwareReset(7)	Use this command to reset the IPGW card from a remote location.
lanLoopBack(8)	Use this command to initiaite a LAN loopback.
bulkFileUpload(9)	Use this command to upload the single configuration file.
bulkFileDownload(10)	Use this command to download the single configuration file.

Trap enables (ipgwTrapEnaTable)

The ipgwTrapEnaTable provides objects for configuring traps on the IPGW cards . There is one entry for each IPGW card in the chassis. A unique index identifies the IPGW card to which the trap enable objects pertain.

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Traps

The IPGW MIB contains the *ipgwTrapEnUiReset* trap enable object to generate a nacUserInterfaceReset(34) trap.

Refer to the chapter titled *Trap Reference* for additional information about traps.

MDM MIB

This chapter contains detailed information about the Modem MIB (MDM MIB).

Purpose of the MIB	The MDM MIB provides configuration management and performance information of all chassis modem entities.
Products using this MIB	This MIB is used by all modem NACs in the chassis.
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.mdm (1.3.6.1.4.1.429.1.6).</p>
Configuration Data	Use the objects in the MDM MIB to configure the chassis Quad modem entities. This MIB is used in conjunction with the HDM MIB to configure HiPer DSP modem entities.

MIB Tables

All tables in the MDM are indexed with the entity index of the entry in the entity table that corresponds to the same modem. When the NMC discovers a modem entity, the proxy agent adds a row for that modem to each of the modem MIB tables. When a card on which the given entity resides is removed from the chassis, these same rows are deleted from each of the MDM MIB tables.

The MDM MIB contains these tables:

- mdmIDTable (modem identification)
- mdmLiTable (line interface)
- mdmDcTable (data compression)
- mdmTfTable (test function)
- mdmDiTable (DTE interface)
- mdmScTable (signal converter)
- mdmCcTable (call control)
- mdmEcTable (error correction)
- mdmCsTable (call statistics)
- mdmEvTable (events)
- mdmEtTable (event threshold)
- mdmCdTable (commands)
- mdmTeTable (trap enables)
- mdmLsTable (link security)
- mdmHsTable (hub security)
- mdmArTable (auto response)
- mdmCeTable (cellular support)
- mdmStsTable (status)
- mdmMaTable (mapping)

Modem Identification (mdmIDTable)

Use the objects in the mdmIDTable to query about general modem identification. The objects in this table are self-explanatory from the MIB text.

- Line Interface (mdmLiTable)** Use the objects in the mdmLiTable to configure the operation of the modem's phone line interface.
- Data Compression (mdmDcData Compression)** This table only contains the object *mdmDcDataCompression*. Use this object to specify when and how data compression is enabled.
- ModemTests (mdmTf)** The modem tests are contained in several tables within mdmTf. Use these commands to perform various modem tests. Refer to the Quad Modem and HiPer DSP reference manuals for additional details about the modem tests.



Objects not listed in this section are not supported by Quad modems.

mdmTfTable

This table contains objects used for modem test functions. These objects are implemented by the Quad modems.

MIB Object	Description	Settings and use
<i>MdmTfTest</i>	Modem self test	S16 = 1; analog loopback test S16 = 4; ASCII test pattern S16 = 8; Remote digital loopback
<i>MdmTfTestTime</i>	&Tn (V54) test timer in seconds.	S18 = 0; by default used for V54 tests (analog, digital, and remote loopback).
<i>MdmTfV54</i>	&Tn V54 test	&T0 ; end tests &T1 ; Analog loopback (ALB) &T3 ; Digital loopback (DLB) &T4 ; Grant remote DLB &T5 ; Deny remote DLB &T6 ; Remote DLB &T7 ; Remote DLB with self test &T8 ; ALB with self test.

MIB Object	Description	Settings and use
<i>MdmTfV54Errors</i>	V54 loopback error test	No Settings. Use this object to query the error count (maximum 255) if incoming data doesn't match V54 test pattern.
<i>MdmTfATG</i>	Memory dump via NMC from the modems	Use the ATG command to query memory data dump starting at the address location defined as ATG = xxxx:yyyy (xxxx = segment value, yyyy = offset value). Total Control Manager is the recommended method of configuration.

mdmTfToneTable

The mdmTfToneTable contains objects used for modem tone tests. These objects are implemented by the Quad modems:

MIB Object	Description	Settings and use
<i>MdmTfTxFreq</i>	Modem Tx tone test frequency	Use this object to configure frequency levels. Total Control Manager is the recommended method of configuration. Transmit frequency limits are 300hz to 4000hz. &T9 = xx,yy (where xx = freq value, yy = negative amp value) command enables tone generation.
<i>MdmTfTxAmpl</i>	Modem Tx tone test amplitude	Use this object to configure amplitude levels. Total Control Manager is the recommended method of configuration. Transmit amplitude limits are 0 to -40. &T9 = xx,yy (where xx = frequency, yy = negative amplitude values) command enables tone generation.

<i>MdmTfRxFreq</i>	Modem Rx tone test frequency	Use this object to query received frequency levels. The &T10 command enables received tone test.
<i>MdmTfRxAmpl</i>	Modem Rx tone test amplitude	Use this object to query received amplitude levels. The b command enables the received tone test.

mdmTfRspndrTable

The mdmTfRspndrTable contains objects used for 105 Dial-in/Dial-out responder tests (remote testing through the NMC and Total Control Manager). Not all supported objects are used for the responder test in dial-in mode as enabled by S72=64 via RS-232 or NMC.



The HiPer DSP does not support responder tests.

These objects are supported by Quad modems:

MIB Object	Description	Settings and use
<i>MdmTf404FarNearLvl</i>	105 dialout responder test	Use this object to query the measured 404hz tone Far to Near dB level.
<i>MdmTf404NearFarLvl</i>	105 dialout/dial-in responder test	Set S72.6=1 to enable dialin responder test. Use this object to query the measured 404hz tone Near to Far dB level.
<i>MdmTf1004FarNearLvl</i>	105 dialout responder test	Use this object to query the measured 1004hz tone Far to Near dB level.
<i>MdmTf1004NearFarLvl</i>	105 dialout/dialin responder test	Set S72.6=1 to enable dialin responder test. Use this object to query the measured 1004hz tone Near to Far dB level.
<i>MdmTf2804FarNearLvl</i>	105 dialout responder test	Use this object to query the measured 2804hz tone Far to Near dB level

MIB Object	Description	Settings and use
<i>MdmTf2804NearFarLvl</i>	105 dialout/dial-in responder test	Set S72.6=1 to enable dialin responder test. Use this object to query the measured 2804hz tone Near to Far dB.
<i>MdmTfCmsgFarNearLvl</i>	105 dialout responder test	Use this object to query the C-message noise Far to Near dB level.
<i>MdmTfCmsgNearFarLvl</i>	105 dial-in/dialout responder test	Set S72.6=1 to enable dialin responder test. Use this object to query the measured C-message noise Near to Far dB level.
<i>MdmTfCnotchFarNearLvl</i>	105 dialout responder test	Use this object to query C-notch noise Far to Near dB level.
<i>MdmTfCnotchNearFarLvl</i>	105 dialout/dialin responder test	Set S72.6 = 1 to enable dialin responder test. Use this object to query C-notch noise Near to Far dB level.
<i>MdmTfSigNoiseFarNearLvl</i>	105 dialout responder test	Use this object to query Signal to Noise Far to Near ratio dB level.
<i>MdmTfSigNoiseNearFarLvl</i>	105 dialout/dial-in responder test	Set S72.6 = 1 to enable dialin responder test. Use this object to query Signal to Noise Near to Far ratio dB level.
<i>MdmTf404FarNearSts</i>	105 dialout responder test	Use this object to query Status. 0 = no test executed. 1 = test executed and valid results. 2 = no responder accessed
<i>MdmTf404NearFarSts</i>	Same as above	Same as above
<i>MdmTf1004FarNearSts</i>	Same as above	Same as above
<i>MdmTf1004NearFarSts</i>	Same as above	Same as above
<i>MdmTf2804FarNearSts</i>	Same as above	Same as above

MIB Object	Description	Settings and use
<i>MdmTf2804NearFarSts</i>	Same as above	Same as above
<i>MdmTfCmsgFarNearSts</i>	Same as above	Same as above
<i>MdmTfCmsgNearFarSts</i>	Same as above	Same as above
<i>MdmTfCnotchFarNearSts</i>	Same as above	Same as above
<i>MdmTfCnotchNearFarSts</i>	Same as above	Same as above
<i>MdmTfSigNoiseFarNearSts</i>	Same as above	Same as above
<i>MdmTfSigNoiseNearFarSts</i>	Same as above	Same as above
<i>MdmTf0dB1004FarNearLvl</i>	105 dialout responder test	Use this object to query measured 1004hz at 0dB Far to Near dB level.
<i>MdmTf0dB1004NearFarLvl</i>	105 dialout/dialin responder test	Set S72.6=1 to enable dialin responder test. Use this object to query measured 1004hz at 0dB Near to Far dB level.
<i>MdmTf0dB1004FarNearSts</i>	105 dialout responder test	Use this object to query status. <i>0 = no test executed.</i> <i>1 = test executed and valid results</i> <i>2 = no responder accessed</i>
<i>MdmTf0dB1004NearFarSts</i>	Same as above	Same as above.

DTE Interface (mdmDiTable)

Use the objects in mdmDiTable to configure the operation of the modem's DTE interface.



The object mdmDiAtString is not supported for Quad or HiPer DSP modems.

Signal Converter (mdmScTable)

Use the objects in the `mdmScTable` to configure operation of the modem's signal converter. This includes control of allowed modulation schemes and specific characteristics for certain modulations. Complex controls are available for faster modulations.

The NMC supports *ScLinkRateAmpU* (the `&U` AT command), which works in conjunction with `&N` command to further control link speeds. It does this by establishing a range of link rates with the `&N` setting as the maximum link rate and the `&U` setting is the minimum. This table shows how the link speed, *L*, is limited by the *N* and *U* settings.

	N = 0	N ≠ 0
U = 0	No limit	$L = N$
U ≠ 0	$U \leq L$	$U \leq L \leq N$

For asymmetric connections (client to server), the `&N` and `&U` settings control the link rate of the high-speed direction of the connection (server to client direction for asymmetric *V.90*). The link rate of the low-speed direction of the connection is controlled by S-Registers 74 and 75 (client to server direction for asymmetric *V.90*).

Configure the `&U` setting at the modem's channel level.

Call Control (mdmCcTable)

Use the objects in the `mdmCcTable` to configure the way the modem sets up and tears down call. Included in this table are ANI, DNIS, and other configurable parameters regarding a modem's T1 interface.

Error Correction (mdmCcTable)

Use the objects in the `mdmCcTable` to configure the modem's operation with regard to MNP and V.42 error correction. Most objects in this table are self-explanatory from the MIB text.

mdmCcDtmTerminationTone

Use *mdmCcDtmTerminationTone* as the equivalent of the `%Gn` AT command, where $n=0-16$. The digits are translated as:

- 0-9; DTMF tones "0" to "9"
- 10-15; DTMF tones "A", "B", "C", "D", "E", and "F"
- 16; Intertone timeout (2 seconds)

mdmCcDataOverVoice

Use *mdmCcDataOverVoice* to enable or disable data over voice (DOVBS). If DOVBS is enabled on the originating side, the modem is instructed to originate speech bearer capability call types.

mdmCc2100AnswerTone

Use *mdmCc2100AnswerTone* to enable or disable data over voice (DOVBS). If DOVBS is enabled on the answering side, the modem is instructed to originate speech bearer capability call types.

**Call Statistics
(mdmCsTable)**

The mdmCsTable contains objects that comprise a variety of statistics pertaining to the last or current call for a given modem. When the modem is not connected, the values are for the last call. Upon making an incoming or outgoing connection, the values for these objects are re-initialized and, at that time, reflect current call statistics. Most objects in this table are self-explanatory from the MIB text.



The HiPer DSP does not support the DTMF detection objects in this table.

mdmCsCollectedDtmfDigits

The Quad modem supports the ability to detect, collect, and route the DTMF tones sent from a client device to a software application that is either included in the chassis (such as the EdgeServer), or through RS-232 or packet bus to an application sitting outside of the chassis.

Use *mdmCsCollectedDtmfDigits* to query a modem to gather all collected DTMF tones in ASCII data format.

Refer to *Modem Call Fail and Fail to Connect Reason Reference* for more information about *mdmCsDisconnectReason* and *mdmCsConnectFailReason*.

**Event counter
(mdmEvTable)**

The mdmEvTable contains objects that count modem events. Objects in this table are initialized to 0 for each modem upon NMC installation or power-on. From then on, each time a designated event occurs, the corresponding counter for that modem and event increments by 1. The counters are never zeroed out unless the NMC is power cycled.

These counters can provide useful history information regarding modem performance.

Event thresholds (mdmEtTable)

Many modem events defined with the NMC have associated thresholds. The associated traps will not be sent if you do not set these thresholds.

Use the objects in the mdmEtTable to configure these thresholds:

- DTE idle (*mdmEtDtIdleThresh*)
- DTR false (*mdmEtDtrFalseThresh*) – Quad modem only
- DTR true (*mdmEtDtrTrueThresh*) – Quad modem only
- BLER error (*mdmEtBlerThresh*)
- Fallback (*mdmEtFallbackThresh*)

Object	Recommended threshold	Description
<i>MdmEtDtIdleThresh</i>	0.255, default 0	Use this object to set the threshold for DTE idle. <i>MdmEtDtIdleThresh</i> defines the length of time for the modem to wait before reporting a DTE transmit data idle event. There must be no activity on the DTE transmit line for the specified quantity of minutes.
<i>MdmEtDtrFalseThresh</i>	0.255, default 0	Use this object to set the threshold for DTR false. (Quad modem only) <i>MdmEtDtrFalseThresh</i> defines the number of seconds that will be used by the modem to qualify a DTR False event.
<i>MdmEtDtrTrueThresh</i>	0.255, default 0	Use this object to set the threshold for DTR true. (Quad modem only) <i>MdmEtDtrTrueThresh</i> defines the number of seconds that will be used by the modem to qualify a DTR True event.
<i>MdmEtConnTimeLimit</i>	N/A	Not supported.
<i>MdmEtBlerThresh</i>	0.255, default 0	Use this object to set the threshold for BLER errors. <i>MdmEtBlerThresh</i> defines the number of BLERs that will be used to qualify the BLER count at threshold event for a given call.

Object	Recommended threshold	Description (continued)
<i>MdmEtFallbackThresh</i>	0.255, default 0	Use this object to set the threshold for fallback. <i>MdmEtFallbackThresh</i> defines the number of fallbacks at which the fallback count at threshold event will be generated for a given call.

Modem commands (mdmCdTable)

The *mdmCdTable* contains objects used to issue and force commands to the Quad modems, as well as to check the results, and interpret the codes resulting from commands.



The HiPer DSP does not support modem commands.

Command function

The *mdmCdFunction* object is used for basic Quad modem functions. This table describes its use. Only the implemented commands are described.

Command	Description
noCommand(1)	This is the default value for <i>mdmCdFunction</i> .
softwareReset(2)	Use this command to reset a single modem entity (for example, one modem on a Quad modem card) without resetting all modems on that card. Use <i>hardwareReset</i> in the CHS MIB to reset all modems on a card. If the NMC knows that the modem is connected or in a state such that it might not be beneficial to reset the modem, the command may fail with a returned <i>mdmCdCode</i> value of <i>connected(14)</i> . In this case, if you still wish to reset the modem, you may resend the softwareReset command, along with mdmCdForce set to "force". If the modem cannot process the softwareReset command, a hardware reset of the card may be required.
storeToNvram(3)	This command is equivalent to &W of the modem AT command set. Use this command to save all of the modem's current settings to the NVRAM on the modem card. Each modem entity of a multi-channel modem card has its own independent NVRAM storage.
restoreFromDflt(4)	This command is equivalent to &F of the modem AT command set. Use this command to restore all modem settings to factory default values.

Command	Description (continued)
restoreFromNvram(5)	Use this command to restore all modem configuration settings to the values that were last saved to the modem's NVRAM (either by the storeToNvram command or by the &W command).
offHook(6)	This command is equivalent to H1 of the modem AT command set. Use this command to make the modem go off-hook and to busy-out the phone line.
onHook(7)	This command is equivalent to H0 command of the modem AT command set. Use this command to make the modem go on-hook and to restore the phone line.
sndTone(8)	Use this command to generate tones at a specific frequency level to test a T1 DS0.
rcvTone(9)	Use this command to test a T1 DS0 and receive tone levels measured at 404hz, 1004hz, and 2804hz.
endTest(10)	Use this command to end all test modes (102, 105, send/receive tones, and &Tn for ALB, DLB, and RLB). This command initiates a test disconnect.
rspndrTest105(11)	Use this command to start a 105 responder dialout test mode for a T1 DS0 channel.
rspndrTest102(12)	Use this command to start a 102 responder dialout test mode for a T1 DS0 channel.
lclAnlgLpbk(13)	<p>This command is equivalent to &T1 of the modem AT command set.</p> <p>Use this command to check test conditions prior to starting the local analog loop back (ALB) test. Valid test conditions are analog line types only, DTR high, no AT parsing or dialing. ATSO=0 test condition performs answer ALB test and SO>0 performs originate ALB test. This test reports status to the NMC (success or failed).</p>
lclDgtlLpbk(14)	<p>This command is equivalent to &T3 of the modem AT command set.</p> <p>Use this command to check test conditions prior to starting the local digital loop back test. Valid test conditions are online mode, non-arq mode. This test reports status to the NMC (success or failed).</p>

Command	Description (continued)
rmtDgtlLpbk(15)	<p>This command is equivalent to &T4 of the modem AT command set.</p> <p>Use this command to check test conditions prior to starting the remote digital loop back test. Valid test conditions are online mode, connect speeds of 1200 and 2400 only, and non-arq mode. This test reports status to the NMC (success or failed).</p>
selfTest(16)	<p>Use this command to report the status for power-up of RAM, ROM, and NVRAM test ("Success" or "Failed").</p>
testRam(17)	<p>This command is equivalent to I2 of the modem AT command set. Use this command to test RAM. This test always reports a status "Success" for test passed.</p>
testRom(18)	<p>Use this command to test ROM. This test reports a status for power-up of ROM test ("Success" or "Failed").</p>
testNVRAM(19)	<p>Use this command to test NVRAM. This test reports a status for power-up of NVRAM test. ("Success" or "Failed" for bad NVRAM hardware condition).</p>
v54LclAnlgLpbk(20)	<p>This command is equivalent to &T8 of the modem AT command set.</p> <p>Use this command to check test conditions prior to starting the local analog loop back (ALB) test with V54 error count. Valid test conditions are analog line types only, DTR high, no AT parsing or dialing, S16 not equal to 4. ATSO=0 test condition performs answer ALB test and SO>0 performs originate ALB test. This test reports status to the NMC (success or failed).</p>
v54RmtDgtlLpbk(21)	<p>This command is equivalent to &T7 of the modem AT command set.</p> <p>Use this command to check test conditions prior to starting the remote digital loop back (RLB) test with V54 error count. Valid test conditions are online mode, connect speeds of 1200 and 2400 only, and non-arq mode. This test reports status to the NMC (success or failed).</p>

Command	Description (continued)
idlePhoneLine(22)	Use this command to report status of an idle phone line. This test reports status as follows: <ul style="list-style-type: none"> ■ Success, normal phone line condition ■ No loop current ■ No dial tone or no line detected ■ No loop current or no dial tone present
loadHwFlowDflt(23)	Use this command to load the hardware flow control template settings (&F1) into the modem.
loadSwFlowDflt(24)	Use this command to load the software flow control template settings (&F2) into the modem.
loadMnp10CllulrDflt(25)	Use this command to load the MNP10 Cellular template settings (&F4) into the modem.
loadV42CllulrMblDflt(26)	Use this command to load the V.42 ETC Mobile Cellular template settings (&F5) into the modem.
loadV42CllulrFxdDflt(27)	Use this command to load the V.42 ETC Fixed Site Cellular template settings (&F6) into the modem.

Trap Enables (mdmTeTable)

The mdmTeTable group provides objects for configuring modem traps. As with all NMC-generated SNMP traps, a trap enable object exists for all modem-related traps. As with the modem MIB tables, a row exists for each modem entity in the chassis, so it is possible to have complete control over which traps are generated from each modem. Typically, it is a good idea to limit NMC-generated modem traps to avoid flooding the NMC with common events. For example, events dealing with connections established and terminated should only be enabled for a short time as a debugging tool.

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Some of the modem events also require threshold settings within the `mdmEtTable`. Associated traps will not be generated if you do not set these thresholds.

Traps for all cards in the chassis are handled by the NMC. If you use *nmcPowerUpAutoCfgEnable*, the NMC will automatically send trap enables to all NACs.

Link Security (`mdmLsTable`)

The Total Control modems provide link-level security that is compatible with the Courier modem product line. Use the `mdmLsTable` to configure each modem separately.

Modem link security and Total Control modem security are mutually exclusive. This means that modems can only be configured for one method of security. If you configure modems for both security methods, Total Control modem security takes precedence on rack modems in the chassis.



The HiPer DSP does not support link security in this release.

Hub Security (`mdmHsTable`)

You can independently configure each modem that supports Total Control security for dial-in security, dial-out security, none, or both using the `mdmHsTable`.

This security only applies if the NIC is using the RS-232 interface; otherwise, the gateway NAC will provide the security options. If you are unable to use Total Control modem security, you may not have purchased that option (contact 3Com Tech Support).

Modem link security and Total Control modem security are mutually exclusive. This means that modems can only be configured for one method of security. If you configure modems for both security methods, Total Control modem security takes precedence on rack modems in the chassis.



The HiPer DSP does not support hub security in this release.

Auto Response

The MDM MIB supplies objects and tables for setting entity-level AutoResponse scripts. This means that an event will trigger an AutoResponse only for the modem on which the event occurs. The other modems on the NAC will not be affected.

Refer to the chapter in this document titled “Auto Response Reference” for additional information.

Cellular Support (mdmCeTable)

Use the `mdmCeTable` to configure modems that support cellular options. This table supports Cellular V.42ETC and MNP10 parameters.



The HiPer DSP does not support cellular options in this release.

Modem Status (mdmStsTable)

The `mdmStsTable` contains one object (`mdmStsPbClock`) that provides information about the current status of the packet bus clock.

Modem Mapping (mdmMaTable)

The `mdmMaTable` contains two objects that map and provide information about the card-level template configuration to which each HiPer DSP modem is mapped. This table is only used for the HiPer DSP templates.

mdmMaChannelConfig

Use `mdmMaChannelConfig` to assign (map) modems to templates. Templates are referred to as groups within this object.

mdmMaChangeIndicator

In some cases, modem channel settings may change due to AT commands, console commands, or other state changes. These commands will alter the setting from what you installed to a channel from the template. Use `mdmMaChangeIndicator` to determine if a modem channel currently contains the template configurations, or if it has changed some parameters. This object will return a value that reports a change occurred, but it will not provide specific change details.

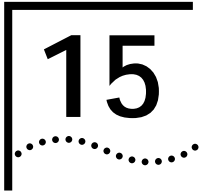
Traps

The MDM MIB contains these trap enable objects:

Trap Enable	Generates this trap
<code>mdmTelConnEstablished</code>	<code>incomingConnectionEstablished(9)</code>
<code>mdmTeOutConnEstablished</code>	<code>outgoingConnectionEstablished(10)</code>
<code>mdmTelConnTerminated</code>	<code>incomingConnectionTerminated(11)</code>
<code>mdmTeOutConnTerminated</code>	<code>outgoingConnectionTerminated(12)</code>
<code>mdmTeConnAttemptFailure</code>	<code>connectAttemptFailure(13)</code>
<code>mdmTeConnLimitExpired</code>	<code>connectTimerExpired(14)</code>

Trap Enable	Generates this trap (continued)
mdmTeDteXmitDataIdle	dteTransmitDataIdle(15) Set threshold using <i>mdmEtDtIdleThresh</i> .
mdmTeDtrTrue	dtrTrue(16) Set threshold using <i>mdmEtDtrTrueThresh</i> .
mdmTeDtrFalse	dtrFalse(17) Set threshold using <i>mdmEtDtrFalseThresh</i> .
mdmTeBlerCountAtThresh	blerCountAtThreshold(18) Set threshold using <i>mdmEtBlerThresh</i> .
mdmTeFallbkCountAtThresh	fallbackCountAtThreshold(19) Set threshold using <i>mdmEtFallbackThresh</i> .
mdmTeNoDialTone	noDialTone(20)
mdmTeNoLoopCurrent	noLoopCurrent(21)
mdmTeResetByDTE	modemResetByDte(27)
mdmTeDialOutCallDur	dialOutCallDuration(45)
mdmTeDialInCallDur	dialInCallDuration(46)
mdmTePbActive	pktBusSessActive(30)
mdmTePbLost	pktBusSessLost(32)
mdmTeDteRingNoAns	dteRingNoAnswer(29)
mdmTePbClockLossEvent	pktBusClkLost(68)
mdmTePbClockRestoreEvent	pktBusClkRestore(69)
mdmTeInConnAttemptFail	inconnectAttemptFailure(86)
mdmTeOutConnAttemptFail	outconnectAttemptFailure(87)
mdmTe105ResponderTest	mdm105ResponderTest(121)

Refer to the chapter titled *Trap Reference* for additional information about traps.



NMC MIB

This chapter contains detailed information about the Network Management Card MIB (NMC MIB).

Purpose of the MIB	The NMC MIB provides management information for the NMC.
Products using this MIB	This MIB is used by the NMC.
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.file (1.3.6.1.4.1.429.1.2).</p>
Configuration Data	<p>The NMC MIB contains these groups:</p> <ul style="list-style-type: none">■ nmcCfg (configuration)■ nmcStat (status)■ nmcTrap (trap destinations)■ nmcCmd (command)■ nmcHs (hub security)■ nmcTe (trap enables)■ nmcUiCfg (user interface configuration)■ nmcAuth (authentication)■ nmcNtp (network time protocol)

NMC Configuration Group

Use the `nmcCfg` group to configure basic NMC management parameters.

Changes you make to read-write objects in the `nmcCfg` group take immediate effect when you issue a **set** command. Make these changes permanent by issuing the *saveToNvram* command (refer to the section on NMC commands). When the NMC boots from power-on or a software reset command, it always initializes these variables from the values that were last saved to NVRAM.

Configurable `nmcCfg` objects not specified in the following sections are self-explanatory from the MIB text.

The NMC's real time clock

This group contains two objects to configure system time and date: *nmcCfgSystemTime* and *nmcCfgSystemDate*. Once you set these objects, related options (including *nmcGmtime*) are automatically calculated by the system.

Set the time zone (*nmcTimezone*) and daylight savings time preference (*nmcDaySavingTime*) before setting system time and date.

Set the system time and date according to the local time of the NMC's physical location. The format for the time string is HH:MM:SS, where:

- HH = hours (0-23)
- MM = minutes (0-59)
- SS = seconds (0-59)

The format for the date string is MM/DD/YY, where:

- MM = month (1-12)
- DD = day (1-31)
- YY = year (2 digits)

Accurately setting the NMC's real time clock is important in obtaining valid timestamps in SNMP trap messages. Traps contain a timestamp given in Greenwich Mean Time (GMT) as an integer indicating the number of seconds since midnight on January 1, 1970.

Configuring the WAN (SLIP) port for dial access

If the WAN port is physically connected to a modem for dial-up access, the port will require object configuration.

When the NMC needs to transmit a SLIP packet out the WAN port, it checks for the presence of Carrier Detect (CD) on the WAN port serial interface. If CD is present, it is assumed that a connection to the desired destination exists, and the packet is transmitted. The NMC typically will see CD present when directly connected to the MS because the MS's Data Terminal Ready (DTR) signal drives CD.

If the NMC tries to transmit a packet out the WAN port and it does not detect CD, it attempts to set up a modem on that port by issuing an AT command.



*Configure this command using `nmcCfgAtString`. This AT string enables command mode local echo, verbal result codes, and hardware flow control. Maintain the default setting for proper functionality with the modem, but you may add additional settings to the string as desired. The default value of `nmcCfgAtString` is **AT&FE0Q0&H1&R2&B1V1**.*

Next, the NMC issues an AT dial command using the phone number configured with `nmcCfgWanDialOutPhoneNum`. If the process of dialing results in a successful connection (CD becomes active), the packet transmits. The time for which the SLIP port can remain inactive before the NMC hangs up the call is configured with `nmcUiCfgInactiveTime`. If at least one packet is received and/or transmitted within the configured time limit while an NMC-initiated dial connection is up, the modems remain online. If the inactivity time passes without any SLIP traffic on the WAN port, the NMC drops DTR and hangs up the call.



This time restriction does not exist on a dial-in call to the NMC.

If the NMC is not successful in making a dial-up connection while attempting to transmit a packet from the WAN port, the NMC waits the number of seconds configured by `nmcCfgWanRetryPause` and retries the dialed number. The NMC retries up to the number of times configured in `nmcCfgNumWanRetries` before it gives up. If the number of retries is exceeded, the NMC discards the packet. If `nmcCfgNumWanRetries` is 0, the NMC tries to send the packet "forever" and will not discard it.

Each unsuccessful dial attempt is counted by the NMC as a connection attempt failure. If the number of connection attempt failures exceeds the value configured in `nmcCfgNumFailBeSuspend`, the NMC suspends dial-out attempts for the time specified by the value configured in `nmcCfgWanRetrySuspendTime`. This suspension allows remote MSs to dial in to the WAN port. For example, if the remote line is always busy

and the NMC continuously tries to dial out, remote MSs may be unable to dial in. When the suspension timer expires, the NMC may continue to make dial-out connections subject to further suspensions by the same rules.



The values for `nmcCfgAtString` and `nmcCfgWanDialOutPhoneNum` can be modified to connect the WAN port to other devices.

TFTP timeout

The NMC uses TFTP to perform file transfers during operations such as software download (SDL). TFTP is a simplified version of FTP (File Transfer Protocol) that transfers files but does not provide password protection or user-directory capability.

This feature allows you to complete an SDL over the Internet and across congested networks. For example, use this timeout in a network where there are long delay times in getting packets to the NMC. Then, prior to doing a software download, increase the timeout to a high value of 30 seconds. This would allow a very long delay in getting a TFTP packet to the NMC.

Use the configurable parameter `nmcCfgTFTPTimeout` to specify the timeout duration before the NMC terminates a TFTP session.

Session ID/Call reference number for RADIUS logging

The session ID feature is used on a HiPer chassis. When enabled, it allows a universal format for the call reference number on a RADIUS server.

Set the variable to **enable** to send a universal call reference number format from all NACs to a RADIUS server. Set the variable to **disable** if you are using a non-HiPer chassis.

Configuring security and accounting servers

The `nmcCfgLog` contains many accounting- and event logging-related parameters. These include configurable server selections. The MIB text associated with each object is self-explanatory. Refer to “Other MIB-specific information” at the end of this chapter for additional details.

NMC Status Group

The nmcStat group provides objects which a MS may use to query the NMC to check status. You may customize these objects for your network and add them to a performance monitoring application.

Many of the objects in the nmcStat group are self-explanatory from the MIB text. Other objects are explained in this section.

Test result bitmaps

The NMC's online test results do not fit neatly into the result mechanism for commands, so they are returned by the object *nmcStatTestResult*. Similarly, the results of the NMC's power-up test are returned with the object *nmcStatPowerUpTstFailBMap*. The definitions for both of these test results consist of INTEGERS. AN INTEGER 1 in the bitmap indicates the test failed.

When the bits are numbered such that 1 is the least significant bit of the INTEGER and 32 is the most significant bit, these are the bit definitions:

Bit number	Test name
1	Slot 1 Management Bus UART
2	Slot 2 Management Bus UART
3	Slot 3 Management Bus UART
4	Slot 4 Management Bus UART
5	Slot 5 Management Bus UART
6	Slot 6 Management Bus UART
7	Slot 7 Management Bus UART
8	Slot 8 Management Bus UART
9	Slot 9 Management Bus UART
10	Slot 10 Management Bus UART
11	Slot 11 Management Bus UART
12	Slot 12 Management Bus UART
13	Slot 13 Management Bus UART
14	Slot 14 Management Bus UART
15	Slot 15 Management Bus UART
16	Slot 16 Management Bus UART
17	NIC Management Bus UART
18	Analog to digital converter
19	EEPROM CRC (over manufacturer's information)

Bit number	Test name (continued)
20	FLASH Vpp programming voltage
21	Power supply presence circuit
22-32	Reserved

Failure of one or more power-up tests causes the NMC RUN/FAIL LED to alternately blink red and green. If all power-up tests pass, the LED remains solid green.



*Use a single **get** request to retrieve `nmcStatTestResult` together with `nmcCmdResult`, `nmcCmdCode`, `nmcCmdMgtStationId`, and `nmcCmdReqId` to ensure that test results were those generated by a given MS.*

MIB compatibility

Total Control Manager software uses the object `nmcStatCompSwVer` to identify MIB compatibility, independent of NMC and Total Control Manager versions.

Hub status LED

Use `nmcStatRedLed` to determine the reason why the NMC's hub status LED is solid red. Querying this object can be a useful diagnostic and trouble clearing tool.

Auxiliary I/O objects

The HiPer NMC card uses the 10/100 Ethernet Aux I/O NIC. Query the objects `nmcAuxIn1Sts`, `nmcAuxIn2Sts`, `nmcAuxOut1Sts`, and `nmcAuxOut2Sts` to determine the status of each auxiliary port.

NMC Trap Group

The `nmcTrap` group contains objects used to configure the NMC's trap reporting mechanism. The objects are self-explanatory in the MIB text. Additional information is provided here, as well as in "Configuring Traps", later in this manual.

Trap destination table

The trap destination identifies the IP address of the server(s) to which the traps are sent. When adding a new trap destination, enter this information for each new MS:

- **Destination IP** — the address of one or more alarm and/or logging servers (*nmcTrapDestIP*)
- **Community string** — SNMP community strings allow the MS to filter trap and log messages that may come from other devices on or outside of the network (*nmcTrapDestCommunity*)
- **Description** — optional additional descriptive text (*nmcTrapDestDescr*)

Use the *nmcTrapDestTable* to specify the destination(s) of the trap messages. When a trap generates, the trap message will be sent to all IP addresses specified in this table. At least one entry must be present in the *nmcTrapDestTable* for traps to be sent to a server. By default, the trap destination table contains no entries. This means that traps and log messages will not be generated until you specify a destination IP address(es).



Setting the community string for an existing row to the value "invalid" will delete the specified row from the trap destination table.

NMC Command Group

The nmcCmd group contains objects used to issue commands to the NMC, force commands, check the results, and interpret the codes resulting from commands.

Configuring and saving settings on the NMC

Each chassis NAC has its own NVRAM. The NMC also has its own NVRAM, which is independent from the NACs. Settings for each card may be saved to either NVRAM, depending on network needs.

Two of the most typical ways of using NVRAM are:

- Configure each chassis device and save the parameters to each NAC's NVRAM. Set the NACs to boot from their own NVRAM (ignoring the NMC NVRAM, except for NMC-specific parameters). Disable *nmcPowerUpAutoCfgEnable* so the NMC does not automatically configure the NACs upon NAC installation, NMC installation, or chassis power-up.
- Configure each device, and save all settings to the NMC's NVRAM (ignoring each NAC's NVRAM). Enable *nmcPowerUpAutoCfgEnable* so the NMC automatically configures the NACs upon NAC installation, NMC installation, or chassis power-up.

The first approach minimizes the effort of the NMC. It also allows the possibility of local NAC configuration changes of which the NMC is unaware. Reduce this degree of uncertainty by disabling other chassis variables, including configuration-related AT commands on the modems.

Because of the difficulty in keeping the NMC in sync with local configuration changes, a cache was developed. The NMC cache operates like this:

- 1 As the NMC discovers NACs on power-up, it loads the cache for each of the 16 slots from values previously saved to NMC NVRAM.
- 2 Cache values for any slots for which there is no configuration in NMC NVRAM are set to default values.
- 3 From this point on, the cache is updated as SNMP **sets** are made to NACs through the NMC's proxy agent. The cache values are saved to the NMC's NVRAM upon invocation of the *saveToNvram(2)* command. These become the values used by the NMC during an autoconfigure operation.

The NMC's parameters are not cached like the NAC-specific parameters. They are always restored from the NMC's NVRAM. The MIB objects defined under *nmcUiCfg* are not cached and never saved to NVRAM. A separate storage facility in the NMC's EEPROM holds these settings.

NMC configuration commands

The *nmcCmdFunction* object is used for basic system functions. This table describes its use.

Command	Description
noCommand(1)	This is the default value for <i>nmcCmdFunction</i> . Set <i>nmcCmdFunction</i> to abort any command currently in progress.
saveToNvram(2)	Use this command to save the current configuration cache for each NAC to the NMC's NVRAM. This command also saves the current state of each of the NMC's configurable parameters, except for parameters within the <i>nmcUiCfg</i> group. Since you may also set these UI port, they are saved to EEPROM by <i>savUiParamsToiEEPROM</i> .

Command	Description (continued)												
restoreFromDefaults(3)	<p>Use this command to restore all configurable NMC parameters (except those in the <i>nmcUiCfg</i> group) to their default state.</p> <p>This command also restores the cache for each chassis NAC to its defaults values for the given card type. After defaulting the cache, the cache contents are used to reconfigure each device to its default state. For the most part, this command restores the entire chassis to the factory default configuration (except for settings made in the <i>nmcUiCfg</i> group).</p>												
restoreFromNvram(4)	<p>Use this command to restore all configurable NMC parameters (except those in the <i>nmcUiCfg</i> group) to the value last saved in the NMC's NVRAM.</p> <p>This command also attempts to restore the cache for each NAC to the values last saved to the NMC's NVRAM. If the parameters in NVRAM for a given slot do not match the currently-installed card type, the cache for that slot is loaded from the default values for the given card type. After restoring all slot caches, the NMC configures each device with the parameter values from that slot's cache.</p>												
nonDisruptSelfTest(5)	<p>Use this command to initiate a nondisruptive self test while the NMC is online.</p> <p>On power-up, the NMC completes a full hardware self test. Once the NMC's operational code is up and running, such a test would disrupt the NMC's ability to preform certain network management functions. For this reason, the online version of the NMC's self-test is nondisruptive, and is comprised of a subset of the power-up tests.</p> <p>For convenience, test results are returned in a bitmapped object (<i>nmcStatTestResult</i>) that is defined exactly like the power-on self test result (<i>nmcStatPowerUpTstFailMap</i>). This example shows which tests (and bits) pertain to the NMC's nondisruptive self test (assuming that the least significant bit of the iNTEGER bitmap is bit #1 and the most significant is bit #32).</p> <table> <tr> <th>Bit number</th><th>Test name</th></tr> <tr> <td>18</td><td>Analog to digital converter</td></tr> <tr> <td>19</td><td>EEPROM CRC</td></tr> <tr> <td>20</td><td>Flash Vpp programming voltage</td></tr> <tr> <td>21</td><td>Power supply presence circuit</td></tr> <tr> <td>22</td><td>Flash file system CRC test</td></tr> </table>	Bit number	Test name	18	Analog to digital converter	19	EEPROM CRC	20	Flash Vpp programming voltage	21	Power supply presence circuit	22	Flash file system CRC test
Bit number	Test name												
18	Analog to digital converter												
19	EEPROM CRC												
20	Flash Vpp programming voltage												
21	Power supply presence circuit												
22	Flash file system CRC test												

Command	Description (continued)
softwareReset(6)	Use this command to reset the NMC from a remote location. This may be useful when changing the IP address of an NMC, or in an emergency when an NMC software malfunction occurs and you need to restore the NMC to a good state.
saveUiParamsToEEPROM(7)	<p>Use this command to save the value of the MIB objects within <i>nmcuiCfg</i> to the NMC's EEPROM.</p> <p>These objects are treated separately for these reasons:</p> <ul style="list-style-type: none"> ■ The IP address of the NMC is unique, so it is stored in the EEPROM, which is physically attached to the NMC card. The FLASH SIMM can then be swapped to other NMC cards. ■ Total Control Manager does not save this group of parameters in the *.whb file when it saves a chassis configuration to disk. <p>This command is needed because IP addressing parameters do not take effect immediately upon issuing a set command. The new values are stored in RAM until you issue the <i>saveUiParamsToEEPROM</i> command. This saves the RAM image to the EEPROM. Subsequently, you must issue a <i>softwareReset</i> command to the NMC to cause the new values for <i>nmcUiCfg</i> objects to take effect.</p>
restoreNmcFromDefaults(8)	<p>Use this command to restore all configurable NMC parameters (except those in the <i>nmcUiCfg</i> group) to their default state.</p> <p>This command only restores the NMC defaults.</p>
restoreNmcFromNvram(9)	<p>Use this command to restore all configurable NMC parameters (except those in the <i>nmcUiCfg</i> group) to the value last saved in the NMC's NVRAM.</p> <p>This command only restores the NMC NVRAM values.</p>
bulkFileUpload(10)	<p>Use this command to :</p> <ul style="list-style-type: none"> ■ Upload the NMC NVRAM image file to a MS for transferring chassis configurations (.nvr file) ■ Upload the single configuration file (in CFM format) to a MS (.cfm file) ■ Upload the modem history file (.hst file)
bulkFileDownload(11)	<p>Use this command to :</p> <ul style="list-style-type: none"> ■ Download the NMC NVRAM image file to a MS for transferring chassis configurations (.nvr file) ■ Download the single configuration file (in CFM format) to a MS (.cfm file)

Command	Description (continued)
openAuxOutputPort1(12)	Use this command to open auxiliary output port 1. (HiPer NMC only)
openAuxOutputPort2(13)	Use this command to open auxiliary output port 2. (HiPer NMC only)
closeAuxOutputPort1(14)	Use this command to close auxiliary output port 1. (HiPer NMC only)
closeAuxOutputPort2(15)	Use this command to close auxiliary output port 2. (HiPer NMC only)

AUX I/O commands

Change port status by issuing the appropriate Aux I/O command. The values of these commands reflect the actual operation values needed to be sent via an SNMP message. When the port status is successfully changed, a trap is sent from the NMC to the MS.

Viewing the results of an NMC command

Use *nmcCmdResult* and *nmcCmdCode* to determine the results of an NMC command. These objects are useful when trouble clearing a command that did not complete.

NMC Hub Security Group

The group *nmCHs* offers minimal hub security, providing a mechanism to deny access based upon community, IP address, and an authorized access list. For dial-in users, the NMC will issue a password prompt for modem access. Hub security is supported by the Quad modems only.

NMC Trap Enable Group

The *nmcTe* group provides objects for configuring NMC traps. You must configure traps in order for them to be sent to the MS alarm server and/or the RADIUS logging server. Configuring traps is a two-step process: set the corresponding trap enable object, then specify the trap destination (as configured in the *nmcTrap* group).

Four options typically exist when setting traps (a few traps allow "enable" and "disable" options only):

Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.

Command	Description (continued)
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Traps for all cards in the chassis are handled by the NMC. If you use *nmcPowerUpAutoCfgEnable*, the NMC will automatically send trap enables to all NACs.

NMC User Interface Configuration Group

For convenience when remotely managing an NMC, you may use the objects in the *nmcUiCfg* group to set the configurable parameters of the local User Interface (UI) port. These parameters include IP addresses, subnet masks, and community strings.

Due to the complexities associated with changing many of these parameters while the NMC is running operational code, a **set** to most of these objects results only in setting a RAM location within the NMC. You must issue the **saveUiParamsToEEPROM** command (*nmcCmdFunction*), followed by **softwareReset** (*nmcCmdFunction*) or power cycle in order for the new values to take effect.

The exception to this above rule occurs when setting community strings. Both the public (read-only) and private (read-write) community strings take effect immediately. If the NMC is reset without issuing a **saveUiParamsToEEPROM** command, they will revert to the values last saved to EEPROM.

NMC Authorized Access Group

The *nmcAuth* group provides a table that allows you to define which MS are allowed to access the NMC. By default, the NMC authorized access table (*nmcAuthAccTable*) is empty. In this state, no MS are restricted from NMC access. Once you add a row to this table, only MSs whose IP addresses are authorized can access the NMC.

The table is indexed by IP address (*nmcAuthAccIpAddr*). The other objects in each row include a required subnet mask (*nmcAuthAccNetMask*) and optional text descriptions (*nmcAuthAccDescr*).

IP address validation

This is the process the NMC uses to perform IP address validation:

- 1 The NMC performs an “and” operation on the source IP address of an incoming packet and the value of *nmcAuthAccNetMask* for the first row in the authorized access list.
- 2 The NMC performs an “and” operation on the *nmcAuthAcclpAddr* with the *nmcAuthAccNetMask* for that same row.
- 3 The NMC compares the “anded” value of step 1 to the value it obtains by performing the second and test. This process allows a single station or group of stations access based upon the mask value.
 - If the two values match, the IP packet passes the authorization test and is passed to higher protocol layers for further processing.
 - If the values do not match, the next row in the authorized access list is verified by the same procedure.
 - If the end of the table is reached without a match, the IP packet fails the authorization test. The value of *ipInDiscards* is incremented, and the IP packet is discarded.

ICMP packets are not validated against these checks. It is often helpful to use an ICMP echo packet to check basic network health from any convenient network terminal (which may not be on the authorized access list). All SNMP and TFTP packets are checked against this authorized access list.

Use the User Interface port to clear the authorized access list if access is locked (for example, the MS is not included on the list). Alternately, DIP switch 5 on the NMC card can be used to boot with an empty list.

NMC Network Time Protocol group

Network Time protocol (NTP) is the Standard Network Time Protocol (also known as SNTP). Use the objects in *nmcNtp* to configure the NMC clock and all other cards in the chassis so they are in sync with other server clocks on a network.

The SNTP implementation in the NMC allows for definition of a primary and secondary NTP server, as well as an interval time. In addition to the configurable settings, objects are defined to allow the system administrator to monitor the health of the NTP function in the NMC. These objects include:

- Time of last successful synchronization

- Server used for last synchronization
- Time of last synchronization failure
- Which server (if any) was last listed as failed

Use *nmcNtpOperationalMode* to enable NTP [set the object to unicast(2)].

MIB Tables

The NMC MIB contains these tables: *nmcTrapDestTable* and *nmcAuthAccTable*. Refer to previous sections of this chapter for additional information.

Traps

The NMC MIB contains these trap enable objects:

Trap enable	Generates this trap
nmcCfgAuthFailTrapEnable	authenticationFail (generic trap 5)
nmcTeDialInLogFail	dialInLoginFail(38)
nmcTeDialOutLoginFail	dialOutLoginFail(37)
nmcTeDialOutRestrictNum	dialOutRestrictedNum(39)
nmcTeDialBackRestrictNum	dialBackRestrictedNum(40)
nmcTeUserBlacklist	userBlacklisted(41)
nmcTeUserBlacklistLogIn	loginAttemptByBlacklistedUser(42)
nmcTeRespAttemptLimExceeded	responseAttemptLimExceeded(43)
nmcTeLoginAttmpLimExceed	mdmLoginAttemptLimExceeded(44)
nmcTeLogSrvrLoss	acctSrvrLoss(49)
nmcTeSecSrvrLoss	securitySrvrLoss(62)
nmcTeSinglePbClockFail	singlePktBusClockFailure(63)
nmcTePbClockSwitch	pktBusClockSwitch(64)
nmcTePbClockFail	pktBusClockFailure(65)
nmcTeDnsSrvrLoss	dnsSrvrLoss(80)
nmcTeNtpSrvrLoss	ntpSrvrLossConn(81)
nmcTeNtpSrvrRestore	ntpSrvrRestConn(82)
nmcTeNtpSrvrDegraded	ntpSrvrDgrConn(85)
nmcTeDnsSrvrRestore	dnsSrvrRestore(94)
nmcTeDnsSrvrDegraded	dnsSrvrDegraded(95)
nmcTeLogSrvrRestore	acctSrvrRestore(96)

Trap enable	Generates this trap (continued)
nmcTeLogSrvrGroupOper	acctSrvrGroupOper(97)
nmcTeLogSrvrGroupDegr	acctSrvrGroupDegr(98)
nmcTeLogSrvrGroupNonOp	acctSrvrGroupNonOp(99)
nmcTeSecSrvrRestore	securitySrvrRestore(100)
nmcTeSecSrvrGroupOper	securitySrvrGrpOper(101)
nmcTeSecSrvrGroupDegr	securitySrvrGrpDegr(102)
nmcTeSecSrvrGroupNonOp	securitySrvrGrpNonOp(103)

Refer to the chapter titled *Trap Reference* for additional information about traps.

Other MIB-specific Information	<p>This section includes information about:</p> <ul style="list-style-type: none">■ Configuring security and accounting servers■ DNS resolver feature■ SNMP forwarding
Configuring Security and Accounting, Servers	<p>The 3Com accounting and event logging application captures and logs a variety of call accounting and event information to simple ASCII log files. This feature relies on the RADIUS (Remote Authentication Dial In User Service) client-server protocol.</p> <p>The NMC client communicates with the PC-based server via an Ethernet or Token-ring LAN. Information, such as modem connectivity and specific chassis events, may be sent by NMC clients across a LAN connection to be logged. The server itself does not generate reports from the data; a post-processing application or database script must be used to format the logged data.</p> <p>The nmcCfgLog and nmcTeLog groups contain many accounting- and event logging-related parameters. These include configurable server selections and traps for any loss of server connection. The MIB text associated with each object is self-explanatory.</p>

Log records

The NMC does not provide internal log space for event records. Rather, it provides an internal queue for event records waiting to be sent to the RADIUS server.

The NMC client removes an event record from its input queue, formats it properly for the RADIUS server, sends it, and waits for the reply before starting the next event record. If the NMC client is unable to communicate with a RADIUS server, the NMC client discards the event records.

1-8 RADIUS Accounting

The NMC supports from one to eight RADIUS accounting server addresses (primary and backups). Configure the addresses with these RADIUS accounting server address MIB objects:

- *nmcCfgLogPriSvrAddr*
- *mcCfgLogSecSvrAddr*
- *nmcCfgLog3SvrAddr*
- *nmcCfgLog4SvrAddr*
- *mcCfgLog5SvrAddr*
- *mcCfgLog6SvrAddr*
- *nmcCfgLog7SvrAddr*
- *nmcCfgLog8SvrAddr*

Enable the Accounting server DNS host name resolver by setting *nmcCfgLogDnsEna* to **enabled** and setting *nmcCfgLogSvrName* to the host name. The DNS entry can include from one to eight addresses per host name.

1-8 RADIUS Security

The NMC supports from one to eight RADIUS security server addresses (primary and backups). Configure the addresses with these RADIUS security server address MIB objects:

- *nmcHsSecuritySvrAddr*
- *nmcHsSecondarySvrAddr*
- *nmcHsSecurity3SvrAddr*

- *nmcHsSecurity4SrvrAddr*
- *nmcHsSecurity5SrvrAddr*
- *nmcHsSecurity6SrvrAddr*
- *nmcHsSecurity7SrvrAddr*
- *nmcHsSecurity8SrvrAddr*

Enable the Security server DNS host name resolver by setting *nmcHsSecuritySrvrDnsEna* to **enabled** and setting *nmcHsSecuritySrvrName* to the host name. The DNS entry can include from one to eight addresses per host name.

MD5 Calculation

MD5 is the encryption mechanism used for RADIUS authentication. Use *nmcCfgLogMD5Calc* to indicate to the accounting client whether to calculate MD5 for the accounting request message.



Total Control Manager *Total Control Manager* does not use MD5.

Call statistics groups

Use the object *nmcCfgLogCallStatGrpSel* to specify which call statistics groups should be sent to a RADIUS log. This object provides a filtering mechanism for statistics.

DNS RADIUS Security/Accounting Host Name

You can make configuration of security and accounting RADIUS server addresses easier to maintain by enabling a RADIUS Server DNS name resolver by enabling *nmcCfgLogDnsEna* or *nmcHsSecuritySrvrDnsEna*. Be aware that the MIB RADIUS server address objects are unread when the DNS resolver is enabled and do not reflect the resolved addresses.

User Server Selection Disabled

The recovery scheme uses an automated mechanism using Status-Server request to poll the communication status of a RADIUS Server. The objects *nmcCfgLogSrvrSelect* or *nmcHsServerSelect* reflects the active RADIUS server in which RADIUS transactions are expected to occur on. The MIB objects are no longer used to change the active server and are now read-only.

Recovery Mechanism uses Status-Server Request

The RADIUS client communication recovery algorithm uses Status-Server requests to poll communication status of priority servers. The RADIUS server must be Status-Server enabled to work effectively.

The 3Com RADIUS server has a configurable option to enable Status-Server in more recent releases. If the RADIUS server does not support Status-Server requests, the RADIUS server will not respond to the requests and the NMC will not recover to the primary server effectively. The primary RADIUS server will only be retried after all backups have failed.

Server trouble clearing

RADIUS Server/Accounting communication repeatedly up and down Symptom: constant Security/Accounting loss and restore traps.

RADIUS MD5 authentication keys should be replicated at the RADIUS server. If authentication keys do not match, RADIUS Security (and potentially Accounting if MD5 is enabled) will not work effectively. The Status-Server request is not MD5 encrypted, therefore, does not test if authentication keys match. Unmatched keys will result in successful Status-Server transactions and the NMC assuming server is up, but RADIUS transactions will fail. This will result in the NMC re-attempting the failed server every Status-Server poll interval, defined by *nmcCfgLogStatusInterval* or *nmCHsSecurityStatusInt*, and failing once again. Although RADIUS transactions continue on the next highest priority available server, processing is slowed and unnecessary traffic is produced.

Status-Server Poll Interval

The Status-Server poll interval is the interval between consecutive Status-Server requests to a RADIUS server known to have communication failures.

DNS Resolver Feature

RADIUS Security & Accounting Host Name

The host names for RADIUS Security and Accounting can either be the same, or each can be unique.

Basic DNS Configuration Suggestions

- The NMC DNS Server primary and secondary addresses should be DNS Server backups to each other.
- Host name Alias resolution is not supported by the NMC. Put host names in name to address DNS entries.
- Carefully plan the Time To Live (TTL) value and account for the potential for future changes. It should be set to a time less than the approximate time to plan a address change (i.e. an estimate of time between the inception planning the address change and the time of the address change). A week is a good TTL to start with, and it should be shortened as the time to address change closes in.
- Multiple IP addresses can be added for a single host name, for RADIUS Accounting and Security server backup purposes. Order these addresses by backup priority (primary should be first entry).

NMC Host Names

When enabling a RADIUS Server host name resolver, use the full domain name (i.e. no default domains are assumed). For example: (valid host name) "nmc55.isp.usr.com.", (invalid host name) "nmc55".

Refresh DNS Cache Suggestion

Updating a host name MIB object, whether there is a change to the name or not, causes the NMC DNS resolver to re-resolve host name addresses. This can be a useful last minute DNS update procedure, if the DNS host name entry at the DNS Server has been modified, such as a TTL update. The RADIUS client observes these changes following a Status-Server poll interval.

SNMP forwarding

The NMC can forward an SNMP packet to any chassis NAC that contains an SNMP agent. This includes the HiPer ARC, EdgeServer, and CDMA cards.

The HiPer ARC card contains MIB objects that can be accessed through the NMC. You must change the NMC community strings to access these objects.

The community strings needed to access the HiPer ARC objects match the NMC community strings, but they must be appended with:

@slot_number_of_HiPerARC*1000

For example, if the HiPer ARC card is in slot 15, and the NMC's read-only community string is "public", the community string needed to do **get** and **get-next** commands on the HiPer ARC is:

public@15000

Similarly, if the NMC's read-write community string is "private", the community string needed to do **set** commands on the HiPer ARC is:

private@15000

This chapter contains information about additional USR MIBS that are used within the Total Control chassis.

PB MIB

The Packet Bus (PB) MIB is an enterprise-specific MIB that provides management for each gateway NAC in the chassis.

Products using this MIB

This MIB is used by the NETServer and HiPer ARC cards.

Registration ID

This MIB is an Internet Enterprise MIB registered under the USR enterprise node.

iso.org.dod.internet.private.enterprises.usr.nas.pb
(1.3.6.1.4.1.429.1.8)

Tables

The PB MIB contains these tables:

- **pbCfgTable** — The Packet Bus Configuration table contains configuration information for each Gateway NAC in the NAS chassis.
- **pbSessionTable** — The Packet Bus Session table contains an entry for each Gateway NAC in the chassis. It provides a means to configure packet bus connections and determine the status of the defined connections. This table resides in the NAC and can be implemented as either a dense or sparse table.
- **pbTrapEnaTable** — The Packet Bus Trap Enable table contains objects to enable traps on each Gateway card in a chassis.

Traps The PB MIB contains these trap enables:

Trap enable	Generates this trap
pbTrapEnaSessActive	pktBusSessActive (30)
pbTrapEnaPktBusCongest	pktBusSessCongestion (31)
pbTrapEnaPktBusSessLost	pktBusSessLost (32)
pbTrapEnaSessionInactive	pktBusSessInactive (33)
pbTrapEnaSessionError	pktBusSessError (47)

Refer to the chapter titled *Trap Reference* for additional information about traps.

Packet Bus Datagram MIB (PBDG MIB)

The Packet Bus Datagram (PBDG) MIB is an enterprise-specific MIB that provides management for NACs using packet bus datagrams.

Products using this MIB

This MIB is used by the NETServer and HiPer ARC cards.

Registration ID

This MIB is an Internet Enterprise MIB registered under the USR enterprise node.

iso.org.dod.internet.private.enterprises.usr.nas.pbdg
(1.3.6.1.4.1.429.1.14)

Tables

The PBDG MIB contains these tables:

- **pbdgDatagramTable** — The Packet Bus Datagram table contains configuration information for each NAC in the chassis that supports packet bus datagrams.
- **pbdgCfgTable** — The Packet Bus Configuration table contains configuration information for each NAC in the chassis that supports packet bus datagrams.

Traps No trap enables are included in this MIB.

ULPB MIB	The ULPD MIB provides management for a Link Access Procedure, Balanced (LAPB) interface. LAPB represents the link layer of the X.25 protocol.
Products using this MIB	This MIB is used by the X.25 NAC.
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.ulpb (1.3.6.1.4.1.429.1.10)</p>
Tables	<p>The ULPB MIB contains these tables:</p> <ul style="list-style-type: none"> ■ ulpbAdmnTable — The ulpbAdmnTable table defines the objects that can be changed to manage a LAPB interface. Changing one of these parameters may take effect in the operating LAPB immediately or may wait until the interface is restarted depending on the details of the implementation. ■ ulpbOperTable — The ulpbOperTable table contains configuration information about interface parameters currently set in the interface. These objects are read-only. ■ ulpbStatTable — The ulpbStatTable <i>table contains</i> statistics information about this LAPB interface.
Traps	No trap enables are included in this MIB.
X.25 Interface MIB (UX25 MIB)	The UX25 MIB provides management for the X.25 NAC.
Products using this MIB	This MIB is used by the X.25 NAC.

Registration ID This MIB is an Internet Enterprise MIB registered under the USR enterprise node.

iso.org.dod.internet.private.enterprises.usr.nas.ux25
(1.3.6.1.4.1.429.1.9)

Tables The UX25 MIB contains these tables:

- **ux25AdmnChannelTable** — The UX25 Administrator Channel table defines objects for the parameters of an X.25 interface which the administrator can read and set.
- **ux25AdmnClassTable** — The UX25 Administrator Class table defines objects for the parameters of an X.25 interface which the administrator can read and set.
- **ux25AdmnPacketTable** — The UX25 Administrator Packet table defines objects for the parameters of an X.25 interface which the administrator can read and set.
- **ux25AdmnSubscriberTable** — The UX25 Administrator Subscriber table defines objects for the parameters of an X.25 interface which the administrator can read and set.
- **ux25AdmnTimerTable** — The UX25 Administrator Timer table defines objects for the parameters of an X.25 interface which the administrator can read and set.
- **ux25OperChannelTable** — The UX25 Operational Channel table defines objects that report the current parameters used by a running interface. These objects are read only.
- **ux25OperClassTable** — The UX25 Operational Class table defines objects that report the current parameters used by a running interface. These objects are read only.
- **ux25OperPacketTable** — The UX25 Operational Packet table defines objects that report the current parameters used by a running interface. These objects are read only.
- **ux25OperSubscriberTable** — The UX25 Operational Subscriber table defines objects that report the current parameters used by a running interface. These objects are read only.
- **ux25OperTimerTable** — The UX25 Operational Timer table defines objects that report the current parameters used by a running interface. These objects are read only.

- **ux25StatTable** — The UX25 Statistics table defines objects that report operational statistics for an X.25 interface.

Traps No trap enables are included in this MIB.

X.25 Gateway MIB	The X.25 Gateway (X25g) MIB provides management for the X.25 gateway NAC.
Products using this MIB	This MIB is used by the X.25 card.
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.x25g (1.3.6.1.4.1.429.1.11)</p>
Tables	<p>The X25g MIB contains these tables:</p> <ul style="list-style-type: none">■ x25gwldTable — The X.25 Gateway Identification table contains information that identifies the hardware and software that make up the specified X.25 card.■ x25gwCmdTable — The X.25 Card Command table contains an entry for each of the manageable X.25 Cards in the chassis. It provides a means through which to take specific actions on one or more X.25 cards in the NAS chassis. The number of entries in this table is given by the value of x25gwldNumber.■ x25gwCfgTable — The X.25 Card Configuration table contains configurable parameters specific to the X.25 Gateway NAC.■ x25gwTrapEnaTable — The X.25 Card Trap Enable table contains objects to enable traps on the X.25 Cards in the chassis.
Traps	The X25g MIB contains the <i>x25gwTrapEnaUiReset</i> object that allows you to enable reporting of NAC Reset by user interface command traps.

Trap enable	Generates this trap
x25gwTrapEnaUiReset	nacUserInterfaceReset (34)

Refer to the chapter titled *Trap Reference* for additional information about traps.

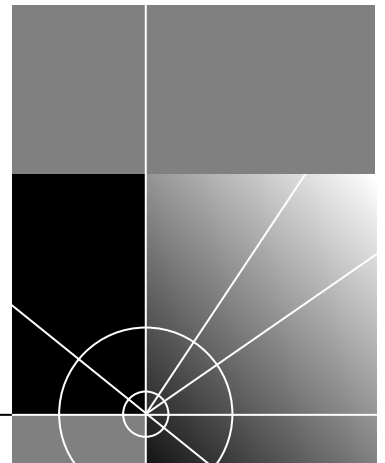
X.25 WAN MIB	The X.25 WAN (X25w) MIB provides management for the X.25 WAN interface and subnet.						
Products using this MIB	This MIB is used by the X.25 card.						
Registration ID	<p>This MIB is an Internet Enterprise MIB registered under the USR enterprise node.</p> <p>iso.org.dod.internet.private.enterprises.usr.nas.x25w (1.3.6.1.4.1.429.1.12)</p>						
Tables	<p>The X25g MIB contains these tables:</p> <ul style="list-style-type: none"> ■ x25wanAdmnTable — The X.25 WAN Administration table contains an entry for each manageable X.25 Subnet in the chassis. It provides a means to configure the serial interface associated with the subnet. ■ x25wanOperTable — The X.25 WAN Operational table. ■ x25wanStatsTable — The X.25 WAN Statistics table contains counters which allow the operator to view activity which can indicate the health of the WAN connection. ■ x25wanTrapEnaTable — The X.25 WAN Trap Enable table contains an entry for each manageable X.25 Subnet in the chassis. It provides a means to enable reporting of traps associated with the subnet. 						
Traps	<p>The X25w MIB contains these trap enables:</p> <table> <tr> <th>Trap enable</th><th>Generates this trap</th></tr> <tr> <td>x25wanTrapEnaOutOfSvc</td><td>gwWanPortOutOfService (35)</td></tr> <tr> <td>x25wanTrapEnaLinkActive</td><td>gwWanPortLinkActive (36)</td></tr> </table>	Trap enable	Generates this trap	x25wanTrapEnaOutOfSvc	gwWanPortOutOfService (35)	x25wanTrapEnaLinkActive	gwWanPortLinkActive (36)
Trap enable	Generates this trap						
x25wanTrapEnaOutOfSvc	gwWanPortOutOfService (35)						
x25wanTrapEnaLinkActive	gwWanPortLinkActive (36)						

Refer to the chapter titled *Trap Reference* for additional information about traps.



CONFIGURATION

- Chapter 17** Configuring Basic NMC Card Parameters
- Chapter 18** Configuring NMC Accounting and Event Logging
- Chapter 19** Configuring NMC Hub Security
- Chapter 20** Configuring NMC NTP Servers
- Chapter 21** Configuring NMC Dial-out



CONFIGURING BASIC NMC CARD PARAMETERS

This chapter contains guidelines for using NMC MIB objects to configure these features on the NMC:

- Basic NMC and user interface (UI) configuration
- WAN SLIP port
- Added cost features
- Basic NMC security
- Authorized access

These parameters are almost always set through the NMC User Interface (UI) port only. The SNMP information is included in this chapter for your reference. Refer to the *NMC Product Reference* for UI configuration information.

NMC Command Table Overview

The *nmcCmdFunction* object is used for basic system functions. You must use these commands when issuing **set** commands to the NMC.

This table describes their use. Refer to the chapter “NMC MIB” for additional details about NMC commands.

Command	Description
noCommand(1)	This is the default value for <i>nmcCmdFunction</i> . Set <i>nmcCmdFunction</i> to abort any command currently in progress.
saveToNvram(2)	Use this command to save the current configuration cache for each NAC to the NMC’s NVRAM. This command also saves the current state of each of the NMC’s configurable parameters, except for parameters within the <i>nmcUiCfg</i> group.

Command	Description (continued)
restoreFromDefaults(3)	<p>Use this command to restore all configurable NMC parameters (except those in the <i>nmcUiCfg</i> group) to their default state.</p> <p>This command also restores the cache for each chassis NAC to its defaults values for the given card type.</p>
restoreFromNvram(4)	<p>Use this command to restore all configurable NMC parameters (except those in the <i>nmcUiCfg</i> group) to the value last saved in the NMC's NVRAM.</p> <p>This command also attempts to restore the cache for each NAC to the values last saved to the NMC's NVRAM.</p>
nonDisruptSelfTest(5)	Use this command to initiate a nondisruptive self test while the NMC is online.
softwareReset(6)	Use this command to reset the NMC from a remote location.
saveUiParamsToEEPROM(7)	Use this command to save the value of the MIB objects within <i>nmcuiCfg</i> to the NMC's EEPROM.
restoreNmcFromDefaults(8)	<p>Use this command to restore all configurable NMC parameters (except those in the <i>nmcUiCfg</i> group) to their default state.</p> <p>This command only restores the NMC defaults.</p>
restoreNmcFromNvram(9)	<p>Use this command to restore all configurable NMC parameters (except those in the <i>nmcUiCfg</i> group) to the value last saved in the NMC's NVRAM.</p> <p>This command only restores the NMC NVRAM values.</p>
bulkFileUpload(10)	<p>Use this command to :</p> <ul style="list-style-type: none"> ■ Upload the NMC NVRAM image file to a MS for transferring chassis configurations (.nvr file) ■ Upload the single configuration file (in CFM format) to a MS (.cfm file) ■ Upload the modem history file (.hst file)
bulkFileDownload(11)	<p>Use this command to :</p> <ul style="list-style-type: none"> ■ Download the NMC NVRAM image file to a MS for transferring chassis configurations (.nvr file) ■ Download the single configuration file (in CFM format) to a MS (.cfm file) ■ Download the modem history file (.hst file)
openAuxOutputPort1(12)	Use this command to open auxiliary output port 1.
openAuxOutputPort2(13)	Use this command to open auxiliary output port 2.
closeAuxOutputPort1(14)	Use this command to close auxiliary output port 1.

Command	Description (continued)
closeAuxOutputPort2(15)	Use this command to close auxiliary output port 2.
closeAuxOutputPort1(14)	Use this command to close auxiliary output port 1.
closeAuxOutputPort2(15)	Use this command to close auxiliary output port 2.

What you Must Configure through the NMC User Interface

You must use the User Interface (UI) to complete initial NMC configuration for these parameters:

- LAN IP address (nmcUiCfgLanIPAddr)
- LAN subnet mask (nmcUiCfgLanSubnetMask)
- WAN IP address (nmcUiCfgWanIPAddr)
- WAN subnet mask (nmcUiCfgWanSubnetMask)
- Default gateway IP address (nmcUiCfgDefaultGwylP)

Any subsequent changes may be made with a MIB browser or Total Control Manager.

Setting Basic NMC System Configuration

Use Total Control Manager or an SNMP browser to configure the NMC. Refer to “NMC MIB” in this document for additional information about MIB objects. Refer to the Total Control Manager documentation set for additional information about Total Control Manager.

Total Control Manager name/function	Object	Notes
System Time	nmcCfgSystemTime 1.3.6.1.4.1.429.1.2.1.1	
System Date	nmcCfgSystemDate 1.3.6.1.4.1.429.1.2.1.2	
Greenwich Mean Time	nmcGmtime 1.3.6.1.4.1.429.1.2.1.3	
Time Zone	nmcTimezone 1.3.6.1.4.1.429.1.2.1.4	
enable traps	nmcCfgAuthFailTrapEnable 1.3.6.1.4.1.429.1.2.1.5	Not part of Total Control Manager

Total Control Manager		
name/function	Object	Notes (continued)
Daylight Savings Time	nmcDaySavingTime 1.3.6.1.4.1.429.1.2.1.6	
Auto Config on Card Initialization *	nmcPowerUpAutoCfgEnable 1.3.6.1.4.1.429.1.2.1.9	<p>Disable <i>nmcPowerUpAutoCfgEnable</i> so the NMC does not automatically configure the chassis NACs upon NAC installation, NMC installation, or chassis power-up.</p> <p>Enable <i>nmcPowerUpAutoCfgEnable</i> so the NMC automatically configures the NACs upon NAC installation, NMC installation, or chassis power-up.</p>
Chassis Name	uchasDescr 1.3.6.1.4.1.429.1.1.3.2	Enter a four-digit chassis identification.
NMC LED Display	uchasDisplayName 1.3.6.1.4.1.429.1.1.3.3	
TFTP Timeout	nmcCfgTFTPTIMEOUT 1.3.6.1.4.1.429.1.2.1.21	The NMC uses TFTP to perform file transfers during operations such as software download (SDL). Configure this parameter to specify the timeout duration before the NMC terminates a TFTP session. For example, use this timeout in a network where there are long delay times in getting packets to the NMC. Prior to completing an SDL, increase the timeout to a high value of 30 seconds. This allows a very long delay in getting a TFTP packet to the NMC.

Total Control Manager name/function	Object	Notes (continued)
Unique Call Reference Number	nmcCfgSessionIDNewFmt 1.3.6.1.4.1.429.1.2.1.38	Enable <i>nmcCfgSessionIDNewFmt</i> to send a universal call reference number format from all NACs to a RADIUS server. Disable <i>nmcCfgSessionIDNewFmt</i> if you are using a non-HiPer chassis.

Configuration notes **nmcPowerUpAutoCfgEnable**

This is what happens when you initialize a HiPer DSP NAC:

1 If you set:

If you set command to enable (1) nmcPowerUpAutoConfigEnable

to enable (1), then you will refresh both the card values and the trap settings.

If you set command to disable (2) If you set:

nmcPowerUpAutoConfigEnable

to disable (2), then only the traps will refresh.

2 This command from the T1H MIB is sent to the HiPer DSP NAC:

refreshCfg1Chans
refreshCfg2Chans
refreshCfg3Chans
refreshCfg4Chans

3 The NAC settings will refresh as follows:

If you...	The NAC refreshes...
Saved templates to NVRAM	to your template settings
Did not save templates	to the factory defaults

Setting User Interface Configuration

For convenience when remotely managing an NMC, you may use the objects in the `nmcUiCfg` group to set the configurable parameters of the local User Interface (UI) port. These parameters include IP addresses, subnet masks, and community strings.

Due to the complexities associated with changing many of these parameters while the NMC is running operational code, a **set** to most of these objects results only in setting a RAM location within the NMC. You must issue the **saveUiParamsToEEPROM** command (*nmcCmdFunction*), followed by **softwareReset** (*nmcCmdFunction*) or power cycle in order for the new values to take effect.

The exception to this above rule occurs when setting community strings. Both the public (read-only) and private (read-write) community strings take effect immediately. If the NMC is reset without issuing a **saveUiParamsToEEPROM** command, they will revert to the values last saved to EEPROM.

Total Control Manager name/function	Object	Notes
LAN IP Address	<code>nmcUiCfgLanIPAddr</code> 1.3.6.1.4.1.429.1.2.8.1	
LAN Subnet Mask	<code>nmcUiCfgLanSubnetMask</code> 1.3.6.1.4.1.429.1.2.8.2	
WAN IP Address	<code>nmcUiCfgWanIPAddr</code> 1.3.6.1.4.1.429.1.2.8.3	
WAN Subnet Mask	<code>nmcUiCfgWanSubnetMask</code> 1.3.6.1.4.1.429.1.2.8.4	
Default Gateway IP Address	<code>nmcUiCfgDefaultGwylP</code> 1.3.6.1.4.1.429.1.2.8.5	
NMC NIC LAN Interface	<code>nmcUiCfgLanIfEnable</code> 1.3.6.1.4.1.429.1.2.8.8	

Total Control Manager name/function	Object	Notes (continued)
Route Traffic between LAN & WAN	nmcUiCfgRouteEnable 1.3.6.1.4.1.429.1.2.8.10	Set routing to "Enable" to allow remote access to the entire LAN through the NMC SLIP port. Set routing to "Disable" to disable LAN routing and only allow remote access to the NMC for chassis management.
UI Port Inactivity Time (minutes)	nmcUiCfgInactiveTime 1.3.6.1.4.1.429.1.2.8.14	If there is no activity on a SLIP port for the specified time duration, the port drops the connection. This applies to both dialed-out and dialed-in calls. If the UI console remains inactive for the duration, the display reverts back to the password request screen. NOTE: If <i>Total Control Manager</i> or another management device connected to the physical port is polling at rates lower than the specified inactivity time, the connection will not be dropped.
Password for UI	nmcUiCfgPassword 1.3.6.1.4.1.429.1.2.8.15.0	

Configuring the Serial Line Internet Protocol Port

This section describes configuring the Serial Line Internet Protocol (SLIP) port to allow remote chassis management. The SLIP port uses the Wide Area Network (WAN) protocol.

Refer to the *HiPer NMC Getting Started Guide* for instructions about how to connect to the NMC's console port and establish a terminal session.



3Com strongly recommends using the NMC user interface to configure the SLIP port. Refer to the HiPer NMC Product Reference for additional information.

If you are using an Ethernet NIC with the NMC, you may configure the CH1 port to serve as a second SLIP port. When CH1 is configured to be a SLIP port, the NMC NAC recognizes it as WAN2. For this port to function correctly, you must configure the WAN2 IP address and subnet mask. This is the address and subnet you will use to access the NMC remotely.

Typically, once you make these settings, you do not need to change them for daily operation.



DIP switch 6 must be set to ON to enable CH1 to serve as a second SLIP port. Refer to the HiPer NMC Getting Started Guide.

M name/function	Object	Notes
Set UI Port to SLIP Port	nmcUiCfgUiSlipCfg 1.3.6.1.4.1.429.1.2.8.11	
Second SLIP Port IP Address	nmcUiCfgWan2IpAddr 1.3.6.1.4.1.429.1.2.8.12	
Second SLIP Port Subnet Mask	nmcUiCfgWan2SubnetMask 1.3.6.1.4.1.429.1.2.8.13	

Configuring added cost features

Several NMC features are available at additional cost from 3Com. These include (but are not limited to): RADIUS, cellular support, and v.90.

If you ordered any of these features, the NMC should ship from the factory with the features enabled. If you need to add additional-cost features, contact your sales representative to obtain the appropriate feature enable string. You will need to provide the serial number of the NMC on which you are enabling the feature.

Total Control Manager name/function	Object	Notes
Hub security	uchasSlotStatFeEna	This object can be read to determine what options have been enabled in the NACs. It uses individual bits to represent the enable status of the features which are NAC specific. Bit Masks: <ul style="list-style-type: none"> ■ 0x1 (hub security) ■ 0x2 (cellular) ■ 0x4 (auto response) ■ 0x20 (x2/V.90) ■ 0x80 (PIAFs)
Cellular	1.3.6.1.4.1.429.1.1.1.1.1.8	
Auto Response		
x2/V.90		
PIAFs		

Setting basic NMC security

You may enable a password to restrict access to the NMC UI and the configuration menus. You must set the community strings before you can enable UI password protection. When the password feature is enabled, two levels of security exist: read-only access (SNMP read string) and read and write access (SNMP write string).



Setting NMC community strings and the UI password

DIP switch 6 must be ON to enable a password. Refer to the NMC Getting Started Guide.

Function	Object	Notes
private community string	nmcUiCfgPrivateString 1.3.6.1.4.1.429.1.2.8.7	Not in Total Control Manager
public community string	nmcUiCfgPublicString 1.3.6.1.4.1.429.1.2.8.9	Not in Total Control Manager
UI password protection	nmcUiCfgPassword 1.3.6.1.4.1.429.1.2.8.15	Not in Total Control Manager

Setting the authorized access list

The authorized access list is a list of all IP addresses that are allowed to access the NMC. IP addresses that are not on this list will not be allowed to access the NMC. You may clear all IP addresses from this list by reinitializing it. All IP addresses will have full access rights to the NMC until you set a new authorized access list. Full access for all IP addresses will make the network less secure.

The *nmcAuth* Authorized Access Group table defines which management stations are allowed to access the NMC. Each MS is a table entry.

Function	Object	Notes
IP address of each authorized MS	<i>nmcAuthACclpAddr</i> 1.3.6.1.4.1.429.1.2.9.1	Not in Total Control Manager
Netmask of each authorized MS	<i>nmcAuthAccNetMask</i> 1.3.6.1.4.1.429.1.2.9.2	Not in Total Control Manager must be set to other than 0.0.0.0
description of each authorized MS	<i>nmcAuthAccDescr</i> 1.3.6.1.4.1.429.1.2.9.3	Not in Total Control Manager optional

The table is indexed by IP address (*nmcAuthACclpAddr*). The other objects in each row include a required subnet mask (*nmcAuthAccNetMask*) and optional text descriptions (*nmcAuthAccDescr*). The network mask (or address mask) masks the corresponding IP address to allow access to the range of stations with IP addresses that fall within the host identification range after the mask is applied. You may add a maximum of 10 rows to the table.

To add a row to the table

- 1 Use *nmcAuthACclpAddr* to add a MS.
- 2 Set a value other than 0.0.0.0 for *nmcAuthAccNetMask*.
- 3 If needed, add an optional description using *nmcAuthAccDescr*.

To delete a row from the table Set the *nmcAuthAccNetMask* to 0.0.0.0.

To clear the entire authorized access list Use the User Interface port to clear the authorized access list. Alternately, DIP switch 5 on the NMC card can be used to boot with an empty list.



*All changes are immediate. However, any change must be saved to NVRAM by issuing the **saveToNvram** command on the NMC in order for the change to remain valid after NMC is reset.*

CONFIGURING NMC ACCOUNTING AND EVENT LOGGING

This chapter contains information about configuring NMC accounting and event logging servers. Additional information about these MIB objects is located in this document in the chapter “NMC MIB”.

Overview

The 3Com accounting and event logging application captures and logs a wide variety of call accounting and event logging information to simple ASCII log files. Like the hub security option, this feature relies on the RADIUS client-server protocol.

The NMC communicates with the accounting server via the LAN connection. The NMC sends the server information about such things as modem connectivity and specific chassis events. The server itself does not generate reports from the data; a post-processing application or database script is required to format the logged data into a readable format.

The NMC MIB contains many accounting and event-logging parameters. These include configurable server selections and traps for any loss of server connection.

Configuring NMC Accounting

To configure accounting, you must configure the NMC client that sends the data, then configure the server(s) that creates the logs. Use Total Control Manager or an SNMP browser to configure the NMC. Use the 3Com accounting and event logging application to configure the server.

Refer to “*NMC MIB*” in this document for additional information about MIB objects. Refer to the Total Control Manager documentation set for additional information about Total Control Manager. Refer to the 3Com security and accounting documentation set for additional information about these servers.

Configuring the NMC Logging Group

Configure the NMC's logging group with these objects:

Total Control Manager Name	Object
Event Logging Server	nmcCfgLogSrvrSelect 1.3.6.1.4.1.429.1.2.1.14
Primary Log Server IP Address	nmcCfgLogPriSrvrAddr 1.3.6.1.4.1.429.1.2.1.15
Secondary Log Server IP Address	nmcCfgLogSecSrvrAddr 1.3.6.1.4.1.429.1.2.1.16
Log Server's UDP Port	nmcCfgLogUdpPortNum 1.3.6.1.4.1.429.1.2.1.17
Logging Client TX Retry	nmcCfgLogRetryCnt 1.3.6.1.4.1.429.1.2.1.18
Log Group Selection	nmcCfgLogCallStatGrpSel 1.3.6.1.4.1.429.1.2.1.19
MD5 Calculation	nmcCfgLogMD5Calc 1.3.6.1.4.1.429.1.2.1.20
Third Backup Logging Server	nmcCfgLog3SrvrAddr 1.3.6.1.4.1.429.1.2.1.24
Fourth Backup Logging Server	nmcCfgLog4SrvrAddr 1.3.6.1.4.1.429.1.2.1.25
Fifth Backup Logging Server	nmcCfgLog5SrvrAddr 1.3.6.1.4.1.429.1.2.1.26
Sixth Backup Logging Server	nmcCfgLog6SrvrAddr 1.3.6.1.4.1.429.1.2.1.27
Seventh Backup Logging Server	nmcCfgLog7SrvrAddr 1.3.6.1.4.1.429.1.2.1.28
Eighth Backup Logging Server	nmcCfgLog8SrvrAddr 1.3.6.1.4.1.429.1.2.1.29
Logging Server's Name	nmcCfgLogSrvrName 1.3.6.1.4.1.429.1.2.1.30
Logging Server DNS Enable	nmcCfgLogDnsEna 1.3.6.1.4.1.429.1.2.1.34
Status-Server Request Interval	nmcCfgLogStatusInterval 1.3.6.1.4.1.429.1.2.1.35

Configuring Logging Servers

Configure logging servers with these objects:

Total Control Manager Name	Object
Event Logging Server	nmcCfgLogSvrSelect 1.3.6.1.4.1.429.1.2.1.14
Primary Log Server IP Address	nmcCfgLogPriSvrAddr 1.3.6.1.4.1.429.1.2.1.15
Secondary Log Server IP Address	nmcCfgLogSecSvrAddr 1.3.6.1.4.1.429.1.2.1.16
Log Server's UDP Port	nmcCfgLogUdpPortNum 1.3.6.1.4.1.429.1.2.1.17
Logging Client TX Retry	nmcCfgLogRetryCnt 1.3.6.1.4.1.429.1.2.1.18
Log Group Selection	nmcCfgLogCallStatGrpSel 1.3.6.1.4.1.429.1.2.1.19
MD5 Calculation	nmcCfgLogMD5Calc 1.3.6.1.4.1.429.1.2.1.20
Third Backup Logging Server	nmcCfgLog3SvrAddr 1.3.6.1.4.1.429.1.2.1.24
Fourth Backup Logging Server	nmcCfgLog4SvrAddr 1.3.6.1.4.1.429.1.2.1.25
Fifth Backup Logging Server	nmcCfgLog5SvrAddr 1.3.6.1.4.1.429.1.2.1.26
Sixth Backup Logging Server	nmcCfgLog6SvrAddr 1.3.6.1.4.1.429.1.2.1.27
Seventh Backup Logging Server	nmcCfgLog7SvrAddr 1.3.6.1.4.1.429.1.2.1.28
Eighth Backup Logging Server	nmcCfgLog8SvrAddr 1.3.6.1.4.1.429.1.2.1.29
Logging Server's Name	nmcCfgLogSvrName 1.3.6.1.4.1.429.1.2.1.30
Logging Server DNS Enable	nmcCfgLogDnsEna 1.3.6.1.4.1.429.1.2.1.34
Status-Server Request Interval	nmcCfgLogStatusInterval 1.3.6.1.4.1.429.1.2.1.35

Configuring RADIUS DNS Servers

Configure RADIUS DNS servers with these objects:

Total Control Manager Name	Object
Primary DNS Server's IP Address	nmcCfgDnsPriSvrAddr 1.3.6.1.4.1.429.1.2.1.22
Secondary DNS Server's IP Address	nmcCfgDnsSecSvrAddr 1.3.6.1.4.1.429.1.2.1.23
Primary DNS Server Retries	nmcCfgDnsRetryCnt 1.3.6.1.4.1.429.1.2.1.31
DNS Server's UDP Port	nmcCfgDnsUdpPortNum 1.3.6.1.4.1.429.1.2.1.32
DNS Server Select	nmcCfgDnsSvrSelect 1.3.6.1.4.1.429.1.2.1.33

Setting Server Traps

If desired, set these traps to inform you of server problems:

Total Control Manager Name /Trap	Object
On Authentication Failure	nmcCfgAuthFailTrapEnable 1.3.6.1.4.1.429.1.2.1.5
On DNS Server Lost dnsSvrLoss(80)	nmcTeDnsSvrLoss 1.3.6.1.4.1.429.1.2.7.14
On NTP Server Lost ntpSvrLossConn(81)	nmcTeNtpSvrLoss 1.3.6.1.4.1.429.1.2.7.15
On NTP Server Restored ntpSvrRestConn(82)	nmcTeNtpSvrRestore 1.3.6.1.4.1.429.1.2.7.16
On Primary NTP Server Failed ntpSvrDgrConn(85)	nmcTeNtpSvrDegraded 1.3.6.1.4.1.429.1.2.7.17
On DNS Server Restored dnsSvrRestore(94)	nmcTeDnsSvrRestore 1.3.6.1.4.1.429.1.2.7.18
On Primary DNS Server Failed dnsSvrDegraded(95)	nmcTeDnsSvrDegraded 1.3.6.1.4.1.429.1.2.7.19
On Logging Server Restored acctSvrRestore(96)	nmcTeLogSvrRestore 1.3.6.1.4.1.429.1.2.7.20
On Logging Server Lost acctSvrLoss(49)	nmcTeLogSvrLoss 1.3.6.1.4.1.429.1.2.7.9
On Logging Server Group Operational acctSvrGroupOper(97)	nmcTeLogSvrGroupOper 1.3.6.1.4.1.429.1.2.7.21
On Logging Server Group Degraded acctSvrGroupDegr(98)	nmcTeLogSvrGroupDegr 1.3.6.1.4.1.429.1.2.7.22

Total Control Manager Name /Trap	Object (continued)
On Logging Server Group Non-Operational acctSvrGroupNonOp(99)	nmcTeLogSvrGroupNonOp 1.3.6.1.4.1.429.1.2.7.23.
On RADIUS Server Restored securitySvrRestore(100)	nmcTeSecSvrRestore 1.3.6.1.4.1.429.1.2.7.24
On RADIUS Server Group Operational securitySvrGrpOper(101)	nmcTeSecSvrGroupOper 1.3.6.1.4.1.429.1.2.7.25
On RADIUS Server Group Degraded securitySvrGrpDegr(102)	nmcTeSecSvrGroupDegr 1.3.6.1.4.1.429.1.2.7.26
On RADIUS Server Group Non-Operational securitySvrGrpNonOp(103)	nmcTeSecSvrGroupNonOp 1.3.6.1.4.1.429.1.2.7.27

Refer to the chapter titled *Trap Reference* for additional information about traps.

CONFIGURING NMC HUB SECURITY

This chapter contains information about configuring NMC hub security. Additional information about these MIB objects is located in this document in the chapter “NMC MIB”.

Overview

Hub security is an option that allows you to prevent unauthorized users from dialing into or out of your chassis. This feature uses the RADIUS protocol to provide user authentication.

User authentication is based upon encrypted or “secret key” information passed between the RADIUS client (NMC) and the server. Successful operation requires the NMC and the server to contain matching encryption information.

Configuring NMC Hub Security

Use Total Control Manager or an SNMP browser to configure the NMC. Refer to “NMC MIB” in this document for additional information about MIB objects. Refer to the Total Control Manager documentation set for additional information about Total Control Manager.

Total Control Manager Name	Object
User Name Prompt	nmcHsDialInOutNamePrompt 1.3.6.1.4.1.429.1.2.6.1
User Password Prompt	nmcHsDialInOutPsswdPrompt 1.3.6.1.4.1.429.1.2.6.2
Dial Back Name Prompt	nmcHsDialBackNamePrompt 1.3.6.1.4.1.429.1.2.6.3
Dial Back Password Prompt	nmcHsDialBackPsswdPrompt 1.3.6.1.4.1.429.1.2.6.4

Total Control Manager Name	Object (continued)
Dial Back Number Prompt	nmcHsDialBackPhonePrompt 1.3.6.1.4.1.429.1.2.6.5
Dial Back Pending Prompt	nmcHsDialBackPendPrompt 1.3.6.1.4.1.429.1.2.6.6
Modem Select Prompt	nmcHsMdmSelectPrompt 1.3.6.1.4.1.429.1.2.6.7
Login Failed Message	nmcHsLoginFailedMsg 1.3.6.1.4.1.429.1.2.6.8
Restricted Number Prompt	nmcHsPhoneRestrictPrompt 1.3.6.1.4.1.429.1.2.6.9
Invalid Modem Select Message	nmcHsInvalidMdmSelecMsg 1.3.6.1.4.1.429.1.2.6.10
No Modems Available Message	nmcHsNoMdnsAvailMsg 1.3.6.1.4.1.429.1.2.6.11
Connect Success Message	nmcHsConnectSuccessMsg 1.3.6.1.4.1.429.1.2.6.12
New Password Message	nmcHsNewPasswordPrompt 1.3.6.1.4.1.429.1.2.6.13
Change Password Message	nmcHsChangePasswordMsg 1.3.6.1.4.1.429.1.2.6.14
Response Timeout	nmcHsPromptRspTimeout 1.3.6.1.4.1.429.1.2.6.15
Response Attempt Limit	nmcHsPromptRspAttempts 1.3.6.1.4.1.429.1.2.6.16
Response Echo Enable	nmcHsPromptRspEchoEna 1.3.6.1.4.1.429.1.2.6.17
Dial Back Delay	nmcHsDialBackDelay 1.3.6.1.4.1.429.1.2.6.18
Dial Back Attempt Limit	nmcHsDialBackAttempts 1.3.6.1.4.1.429.1.2.6.19
Primary Security Server IP Address	nmcHsSecuritySrvrAddr 1.3.6.1.4.1.429.1.2.6.20
Security Server UDP Port	nmcHsSecuritySrvrPort 1.3.6.1.4.1.429.1.2.6.21

Total Control Manager Name	Object (continued)
Security Server Retries	nmcHsSecuritySvrRetries 1.3.6.1.4.1.429.1.2.6.22
Modem Attempt Limit	nmcHsMdmAttemptLimit 1.3.6.1.4.1.429.1.2.6.23
Security Server Unavailable	nmcHsServerUnavailable 1.3.6.1.4.1.429.1.2.6.24
Secondary Security Server IP Address	nmcHsSecondarySvrAddr 1.3.6.1.4.1.429.1.2.6.26
Third RADIUS Security Backup Server	nmcHsSecurity3SvrAddr 1.3.6.1.4.1.429.1.2.6.28
Fourth RADIUS Security Backup Server	nmcHsSecurity4SvrAddr 1.3.6.1.4.1.429.1.2.6.29
Fifth RADIUS Security Backup Server	nmcHsSecurity5SvrAddr 1.3.6.1.4.1.429.1.2.6.30
Sixth RADIUS Security Backup Server	nmcHsSecurity6SvrAddr 1.3.6.1.4.1.429.1.2.6.31
Seventh RADIUS Security Backup Server	nmcHsSecurity7SvrAddr 1.3.6.1.4.1.429.1.2.6.32
Eighth RADIUS Security Backup Server	nmcHsSecurity8SvrAddr 1.3.6.1.4.1.429.1.2.6.33
RADIUS Security Server Host Name	nmcHsSecuritySvrName 1.3.6.1.4.1.429.1.2.6.34
RADIUS Security Server DNS	nmcHsSecuritySvrDnsEna 1.3.6.1.4.1.429.1.2.6.35
Status-Server Request Interval	nmcHsSecurityStatusInt 1.3.6.1.4.1.429.1.2.6.36
Password Prompt	nmcHsDiPasswdEnaDis 1.3.6.1.4.1.429.1.2.6.27

Configuring Hub Security Traps

If desired, set these traps to inform you of hub security problems:

Total Control Manager name / trap	Object
Dial Back Restrict Number Trap dialBackRestrictedNum(40)	nmcTeDialBackRestrictNum 1.3.6.1.4.1.429.1.2.7.4
User Blacklist Trap userBlacklisted(41)	nmcTeUserBlacklist 1.3.6.1.4.1.429.1.2.7.5
User Blacklist Login Trap loginAttemptByBlacklistedUser(42)	nmcTeUserBlacklistLogin 1.3.6.1.4.1.429.1.2.7.6
Response Attempt Limit Exceeded Trap responseAttemptLimExceeded(43)	nmcTeRespAttemptLimExceeded 1.3.6.1.4.1.429.1.2.7.7
Login Attempt Limit Exceeded Trap mdmLoginAttemptLimExceeded(44)	nmcTeLoginAttmptLimExceed 1.3.6.1.4.1.429.1.2.7.8
Security Server Lost securitySrvrLoss(62)	nmcTeSecSrvrLoss 1.3.6.1.4.1.429.1.2.7.10

Refer to the chapter titled *Trap Reference* for additional information about traps.

CONFIGURING NMC NTP SERVERS

This chapter contains information about configuring NMC NTP servers. Additional information about these MIB objects is located in this document in the chapter “NMC MIB”.

Overview

NTP is the Standard Network Time Protocol (also known as SNTP). Use the objects in the nmcNtp table to configure the NMC and all other cards in the chassis so they are in sync with other servers on a network.

Configuring NTP servers

Use Total Control Manager or an SNMP browser to configure the NMC. Refer to “NMC MIB” in this document for additional information about MIB objects. Refer to the Total Control Manager documentation set for additional information about Total Control Manager.

Total Control Manager Name	Object
Primary NTP Server's IP Address	nmcNtpSvrPrimAddr 1.3.6.1.4.1.429.1.2.10.1
Secondary NTP Server's IP Address	nmcNtpSvrSecdAddr 1.3.6.1.4.1.429.1.2.10.2
Synchronization Interval (sec)	nmcNtpSyncInterval 1.3.6.1.4.1.429.1.2.10.3
Operational Mode	nmcNtpOperationalMode 1.3.6.1.4.1.429.1.2.10.4
<i>Note: Set to unicast(2) to enable NTP</i>	

Setting server traps

If desired, set these traps to inform you of NTP server problems:

Total Control Manager name / trap	Object
On NTP Server Lost ntpSvrLossConn (81)	nmcTeNtpSvrLoss 1.3.6.1.4.1.429.1.2.7.15
On NTP Server Restored ntpSvrRestConn (82)	nmcTeNtpSvrRestore 1.3.6.1.4.1.429.1.2.7.16
On Primary NTP Server Failed ntpSvrDegrConn (85)	nmcTeNtpSvrDegraded 1.3.6.1.4.1.429.1.2.7.17

Refer to the chapter titled *Trap Reference* for additional information about traps.

CONFIGURING NMC DIAL-OUT

This chapter contains information about configuring NMC dial-out. Additional information about these MIB objects is located in this document in the chapter “NMC MIB”.

Overview

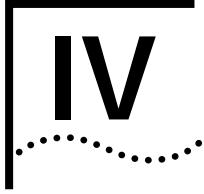
Use these objects to configure dial-up access when the NMC's WAN port is physically connected to a modem.

Configuring NMC Dial-out

Use Total Control Manager or an SNMP browser to configure the NMC. Refer to “NMC MIB” in this document for additional information about these MIB objects. Refer to the Total Control Manager documentation set for additional information about Total Control Manager.

Total Control Manager Name	Object
AT Init String	<div>nmcCfgAtString</div> <div>1.3.6.1.4.1.429.1.2.1.8</div> <div><i>Note: This AT string enables command mode local echo, verbal result codes, and hardware flow control. Maintain the default setting for proper functionality with the modem, but you may add additional settings to the string as desired. The default value is AT&FEOQ0&H1&R2&B1V1</i></div>
WAN Connect Number	<div>nmcCfgWanDialOutPhoneNum</div> <div>1.3.6.1.4.1.429.1.2.1.7</div> <div><i>Note: Use nmcUiCfgInactiveTime to configure the time for which the SLIP port can remain inactive before the NMC hangs up the call.</i></div>

Total Control Manager Name	Object (continued)
WAN Dial Out Attempt Limit	nmcCfgNumWanRetries 1.3.6.1.4.1.429.1.2.1.10 <i>Note: If the number of retries is exceeded, the NMC discards the packet. If nmcCfgNumWanRetries is 0, the NMC tries to send the packet "forever" and will not discard it.</i>
Pause between Retries(sec)	nmcCfgWanRetryPause 1.3.6.1.4.1.429.1.2.1.11
Retries Suspension Interval(sec)	nmcCfgWanRetrySuspendTime 1.3.6.1.4.1.429.1.2.1.12
Connection Failure Limit	nmcCfgNumFailBefSuspend 1.3.6.1.4.1.429.1.2.1.13



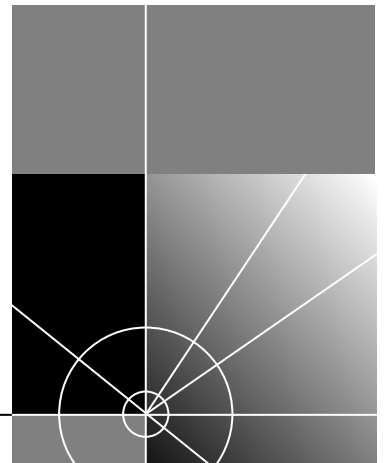
MONITORING THE SYSTEM

Chapter 22 Traps Overview

Chapter 23 Trap Reference

Chapter 37 Modem Disconnect and Connect Fail Reasons

Chapter 25 AutoResponse Reference



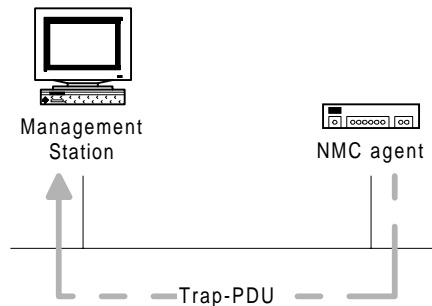
22

TRAPS OVERVIEW

This chapter contains an overview of trap management within the Total Control chassis. Refer to subsequent chapters for trap listings and configuration information.

Understanding Events and Traps

The NMC and all other manageable chassis NACs have an SNMP mechanism for reporting any events on the card to an external Management Station (MS). These event messages are called traps.



Events An event is something that happens to the operating status of a chassis device. This event may include a state change, error, or device reset.

With very few exceptions, an "event" can be reported as either an SNMP trap or a RADIUS log record. The configurable options are: disable all, enable trap, enable log, and enable all. When an event occurs on a NAC, it is forwarded to the NMC. If you "enable" the event when you configure the NAC, the event causes the NMC to generate a trap and/or RADIUS log record. If you "disable" the event, it is discarded and not sent to the NMC.

In some cases, like Management Bus Failure and Watchdog Timeout, the event is related to the NAC, but is detected and reported directly by the NMC.

Traps Traps are unsolicited SNMP messages sent from a network device to a MS to signal that a specific event, or fault, has occurred on or within that network device. Within the chassis, the NMC sends a trap message to a specified MS. The MS station may be the 3Com Alarm Server, or another MS running third-party trap handler/alarm server software.

The data packet containing the event information is called the Trap-PDU, or trap message. Traps are different from device statistics in that they represent one-time events, while statistics show trends of events recurring over a specified time period.

Why Use Traps? Traps allow you to detect, isolate, and correct problems or to monitor events that occur on a NAC or other chassis device. They may be configured through a MIB browser or Total Control Manager.

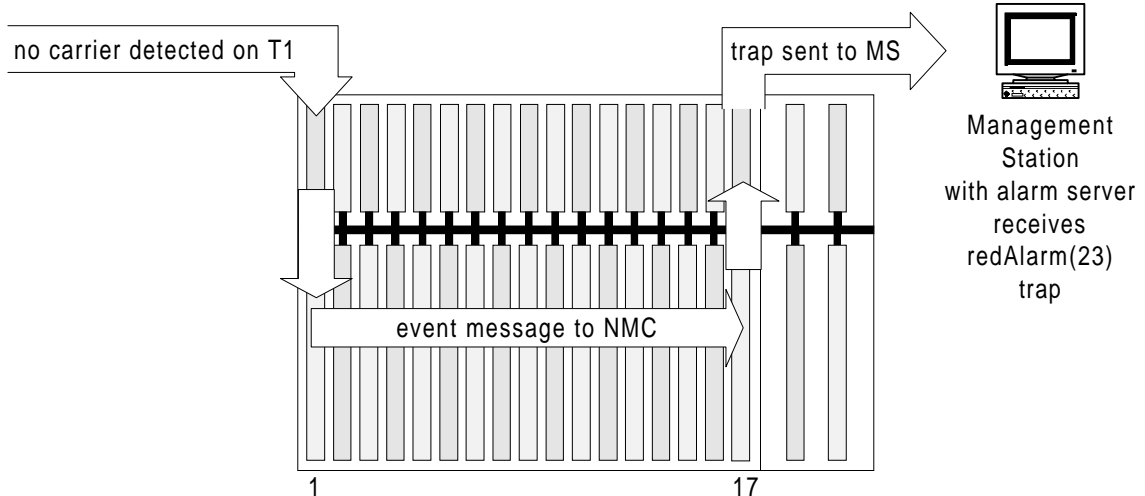
After a MS receives a trap, the subsequent actions are site-specific and independent of 3Com software. Many customers configure their MS software to issue an alarm in response to specific traps. Possible alarms include reports, audible buzzers, and pages to the network administrator.

Within the Total Control chassis, you can enable traps to do the following:

- Physically monitor chassis network devices
- Physically monitor the chassis
- Monitor security
- Use AutoResponse timers to send a trap when a specific event occurs

Example of a USR (3Com) Trap

This illustration shows how a trap is generated when no carrier is detected on the T1. This condition is called a Red Alarm, and indicates the PRI/T1 has lost carrier detect.



Understanding Alarms

An alarm is the action the MS takes in response to the trap message it receives. Alarms are specific to third-party software packages and customer applications. They may include reports, buzzers, or pages to the network manager.

Alarm Servers

In generic terms, an alarm server is a software application that operates on a MS to report traps. These applications are packaged with many popular SNMP browser packages.

3Com's Alarm Server product is an additional software application that ships with Total Control Manager. It logs the traps sent by any chassis under Total Control Manager management on the same LAN or WAN. Alarm Server allows the MS operator to print, acknowledge, and delete

alarm logs. Refer to the Total Control Manager documentation for additional information about 3Com Alarm Server.

Configuring Traps

Trap Enable and Disable

Most traps provide you with an **enable trap/disable all** choice. Some modem events also require you to set an associated threshold that specifies limits for activating the trap. For many traps, it is also possible to log the occurrence of the trap using the chassis Accounting/Event Logging feature via **enable log** or **enable all**.

Traps settings are configured and saved on the NMC card only. When an event occurs within the chassis, a message is sent to the NMC. Events are checked against a trap enable/disable table kept in NMC memory. If the trap is enabled within the NMC, an event message is sent to the MS. If logging is enabled, a log message is sent to the logging server. If traps and/or logging are disabled, the event is discarded.

Configuration Options

Four options typically exist when setting traps (a few traps allow “enable” and “disable” options only):

Command	Description
enableTrap (1)	Enabling the trap allows the NMC to send the trap to the MS alarm server.
disableAll (2)	Disabling all prevents the NMC from sending the trap to the MS alarm server and RADIUS logging server.
enableLog (3)	Enabling the log allows the NMC to send the trap to the RADIUS logging server.
enableAll (4)	Enabling all allows the NMC to send the trap to both the MS alarm server and RADIUS logging server.

Use discretion when configuring traps:

- When **enableAll** is set, two messages are always sent from the NMC (both a trap and a log)
- The NMC can handle a maximum of 65 trap messages simultaneously; additional trap messages may be lost
- Too many traps and logs may cause unacceptable traffic levels on a LAN; you may need to perform a “traffic analysis” to determine the acceptable traffic level

Trap Destination
Table

Traps are sent to all MS that are registered in the NMC's trap destination table. RADIUS logs use a separate destination table. A maximum of ten destinations may be set within the NMC. Use an SNMP MIB browser or Total Control Manager to set the trap destinations. Refer to "Configuring Traps" in this manual, and the Total Control Manager documentation, for additional information.

Entries in the trap destination table include:

- **Destination IP** — the address of one or more alarm and/or logging servers
- **Community string** — SNMP community strings allow the MS to filter trap and log messages that may come from other devices on or outside of the network
- **Description** — additional descriptive text (optional)

By default, the trap destination table contains no entries. This means that traps and log messages will not be generated until you specify a destination IP address(es).

Transient Events

Many events are considered to be transient in that they may only last for a short period of time before they are cleared. For example, noise on a T1 span may cause a temporary line condition that generates a trap. When the condition improves, it generates a "clear" event.

Examples of transient and subsequent clear events include:

Transient Event	"Clear" event
yellowAlarm	yellowAlarmClear
redAlarm	redAlarmClear
lossOfSignal	lossOfSignalClear

Although these traps represent potentially serious conditions, they may clear themselves within a short time. To avoid unnecessary alarms, you may want to build a delay mechanism into your alarm server software. When configuring an alarm server, you will need to determine the appropriate severity level of these alarms based upon your specific network needs. Use transient events to monitor the condition of your network. Performance monitor packages will accumulate these events as statistics, allowing you to determine the quality of a span line by event frequency.

Chassis Trap

Most traps generated within a Total Control chassis are specific to 3Com devices. Five SNMP generic traps are also generated.

Types of Chassis Traps

Use chassis traps to monitor physical network devices within the chassis, and to track and monitor a call's progress through the chassis. You can also use traps for formal performance monitoring through specialty software packages on the MS.

NMC traps

Enable NMC traps to monitor physical events in the chassis. NMC traps can indicate such things as when a card is inserted into the chassis, when a fan fails, and which security servers are in use.

T1, E1, and PRI-ISDN traps

Enable span-level traps to monitor changes occurring on a T1/E1/PRI-ISDN span. Events can indicate physical problems in the Telco network, as well as problems within the chassis. Some traps can also be used to track calls through the chassis.

Modem traps

Enable modem traps to:

- Track abnormal calls
- Perform call accounting
- Monitor physical modem devices
- Monitor packet bus activity

Multiple Traps

In many cases, several events will occur in sequence within the chassis. Depending upon the events, this may generate multiple traps. For example:

Event	Generates this trap
An incoming call arrives from the telco and terminates normally on the HiPer DSP span.	callArriveEventHdsp
A modem establishes an incoming connection	ctlIncomingConnectionEstablished
The telco terminates the call normally.	callTerminateEventHdsp
The modem terminates the incoming connection.	ctlIncomingConnectionTerminated

SNMP Generic Traps

Several generic SNMP traps are predefined in RFC 1157. Five of these traps are used by the NMC card. In addition, 3Com-specific traps are assigned as generic enterprise traps.

These are the SNMP generic traps used by the NMC:



Trap 5 is not used by the NMC.

SNMP Trap	Description
coldStart(0) Cold Start	Trap is generated when the NMC boots
warmStart(1) Warm Start	Trap is generated when the software resets.
linkDown(2) Link Down	Trap occurs when a communication link goes down
linkUp(3) Link Up	Trap occurs when a communication link comes up
authenticationFailure(4) Authentication Failure	Trap occurs if PDU authentication fails or the community string is incorrect. The NMC recieved a requested from an unauthorized MS.
enterpriseSpecific(6) Enterprise Specific	3Com-defined traps. These traps are further defined in the CHS_TRAP MIB.

Chassis Trap MIB

The Chassis Trap (CHS_TRAP) MIB provides a set of traps for management of the entire chassis. The CHS_TRAP MIB tells the SNMP browser trap parser how to display the information it receives in the trap PDU.

Use an SNMP browser or Total Control Manager to display trap messages. Use the information contained within the Chassis Trap MIB to understand the data behind the trap.

Understanding the Chassis Trap MIB

This section explains how to read the information contained within the Chassis trap MIB.



Always make sure the correct version of the Chassis Trap MIB is compiled into the management station. A Chassis Trap MIB built for a previous version of the NMC software may be incompatible and return unreadable data.

This is an excerpt from the CHS_MIB text file, showing the *ctIncomingConnectionTerminated* trap.

```

1  ctIncomingConnectionTerminated TRAP-TYPE
    ENTERPRISE usr
    VARIABLES{
        nmcTrapSequenceNumber,
        nmcStatEventId,
        nmcGmtime,
        uchasSlotIndex,
        uchasEntityIndex,
        uchasEntityObjectID,
        mdmCsCallRefNum,
        mdmCsCallDuration,
        mdmCsDisconnectReason
    }
2  — DESCRIPTION
    "Incoming connection terminated on modem."
3  — --#TYPE "Incoming Connection Terminated"
4  — --#SUMMARY "%d; Incoming Connection Terminated on Slot %d, Chan %d,
        Ref %d, Dur %d, Disc %d"
5  — --#ARGUMENTS { 1 3 4 6 7 8 }
6  — --#SEVERITY INFORMATIONAL
    --#TIMEINDEX 2
    --#HELP "nmm.hlp"
    --#HELPTAG 9999
    --#STATE OPERATIONAL
7  — ::= 56

```

Number	Description
1	<p>These are the variable bindings (also called Var Binds). A variable is an instance of an object type defined according to its OID. A variable binding is the pairing of the name of a variable to the variable's value. This field is a list of OIDs and their corresponding values.</p> <p>Var Binds indicate the values returned in the trap message. The specific Var Binds will be different for each trap message.</p>
2	<p>DESCRIPTION - This is a brief description of the trap, written only within the CHS_MIB text file.</p>
3	<p># TYPE - Provides the trap name. This value is the trap identifier that tells the MS the type of trap it is receiving. When the correct Chassis Trap MIB is integrated in to the MS software, the #Type will be translated to display the human-readable trap name.</p> <p>For example: Trap type 16 = dtrTrue</p>

Number	Description (continued)
4	<p># SUMMARY - This is the message the trap parser will print in the GUI. The value defines for how to display the information contained within the trap PDU. The “%d” placeholders are replaced with the respective arguments.</p> <p>For this trap, the following is true:</p> <p>nmcTrapSequenceNumber = 0</p> <p>nmcStatEventId = 1</p> <p>nmcGmtime = 2</p> <p>uchasSlotIndex = 3</p> <p>uchasEntityIndex = 4</p> <p>uchasEntityObjectID = 5</p> <p>mdmCsCallRefNum = 6</p> <p>mdmCsCallDuration = 7</p> <p>mdmCsDisconnectReason = 8</p> <p># ARGUMENTS { 1 3 4 6 7 8 } indicates that Var Binds 1,3,4,6,7, and 8 will respectively replace %d in the # SUMMARY message.</p> <p>Therefore, the message in the GUI will be constructed like this:</p> <p>“ (value of <i>nmcTrapSequenceNumber</i>); Incoming Connection Terminated on Slot (value of <i>uchasSlotIndex</i>), Channel (value of <i>uchasEntityIndex</i>), Ref (value of <i>mdmCsCallRefNum</i>), Dur (value of <i>mdmCsCallDuration</i>), Disc (value of <i>mdmCsDisconnectReason</i>)”</p>
5	<p># ARGUMENTS - These values refer to the “%d” placeholders in the # SUMMARY. Values for the placeholders are obtained from the variable bindings.</p>
6	<p># SEVERITY - assigns one of three internal severity ratings: informational, minor, and major. Refer to “Trap severity” in this chapter for additional information.</p>
7	<p>This is the assigned trap number.</p>

Understanding the Structure of a SNMP Trap PDU

Each trap message sent by the NMC adheres to a predefined SNMP structure. The common framework allows the trap handler software on the MS to interpret the trap message. When the CHS_TRAP MIB is compiled within the MS, the software is then able to translate the trap message and display the specific USR trap information. This PDU is not acknowledged when it is received by the agent.

SNMPv1 trap PDUs have this format:

Item	Description
PDU type	This field contains a value of "4", indicating that this is a trap PDU.
enterprise	Identifies the value of the NMC's <i>sysObjectId</i>
agent address	The IP address of the NMC that sent the trap
generic trap ID	Identifies any generic SNMP trap; for all USR traps, the value will always be <i>enterpriseSpecific(6)</i> ; refer to the "SNMP generic traps" table in this chapter
specific trap ID	Provides the specific number of the USR trap; if the trap is generic, the number will be "0"
time stamp	The time when the trap occurred, provided by the value of the NMC's <i>sysUpTime</i>
variable bindings	A variable is an instance of an object type defined according to its OID. A variable binding is the pairing of the name of a variable to the variable's value. This field is a list of OIDs and their corresponding values.

Event Messages in an SNMP Browser

Event messages are human-readable reports displayed on an SNMP browser that provide a brief description of the event, along with coded information about the event. An event message provides a brief description of an event, along with the event code associated with that event. How the message is displayed will depend upon the browser.

Numbering the Events

Each trap is individually numbered in the event message to allow MS maintenance and logging.

Sequence number

Every trap that is generated by the NMC is assigned a sequence number by the object *nmcTrapSequenceNumber*. This object increments every time a trap is generated by the NMC. This object is sent as part of the "variable-bindings" list in each trap (the variable-binding is part of the SNMP structure of the trap message). A MS can use the sequence number to detect when a trap has been lost.

The *nmcTrapSequenceNumber* resets every time the NMC reboots.

Event number

Every trap generated by the NMC is also assigned an event number, called *nmcStatEventId*. This object allows you to determine if all generated traps were received by the MS. This value increments once for each event detected by the NMC.

Trouble Clearing Trap Packets

This section provides examples and explanations of trap packets received by an SNMP trap parser that contain unusable trap data.

Example 1

This example shows the result from SNMP when an older version of the CHS_TRAP.MIB is used. This older version does not contain the latest traps, so the MS cannot understand the incoming trap (118). In addition, not all current MIBs were loaded into the parser, so some OIDs could not be interpreted.

```
Tue Jun  2 15:06:28 1998 [ ip.xxx.xxx.xx.xxx.nn.nnn.com ] :
Trap: sequence=2
receive-time=Tue Jun  2 15:06:28 1998
version=0
community=public
source-time=00:28:17.68
trap-type=enterprise
enterprise=U.S. Robotics, Inc.
trap-no=118
trap-name=enterprise specific trap: 118
priority=low

nmcTrapSequenceNumber=77
nmcStatEventId=77
nmcGmtime=793845574

KEY=1
uchasSlotIndex=15
uchasEntityIndex=1

1.3.6.1.4.1.429.1.28.2.1.2.1=3

mdmCsCallRefNum=0

1.3.6.1.4.1.429.1.28.2.1.8.1=0
```

Example 2 This trap is also misunderstood by the trap parser.

```
Tue Jun  2 15:20:45 1998 [ ip.xxx.xxx.xx.xxx.nn.nnn.com ] :  
Trap:
```

```
sequence=3  
receive-time=Tue Jun  2 15:20:45 1998  
version=0  
community=public  
source-time=00:42:37.00  
trap-type=enterprise  
enterprise=U.S. Robotics, Inc.  
trap-no=119  
trap-name=enterprise specific trap: 119  
priority=low
```

```
nmcTrapSequenceNumber=78  
nmcStatEventId=78  
nmcGmtime=793846431
```

```
KEY=1  
uchasSlotIndex=15  
uchasEntityIndex=3
```

```
1.3.6.1.4.1.429.1.28.2.1.2.1=3
```

```
mdmCsCallRefNum=235012143
```

```
1.3.6.1.4.1.429.1.28.2.1.4.1=3  
1.3.6.1.4.1.429.1.6.9.1.1.85.1=793845584  
1.3.6.1.4.1.429.1.6.9.1.1.86.1=793846430  
1.3.6.1.4.1.429.1.6.9.1.1.84.1=0  
1.3.6.1.4.1.429.1.28.2.1.8.1=0
```

```
mdmCsLastNumberDialedIn=  
mdmCsDisconnectReason=0  
mdmCsLastCallingPartyNum=
```



TRAP REFERENCE

This chapter contains a list of all chassis traps, including trap number, trap type, applicable card(s), and trouble clearing information.

Trap Table

The table in this chapter contains a listing of all chassis traps handled by the NMC for release version 6.X.

These explanations apply to the table:

- The trap number represents the trap identification reported in the Chassis Trap (chs_trap) MIB.
- The description column provides the trap name as adapted from the chassis trap MIB, as well as the trap enable object or command.
- The cards affected column lists those chassis NACs affected by the trap.
- Additional text, if provided, offers explanatory and trouble clearing information.



Some traps are labeled "obsolete". These are provided to ensure backward compatibility.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing
1	moduleInserted Trap enable object: <i>uchasModuleInsertedTrapEna</i>	all except NMC	Condition — a card was inserted into an empty slot in the hub. Possible causes and trouble clearing — No corrective action needed unless the event is not expected.
2	moduleRemoved Trap enable object: <i>uchasModuleRemovedTrapEna</i>	all except NMC	Condition — a card was removed from the hub. Possible causes and trouble clearing — No corrective action needed unless the event is not expected.
3	psuWarning Trap enable object: <i>uchasPSUWarningTrapEna</i>	PSU	Condition — the power supply is out of the normal operating range. Possible causes and trouble clearing — The NMC detected a PSU which is not responding to periodic polls. ■ Check the chassis PSU.
4	psuFailure Trap enable object: <i>uchasPSUFailureTrapEna</i>	PSU	Condition — the power supply has failed. Possible causes and trouble clearing — The NMC detected a PSU which is not responding to periodic polls. ■ Check the chassis PSU.
5	tempWarning Trap enable object: <i>uchasTempWarningTrapEna</i>	chassis via NMC	Condition — the internal hub temperature is out of the normal operating range. Possible causes and trouble clearing — The chassis over-temperature sensor detected a high operating temperature. The chassis is either too hot, the NMC card is reading the temperature incorrectly, or the over-temperature sensor failed. ■ To determine the chassis temperature, use Total Control Manager (check <i>Chassis Temperature</i> in the NMC Identification Group) or a MIB browser. The normal operating temperature must be below 41° C (104° F). If the temperature reading stays above 40° C, check the actual site temperature. If the reading is inaccurate, replace the sensor or the NMC card.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
6	fanFailure Trap enable object: <i>uchasFanFailureTrapEna</i>	chassis via NMC	<p>Condition — the hub cooling fan failed.</p> <p>Possible causes and trouble clearing — The chassis fan sensor detected a non-operational fan, the fan sensor failed, or the NMC misread the sensor.</p> <ul style="list-style-type: none"> ■ Make sure the fan is functioning. ■ Make sure the fan tray is connected. ■ Check the NMC card for a red Hub Status LED. Swap the NMC with a known good NMC, then check to see if problem continues. Replace the NMC if needed.
7	entityWatchdogTimeout Trap enable object: <i>uchasEntityWatchdogTrapEna</i>	all except NMC	<p>Condition — a watchdog timeout was detected. This may be an indication of a software failure.</p> <p>Possible causes and trouble clearing — A card in the chassis detected a software failure and rebooted itself.</p> <ul style="list-style-type: none"> ■ Use Total Control Manager or a MIB browser to check card status. Replace a failed card.
8	entityMgtBusFailure Trap enable object: <i>uchasEntityMgtBusFailTrapEna</i>	all except NMC	<p>Condition — a chassis NAC (entity) failed to respond to the NMC.</p> <p>Possible causes and trouble clearing — Typically, this trap indicates a session between the NMC and a chassis NAC has failed and re-established itself. If you continue to receive this trap, and it is associated with a specific NAC, the NAC may need to be replaced.</p>
9	incomingConnectionEstablished Trap enable object: obsolete	Dual modem Quad modem	<p>Condition — an incoming connection was established on a modem.</p> <p>Obsolete — replaced by trap 54, which provides tracking.</p>
10	outgoingConnectionEstablished Trap enable object: obsolete	Dual modem Quad modem	<p>Condition — an outgoing connection was established on a modem.</p> <p>Obsolete — replaced by trap 55, which provides tracking.</p>
11	incomingConnectionTerminated Trap enable object: obsolete	Dual modem Quad modem HiPer DSP	<p>Condition — an incoming connection was terminated on a modem.</p> <p>Obsolete — replaced by trap 56.</p>

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
12	outgoingConnectionTerminated Trap enable object: obsolete	Dual modem Quad modem HiPer DSP	Condition — an outgoing connection was terminated on a modem. Obsolete — replaced by trap 57.
13	connectAttemptFailure Trap enable object: obsolete	Dual modem Quad modem	Condition — a modem in the chassis failed to connect a call with another modem. Obsolete — replaced in NMC v5.0 with traps 86 and 87, which provide directionality.
14	connectTimerExpired Trap enable object: <i>mdmTeConnTimeLimit</i> <i>hdmTeConnTimeLimit</i> (DSP)	Dual modem Quad modem HiPer DSP <i>modem-level trap</i>	Condition — the modem's connection time limit expired. A connection on a modem has not passed any data for a specified period, so the modem disconnected the call.
15	dteTransmitDataIdle Trap enable object: <i>mdmTeDteXmitDataIdle</i> <i>hdmTeDteXmitDataIdle</i> (DSP)	Dual modem Quad modem HiPer DSP <i>modem-level trap</i>	Condition — the modem has not received any packets from the attached DTE for a period longer than its idle time threshold.
16	dtrTrue Trap enable object: <i>mdmTeDtrTrue</i>	Dual modem Quad modem	Condition — the attached DTE is asserting the DTR signal in a "true" condition.
17	dtrFalse Trap enable object: <i>mdmTeDtrFalse</i>	Dual modem Quad modem	Condition — the attached DTE is asserting the DTR signal in a "false" condition. Possible causes and trouble clearing — <ul style="list-style-type: none"> ■ Check DTE or connected terminal. ■ If using a NETServer, try resetting the corresponding S-port. ■ Try a software reset on the modem. ■ Reboot the modem NAC. ■ Swap the modem to see if the problem follows the card. Replace if needed.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
18	blerCountAtThreshold Trap enable object: <i>mdmTeBlerCountAtThresh</i> <i>hdmTeBlerCountAtThresh</i> (DSP)	Dual modem Quad modem HiPer DSP <i>modem-level trap</i>	<p>Condition — the modem has exceeded the maximum number of block errors (BLERs) configured for this call. The threshold is defined in the modem's BLER counter (within the template on a HiPer DSP).</p> <p>Possible causes and trouble clearing —</p> <ol style="list-style-type: none"> 1 Line noise 2 Line interference 3 Bad cabling <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Reset the threshold value in the modem's BLER counter. ■ Check line status. ■ Check cabling.
19	fallbackCountAtThreshold Trap enable object: <i>mdmTeFallbkCountAtThresh</i> <i>hdmTeFallbkCountAtThresh</i> (DSP)	Dual modem Quad modem HiPer DSP <i>modem-level trap</i>	<p>Condition — the modem has exceeded the maximum number of fallbacks (retrains) configured for this call. The threshold is defined in the modem's fallback counter (within the template on a HiPer DSP).</p> <p>Possible causes and trouble clearing —</p> <ol style="list-style-type: none"> 1 Line noise 2 Line interference 3 Bad cabling <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Reset the threshold value in the modem's fallback counter. ■ Check line status. ■ Check cabling.
20	noDialTone Trap enable object: <i>mdmTeNoDialTone</i>	Dual modem Quad modem	<p>Condition — the modem did not detect a dial tone on its NIC interface when it went off-hook to dial.</p> <p>Possible causes and trouble clearing —</p> <ul style="list-style-type: none"> ■ Check for dialtone on the line; switch lines if needed. ■ Replace the NIC if needed.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
21	noLoopCurrent Trap enable object: <i>mdmTeNoLoopCurrent</i>	Dual modem Quad modem	<p>Condition — the modem did not detect loop current on its NIC interface when it went off-hook to dial.</p> <p>Possible causes and trouble clearing —</p> <ul style="list-style-type: none"> ■ Check the line; switch lines if needed. ■ Replace the NIC if needed.
22	yellowAlarm Trap enable object: <i>uds1TrapEnaYellowAlarm</i> (T1) <i>usrds1EventYellowAlarm</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	<p>Condition — a yellow alarm condition exists when the remote end of a DS1 is experiencing an “out of frame” (OOF) condition.</p> <p>This trap is used with yellowAlarmClear (50).</p> <p>Possible causes and trouble clearing —</p> <ol style="list-style-type: none"> 1 The T1-E1/PRI card does not detect a valid received signal, which may indicate the telco switch is malfunctioning or there is a line problem: <ul style="list-style-type: none"> ■ Check the T1-E1/PRI span configuration. This includes the framing type (D4/ESF) for T1, or E1 frame with CRC-4 or plain straight E1 framing (non-CRC-4) for E1. ■ Check line coding. ■ Replace the NIC if needed. 2 There may be a line problem: <ul style="list-style-type: none"> ■ Check the cables. Do not use flat cable. Use standard twisted pair T1 cable that meets telco specifications. ■ Check impedance matching. ■ Check for ground loops. ■ Check signal strength and pulse mask. ■ Check digital clock rate. ■ Swap out cables if needed. 3 The telco may have put the line out of service: <ul style="list-style-type: none"> ■ Make sure the DS1 interfaces are LIU terminated and kept-alive. ■ Check the telco configuration.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
23	redAlarm Trap enable object: <i>uds1TrapEnaRedAlarm</i> (T1) <i>usrds1EventRedAlarm</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	<p>Condition — Red alarm; the framing pattern was lost on the DS1's receiver. This is also known as an "out of frame" (OOF) condition. This trap is used with redAlarmClear (51).</p> <p>Possible causes and trouble clearing —</p> <ol style="list-style-type: none"> 1 The T1-E1/PRI has lost carrier detect. Cannot extract aligned digital data from the remote end. A yellow pattern will be sent to the remote end as a warning. <ul style="list-style-type: none"> ■ Check the T1-E1/PRI span configuration. This includes the framing type (D4/ESF) for T1, or E1 frame with CRC-4 or plain straight E1 framing (non-CRC-4) for E1. ■ Check the line quality. ■ Check line coding. 2 There may be a line problem: <ul style="list-style-type: none"> ■ Check the cables. Do not use flat cable. Use standard twisted pair T1 cable that meets telco specifications. ■ Check impedance matching. ■ Check for ground loops. ■ Check signal strength and pulse mask. ■ Check digital clock rate. ■ Swap out cables if needed. 3 The telco may have put the line out of service: <ul style="list-style-type: none"> ■ Make sure the DS1 interfaces are LIU terminated and kept-alive. <p>Check the telco configuration.</p>

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
24	lossOfSignal Trap enable object: <i>uds1TrapEnaLossOfSignal</i> (T1) <i>usrds1EventLossOfSignal</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	<p>Condition — Loss of signal; the DS1 receiver has received 175 consecutive 0s. The DS1 is unable to recover a receive signal.</p> <p>This trap is used with <i>lossOfSignalClear</i> (52).</p> <p>Possible causes and trouble clearing — The T1-E1/PRI has lost electrical communication or synchronization with the telco switch. This may occur after inserting a NIC.</p> <ul style="list-style-type: none"> ■ Ensure the NIC is seated and configured correctly. ■ Check cables and breakout boxes. ■ Check line status with the telco. ■ Ensure the modular cables are installed correctly. ■ Check receiver gain. ■ Swap cables if needed. ■ Replace the NIC if needed.
25	alarmIndicationSignal Trap enable object: <i>uds1TrapEnaAlarmIndSignal</i> (T1) <i>usrds1EventAlarmIndSignal</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	<p>Condition — Alarm indication signal (blue alarm); the DS1 is receiving an all 1s pattern. This is an indication that the remote end has lost its receive signal.</p> <p>This trap is used with <i>alarmIndicationSignalClear</i> (53).</p> <p>Possible causes and trouble clearing —</p> <ol style="list-style-type: none"> 1 The remote end lost its receive signal: <ul style="list-style-type: none"> ■ Check the telco to verify that the remote end has not lost its receive signal. ■ Check the telco to verify that the line was not placed out of service. 2 The T1-E1/PRI card received an indication that the Telco switch is receiving alarms from a higher-order device. <ul style="list-style-type: none"> ■ Check if a T2 or higher transmission facility is in an alarm state.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
26	transmitTimingSourceSwitch Trap enable object: <i>dt1TrapEnaTxTmgSrcSwitch</i>	T1-E1/PRI	<p>Condition — The specified dual T1 card switched to an alternate timing source.</p> <p>Possible causes and trouble clearing —</p> <ol style="list-style-type: none"> 1 The primary timing source is slaved to the first T1-E1/PRI: <ul style="list-style-type: none"> ■ Check the out-of-band management for the T1-E1/PRI card. <ol style="list-style-type: none"> 2 T1 alarm condition: <ul style="list-style-type: none"> ■ Look for alarm conditions on either T1. Contact the Telco if needed. ■ If you do not find alarms, set the primary timing source back to span 1. If span 1 will not provide timing, have the Telco check the timing pair on the T1. If you still cannot get span 1 to provide timing, you may need to replace the card.
27	modemResetByDte Trap enable object: <i>mdmTeResetByDTE</i> <i>t1hTeResetByDTE</i> (HiPer DSP)	Dual modem Quad modem HiPer DSP	<p>Condition — Modem reset by DTE; the modem received a reset command from the DTE.</p> <p>HiPer DSP – Modem reset by the console.</p>
28	modemRingNoAnswer Trap enable object: <i>anicCfgMdmRingNATrapEna</i>	Dual modem Quad modem	<p>Condition — Modem ring no answer; the modem failed to answer a call. This trap is generated if the DTR is present and the S0 register on the NAC is not equal to zero.</p> <p>This trap applies to analog only.</p> <p>Possible causes and trouble clearing — Check the S0 register value.</p>

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
29	dteRingNoAnswer Trap enable object: <i>mdmTeDteRingNoAnswer</i> <i>anicCfgDteRingNATrapEna</i> (analog)	Dual modem Quad modem HiPer DSP <i>modem-level trap</i>	<p>Condition —</p> <p>Dual and Quad modems – The DTE is not responding to the modem, even though the modem answered the call. This trap is generated when the DTR is not present and the S0 register on the NAC is not equal to zero. Typically, the port on the gateway card is inactive.</p> <p>This trap is sent when the modem receives notice of a call but fails to answer because was not directed by the DTE to answer the call. This is considered a “lost call”. When the modem is first notified of a call, it sets a 60-second timer. If the modem does not train within 60 seconds, it will send the trap. If another call arrives on that modem, the trap is sent for the first call. On heavily used hubs, an occasional “lost call” may occur and send this trap. If the threshold or pattern of <i>dteRingNoAnswer</i> traps is high, this may signal a problem with the modem.</p> <p>HiPer DSP – A packet bus session could not be established with a gateway card, the call clears before it is answered by a gateway card, or the gateway card times out before answering the call.</p> <p>Possible causes and trouble clearing —</p> <ol style="list-style-type: none"> 1 Check to see if the port on the gateway card is active. 2 The hub may be in a state of heavy use; check the threshold of the <i>modemRingNoAnswer</i> trap.
30	pktBusSessActive Trap enable object: <i>pbTrapEnaSessActive</i> (gateway) <i>hdmTePbActive</i> (DSP)	HiPer DSP NETServer X.25 EdgeServer	<p>Condition — Packet bus session active; the DTE/gateway NAC established a session with the modem.</p> <p>Possible causes and trouble clearing — This trap is normal after a card is reset or after a power cycle.</p> <p>Informational only.</p>
31	pktBusSessCongestion Trap enable object: <i>pbTrapEnaPktBusCongest</i>	NETServer X.25 EdgeServer	<p>Condition — Packet bus session congestion; a session between the modem and the DTE/gateway NAC encountered congestion on the packet bus.</p>

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
32	pktBusSessLost Trap enable object: <i>pbTrapEnaPktBusSessLost</i> (gateway) <i>hdmTePbLost</i> (DSP)	HiPer DSP X.25 EdgeServer	Condition — Packet bus session lost; a session between the modem and the DTE/gateway NAC was disconnected.
33	pktBusSessInactive Trap enable object: <i>pbTrapEnaSessionInactive</i>	NETServer X.25 EdgeServer	Condition — Packet bus session inactive; a session between the modem and the DTE/gateway NAC has become inactive. Possible causes and trouble clearing — The NETServer automatically recovers from this event and re-opens the packet bus connection.
34	nacUserInterfaceReset Trap enable object: <i>ipgwTrapEnUiReset</i> <i>x25gwTrapEnaUiReset</i> (X.25)	HiPer ARC T1-E1/PRI X.25	Condition — NAC user interface reset; the NAC was reset from the user interface. Possible causes and trouble clearing — This trap is generated by non-modem NACs to indicate that an out-of-band connection reset the NAC. Informational only.
35	gwWanPortOutOfService Trap enable object: <i>x25wanTrapEnaOutOfSvc</i>	X.25	Condition — GW WAN port out of service; a gateway WAN port has changed from Link Active to Out of Service.
36	gwWanPortLinkActive Trap enable object: <i>x25wanTrapEnaLinkActive</i>	X.25	Condition — GW WAN port link active; a gateway WAN port has changed from Out of Service to Link Active.
37	dialOutLoginFail Trap enable object: <i>nmcTeDialOutLogFail</i>	NMC Dual modem Quad modem	Condition — Dial out login failure; a dialout login security session failed and the call was not placed. This informational trap is for NMC-based security for Dual and Quad modems.
38	dialInLoginFail Trap enable object: <i>nmcTeDialInLoginFail</i>	NMC Dual modem Quad modem	Condition — Dial in login failure; a dialin login security session failed and the call was not placed. This informational trap is for NMC-based security for Dual and Quad modems.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
39	dialOutRestrictedNum Trap enable object: <i>nmcTeDialOutRestrictNum</i>	NMC Dual modem Quad modem	<p>Condition — Dial out restricted phone number; a dialout security session failed as a result of attempting to dial a restricted phone number. The call was not placed.</p> <p>This informational trap is for NMC-based security for Dual and Quad modems.</p> <p>Possible causes and trouble clearing — Check the number and retry.</p>
40	dialBackRestrictedNum Trap enable object: <i>nmcTeDialBackRestrictNum</i>	NMC Dual modem Quad modem	<p>Condition — Dial back restricted number; a dialback security session failed as a result of attempting to dial a restricted number.</p> <p>This informational trap is for NMC-based security for Dual and Quad modems.</p> <p>Possible causes and trouble clearing — Check the number and retry.</p>
41	userBlacklisted Trap enable object: <i>nmcTeUserBlacklist</i>	NMC Dual modem Quad modem	<p>Condition — User blacklisted; a security user reached their final failed login attempt number and is now blacklisted.</p> <p>This trap is for NMC-based security for Dual and Quad modems.</p> <p>Possible causes and trouble clearing — Check the password and reset the user if required.</p>
42	loginAttemptByBlacklistedUser Trap enable object: <i>nmcTeUserBlacklistLogin</i>	NMC Dual modem Quad modem	<p>Condition — Attempted login blacklisted; a currently blacklisted security user attempted to login.</p> <p>This trap is for NMC-based security for Dual and Quad modems.</p> <p>Possible causes and trouble clearing — This may be considered a warning condition; you may wish to monitor for a potential security problem.</p>
43	responseAttemptLimExceeded Trap enable object: <i>nmcTeRespAttemptLimExceeded</i>	NMC Dual modem Quad modem	<p>Condition — Response attempt limit exceeded; a security user failed to issue a valid response to a particular security prompt before the configured limit.</p> <p>This trap is for NMC-based security for Dual and Quad modems.</p>

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
44	mdmLoginAttemptLimExceeded Trap enable object: <i>nmcTeLoginAttemptLimExceed</i>	NMC Dual modem Quad modem	Condition — Login attempt limit exceeded; a user does not appear in the security user database. This trap is for NMC-based security for Dual and Quad modems. Possible causes and trouble clearing — Add the user to the security user database.
45	dialOutCallDuration Trap enable object: <i>mdmTeDialOutCallDur</i>	Dual modem Quad modem	Condition — a dial-out call ended. This trap is sent to inform the accounting server of the duration of a dial-out call.
46	dialInCallDuration Trap enable object: <i>mdmTeDialInCallDur</i>	Dual modem Quad modem	Condition — a dial-in call ended. This trap is sent to inform the accounting server of the duration of a dial-in call.
47	pktBusSessError Trap enable object: <i>pbTrapEnaSessionError</i>	X.25	Condition — a session between a modem and the DTE/gateway NAC is giving errors. Possible causes and trouble clearing — A modem card or the gateway card may need to be replaced.
48	nmcArCustomTrap Trap enable object: Enable through AutoResponse	NMC	Condition — an NMC SNMP autoreponse trap was sent. Possible causes and trouble clearing — An event in the chassis triggered an autoreponse event. This is a custom trap. The meaning of this trap depends upon how your system administrator has defined it.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
49	acctSvrLoss Trap enable object: <i>nmcTelogSvrLoss</i>	NMC	<p>Condition — the NMC lost its connection to the accounting server. Check the server security failure code returned with the trap for additional information.</p> <p>Possible causes and trouble clearing — The NMC has eight RADIUS accounting servers, including a primary server and a set of secondary servers. The NMC polls the RADIUS server to update server status. When this trap occurs, there was no response from the RADIUS server after several polls. This may be caused because:</p> <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The server is too busy to respond. 4 The server is down. <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Check the network connections. ■ Ensure the server is active. ■ Check the server configuration. ■ Restart the server. ■ Check the communication link.
50	yellowAlarmClear Trap enable object: <i>uds1TrapEnaYellowAlarmClr</i> (T1) <i>usrds1EventYellowAlarmClr</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	<p>Condition — the T1-E1/PRI card received a yellow alarm cleared indication from the Telco switch. This informs you that the previous yellowAlarm trap is cleared.</p> <p>Possible causes and trouble clearing — Informational only.</p>
51	redAlarmClear Trap enable object: <i>uds1TrapEnaRedAlarmClr</i> (T1) <i>usrds1EventRedAlarmClr</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	<p>Condition — the T1-E1/PRI card received a red alarm cleared indication from the Telco switch. This informs you that the previous redAlarm trap is cleared.</p> <p>Possible causes and trouble clearing — Informational only.</p>
52	lossOfSignalClear Trap enable object: <i>uds1TrapEnaLossOfSgnlClr</i> (T1) <i>usrds1EventLossOfSgnlClr</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	<p>Condition — the T1-E1/PRI card received a loss of signal alarm cleared indication from the Telco switch. This informs you that the previous lossOfSignal trap is cleared.</p> <p>Possible causes and trouble clearing — Informational only.</p>

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
53	alarmIndicationSignalClear Trap enable object: <i>uds1TrapEnaAlrmIndSgnlClr</i> (T1) <i>usrds1EventAlrmIndSgnlClr</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	Condition — the T1-E1/PRI card received a blue alarm cleared indication m the Telco switch. This informs you that the previous alarmIndicationSignal trap is cleared. Possible causes and trouble clearing — Informational only.
54	ctIncomingConnectionEstablished Trap enable object: <i>mdmTelnConnEstablished</i> <i>hdmTelnConnEstablished</i> (DSP)	Quad modem HiPer DSP <i>modem-level trap</i>	Condition — a modem successfully established an incoming call and sent a call reference number. This trap replaced incomingConnectionEstablished (9).
55	ctOutgoingConnectionEstablished Trap enable object: <i>mdmTeOutConnEstablished</i> <i>HdmTeOutConnEstablished</i> (DSP)	Quad modem HiPer DSP <i>modem-level trap</i>	Condition — a modem successfully established an outgoing call. This trap replaced outgoingConnectionEstablished (10).
56	ctIncomingConnectionTerminated Trap enable object: <i>mdmTelnConnTerminated</i> <i>hdmTelnConnTerminated</i> (DSP)	Quad modem HiPer DSP <i>modem-level trap</i>	Condition — a modem successfully ended an incoming call. This trap replaced incomingConnectionTerminated (11).
57	ctOutgoingConnectionTerminated Trap enable object: <i>mdmTeOutConnTerminated</i> <i>hdmTeOutConnTerminated</i> (DSP)	Quad modem HiPer DSP <i>modem-level trap</i>	Condition — a modem successfully ended an outgoing call. This trap replaced outgoingConnectionTerminated (12).
58	ctConnectAttemptFailure Trap enable object: obsolete	Quad modem	Condition — a modem was unable to successfully establish a call. Obsolete — replaced in NMC v5.0 with traps 86 and 87, which provide directionality.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
59	contCrcAlarm Trap enable object: <i>uds1TrapEnaContCrcAlrm</i> (T1) <i>usrds1EventContCrcAlrm</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	Condition — a continuous CRC error condition occurred on the DS1. Possible causes and trouble clearing — The T1-E1/PRI is receiving a continuous stream of CRC errors from the T1 span: <ul style="list-style-type: none"> ■ Verify layer 1 configuration, particularly framing mode. ■ Check cables. ■ Make sure the modular jack is inserted properly. ■ Make sure the NIC is installed correctly. ■ Check cable quality. Do not use flat cable. Use only twisted-pair cable. ■ Check impedance matching. ■ Check for ground loops. ■ Check signal strength and pulse mask. ■ Check the digital clock rate and verify with the telco.
60	contCrcAlarmClear Trap enable object: <i>uds1TrapEnaContCrcAlrmClr</i> (T1) <i>usrds1EventContCrcAlrmClr</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	Condition — a continuous CRC error condition cleared from the DS1. Possible causes and trouble clearing — Informational only.
61	phyStateChng Trap enable object: <i>uds1TrapEnaPhysStateChng</i> (T1) <i>usrds1EventPhysStateChng</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level span monitor trap</i>	Condition — a physical state change occurred on the DS1. Possible causes and trouble clearing — A T1-E1/PRI connected to the chassis changed physical state: <ul style="list-style-type: none"> ■ Check out-of-band for both spans to determine if either span is in an alarm condition. ■ Contact the Telco if needed.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
62	securityServerLoss Trap enable object: <i>nmcTeSecSvrLoss</i>	NMC	<p>Condition — the NMC is unable to contact the security server.</p> <p>Possible causes and trouble clearing — The NMC has eight security servers. There is a primary server and a set of secondary servers. The NMC polls the server to update server status. When this trap occurs, there was no response from the server after several polls. This may be caused because:</p> <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The server is too busy to respond. 4 The server is down. <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Check the network connections. ■ Ensure the server is active. ■ Check the server configuration. ■ Restart the server.
63	singlePktBusClockFailure Trap enable object: <i>nmcTeSinglePbClockFailure</i>	NMC	<p>Condition — a backplane packet bus clock failed in a single NAC slot.</p> <p>Possible causes and trouble clearing —</p> <ul style="list-style-type: none"> ■ Check the NMC card for a red Hub Status LED. ■ Swap the NMC with a known good NMC, then check to see if problem continues.
64	pktBusClockSwitch Trap enable object: <i>pktBusClockSwitch</i>	NMC	<p>Condition — a backplane packet bus clock failed in multiple NAC slots. The NMC assumed the role of clock master.</p> <p>Possible causes and trouble clearing —</p> <ul style="list-style-type: none"> ■ Check the NMC card for a red Hub Status LED. ■ Swap the NMC with a known good NMC, then check to see if problem continues.
65	pktBusClockFailure Trap enable object: <i>pktBusClockFailure</i>	NMC	<p>Condition — the packet bus clock on the NMC daughter board failed.</p> <p>Possible causes and trouble clearing —</p> <ul style="list-style-type: none"> ■ Ensure the daughter board is in good condition (386- and 486-based NMCs only). Replace the NAC if it is not.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
66	gwNetworkFailed Trap enable object: <i>gwTeArNetFailed</i> (gateway) <i>gwTegwNetworkFailed</i> (NAC)	NETServer HiPer ARC	<p>Condition — the gateway network failed because authentication and accounting failed.</p> <p>Possible causes and trouble clearing — When this trap occurs, there was no response from the RADIUS server after several polls. This may be caused because:</p> <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The RADIUS server is too busy to respond. 4 The RADIUS server is down. 5 The DTE/gateway NAC lost communication with all RADIUS servers: <ul style="list-style-type: none"> ■ Check the RADIUS server. ■ Check the cable from the DTE to the server. ■ Check the NETServer or HiPer ARC card.
67	gwNetworkRestored Trap enable object: <i>gwTeArNetRestored</i> (gateway) <i>gwTegwNetworkRestored</i> (NAC)	NETServer HiPer ARC	<p>Condition — the gateway network is restored and contact with the RADIUS server is re-established.</p> <p>Possible causes and trouble clearing — This trap indicates gwNetworkFailed (66) cleared.</p> <p>Informational only.</p>

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
68	pktBusClkLost Trap enable object: <i>pbdgCfgClockLossEvent</i>	all except NMC	<p>Condition — the NMC determined that the packet bus clocking source failed, either for a single NAC or for the entire chassis.</p> <p>Possible causes and trouble clearing —</p> <ol style="list-style-type: none"> 1 The T1-E1/PRI determined that the packet bus clock is no longer present: <ul style="list-style-type: none"> ■ Verify chassis type. If clocked backplane type, make sure the NMC status indicates a single clock failure. If not, replace the NAC. ■ Identify which slot(s) failed. Swap the NAC with a known good NAC. This condition may clear itself. Do not replace the NAC in question until this trap occurs multiple times. 2 The problem may be with the NMC or the backplane: <ul style="list-style-type: none"> ■ Check the clock source. ■ If the problem is related to slot 17, the NMC is providing timing for the backplane. Contact 3Com tech support.
69	pktBusClkRestore Trap enable object: <i>pbdgCfgClockRestoreEvent</i>	all except NMC	<p>Condition — the condition causing the Packet bus clock lost trap is cleared.</p> <p>Possible causes and trouble clearing — This trap indicates pktBusClkLost(68) cleared. Informational only.</p>
70	dChanInService Trap enable object: <i>uds1TrapEnaDchanInSrv (T1-E1/PRI)</i> <i>usrds1EventDchanInSrv (DSP)</i>	T1-E1/PRI HiPer DSP <i>span-level call control trap</i>	<p>Condition — the condition causing a D-Channel out of service trap is cleared.</p> <p>Possible causes and trouble clearing — This trap indicates dChanOutOfService(71) cleared. Informational only.</p>
71	dChanOutOfService Trap enable object: <i>uds1TrapEnaDchanOutOfSrv (T1-E1/PRI)</i> <i>usrds1DchanOutOfSrv (DSP)</i>	T1-E1/PRI HiPer DSP <i>span-level call control trap</i>	<p>Condition — the D-Channel is out of service.</p> <p>Possible causes and trouble clearing — The T1-E1/PRI card has lost D-Channel signalling on one or both spans.</p> <ul style="list-style-type: none"> ■ Verify the setting of <i>udsiStatDChannel</i>. ■ Call the Telco if needed.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
72	ds0InService Trap enable object: <i>uds1TrapEnaDs0InSrv</i> (T1-E1/PRI) <i>usrds1EventDs0InSrv</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level call control trap</i>	Condition — the condition causing a DS0 out of service trap is cleared. Possible causes and trouble clearing — This trap indicates ds0sOutOfService(73) cleared. This is the only time this trap generates. This trap may be sent when the chassis reboots after a power cycle. This trap will only occur following ds0sOutOfService(73).
73	ds0sOutOfService Trap enable object: <i>uds1TrapEnaDs0OutOfSrv</i> (T1-E1/PRI) <i>usrds1EventDs0OutOfSrv</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level call control trap</i>	Condition — this trap provides a list of the DS0s going out of service. Possible causes and trouble clearing — The NAC received an indication that one or more DS0s are in an out of service condition. <ul style="list-style-type: none">■ Use out-of-band checking or <i>ids0StatDs0</i> to see if any DS0s are busied-out. Unbusy them if they are out of service.■ Call the Telco if they are out of service or MaintBusy.
74	callEvent Trap enable object: obsolete	T1 (obsolete)	Condition — T1, T1-E1/PRI call event. Obsolete — replaced by traps 76-7
75	psuIncompatible Trap enable object: <i>uchasPsuIncompatible</i>	NMC	Condition — the power supply is incompatible with the chassis. Possible causes and trouble clearing — Use a compatible power supply. Power supply outputs must match each other (for example, you cannot mix a 45 amp and a 70 amp PSU in the same chassis).
76	callArriveEvent Trap enable object: <i>dt1TrapEnaCallArriveEvent</i> (T1-E1/PRI)	T1-E1/PRI	Condition — T1, T1-E1/PRI call arrive event. This trap is the first indication of a new call on the T1 span. Informational only.
77	callConnectEvent Trap enable object: obsolete	T1-E1/PRI	Condition — T1, T1-E1/PRI call connect event. Possible causes and trouble clearing — This trap is sent when the call is connected to a modem. Informational only.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
78	callTermNormalEvent Trap enable object: dt1TrapEnaCallTermEvent (T1-E1/PRI))	T1-E1/PRI T1-E1/PR	Condition — T1, T1-E1/PRI normal call termination event. Possible causes and trouble clearing — This trap is sent when the call terminates normally at the DS0 level. Informational only.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
79	callTermFailedEvent Trap enable object: obsolete	T1-E1/PRI	<p>Obsolete — replaced in NMC v5.0 by traps 114 and 115, which provide directionality.</p> <p>Condition — T1, T1-E1/PRI call termination (fail) event.</p> <p>Possible causes and trouble clearing — This trap is sent when a call arrives in the chassis but fails to be delivered to a modem.</p> <ol style="list-style-type: none"> 1 A call failed to connect: <ul style="list-style-type: none"> ■ Check the cause code sent with the event to begin trouble clearing. 2 The call arrived when the T1-E1/PRI was configured to block all calls: <ul style="list-style-type: none"> ■ Check that DS0 and DS1 configuration is not set to block all calls of the desired type. 3 Calling party number is not valid: <ul style="list-style-type: none"> ■ Check the configuration of acceptable calling party numbers. 4 Called party number is not valid: <ul style="list-style-type: none"> ■ Check the configuration of acceptable called party numbers. 5 Bearer capability is not valid: <ul style="list-style-type: none"> ■ Check configuration. 6 No modem is available: <ul style="list-style-type: none"> ■ Check for a modem that may be in a failed operational state. ■ Also look for a modem that does not seem to be taking calls. 7 Modem or gateway is not responding to a set-up message: <ul style="list-style-type: none"> ■ Look for failed or out-of-service modems or gateway. 8 Call was directed to an out-of-service DS0: <ul style="list-style-type: none"> ■ Look for out-of-service DS0s and return them to service. 9 No free B-channel is available for outgoing calls: <ul style="list-style-type: none"> ■ Look for out-of-service DS0s and return them to service.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
80	dnsSrvrLoss Trap enable object: <i>nmcTeDnsSrvrLoss</i>	NMC	<p>Condition — unable to contact the primary DNS server.</p> <p>Possible causes and trouble clearing — When this trap occurs, there was no response from the server after several polls. This may be caused because:</p> <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The DNS server is too busy to respond. 4 The DNS server is down. <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Check the network connections. ■ Ensure the DNS server is active. ■ Check the DNS server configuration. ■ Restart the DNS server. ■ Check the communication link.
81	ntpSrvrLossConn Trap enable object: <i>nmcTeNtpSrvrLoss</i>	NMC	<p>Condition — unable to contact the primary NTP server.</p> <p>Possible causes and trouble clearing — When this trap occurs, there was no response from the server after several polls. This may be caused because:</p> <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The NTP server is too busy to respond. 4 The NTP server is down. <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Check the network connections. ■ Ensure the NTP server is active. ■ Check the NTP server configuration. ■ Restart the NTP server.
82	ntpSrvrRestConn Trap enable object: <i>nmcTeNtpSrvrRestore</i>	NMC	<p>Condition — the connection to the NTP server is restored.</p> <p>Possible causes and trouble clearing — Informational only.</p>

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
83	ipgwLinkUp Trap enable object: <i>linkTrapUp</i> (MIB-2)	HiPer ARC	Condition — the IP gateway link is up. Possible causes and trouble clearing — Informational only.
84	ipgwlinkDown Trap enable object: <i>linkTrapDown</i> (MIB-2)	HiPer ARC	Condition — the IP gateway link is down. Possible causes and trouble clearing — When this trap occurs, there was no response from the server after several polls. This may be caused because: <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The server is too busy to respond. 4 The server is down. Take these trouble clearing actions: <ul style="list-style-type: none"> ■ Check the network connections. ■ Ensure the server is active. ■ Check the server configuration. ■ Restart the server.
85	ntpSrvrDegrConn Trap enable object: <i>nmcTeNtpSrvrDegraded</i>	NMC	Condition — unable to contact the primary NTP server.
86	inconnectAttemptFailure Trap enable object: <i>mdmTeInConnAttemptFail</i>	HiPer DSP <i>modem-level trap</i>	Condition — the inbound call failed to connect with a modem. Possible causes and trouble clearing — <ol style="list-style-type: none"> 1 The DS0 may be busy or out of service. 2 The modem training sequence may have failed. Refer to the modem disconnect and fail to connect reasons.
87	outconnectAttemptFailure Trap enable object: <i>mdmTeOutConnAttemptFail</i>	HiPer DSP <i>modem-level trap</i>	Condition — the outbound call failed to connect with a modem. Possible causes and trouble clearing — The DS0 may be busy or out of service. <ul style="list-style-type: none"> ■ Refer to the modem disconnect and fail to connect reasons.
88	applicationProcessorReset Trap enable object: Not used.	HiPer DSP	Not used.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
89	dspReset Trap enable object: Not used.	HiPer DSP	Not used.
90	changedtoMainSrvsStat Trap enable object: <i>usrds1EventDs0ServStateMt</i>	HiPer DSP <i>span-level call control trap</i>	Condition — the telco placed the span into a maintenance server state (out of service). Possible causes and trouble clearing — This reason will only occur with some telco switches.
91	loopbackcleared Trap enable object: <i>usrds1EventloopBackCleared</i>	HiPer DSP <i>span-level span monitor trap</i>	Condition — the loopback condition on the span line cleared. Possible causes and trouble clearing — Informational only.
92	loopbacktrap Trap enable object: <i>usrds1EventloopBack</i>	HiPer DSP <i>span-level span monitor trap</i>	Condition — a loopback has occurred on the span line. Possible causes and trouble clearing — Informational only.
93	telcoAbnormalResp Trap enable object: <i>uds1EvtelcoAbnormalResp</i>	HiPer DSP <i>span-level call control trap</i>	Not used.
94	dnsSrvrRestore Trap enable object: <i>nmcTeDnsSrvrRestore</i>	NMC	Condition — contact is restored with the primary DNS server. Possible causes and trouble clearing — Informational only.
95	dnsSrvrDegraded Trap enable object: <i>nmcTeDnsSrvrDegraded</i>	NMC	Condition — contact was lost with the primary DNS server, but the secondary DNS server is in service. Possible causes and trouble clearing — Informational only.
96	acctSrvrRestore Trap enable object: <i>nmcTeLogSrvrRestore</i>	NMC	Condition — contact is restored with the RADIUS accounting server. Possible causes and trouble clearing — Informational only.
97	acctSrvrGroupOper Trap enable object: <i>nmcTeLogSrvrGroupOper</i>	NMC	Condition — the RADIUS accounting server group is operational. Contact with the primary RADIUS accounting server is restored. Possible causes and trouble clearing — Informational only.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
98	acctSvrGroupDegr Trap enable object: <i>nmcTeLogSvrGroupDegr</i>	NMC	<p>Condition — the RADIUS accounting server group is degraded. Contact with the primary server has been lost.</p> <p>Possible causes and trouble clearing — When this trap occurs, there was no response from the server after several polls. This may be caused because:</p> <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The server is too busy to respond. 4 The server is down. <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Check the network connections. ■ Ensure the server is active. ■ Check the server configuration. ■ Restart the server.
99	acctSvrGroupNonOp Trap enable object: <i>nmcTeLogSvrGroupNonOp</i>	NMC	<p>Condition — the RADIUS accounting server group is non-operational. Contact is lost to all servers.</p> <p>Possible causes and trouble clearing — When this trap occurs, there was no response from the server after several polls. This may be caused because:</p> <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The server is too busy to respond. 4 The server is down. <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Check the network connections. ■ Ensure the server is active. ■ Check the server configuration. ■ Restart the server.
100	securityServerRestore Trap enable object: <i>nmcTeSecSvrRestore</i>	NMC	<p>Condition — contact is restored with the security server.</p> <p>Possible causes and trouble clearing — Informational only.</p>

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
101	securitySvrGrpOper Trap enable object: <i>nmcTeSecSvrGroupOper</i>	NMC	<p>Condition — the RADIUS security server group is operational. The primary server is restored.</p> <p>Possible causes and trouble clearing — When this trap occurs, there was no response from the server after several polls. This may be caused because:</p> <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The server is too busy to respond. 4 The server is down. <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Check the network connections. ■ Ensure the server is active. ■ Check the server configuration. ■ Restart the server.
102	securitySvrGrpDegr Trap enable object: <i>nmcTeSecSvrGroupDegr</i>	NMC	<p>Condition — the RADIUS security server group is degraded. The primary server is lost, the back-up server is in service.</p> <p>Possible causes and trouble clearing — When this trap occurs, there was no response from the server after several polls. This may be caused because:</p> <ol style="list-style-type: none"> 1 The communication link is down. 2 The network is congested. 3 The server is too busy to respond. 4 The server is down. <p>Take these trouble clearing actions:</p> <ul style="list-style-type: none"> ■ Check the network connections. ■ Ensure the server is active. ■ Check the server configuration. ■ Restart the server. ■ Make sure security server group is alive.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
103	securitySvrGrpNonOp Trap enable object: <i>nmcTeSecSvrGrpNonOp</i>	NMC	Condition — the RADIUS security server group is non-operational. Possible causes and trouble clearing — Contact is lost to all servers: <ul style="list-style-type: none">■ Check communication link.■ Check network.
104	uds1MultiFrame Trap enable object: <i>uds1TrapEnaMultiFrame</i> (T1-E1/PRI) <i>hdr2TeMultiFrame</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level call control trap</i>	Condition — multiframe misalignment occurred on a specified DS1. Possible causes and trouble clearing — Re-initialize the switch span to attempt to clear the condition. This is an R2 trap.
105	uds1RemoteMultiFrame Trap enable object: <i>uds1TrapEnaRemMultiFrame</i> (T1-E1/PRI) <i>hdr2TeRemMultiFrame</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level call control trap</i>	Condition — the remote terminal detected a multiframe misalignment on the specified DS1. Possible causes and trouble clearing — Re-initialize the E1/CAS span. This is an R2 trap.
106	uds1MultiFrameClr Trap enable object: <i>uds1TrapEnaMultiFrmClr</i> (T1-E1/PRI) <i>hdr2TeMultiFrameClr</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level call control trap</i>	Condition — a multiframe misalignment condition cleared on the specified DS1. This is an R2 trap. Possible causes and trouble clearing — Informational only.
107	uds1RemoteMultiFrameClr Trap enable object: <i>uds1TrapEnaRemMultiFrmClr</i> (T1-E1/PRI) <i>hdr2TeRemMultiFrameClr</i> (DSP)	T1-E1/PRI HiPer DSP <i>span-level call control trap</i>	Condition — the remote terminal detected the multiframe misalignment on the specified DS1 cleared. This is an R2 trap. Possible causes and trouble clearing — Informational only.
108	gwyTdmClockUp Trap enable object: <i>gwyTdmClockUp</i>	HiPer ARC EdgeServer	Condition — the gateway card detected the TDM clock is present. Possible causes and trouble clearing — Informational only.
109	gwyTdmClockDown Trap enable object: <i>gwyTdmClockDown</i>	HiPer ARC EdgeServer	Condition — the gateway card detected the TDM clock is lost. Possible causes and trouble clearing — Informational only.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
110	gwyTdmClockError Trap enable object: <i>gwyTdmClockError</i>	HiPer ARC EdgeServer	Condition — the gateway card detected an error in the TDM clock. Possible causes and trouble clearing — Informational only.
111	cdmaFrSrvViolation	EdgeServer	This trap is not used in the TCS 3.5 system release.
112	cdmaAtViolation	EdgeServer	This trap is not used in the TCS 3.5 system release.
113	cdma42bisViolation	EdgeServer	This trap is not used in the TCS 3.5 system release.
114	usrDs1InCallFailedEvent Trap enable object: <i>usrds1EventDs0InConnFail</i>	HiPer DSP <i>span-level call control trap</i>	Condition — an incoming T1-E1/PRI call failed.
115	usrDs1OutCallFailedEvent Trap enable object: <i>usrds1EventDs0OutConnFail</i>	HiPer DSP <i>span-level call control trap</i>	Condition — an outgoing T1-E1/PRI call failed.
116	rmmieRetrainEvent Trap enable object: <i>rmdmTeRetrainEv</i>	Quad modem	Condition — a modem detected a retrain occurring with the remote modem.
117	rmmieSpeedShiftEvent Trap enable object: <i>rmdmTeSpeedShiftEv</i>	Quad modem	Condition — a modem detected a speed shift occurring with the remote modem.
118	callArriveEventHdsp Trap enable object: <i>usrds1EventCallArrive</i>	HiPer DSP <i>span-level call control trap</i>	Condition — this is a normal event. This trap is the first indication of a new incoming call on the HiPer DSP span. Possible causes and trouble clearing — This trap is used with callTerminateEventHdsp(119). Together, these traps are useful for matching up span level failures and call accounting. The traps are still generated when the call enters the span, even if a modem did not answer the call.
119	callTerminateEventHdsp Trap enable object: <i>usrds1EventCallTerm</i>	HiPer DSP <i>span-level call control trap</i>	Condition — this is a normal event. This trap occurs when the call terminates normally on the HiPer DSP span. This trap is used with callArriveEventHdsp(118).

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
120	usrDs1HdspInCallFailedEvent Trap enable object: <i>usrds1EventDs0InConnFail</i>	HiPer DSP <i>span-level call control trap</i>	Condition — an incoming call failed because a modem could not be associated with the call. Possible causes and trouble clearing — Make sure the call was not made to an inactive port, a busy DS0, or an out of service DS0.
121	usrDs1HdspOutCallFailedEvent Trap enable object: <i>usrds1EventDs0OutConnFail</i>	HiPer DSP <i>span-level call control trap</i>	Condition — an outgoing call failed. Possible causes and trouble clearing — Make sure the call was not made from an inactive port, a busy DS0, or an out of service DS0.
122	rds1EvDchSwitchOverStart Trap enable object: <i>usrds1EventNfasDchSwStart</i>	HiPer DSP <i>span-level call control trap</i>	Condition — the primary NFAS D-channel was taken out of service and control started to switch over to the back-up NFAS D-channel. Possible causes and trouble clearing — The switchover can occur because: <ul style="list-style-type: none"> ■ The span line is down ■ The span line was removed ■ The NAC failed This trap will always be generated with rds1EvDchSwitchOverEnd(123) or rds1EvDchSwitchOverFailure(124).
123	rds1EvDchSwitchOverEnd Trap enable object: <i>usrds1EventNfasDchSwEnd</i>	HiPer DSP <i>span-level call control trap</i>	Condition — the primary NFAS D-channel completed the switchover to the back-up NFAS D-channel. This trap will always be generated with rds1EvDchSwitchOverStart(122).

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
124	rds1EvDchSwitchOverFailure Trap enable object: <i>usrds1EventNfasDchSwfail</i>	HiPer DSP <i>span-level call control trap</i>	<p>Condition — the primary NFAS D-channel failed to switch to the back-up NFAS D-channel.</p> <p>Possible causes and trouble clearing — Failure can occur because:</p> <ul style="list-style-type: none"> ■ The span line is down ■ The span line was removed ■ The back-up NFAS D-channel is not configured properly (see <i>usrds1CfgNFASspanType</i> and <i>usrds1CfgSigGroupType</i>) ■ The NAC failed ■ A telco problem occurred <p>This trap will always be generated with <i>rds1EvDchSwitchOverStart(122)</i>.</p>
125	mdm105ResponderTest Trap enable object: <i>mdmTe105ResponderTest</i>	Quad	Condition — a Modem 105 Responder test occurred in Slot %d, Channel %d.
126 to 165	EdgeServer and CDMA/IWF traps	N/A	These traps are not used in the TCS 3.5 system release.
166 to 177	HiPer TRAX traps	N/A	These traps are not used in the TCS 3.5 system release.
178	AuxIN1Open Trap enable object: <i>nmcCmdFunction</i>	NMC	Condition — the Aux Input Port 1 on the 10/100 Ethernet AUX I/O NIC was put in an open state.
179	AuxIN2Open Trap enable object: <i>nmcCmdFunction</i>	NMC	Condition — the Aux Input Port 2 on the 10/100 Ethernet AUX I/O NIC was put in an open state.
180	AuxIN1Closed Trap enable object: <i>nmcCmdFunction</i>	NMC	Condition — the Aux Input Port 1 on the 10/100 Ethernet AUX I/O NIC was put in a closed state.
181	AuxIN2Closed Trap enable object: <i>nmcCmdFunction</i>	NMC	Condition — the Aux Input Port 2 on the 10/100 Ethernet AUX I/O NIC was put in an open state.

Trap #	Trap description/ Trap enable object	Card(s) affected	Condition/Probable Cause/ Trouble Clearing (continued)
182	AuxOut1Open Trap enable object: nmcCmdFunction openAuxOutputPort1(12)	NMC	Condition — the Aux Output Port 1 on the 10/100 Ethernet AUX I/O NIC was put in a closed state.
183	AuxOut1Closed Trap enable object: nmcCmdFunction closeAuxOutputPort1(14)	NMC	Condition — the Aux Output Port 1 on the 10/100 Ethernet AUX I/O NIC was put in a closed state.
184	AuxOut2Open Trap enable object: nmcCmdFunction openAuxOutputPort2(13)	NMC	Condition — the Aux Output Port 2 on the 10/100 Ethernet AUX I/O NIC was put in an open state.
185	AuxOut2Closed Trap enable object: nmcCmdFunction closeAuxOutputPort2(15)	NMC	Condition — the Aux Output Port 2 on the 10/100 Ethernet AUX I/O NIC was put in a closed state.
186 to 188	CDMA/IWF traps	N/A	These traps are not used in the TCS 3.5 system release.
189	rds1EnterMaintMode Enter D-channel disconnect maintanance mode Trap enable object: usrds1CmdFunction enterDChaDisConnMaintMode(6)	HiPer DSP	This trap is not used in the TCS 3.5 system release.
190	rds1ExitMaintMode Exit D-channel disconnect maintanance mode Trap enable object: usrds1CmdFunction exitDChaDisConnMaintMode(7)	HiPer DSP	This trap is not used in the TCS 3.5 system release.
191	Enter blue alarm maintenance mode Trap enable object: usrds1CmdFunction enterBlueAlmMaintMode(8),	HiPer DSP	This trap is not used in the TCS 3.5 system release.
192	Exit blue alarm maintenance mode Trap enable object: usrds1CmdFunction exitBlueAlmMaintMode(9)	HiPer DSP	This trap is not used in the TCS 3.5 system release.

MODEM DISCONNECT AND FAIL TO CONNECT REASON REFERENCE

This chapter contains a complete listing of Quad and HiPer DSP modem disconnect and call fail reasons.

Overview of Disconnect and Fail to Connect

Disconnect and fail to connect reasons are automatically generated for incoming and outgoing calls.

Outgoing Calls For outgoing calls, when the modem is parsing the “D” of the dial string (ATDT982-5100), it initializes the disconnect and fail to connect reason to None(32).

Incoming Calls For incoming calls, the modem initializes the reasons on the incoming call setup message when in PRI mode, and after starting to answer for POTs and T1 modes.

On Every Call At the end of every call, the modem determines if the result is either a call disconnect reason or a fail to connect reason, and assigns a value. The value for the other reason remains as None(32).

The point at which the modem decides whether there was a connection, and therefore whether it should be a disconnect or a fail to connect reason, is as close as possible to when it creates a connect message, raises carrier detect for RS-232 calls, and starts to pass data. In some cases of dial security, the line is slightly blurred, because the modems may need to make a connection to allow a password to be entered, but delay creating a connect message or raising carrier detect. In this case, an incorrect password is reported as a disconnect reason, and not as a fail to connect reason.

At the end of each call, the modem copies these reasons into the call history structure. The modem keeps statistics from the last five calls, available as objects for the NMC to query. At the same time the modem initializes the disconnect and fail to connect reason, it also increments the value of modulo 5 (the pointer to the call history structure).

When the modem connects, it generates an incoming or outgoing connection established event which contains the value of this call history structure pointer. At the end of the call, the modem generates a connection terminated event, which also contains the value of the call history structure pointer. Both events are sent to the NMC. If the call was a call failed event, instead of a connection established event, the modem generates a connection attempt fail event, again with the value of the call history structure pointer.

The last five call statistics are kept to prevent losing call information if a new call is attempted before the NMC queries the statistics from the previous call. The modem provides means to query the disconnect reason and fail to connect reason instantaneously for the Total Control Manager performance monitor. For accounting purposes, it is best to query the reasons by using the call statistics query after the call using the call history structure pointer.

There are a few exceptions where disconnect reasons and fail to connect reasons are not used. For V.25bis operations, Leased Line operations, and Fax modes, this information may not be available or may be inaccurate. For general dialup data connections over T1, PRI, or POTs, the disconnect and fail to connect reasons should be accurate.

Modem Disconnect
and Fail to Connect
Reasons



This table contains a complete list of modem disconnect and connect fail reasons.

The number assigned to the reason is base 0 in the NETServer syslog. This same reason number is base 1 in the MDM MIB. Therefore, be sure to increase the syslog values by one before using this chart.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes
dtrDrop(1) DTR dropped	Quad	The DTE dropped the Data Terminal ready (DTR) signal, terminating the call.	<ul style="list-style-type: none">■ This message reflects normal operation and does not have a fix.■ This reason only applies to the RS-232 NIC interface. Modems on the packetbus ignore this signal.■ Use DIP switch 1 to override the DTR signal for modems on the packet bus.■ This value may occur incorrectly within Total Control Manager because it is the default value until initialized to none(32).

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
escapeSequence(2) Escape code	Quad HiPer DSP	<p>The user or application software sent the +++ escape code (or other character as set in S2) to the modem, and the modem is configured to disconnect the call when it receives the escape code (refer to DIP switch 9/Register S14.0).</p> <p><i>Note: the DIP switch only applies to the Quad modem.</i></p>	<ul style="list-style-type: none"> ■ This reflects normal operation and may not require a fix. ■ This message is affected by S2, the escape code character; S12, the escape code guard time; and DIP switch 9 (S14.0), escape to Online Command Mode or disconnect. <i>Note: the DIP switch only applies to the Quad modem.</i> Set DIP switch 9 to ON so that it does not disconnect the modem on receipt of the disconnect message. ■ You may need to reconfigure your application software using S2 to send a different escape code. ■ Total Control Manager has a third option of disabling +++ detection, which is equivalent to setting S2 to greater than 127.
athCommand(3) ATH0 Command	Quad HiPer DSP	<p>The user or application software sent the ATH (or ATH0) command to the modem while the modem was in online command mode.</p> <p><i>Note: The online command mode only applies to the Quad modems.</i></p>	This reflects normal operation and does not have a fix.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
carrierLoss(4) Loss of Carrier	Quad HiPer DSP	The modem detected a loss of the remote modem's carrier and waited the time duration specified in Register S10. This register's default is 10 seconds.	<ul style="list-style-type: none"> ■ This message reflects normal operation and does not have a fix. ■ For higher speed modulations, carrier loss detection is disabled when other recovery mechanisms are being used (such as retraining). It is rare to see this disconnect or fail to connect reason because other reasons are usually used or are detected sooner. ■ Increase the duration of Register S10 to two seconds.
inactivityTimeout(5) Inactivity Timeout	Quad HiPer DSP	The modem detected no activity (characters sent or received) for the time duration specified in Register S19. This register's default is 0 with the timer disabled.	<ul style="list-style-type: none"> ■ This message reflects normal operation and does not have a fix. ■ Disable the inactivity timeout by setting Register S19 to 0 (ATS19=0)

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
mnplncompatible(6) MNP Incompatibility	Quad	<p>The modem is set to &M5, forced ARQ mode, and the remote modem does not support V.42 or MNP capability, or there was an error negotiating ARQ.</p> <p>See also S27.4 and S27.5 for selectively enabling MNP-only, V.42-only, or V.42-without detection phase.</p>	<ul style="list-style-type: none"> ■ Either do not force ARQ mode or connect to a remote modem that supports V.42 or MNP capability. ■ Refer to registers S27.4 and S27.5 for selectively enabling MNP-only, V.42-only, or V.42 without Detection phase. ■ If you know that both modems support V.42 and/or MNP, there may be too many bit errors during the negotiation to establish an error-connected link. It may be possible to use &N/&U or to selectively disable higher symbol rates or modulations to limit the modems to a speed where they can negotiate ARQ successfully.
undefined(7)	none	Not used.	
remotePassword(8)	none	Not used.	
linkPassword(9)	none	Not used.	
retransmitLimit(10) Retransmit Limit	Quad HiPer DSP	The modems reached the maximum number of attempts to transfer a frame under ARQ.	<p>Quad – The number of attempts is 12 for low speeds and 48 for high speeds. Under cellular protocols, the number of attempts may be 10 or 20. This may be the result of poor line conditions. Contact the telco.</p> <p>HiPer DSP – The number of attempts is 12. This may be the result of poor line conditions. Contact the telco.</p>

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
linkDisconnectMsgReceived(11) LD Received	Quad HiPer DSP	The remote modem sent an MNP error control Link Disconnect request.	<ul style="list-style-type: none"> ■ This is a normal disconnect procedure under MNP error control when the remote modem is the indicator of the disconnect. ■ This message reflects normal operation and does not have a fix. ■ This error can occur if the user or software application sent +++ ATH, or dropped DTR on the remote modem.
noLoopCurrent(12) Loop loss disconnect	Quad	The modem detected a loss of current on the loop connecting it with the telephone company central office. This usually occurs because the remote modem has hung up; the central office drops current momentarily when there is a disconnect at the other end of the call.	<ul style="list-style-type: none"> ■ This reason reflects normal operation. ■ This reason applied only to POTS calls, and not T1 or PRI calls. ■ This is a catch-all reason on POTS calls. It is the first reason triggered when the phone line is disconnected on the local side. It can mean the remote modem went on-hook abruptly without sending a V.42 Disc, MNP LD, or GSTN Cleardown. Even if the remote modem sends these before going on-hook, there is a race condition over whether it is received locally before the loop loss is detected. ■ Loop loss can also be triggered by Call Waiting on the local side, or by Call Waiting on the remote side if the remote modem hardware supports loop loss detection.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
invalidSpeed(13) Invalid Speed	Quad	<ul style="list-style-type: none"> ■ &N, &U, or both non-zero — the modems were not able to establish a connection in the valid range of data rates ■ &U0, &N>0 — the modems could not establish a connection at the exact rate specified by &N ■ &N0, &U>0 — the modems could not establish a connection at a rate greater or equal to the rate specified by &U ■ 0<&U<&N — the modems could not establish a connection between the rates specified by &U and &N ■ This reason is also used if S48.0, S48.1, or S48.2 is used to disable 300, 1200, or 2400 baud and the modems could not make a connection at any other rate 	Modify lower limit &N or upper limit &U values (AT&N0 and AT&U0). This will allow for connection at the highest possible speed.
unableToRetrain(14) Unable to Retrain	Quad HiPer DSP	<p>During initial training , the modems lost contact and tried to retrain to finish the initial connection.</p> <p>After several attempts, disturbances on the phone line prevented the modems from retraining, and they could no longer transmit or receive data.</p>	This may be the result of poor line conditions. Contact your phone company.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
managementCommand(15) TOTAL CONTROL	Quad	The modem was issued an On-hook or Software Rest command via management. This reason is also used for other reasons, including terminating loopback tests via management, and certain unusual cases of hub security.	This reason reflects normal operation.
noDialTone(16) No Dialtone	Quad HiPer DSP	The modem was set for X2,4,6, or 7, and the modem was unable to detect dial tone from the central office before dialing, or after the "W" (wait for second dial tone) dial modifier.	<ul style="list-style-type: none"> Some countries require detection of a dial tone before dialing. Check %D and verify the modem is configured correctly for analog (%D), T1 (%D1), PRI (%D2) operation. If this is a PRI call, check Register S73 for PRI dialout slot. If this is a POTS call, check the phone line connected to the NIC.
keyAbort(17) Keypress Abort	Quad HiPer DSP	The modem detected a key press while training.	This reflects normal operation and does not have a fix.
lineBusy(18) Busy	Quad HiPer DSP	The modem detected a busy tone because the number dialed was busy.	This reflects normal operation and does not have a fix.
noAnswer(19) No Answer	Quad	The modem was dialing with the @ modifier and did not detect an answer.	<p>This reflects normal operation and does not have a fix.</p> <p>An answer is defined as one or more rings, followed by five or more seconds or silence.</p>
voice(20)	none	Not used.	
noAnswerTone(21)	none	Not used.	
noCarrier(22) No Carrier	HiPer DSP	Obsolete. The modem was not able to connect to the remote modem within the value set with Register S7.	<p>This reflects normal operation and does not have a fix.</p> <p>Use Register S7 to increase connect time.</p>

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
undetermined(23) Undetermined	Quad HiPer DSP	This value is only reported to the NMC on a query during modem training.	This value is not reported on the I6 screen. Try placing the call again.
v42SabmeTimeout(24) SABME Timeout	Quad HiPer DSP	The modems failed the SABME portion of the V.42 link negotiation.	SABME = Set Asynchronous Balance Mode Extended. Try placing the call again.
v42BreakTimeout(25) Break Timeout	Quad	Incompatible processing of a break signal occurred.	Try placing the call again.
v42DisconnectCmd(26) DISC Received	Quad HiPer DSP	The remote modem sent a V.42 error control disconnect request.	<ul style="list-style-type: none"> ■ This is a normal disconnect procedure under V.42 error control when the remote modem is the initiator of the disconnect. ■ This reflects normal operation and does not have a fix. ■ This error can occur if the user or application software sent +++, ATH, or dropped DTR on the remote modem.
v42IdExchangeFail(27) XID Timeout	Quad HiPer DSP	The modems failed to negotiate the V.42 XID exchange phase.	Check modem V.42 configuration. Most likely due to a bad phone line.
v42BadSetup(28) Extra Stepup	Quad	The modem received an invalid V.42 bis (compression) frame. This enumeration should read <i>v42ExtraStepup</i> .	Try placing the call again.
v42InvalidCodeWord(29) Illegal Command Code	Quad HiPer DSP	The modem received an invalid V.42 bis (compression) frame.	This reason will likely never occur. If it does, try placing the call again.
v42StringToLong(30) A Rootless Tree	Quad HiPer DSP	The modem received an invalid V.42 bis (compression) frame.	Try placing the call again.
v42InvalidCommand(31) Invalid Codeword	Quad HiPer DSP	The modem received an invalid V.42 bis (compression) frame.	Try placing the call again.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
none(32) No Connection Online No Fail (HiPer DSP)	Quad HiPer DSP	This is the value reported to the NMC on a query or disconnect reason during modem training or while connected. This may also be a query of the call fail reason when the call did not fail.	<ul style="list-style-type: none"> ■ This reflects normal operation. ■ In the screen mode, the modem will report "No Connection" after being reset on power-up. ■ In the online command mode, the modem will display "Online" instead of a disconnect reason.
v32Cleardown(33) GSTN Cleardown	Quad HiPer DSP	<p>The modems were unable to find a common rate at which to connect.</p> <p>For V.90 connections — fallbacks to V.34 (S76.3) were disabled and the line was unable to support a V.90 connection.</p> <p>Disconnect reason — this is a normal disconnect method for high speed connections.</p>	<p>Originally defined for V.32 calls, this reason is extended to V.34, V.FC, and V.90 connections. It is generally called the "GSTN (General Switch Telephone Network) Clear Down."</p> <p>This reason in part reflects normal operation.</p> <p>Make sure both modems are set to variable link connection speeds (AT&N0 and AT&U0).</p>
dialSecurity(34) Dial Security	Quad	<p>Hub security failed for one of several reasons, including:</p> <ul style="list-style-type: none"> ■ Invalid password ■ The modem was unable to communicate with the NMC for a hub security session and was set to refuse calls ■ The modem is disconnecting the initial call in preparation for dialback security 	<ul style="list-style-type: none"> ■ Verify all passwords. ■ Verify chassis communication. ■ If used for dialback security, this reason reflects normal operation.
remoteAccessDenied(35)	none	Not used.	
loopLoss(36)	none	Not used.	

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
ds0Teardown(37) Call Teardown	Quad	<p>The T1 or PRI card initiated the disconnect. This usually occurs because the remote modem hung up: the central office signals to the T1 or PRI card when there is a disconnect at the other end of the call (telco disconnect).</p> <p>The modem detected that the phone network told it the call had hung up. This normally occurs in response to a client modem hang up.</p>	<ul style="list-style-type: none"> ■ This reason only applies to T1 or PRI calls, and not POTS calls. ■ This message is generated by the telco switch. ■ This is a catch-all reason for T1 and PRI calls. Typically, the hang-up was remotely initiated from the remote end of the call, and is viewed as a disconnect request from the telco. The remote user may have broken the call, the remote modem may have hung up, or something may be wrong within the phone network. It can mean the remote modem went on-hook abruptly without sending a V.42 Disc, MNP LD, or Cleardown. Even if the remote modem sends these before going on-hook, there is a race condition over whether it is received locally before the DS0 teardown is received. ■ As a failure to connect reason, this may mean that a non-modem call was received. ■ The reason <i>failToTrain</i> was added in TCS release 3.0 to help isolate non-modem calls.
promptNotEnabled(38) Prompting Not Enabled	Quad	<p>With Link Security enabled, the modem hung up because the originating modem did not send an autopass password, and prompting was not enabled.</p>	<p>Disable link security with ATS53.0=0.</p> <p>Enable Fallback Password Prompting with ATS53.1=1.</p>

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
noPromptingInSync(39) No Prompting in Sync	Quad	With Link Security enabled, the originating modem did not send an autopass password, and the answering modem cannot prompt for a password in any synchronous mode.	Disable link security with ATS53.0=0 . Configure the autopass password on the originating modem.
nonArqMode(40) Non ARQ Mode	none	Not used.	
modeIncompatible(41) Mode Incompatible	Quad	With Link Security enabled, the modem hung up because both modems were not set to the same error control setting.	Make sure the originating and answering modems are set for the same error control setting.
noPromptInNonARQ(42) No Prompting in Non-ARQ	none	With link security enabled, prompting was enabled, but the modem hung up because the originating modem was set for answer control while the answering modem was set for non-error control.	The answering modem cannot prompt when it is set for non-error control. Set the originating modem for error control (AT&M4).
dialBackLink(43) Dial Back Security	none	Not used.	
linkAbort(44) Security Abort	Quad	<ul style="list-style-type: none"> With link security enabled, the modem hung up for one of these reasons: It received an invalid password too many times There was a timeout waiting for the user to enter a password The modem disconnected preparing to dial back security 	Verify the user password in the RADIUS server or gateway.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
autopassfailed(45) Autopass Failed	Quad	With link security enabled, the originating modem did not send an autopass password, and the answering modem did not have prompting enabled but did have forced security mode enabled.	Verify the user autopass password in the RADIUS server or the gateway.
pbGenericError(46) PACKET BUS - Generic Error	Quad HiPer DSP	The packet bus link to the modem was brought down and the modem was not able to determine the reason.	<p>Quad – This reason should not occur; reasons 47-61 cover all known cases of packetbus errors. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.</p> <p>HiPer DSP – This reason is reported for the majority of packetbus errors. Check to see that packetbus links are active on the gateway card.</p>
pbLinkErrTxPreAck(47) PACKET BUS LINK ERROR - (Transmit Pre ACK)	Quad	An error occurred in the packet bus link between the modem and a gateway card which was detected by the packet bus protocol.	This is an unlikely reason. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.
pbLinkErrTxTardyACK(48) PACKET BUS LINK ERROR - (Transmit Tardy ACK)	Quad	An error occurred in the packet bus link between the modem and a gateway card which was detected by the packet bus protocol.	This is an unlikely reason. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
pbTransmitBusTimeout(49) PACKET BUS - Transmit Bus Timeout	Quad	An error occurred in the packet bus physical layer.	<ul style="list-style-type: none"> ■ This can be caused by removing or resetting gateway or PRI cards in the chassis. ■ This can also be caused if the receiver of the gateway card is busy and cannot accept any more data. ■ The timeout is 409 microseconds. ■ This error is fairly likely to occur.
pbReceiveBusTimeout(50) PACKET BUS - Receive Bus Timeout	Quad	Obsolete.	<p>This error is ignored.</p> <p>This value is used in debugging Quad modem hardware. When PRI was added with the capability of multiple PRI and gateway cards, it was no longer desirable to drop the data link on a receiver error because the error might be from a card that is not related to the connection.</p>
pbLinkErrTxTAL(51) PACKET BUS LINK ERROR - (Transmit TAL)	Quad	Obsolete.	
pbLinkErrRxTAL(52) PACKET BUS LINK ERROR - (Receive TAL)	Quad	Obsolete.	

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
pbTransmitMasterTimeout(53) PACKET BUS - Transmit Master Timeout	Quad	An error occurred in the packet bus physical layer. Master machine timed out by driver. May also be due to a modem that is not functioning correctly.	<ul style="list-style-type: none"> ■ This value is related to <i>pbTransmitBusTimeout(49)</i>. ■ Value 49 is detected by the packetbus hardware and depends on the presence of the destination gateway or PRI card. ■ The Quad modem has an additional software timeout to handle the case of the gateway or PRI card being removed from the chassis. ■ The timeout length is 500 milliseconds. ■ This is an unlikely reason. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.
pbClockMissing(54) PACKET BUS - Clock Missing	Quad	Obsolete.	
pbReceivedLsWhileLinkUp(55) PACKET BUS - Received LS while Link Up Link Start Received (HiPer DSP)	Quad HiPer DSP	An error occurred in the packet bus physical layer. The modem received a request to start a new link while it was in a link. This caused the current link to drop and a new link to be attempted.	This is an unlikely reason. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.
pbOutOfSequenceFrame(56) PACKET BUS - Out of Sequence Frame	Quad HiPer DSP	An error occurred in the packet bus physical layer. The modem received a frame out of sequence, or a frame was missing.	This is an unlikely reason. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
pbBadFrame(57) PACKET BUS - Bad Frame	Quad	An error occurred in the packet bus link layer. The modem received a frame with an invalid frame type. The frame was neither a data frame nor a recognized control frame.	This is an unlikely reason. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.
pbAckWaitTimeout(58) PACKET BUS - ACK Wait Timeout	Quad HiPer DSP	An error occurred in the packet bus link layer. The modem did not receive an acknowledgment frame for a data frame it had sent.	This is an unlikely reason. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.
pbReceivedAckSequenceErr(59) PACKET BUS - Received ACK Sequence Error	Quad	An error occurred in the packet bus link layer. The modem received an acknowledgment frame out of sequence, or an ack frame was missing.	This is an unlikely reason. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.
pbRecieveOvrflwRNRFailed(60) PACKET BUS - Receive Overflow, RNR Failed	Quad HiPer DSP	An error occurred in the packet bus link layer. The modem ran out of buffer space for received data frames.	This error should not occur because the modem sends Receiver Not Ready control frames to tell gateway cards to stop sending data before it runs out of buffer space. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.
pbReceiveMsgBufOvrflw(61) PACKET BUS - Receive Message Buffer Overflow	Quad	An error occurred in the packet bus link layer. The modem ran out of buffer space for received control frames.	This error should not occur because the modem control frame buffer space is larger than the maximum number of control frames that can be sent. If this reason does occur with any frequency, reboot the NETServer. If it still occurs, try to isolate which NIC is having the problem. Test and replace the NIC if necessary.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
rcvdGatewayDiscCmd(62) Gateway Disconnect Command Received	Quad HiPer DSP	The gateway card sent the modem a disconnect command.	This is a normal method of terminating a call locally. This reason only applies to calls on the packetbus and is analogous to dtrDrop on the RS-232 NIC interface. Common causes for this reason include idle time-out, session limit, failure to login within the specified time limit, and failure to authenticate.
tokenPassingTimeout(63) Token Passing Timeout	Quad	An error occurred in the internal communications with the Digital Signal Processor (DSP) of the modem.	Try placing the call again.
dspInterruptTimeout(64) DSP Interrupt Timeout	Quad	An error occurred in the clock signal generated by the Digital Signal Processor (DSP) of the modem.	Try placing the call again.
mnpProtocolViolation(65) MNP Protocol Violation	Quad HiPer DSP	An error occurred in the MNP error control protocol. The modem received a frame out of sequence.	Try placing the call again.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
class2FaxHangupCmd(66) Class 2 Fax Hangup Command	Quad	The fax application software sent the modem the Class 2.0 Fax command to terminate the call.	<ul style="list-style-type: none"> ■ This is a normal disconnect message from a Class 2.0 Fax call. ■ The modem does not always report disconnect reasons for Fax connections, especially under Class 1 fax calls. This is due to the nature of Fax sessions: the modem is actually making several connections in a Fax session to send control information, then the actual page data, and finally the end of page control information. ■ This reason is also due to the variances in Fax machines and Fax application software programs, which do not always send disconnect commands, or send them multiple times or in the wrong order./
hstSpeedSwitchTimeout(67) HST Speed Switch Timeout	Quad	Under an HST connection, the modem was not able to complete a speed shift call.	This may be caused by too many disturbances on the line.
tooManyUnacked(68) 128 Unacked LMIs	Quad	Under an MNP10 Cellular connection, the modem was not receiving acknowledgment messages from the remote modem.	This may be caused by too many disturbances on the line.
timerExpired(69) RDL Timer Expired	Quad	A Remote Digital Loopback (RDL) test was terminated using the Register S18 timer.	Increase the Register S18 timer to 30 (ATS18=30).

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
t1Glare(70) T1 Glare	Quad	A Glare (trying to dial out at the same time as an incoming call) condition occurred on a T1 line, causing the modem to abort the dial-out to accept the incoming call.	This reason reflects normal operation.
priDialoutRqTimeout(71) PRI Dialout Request Timeout	Quad	For PRI calls, the modem was unable to get a B-channel allocated from the PRI card.	This reason in part reflects normal operation if all B-channels are active. Verify that the PRI is in the correct slot. Verify Register S73, the default PRI card slot.
abortAnlgDstOvrIsdn(72) Abort Analog Destination over ISDN Network Connection	Quad HiPer DSP	The modem was originating a PRI-ISDN call in universal connect mode. It detected an analog modem answering, but was configured not to accept analog calls (S68.0=1).	Modem not configured for analog. Change setting if you want to accept analog (ATS68.0=1).
normalUserCallClear(73) Normal User Call Clear	HiPer DSP	The remote modem hung up.	Quad – The Quad modem uses <i>ds0Teardown</i> for PRI disconnects. HiPer DSP – This is a Q931 telco clear condition.
normalUnspecified(74) Normal, Unspecified	HiPer DSP	The remote modem hung up.	Quad – The Quad modem uses <i>ds0Teardown</i> for PRI disconnects. HiPer DSP – This is a Q931 telco clear condition.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
bearerIncompatibility(75) Bearer Incompatible	HiPer DSP	The modem was configured for analog, but detected a digital call and rejected it. Or the modem was configured for digital, but detected an analog call and rejected it. Or the modem was not correctly configured for a specific type of digital call.	<p>Quad – The Quad modem uses <i>ds0TearDown</i> for PRI disconnects.</p> <p>HiPer DSP – This value indicates the modem was trying to complete a PRI call with a bearer capability unavailable on that line. This error should not occur if the modem is configured for universal connect.</p> <p>Check the configuration of span parameters. Set AT*V2 to "0" for universal connect. If the configuration is set to universal connect, contact the telco.</p> <p>The call likely failed at the telco level for one of these Q931 reasons:</p> <ul style="list-style-type: none"> ■ Unassigned number ■ No user responding ■ Call rejected ■ Service not available ■ Capability not implemented ■ Bearer capability not available ■ Incomplete destination ■ Interworking unspecified

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
protocolErrorEvent(76) Protocol Error	HiPer DSP	<p>This error can be caused by these conditions:</p> <ul style="list-style-type: none"> Invalid HDLC buffer received from the remote modem. This reason typically applies to V.42, V.10, X.75, or sync PPP calls. V.42 or V.120 MLL (modem link layer) sequence error. The modem received a frame with an invalid sequence number. The modem received an invalid HDLC frame The modem received a DM (disconnect mode) frame while connected to a V.42 analog call. This situation is also likely to cause a <i>linkDisconnect</i>. 	<p>Quad – The Quad modem uses <i>ds0Teardown</i> for PRI disconnects.</p> <p>This value is used for any PRI telco disconnect other than the reasons the modem displays explicitly:</p> <ul style="list-style-type: none"> <i>normalUserCallClear</i> <i>normalUnspecified</i> <i>bearerIncompatibility</i> <i>abnormalDisconnect</i> <i>invalidCauseValue</i> <p>HiPer DSP – This may be an intermittent error from the remote modem. If the problem persists, the DSP portion of the remote modem may be corrupt.</p>
abnormalDisconnect(77) Abnormal Disconnection	HiPer DSP	<p>An ANI/DNIS-based modem initialization string failed. Upon a call attempt, the modem rejected the initialization string.</p> <p>The DSP did not detect a valid address.</p>	<p>Quad – The Quad modem uses <i>ds0Teardown</i> for PRI disconnects.</p> <p>HiPer DSP – Check the integrity of the ANI/DNIS initialization strings. The look-up table may be corrupt. Check the AT strings. Also make sure the number of digits in the ANI/DNIS string is correct.</p>
invalidCauseValue(78) No cause value received or invalid cause value	none	Not used.	

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
resourceUnavailable(79) Resources Unavailable	HiPer DSP	No internal DSP resource is available. A DSP likely failed an internal "keep alive" test and was removed from the resource pool. ANY call mapped to that DSP will fail (this shows as a channel that will not accept calls).	<ul style="list-style-type: none"> ■ This reason reflects normal operation. ■ Try placing the call again. ■ Reboot the card. If the card will not reboot, replace it. If the card will reboot, but the problem persists, replace the card.
remotHungUpDuringTraining(80) Remote On Hook Timeout	Quad	The remote modem was detected, but went on-hook (hung up) before the modems could finish training.	<ul style="list-style-type: none"> ■ This reason replaces <i>ds0TearDown</i> (or <i>loopLoss</i> for POTS) when a remote modem has been detected. This object was added to help isolate call failure reasons. ■ Verify that Register S7 is not set to "short". ■ This reason can also occur if the modems are forced to different modulations or to a modulation that is not supported by one of the modems.
trainingTimeout(81) Training Timeout		The remote modem was detected, but went on-hook (hung up) because the S7 timer expired before the modems could finish training.	<ul style="list-style-type: none"> ■ This reason replaces <i>noCarrier</i> when a remote modem has been detected. ■ Verify that Register S7 is not set to "short". ■ This reason can also occur if the modems are forced to different modulations or to a modulation that is not supported by one of the modems.
incomingModemNotAvailable(82) Incoming Call - Modem not available	HiPer DSP	A modem was not available to an incoming call.	
incomingInvalidBearerCap(83) Incoming Call - Invalid bearer capability	none	Not used.	
incomingInvalidChannelID(84) Incoming Call - Invalid Channel ID	none	Not used.	

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
incomingInvalidProgInd(85) Incoming Call - Invalid Progress Indication	none	Not used.	
incomingInvalidCallingPty(86) Incoming Call - Invalid Calling Party	none	Not used.	
incomingInvalidCalledPty(87) Incoming Call - Invalid Called Party	none	Not used.	
incomingCallBlock(88) Incoming Call - Call Blocked	HiPer DSP	<p>This reason applies only to channelized E1. The particular call type is blocked. Either:</p> <ul style="list-style-type: none"> ■ The DS0 or entire span is configured to block calls ■ The call is determined to be analog and the DS0 or span is configured to block analog calls ■ The call is determined to be digital and the DS0 or span is configured to block digital calls 	Check DS0 and span configuration.
incomingLoopStNoRingOff(89) Incoming Call - Loop Start No Ring Off	none	Not used.	
outgoingTelcoDisconnect(90) Outgoing Call - Telco Disconnect	none	Not used.	
outgoingEMWinkTimeout(91) Outgoing Call - E&M Wink Timeout	none	Not used.	
outgoingEMWinkTooShort(92) Outgoing Call - E&M Wink Too Short	none	Not used.	
outgoingNoChannelAvail(93) Outgoing Call - No Channel Available	none	Not used.	
dspReboot(94) DSP rebooted	HiPer DSP	The DSP rebooted because of an internal condition. All calls were dropped.	This is likely only an intermittent condition. If the problem persists, the card may need to be replaced.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
noDSPRespToKA(95) DSP Did not respond to Keep Alive message	HiPer DSP	The DSP failed an internal "keep alive" test and rebooted, dropping all calls.	This is likely only an intermittent condition. If the problem persists, the card may need to be replaced.
noDSPRespToDisc(96) DSP Did not respond to disconnect request	HiPer DSP	After a disconnect request, the DSP did not respond within the allocated time. The DSP will automatically reboot, dropping all calls..	This is likely only an intermittent condition. If the problem persists, the card may need to be replaced.
dspTailPtrInvalid(97) DSP Tail Pointer Invalid	HiPer DSP	The DSP driver detected an error in its tail pointer ("tail pointer out of range"). The call will drop, but the DSP will not reboot.	This is likely only an intermittent condition. If the problem persists, the card may need to be replaced.
dspHeadPtrInvalid(98) DSP Head Pointer Invalid	HiPer DSP	The DSP driver detected an error in its head pointer ("head pointer out of range"). The call will drop, but the DSP will not reboot.	This is likely only an intermittent condition. If the problem persists, the card may need to be replaced.
dataProcessingGenericErr(99) Data Processing Generic Error	HiPer DSP	Internal processing error in the MRL (modem reliable link).	
timeslotUnavailable(100) Timeslot Un-available	HiPer DSP	No timeslots are available. Either the span is down, or no DS0s are available to take the call.	<ul style="list-style-type: none"> ■ Make sure the span is not unplugged or down. ■ Check if all channels are busy. ■ Try to test a call on a connected DS0 to rule out a DS0 problem. ■ Contact the telco.
GMTTimeNotSet(101) GMT (Greenwich Mean Time)	none	Not used.	
chasAwarenessNotAvailable(102) Chassis Awareness Unavailable from NMC	none	Not used.	
R2InvalidChannelDirection(103) R2 Line Direction	HiPer DSP	An outgoing call was attempted on a span configured for incoming calls only.	Check the line direction setting in <i>hdr2CfgLineDirection</i> .

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
R2ChannelBlockedByNetwork(104) R2 Channel Blocked	HiPer DSP	A blocking signal from the telco is present on the span.	Contact the telco.
R2Glare(105) R2 Glare	HiPer DSP	Both an incoming and an outgoing call were attempted simultaneously on the same channel. One call will be dropped; the incoming call typically takes priority.	
R2OutgoingCallBlocked(106) R2 Outcall Call Blocked	HiPer DSP	An outgoing call was attempted on a span or timeslot that is configured to be blocked.	Use the object <i>hdr2CfgBlkToBlk</i> to change call blocking configuration.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
R2DNISNotFound(107) R2 DNIS not Found	HiPer DSP	<p>When an incoming call arrives, the DNIS number (if present), is checked against the DNIS table. If a match is found, the call is accepted. If no match is found, a decision must be taken on how to handle the call. You can configure whether to a) accept the call, b) reject the call, c) accept the call as an analog call or d) accept the call as a digital call.</p> <p>The Call Type Override table provides this functionality (using the command <i>hdr2InCatMapCallType</i>). The command <i>hdr2CfgBNumNFnd</i> allows you to select the action to be taken when no match is found for the DNIS number on an incoming call.</p> <p><i>Note: If the call is rejected, the cause code used to reject the call is determined by the value of the <i>hdr2CfgWrongNumber</i> configurable parameter.</i></p>	Check the setting of <i>hdr2CfgBNumNFnd</i> .
R2SigCauseCongestion(108) R2 Congestion	HiPer DSP	<ul style="list-style-type: none"> ■ An incoming call is attempted on a span configured for outgoing only. ■ No modem is available. ■ A blocking signal is being transmitted from the NAC. ■ The span is configured to block the call type. 	<ul style="list-style-type: none"> ■ This is the default R2 reason for rejection of an incoming call. ■ Check the line direction setting in <i>hdr2CfgLineDirection</i>. ■ Check the setting in <i>hdr2CfgBNumNFnd</i>.

MIB enumeration/Reason	Modems affected	Description	Trouble clearing notes (continued)
R2SigCauseUnallocNumber(109) Unallocated Number	HiPer DSP	The incoming number was not found in the DNIS table. When no match is found for the DNIS on an incoming call against the DNIS table, and the value of <i>hdr2CfgBNumNFnd</i> is set to reject the call, this reason is used to reject the call.	Check the setting for <i>hdr2CfgBNumNFnd</i> . This object determines the action on a B-Number not found in the DNIS handling table.
R2DSPFatalError(110) Fatal Error	HiPer DSP	A fatal error occurred.	
callBlacklisted(111) Blacklist Reject	HiPer DSP	The attempted phone number is blacklisted.	Check to see if the number is blacklisted. If it is not, reboot the NAC.

AUTORESPONSE REFERENCE

This chapter contains information about the AutoResponse feature.

Overview

AutoResponse lets you automatically take action (such as triggering alarms) against common chassis events. You can also use AutoResponse actions to monitor chassis performance and trouble clear problems.

To enable AutoResponse, you must:

- Create a script using Total Control Manager or an SNMP browser
- Enable associated traps and/or log events

AutoResponse scripts are saved to NMC NVRAM. You may use Total Control Manager or an SNMP browser to configure the scripts. The actual scripts are parsed by the NMC, but many contain actions that are executed by other devices in the chassis (for example, “busy out phone line” is executed on a modem).

Basic Operation

AutoResponse allows you to define a set of actions (an AutoResponse script) to be taken automatically when a specified event occurs within the chassis. The event may be specific to a particular module (NMC or NAC) in a given slot of the chassis, or specific to a particular entity (such as a given modemchannel).

Total Control Manager provides a convenient graphical user interface (GUI) through which you can configure these response scripts. Access the AutoResponse configuration through Total Control Manager by selecting **Auto Response** after clicking on a specific card.



Using Total Control Manager is the easiest way to program AutoResponse. You may also use anSNMP browser to configure the scripts if your browser allows you to manually enter hexadecimal (hex) octets. Many browsers do not allow you to enter “raw” hex values, and cannot be used to create AutoResponse scripts. Most browsers that do allow you to enter hex values require the values entered in left-to-right order. For example, to add the “Remove DS0 from Service” response (22), encode the hex values as: 22 01 02 05.

Total Control Manager does not need to be running at the time an event occurs in order for the NMC to invoke the appropriate response script, since these scripts are pre-programmed into the NMC and saved to NMC's NVRAM. The NMC must be present and functioning for the scripts to execute.

A user-modifiable AutoResponse SNMP TRAP (*nmcArCustomTrap*) can be sent from the NMC in response to any event.

Events and Responses

AutoResponse events are defined as something that occurs within the chassis that deviates from normal operation. AutoResponse events are grouped into three categories: generic chassis-level events, chassis slot-level events, and entity-level events.

Responses are the actions the affected NAC will take as a result of a specific event. AutoResponse responses must be programmed in order for them to operate.

Chassis Events

Two types of chassis events can occur: those that are generic to the chassis-level (PSU, temperature, fan, and global timer events) and those that apply to a specific slot in the chassis. The generic chassis events and the chassis slot-level events use the same list of responses.

Generic Chassis-level Events

Generic chassis-level events apply to the NMC and all NACs, as well as to the slot-level Quad and HiPer DSP modem settings. These are the generic chassis-level events that will trigger an AutoResponse script:

Event (Total Control Manager name)	MIB object
PSU Voltage Out of Range	uchasArPsuVoltOutOfRange 1.3.6.1.4.1.429.1.1.9.1
PSU Failed	uchasArPsuFailed 1.3.6.1.4.1.429.1.1.9.2
Fan Failed	uchasArFanFailed 1.3.6.1.4.1.429.1.1.9.3
HUB Temperature Out Of Range	uchasArHubTempOutOfRange 1.3.6.1.4.1.429.1.1.9.4
Global Timer 1 Expired	uchasArTimer1 1.3.6.1.4.1.429.1.1.9.5 The AutoResponse action executes after the timer expires. Build the AutoResponse script so that an action occurs after the timer expires. In some cases, you may wish to start a second timer to stop the AutoResponse script or to start execution of a new script. In this case, when global timer 1 expires, script A runs. At the same time, global timer 2 starts. When timer 2 expires, the AutoResponse script stops.
Global Timer 2 Expired	uchasArTimer2 1.3.6.1.4.1.429.1.1.9.6 The AutoResponse action executes after the timer expires. Refer to additional notes for Global Timer 1.
Global Timer 3 Expired	uchasArTimer3 1.3.6.1.4.1.429.1.1.9.7 The AutoResponse action executes after the timer expires. Refer to additional notes for Global Timer 1.
Global Timer 4 Expired	uchasArTimer4 1.3.6.1.4.1.429.1.1.9.8 The AutoResponse action executes after the timer expires. Refer to additional notes for Global Timer 1.

Chassis Slot-level Events

Chassis slot-level events are defined within the `uchasArSlotTable` in the Chassis MIB. Access this table using the same slot numbers that you use to access the `uchasSlotTable`. The table allows maximum flexibility by allowing different scripts to be applied to different slots. For example, a script applied to a slot that will hold modem devices might also be configured to busy out trunk cards when the modem is removed. The same slot could also be configured so that all DS1s on the trunk card will be removed from service when a gateway card is removed.

These chassis slot-level events will trigger an AutoResponse script.

Event (Total Control Manager name)	MIB object	Description
Module Inserted	<code>uchasArModuleInserted</code> 1.3.6.1.4.1.429.1.1.9.9.1.2	This event occurs when a module is inserted in to the chassis both NICs and NACs.
Module Reinitialized	<code>uchasArModuleReinit</code> 1.3.6.1.4.1.429.1.1.9.9.1.3	This event occurs when any of these conditions are met: <ul style="list-style-type: none"> ■ Chassis power transitions from off to on. ■ A module is inserted into the chassis. ■ A software download to a module has completed. ■ A module is restored to service. ■ A module has a hardware reset.
Module Removed	<code>uchasArModuleRemoved</code> 1.3.6.1.4.1.429.1.1.9.9.1.4	This event occurs when the NMC detects that a NIC or NAC was not physically removed from the chassis. This script does not apply to power supplies.
Module Non operational	<code>uchasArModuleNonoper</code> 1.3.6.1.4.1.429.1.1.9.9.1.5	This event occurs when any of these conditions are met <ul style="list-style-type: none"> ■ A software download to a module starts. ■ A module is removed from service. ■ A module failed. ■ All entities on a module failed.

Event (Total Control Manager name)	MIB object	Description (continued)
Module Watchdog Timeout	uchasArModuleWatchdog 1.3.6.1.4.1.429.1.1.9.9.1.6	This event occurs when one or more module entities experience a watchdog time-out.

Chassis-level Responses

The generic chassis-level events and the chassis slot-level events use the same responses. Once a response script has been constructed from the following list of responses, it acts upon all entities on a card that is plugged into a specific slot.

Chassis-level responses apply to the NMC and any NAC, as well as the slot-level Quad and HiPer DSP modem settings. Twenty-six chassis-level events are defined. Total Control Manager does not use all of these responses; you may use the additional responses if you write your own scripts.

Possible Chassis-level Responses

These are the possible chassis-level responses to an event:

Response	ID	Description
Generate AutoResponse SNMP TRAP ID [N]	1	<p>A single user-defined value can be configured to be sent in this TRAP.</p> <p>The MS or some other higher level AutoResponse function can use this feature to convey information regarding the fact that a specific logic state had been detected (i.e. a specific point in the response script had been reached) for performing additional responses to better handle the specific event that occurred.</p> <p>Response Descriptor(s):</p> <p>User-defined AutoResponse SNMP TRAP ID (0–127)</p>
Delay Script Execution [N] Seconds	2	<p>This response causes a given script to suspend execution for the amount of seconds specified in the response descriptor.</p> <p>Response Descriptor(s):</p> <p>Time Delay in seconds (0–127)</p>

Response	ID	Description
Terminate Script Execution	3	This response immediately stops AutoResponse script execution. This provides a simple mechanism for disabling scripts without deleting their entire definition. For example, inserting this response at the front of an existing script will not cause this script to execute. Then, at some later date, the script can be re-enabled by removing this response. There are no restrictions on where in the script this response can be placed.
Continue If Test Passes	4	When a test response is executed, the result of the test(s) effects further execution of the script. If the test fails, the script execution continues at the response that immediately follows the test. If the test passes, the script parser searches linearly through the script until it finds the next "Continue if Test Passes" response. Once this response is found, the script execution continues at the response that immediately follows the "Continue if Test Passes" response. This provides the AutoResponse function with some primitive decision making capabilities without need for a full script language.
Configure Module from NMC NVRAM	5	A module is defined as a NAC or the NMC. This response configures all parameters for every entity on a module. This includes card-level and channel-level entities. The values of all parameters to be configured are those that were last saved to the NMC's NVRAM through the "Save To NVRAM" command. This response will perform the same level of configuration that is supported by the auto-configure feature of the NMC.
Configure Module From NMC Factory Default	6	A module is defined as a NAC or the NMC. This response configures all parameters for every entity on a module. This includes card-level and channel entities. The default values are used to configure all parameters.
Remove Module from Service (not part of Total Control Manager)	7	<p>A module is defined as a NAC or the NMC. This response holds all processors and circuits, on the module, in a reset state via the hardware reset circuit on the management bus interface.</p> <p><i>Note: All modems on a multi-modem card are held in reset state via this response.</i></p>
Restore Module to Service (not part of Total Control Manager)	8	<p>A module is defined as a NAC or the NMC. This response removes all processors and circuits on the module from a reset state, via the hardware reset circuit on the management bus interface.</p> <p><i>Note: All modems on a multi-modem card are removed from the reset state via this response.</i></p>

Response	ID	Description
Test Module	9	A module is defined as a NAC or the NMC. This response tests all testable entities on the given module. In the case of Quad modems, all modems on that card are run through the self test. Responses that perform tests are the mechanisms through which branching can be done in the response script execution. Refer to the response "Continue if Test Passes" for a further description of this branching.
Reset Module	10	<p>A module is defined as a NAC or the NMC. This response resets all processors and circuits on the module to a known state via the hardware reset circuit on the management bus interface.</p> <p>Hex value: 0A</p> <p><i>Note: All modems on a multi-modem card are reset via this response.</i></p>
Busy Out Phone Line	11	<p>This analog Dual and Quad Modem script applies only to the slot to which it is assigned. This response is useful for monitoring insert and remove events. It issues a "busy out phone line" command to all modems in the assigned slot.</p> <p>Hex value: 0B</p>
Restore Phone Line	12	<p>This analog Dual and Quad Modem script applies only to the slot to which it is assigned. This response is useful for monitoring insert and remove events. It issues a "restore phone line" command to all modems in the assigned slot.</p> <p>Hex value: 0C</p>
Remove DS1 from Service	13	<p>Instructs the T1-E1/ PRI card at slot (N) to remove the DS1 span from service.</p> <p>This response is followed by two additional octets that specify slot and DS1, respectively. This response has an identical command in the <i>uds1CmdFunction</i> object.</p> <p>Hex value: 0D</p> <p>Equivalent <i>uds1CmdFunction</i> enumeration: <i>localOutOfService(6)</i></p>

Response	ID	Description
Restore DS1 to Service	14	<p>Instructs the T1-E1/PRI card to put the DS1 in service and restore all channels on the span to an in-service state.</p> <p>This response is followed by two additional octets that specify slot and DS1, respectively. This response has an identical command in the <i>uds1CmdFunction</i> object.</p> <p>Hex value: 0E</p> <p>Equivalent <i>uds1CmdFunction</i> enumeration: <i>inService(5)</i></p>
Block DS1 Analog Calls	15	<p>Instructs the T1-E1/PRI card to block all analog calls and only allow incoming BRI/PRI digital calls.</p> <p>This response is followed by two additional octets that specify slot and DS1, respectively. This response has an identical command in the <i>uds1CmdFunction</i> object.</p> <p>Hex value: 0F</p> <p>Equivalent <i>uds1CmdFunction</i> enumeration: <i>blockAnalogCalls(7)</i></p>
Block DS1 Digital Calls	16	<p>Instructs the T1-E1/PRI card to block all incoming digital BRI/PRI calls and only allow analog calls.</p> <p>This response is followed by two additional octets that specify slot and DS1, respectively. This response has an identical command in the <i>uds1CmdFunction</i> object.</p> <p>Hex value: 10</p> <p>Equivalent <i>uds1CmdFunction</i> enumeration: <i>blockDigitalCalls(8)</i></p>
Block DS1 All Calls	17	<p>Instructs the T1-E1/PRI card to block both analog and digital calls.</p> <p>This response is followed by two additional octets that specify slot and DS1, respectively. This response has an identical command in the <i>uds1CmdFunction</i> object.</p> <p>Hex value: 11</p> <p>Equivalent <i>uds1CmdFunction</i> enumeration: <i>blockAllCalls(9)</i></p>

Response	ID	Description
Block DS1 No Calls	18	<p>Instructs the T1-E1/PRI card to restore all analog and digital calls.</p> <p>This response is followed by two additional octets that specify slot and DS1, respectively. This response has an identical command in the <i>uds1CmdFunction</i> object.</p> <p>Hex value: 12</p> <p>Equivalent <i>uds1CmdFunction</i> enumeration: <i>blockNoCalls(10)</i></p>
Aux I/O Input 1 Open	19	<p>Opens the specified input port. Used for the HiPer NMC only.</p> <p>Hex value: 13</p>
Aux I/O Input 2 Open	20	<p>Opens the specified input port. Used for the HiPer NMC only.</p> <p>Hex value: 14</p>
Aux I/O Input 1 Closed	21	<p>Closes the specified input port. Used for the HiPer NMC only.</p> <p>Hex value: 15</p>
Aux I/O Input 2 Closed	22	<p>Closes the specified input port. Used for the HiPer NMC only.</p> <p>Hex value: 16</p>
Aux I/O Output 1 Open	23	<p>Opens the specified output port. Used for the HiPer NMC only.</p> <p>Hex value: 17</p>
Aux I/O Output 2 Open	24	<p>Opens the specified output port. Used for the HiPer NMC only.</p> <p>Hex value: 18</p>
Aux I/O Output 1 Closed	25	<p>Closes the specified output port. Used for the HiPer NMC only.</p> <p>Hex value: 19</p>
Aux I/O Output 2 Closed	26	<p>Closes the specified output port. Used for the HiPer NMC only.</p> <p>Hex value: 1A</p>

Modem Events

Modem events only occur on a per-entity basis. This means that an event will trigger an AutoResponse only for the modem channel on which the event occurs. The other modems on that NAC will not be affected.

These are the modem channel-level events that will trigger an AutoResponse script:

Event (Total Control Manager name)	MIB object
Incoming Connection Established	mdmArIncomConnectEstab 1.3.6.1.4.1.429.1.6.16.1.1.2
Outgoing Connection Established	mdmArOutgoConnectEstab 1.3.6.1.4.1.429.1.6.16.1.1.3
Incoming Connection Terminated	mdmArIncomConnectTerm 1.3.6.1.4.1.429.1.6.16.1.1.4
Outgoing Connection Terminated	mdmArOutgoConnectTerm 1.3.6.1.4.1.429.1.6.16.1.1.5
Connection Attempt Failed	mdmArConnectAttemptFail 1.3.6.1.4.1.429.1.6.16.1.1.6
Connection Time Limit Expired	mdmArConnectTimeExpire 1.3.6.1.4.1.429.1.6.16.1.1.7
Reset by DTE	mdmArResetByDte 1.3.6.1.4.1.429.1.6.16.1.1.8
DTE Transmit Idle	mdmArDteXmitIdle 1.3.6.1.4.1.429.1.6.16.1.1.9
Block Error Count at Threshold	mdmArBlersAtThresh 1.3.6.1.4.1.429.1.6.16.1.1.10
Fallback Count at Threshold	mdmArFbacksAtThresh 1.3.6.1.4.1.429.1.6.16.1.1.11
Dial Out Login Failure	mdmArDialOutLoginFail 1.3.6.1.4.1.429.1.6.16.1.1.12
Dial Out Restricted Number	mdmArDialOutRestrNum 1.3.6.1.4.1.429.1.6.16.1.1.13
Dial In Login Failure	mdmArDialInLoginFail 1.3.6.1.4.1.429.1.6.16.1.1.14
Dial Back Restricted Number	mdmArDialBackRestrNum 1.3.6.1.4.1.429.1.6.16.1.1.15
Dial Back Using Restricted Modem	mdmArDialBackRestModem 1.3.6.1.4.1.429.1.6.16.1.1.16
Login Attempt Limit Exceeded	mdmArLoginAttemptsExceed 1.3.6.1.4.1.429.1.6.16.1.1.17

Event (Total Control Manager name)	MIB object (continued)
User Blacklisted	mdmArUserBlacklisted 1.3.6.1.4.1.429.1.6.16.1.1.18
Attempted Login By Blacklisted User	mdmArAttmpLoginByBlistUsr 1.3.6.1.4.1.429.1.6.16.1.1.19
Response Attempt Limit Exceeded	mdmArRspAttemptLimExceed 1.3.6.1.4.1.429.1.6.16.1.1.20
Modem Watchdog Reset	mdmArWatchdog 1.3.6.1.4.1.429.1.6.16.1.1.21
Management Bus Failure	mdmArMgtBusFailure 1.3.6.1.4.1.429.1.6.16.1.1.22
DTR True	mdmArDtrTrue 1.3.6.1.4.1.429.1.6.16.1.1.23
DTR False	mdmArDtrFalse 1.3.6.1.4.1.429.1.6.16.1.1.24
Modem Ring No Answer	mdmArMdmRingNoAnswer 1.3.6.1.4.1.429.1.6.16.1.1.25
DTE Ring No Answer	mdmArDteRingNoAnswer 1.3.6.1.4.1.429.1.6.16.1.1.26
No Dial Tone	mdmArNoDialTone 1.3.6.1.4.1.429.1.6.16.1.1.27
No Loop Current Detected	mdmArNoLoopCurrent 1.3.6.1.4.1.429.1.6.16.1.1.28
Global Timer 1 Expired	mdmArTimer1 1.3.6.1.4.1.429.1.6.16.1.1.29
Global Timer 2 Expired	mdmArTimer2 1.3.6.1.4.1.429.1.6.16.1.1.30
Global Timer 3 Expired	mdmArTimer3 1.3.6.1.4.1.429.1.6.16.1.1.31
Global Timer 4 Expired	mdmArTimer4 1.3.6.1.4.1.429.1.6.16.1.1.32
Packet Bus Active	mdmArPacketBusActive 1.3.6.1.4.1.429.1.6.16.1.1.33
Packet Bus Lost	mdmArPacketBusLost 1.3.6.1.4.1.429.1.6.16.1.1.34

Entity-level responses

Responses act on a per-entity basis. Because a Quad modem contains four entities, you can write up to four different response scripts for a given chassis slot.

These responses apply to the Quad modem only. Twenty-eight channel-level events are defined. Total Control Manager does not use all of these responses; you may use the additional responses if you write your own scripts.

Example of a entity-level response

For example, the NMC detects a blacklisted user attempting to log into one of the modems in slot 2. The “User Blacklisted” event triggers a “Busy Out Analog Phone Line” AutoResponse script for that modem channel. The other modems on that NAC are not affected.

Possible entity-level responses

These are the possible modem channel-level responses to an event:

Response	ID	Description
Generate AutoResponse SNMP TRAP ID [N]	1	<p>A single user-defined value can be configured to be sent in this TRAP.</p> <p>The MS or some other higher level AutoResponse function can use this feature to convey information regarding the fact that a specific logic state had been detected (i.e. a specific point in the response script had been reached) for performing additional responses to better handle the specific event that occurred.</p> <p>Response Descriptor(s):</p> <p>User-defined AutoResponse SNMP TRAP ID (0–127)</p>
Delay Script Execution [N] Seconds	2	<p>This response causes a given script to suspend execution for the amount of seconds specified in the response descriptor.</p> <p>Response Descriptor(s):</p> <p>Time Delay in seconds (0–127)</p>
Terminate Script Execution	3	<p>This response immediately stops AutoResponse script execution. This provides a simple mechanism for disabling scripts without deleting their entire definition. For example, inserting this response at the front of an existing script will not cause this script to execute. Then, at some later date, the script can be re-enabled by removing this response. There are no restrictions on where in the script this response can be placed.</p>

Response	ID	Description
Continue If Test Passes	4	When a test response is executed, the result of the test(s) effects further execution of the script. If the test fails, the script execution continues at the response that immediately follows the test. If the test passes, the script parser searches linearly through the script until it finds the next "Continue if Test Passes" response. Once this response is found, the script execution continues at the response that immediately follows the "Continue if Test Passes" response. This provides the AutoResponse function with some primitive decision making capabilities without need for a full script language.
Configure Module from NVRAM	5	This response configures all parameters for the modem entity. The values to which the parameters will be configured are those that were last saved to the modem's NVRAM via the "Store Save To NVRAM" command. If this request is denied, it will be periodically retried until a modem accepts the request at some future time.
Configure Module From Modem Factory Default	6	This response configures all parameters for the modem entity. The values to which the parameters will be configured will be the modem factory defaults.
Test Modem	7	This response performs a complete modem self test for a particular modem entity where an event has occurred. This is not a complete "card" test, but rather a complete test of one modem on a given card. This test includes a RAM test and ROM CRC test. If any single test on the modem fails, the entire test fails. Success or failure of this test effects which response in the script runs next (refer to "Continue if Test Passes" response).
Test Analog NIC	8	This response performs a complete analog NIC self test for a particular modem entity where an event has occurred. This is a complete test for the entire NIC rather than only the portion of the NIC that corresponds to the modem entity. This test includes a RAM test and ROM CRC test. If any single test on the NIC fails, the entire test fails. Success or failure of this test effects which response in the script runs next (refer to "Continue if Test Passes" response).
Test Analog Phone Line	9	This response performs a complete analog phone line test for a particular modem entity where an event has occurred. This test includes a test for dial tone and loop current. If any single test on the modem fails, the entire test fails. Success or failure of this test effects which response in the script runs next (refer to "Continue if Test Passes" response).

Response	ID	Description
Restore Analog Phone Line	10	This response restores to service a previously busied-out phone line on the analog NIC associated with this modemchannel. This is the complement of the "Busy out Analog Phone Line" response. Hex value: 0A
Busy OutDS0 - T1 Slot [N] Span [N] Channel [N] from Service	11	Execution of this response instructs the T1-E1/PRI card to busy out one of its DS0 channels that corresponds to the modem for which the given event has occurred. Hex value: 0B
Restore DS0 - T1 Slot [N] Span [N] Channel [N] to Service	12	This is the complement of the "Busy Out DS0 - T1 Slot [N] Span [N] Channel [N] from Service" response. Execution of this response restores a DS0 channel that has been busied out. Hex value: 0C
Modem Software Reset	13	This response attempts to reset the processors associated with the modem entity through message exchange on the management bus. This relies on the given modem being at least partly functional to the extent that it can communicate over the management bus. It is conceivable that a modem may, in an extreme situation, not be able to complete the request. Hex value: 0D
Terminate Connection	14	Execution of this response will hang up the current call if one is in progress. Hex value: 0E
Busy-Out Module's Analog Phone Lines	15	Execution of this response instructs the analog NIC to busy out the analog line that corresponds to the modem for which the given event occurred. Hex value: 0F

Actions on DS1 Spans

These responses take action on specific DS1 spans. When writing a script, each response requires that three bytes/octets be specified:

- The first byte represents the response number.
- The second byte represents the slot number on which to take action on a span.
- The third byte represents the span that is to be busied.

For example, if you wrote a script to busy out the second DS1 on a dual trunk card in slot 1, the script would be: **16 01 02**. In this example, 16 represents the response, 01 represents the slot, and 02 represents the second DS1.

Responses 16 and 17 apply to both Channelized T1-E1 and PRI.
Responses 18–21 apply to PRI only.

Remove DS1 Slot [N] Span [N] from Service	16	Execution of this response instructs the T1-E1/PRI card at slot (N) to remove the DS1 span from service. Hex value: 10 Equivalent <i>uds1CmdFunction</i> enumeration: localOutOfService(6)
Restore DS1 Slot [N] Span [N] to Service	17	Execution of this response instructs the T1-E1/PRI card in slot (N) to restore the DS1 span to service. Hex value: 11 Equivalent <i>uds1CmdFunction</i> enumeration: inService(5)
Block Analog Calls on DS1 Slot [N] Span [N]	18	Execution of this response instructs the PRI card in slot (N) to block all analog calls on the DS1. Hex value: 12 Equivalent <i>uds1CmdFunction</i> enumeration: blockAnalogCalls(7)
Block Digital Calls on Ds1 Slot [N] Span [N]	19	Execution of this response instructs the PRI card in slot (N) to block all digital calls on the DS1. Hex value: 13 Equivalent <i>uds1CmdFunction</i> enumeration: blockDigitalCalls(8)
Block All Calls on Ds1 Slot [N] Span [N]	20	Execution of this response instructs the PRI card in slot (N) to block all calls on the DS1. Hex value: 14 Equivalent <i>uds1CmdFunction</i> enumeration: blockAllCalls(9)
Block No Calls on Ds1 Slot [N] Span [N]	21	Execution of this response instructs the PRI card in slot (N) to block no calls on the DS1. Hex value: 15 Equivalent <i>uds1CmdFunction</i> enumeration: blockNoCalls(10)

Actions on DS0s

These responses take action on specific DS0s. When writing a script, each response requires that four bytes/octets be specified:

- The first byte represents the response number (22–27).
- The second byte represents the slot number (1–16), which should only be a slot with trunk capability.
- The third byte represents the DS1 span line on the specified slot (1–2)
- The fourth byte represents the DS0 (1–30) on the specified DS1.

For example, if you wrote a script to remove the fourth DS0 on the second DS1 on a dual trunk card in slot 1, the script would be: **22 01 02 04**. In this example, 22 represents the response, 01 represents the slot, 02 represents the second DS1, and 04 represents the fourth DS0.

Responses 22 and 23 apply to both Channelized T1-E1 and PRI.
Responses 24–27 apply to PRI only.



These responses do not apply to HiPer DSP cards.

Remove DS0 Slot [N] Span [N] Channel [N] from Service	22	Execution of this response instructs the T1-E1/PRI card at slot (N) to remove the DS0 on span [N] and channel [N] from service. Hex value: 16
Restore DS0 Slot [N] Span [N] Channel [N] to Service	23	Execution of this response instructs the T1-E1/PRI card at slot (N) to restore the DS0 on span [N] and channel [N] to service. Hex value: 17
Block Analog Calls on DS0 Slot [N][Span [N] Channel [N]	24	Execution of this response instructs the PRI card at slot (N) to block all analog calls on the DS0 on span [N] and channel [N]. Hex value: 18
Block Digital Calls on DS0 Slot [N][Span [N] Channel [N]	25	Execution of this response instructs the PRI card at slot (N) to block all digital calls on the DS0 on span [N] and channel [N]. Hex value: 19
Block All Calls on DS0 Slot [N][Span [N] Channel [N]	26	Execution of this response instructs the PRI card at slot (N) to block all calls on the DS0 on span [N] and channel [N]. Hex value: 1A

Block No Calls on DS0 Slot [N][Span [N] Channel [N]	27	Execution of this response instructs the PRI card at slot (N) to block no calls on the DS0 on span [N] and channel [N]. Hex value: 1B
--	----	--

Off-Hook Response

This response requires only the response number to be specified in the script. It applies only to modem NACs and is limited to the modem to which the script is assigned.

Off Hook	28	This response causes the modem to enter an off-hook state, which is useful for modem channels connected via DS1 trunk cards (the other modem busy out response is only for an analog modem NIC). In some situations, this response is more effective than scripts that busy out a DS0, such as in PRI environments where busying out or removing a DS0 may cause a hunt group to return a busy signal. In this case, an off-hook allows the hunt group to skip to the next DS0. This affect is dependent upon the central office switch type and configuration. Hex value: 1C
----------	----	--

Gateway Events

Two gateway AutoResponse events are defined in the GW MIB. They apply to the NETServer and HiPer ARC. The scripts are triggered by *gwNetworkFailed* and *gwNetworkRestored* events. These scripts use identical responses as those that apply to modem events.



Do not use modem- and analog-specific responses because they may terminate the script.

Event (Total Control Manager name)	MIB object	Description
Gateway Network Failed	gwTeArNetFailed 1.3.6.1.4.1.429.1.18.1.1.5	This event occurs when the gateway loses communication with its defined security and accounting servers.
Gateway Network Restored	gwTeArNetRestored 1.3.6.1.4.1.429.1.18.1.1.6	This event occurs when the gateway restores communication with its defined security and accounting servers.

Configuring AutoResponse

Total Control Manager provides the easiest and the recommended method by which to create AutoResponse scripts. Refer to the Total Control Manager documentation for additional information and configuration procedures.

Although it is not recommended, you may use an SNMP browser to create scripts if the browser allows you to manually enter hexadecimal (hex) octets. Many browsers do not allow you to enter “raw” hex values, and cannot be used to create AutoResponse scripts.

Most browsers that do allow you to enter hex values require the values entered in left-to-right order. For example, to add the “Remove DSO from Service” response (22), encode the hex values as: **22 01 02 05**.

Using a browser to create your scripts is a labor-intensive process that requires precise hex-value scripting.

Trouble Clearing AutoResponse

Custom AutoResponse traps (Generate AutoResponse SNMP TRAP ID [N]) are a valuable tool for diagnosing script problems. Use them to verify if the script progresses to the desired responses by adding the traps at strategic positions with the script and assigning each a unique number.

For example, if the desired script for a module inserted event consists of these elements:

```
test module, generate trap 1, continue if test passes, test  
phone line, generate trap 2, continue if test passes,  
configure to programmed settings, generate trap 3, and so  
forth...
```

traps 1 and 2 will generate when the script completes a response. As each trap generates, you can see the specific progress of the script. If the script aborts when a test does not pass, the trap will not generate, and you likely will be able to debug the script with less effort required.

Software Download Errors

Scripts will terminate if a Software Download (SDL) starts while they are executing.

Simultaneous Event Occurrences

Since the AutoResponse function only operates on a single response script at a time, events that occur during execution of a prior event's script are queued and handled sequentially.

A maximum of ten outstanding event messages may be queued up for a single AutoResponse thread without loss of information. If an eleventh event occurs and the AutoResponse thread is not yet finished processing the current script, there will be not space in which to store this eleventh event and it will be ignored.

Script Errors

Scripts are stored as a string of octets. When this string is configured, it is not validated by the NMC. If a script is configured with non-existent response IDs, or the required response descriptors are not present, the error(s) will be detected upon execution of the script by the AutoResponse function. Upon detection of such an error, the AutoResponse function will ignore them.

Invalid hex number in the script

Scripts that contain invalid hex numbers will terminate. This type of error will typically only occur in a script created through an SNMP browser.

**Errors During
AutoResponse
Execution**

Errors that occur during response execution, such as management bus failures or refusal of a command, are ignored by the AutoResponse function. The response script continues execution on the next subsequent response. Response errors will never terminate the script.

INDEX

Numbers

128 Unacked LMIs 24-19
 1-8 RADIUS Accounting 15-16
 1-8 RADIUS Security 15-16

A

A Rootless Tree 24-10
 abnormalDisconnect(77) 24-22
 Abort Analog Destination over ISDN Network
 Connection 24-20
 abortAnlgDstOvrIsdn(72) 24-20
 Abstract Syntax Notation One (ASN.1) 2-12
 Access rights 17-10
 Accounting server
 acctSrvrGroupDegr(98) 15-15, 18-4, 23-26
 acctSrvrGroupNonOp(99) 15-15, 18-5, 23-26
 acctSrvrGroupOper(97) 15-15, 18-4, 23-25
 acctSrvrLoss(49) 15-14, 18-4, 23-14
 acctSrvrRestore(96) 15-14, 18-4, 23-25
 Additional USR MIBS 26, 16-1
 Agent address 22-10
 Agents, proxy 1-4
 Alarm Servers 22-3, 22-4
 alarmIndicationSignal (25) 6-14, 9-9, 23-8, 23-15
 alarmIndicationSignal (53) 9-9
 alarmIndicationSignalClear(53) 6-15, 23-8, 23-15
 Alarms 22-3
 ANIC MIB 3-2
 anicCfgDteRingNATrapEna 23-10
 anicCfgmdmRingNATrapEna 23-9
 applicationProcessorReset 23-24
 ARGUMENTS 22-9
 ARP 3-7
 Asymmetric 14-8
 AT Init String 21-1
 Attempted Login By Blacklisted User 25-11
 Authentication Failure 22-7
 Authentication scheme 1-8
 authenticationFail 15-14
 authenticationFailure(4) 22-7
 authorized access list 15-11, 15-13, 17-10, 17-11
 Auto Config on Card Initialization 17-4
 Auto response group 4-17
 Auto response slot table 4-18

Auto response timer table 4-18
 Autopass Failed 24-14
 autopass password 24-13
 autopassfailed(45) 24-14
 AutoResponse 3-2, 3-3, 4-8, 4-14, 4-15, 8-2, 11-5,
 11-6, 14-15, 14-17, 4-18, 14-2, 17-9, 22-2, 23-13,
 25-10
 AutoResponse(hdmArTable) 11-6
 Aux I/O
 AUX I/O commands 15-11
 Aux I/O Input 1 Closed 25-9
 Aux I/O Input 1 Open 25-9
 Aux I/O Input 2 Closed 25-9
 Aux I/O Input 2 Open 25-9
 Aux I/O Output 1 Closed 25-9
 Aux I/O Output 1 Open 25-9
 Aux I/O Output 2 Closed 25-9
 Aux I/O Output 2 Open 25-9
 Auxiliary I/O objects 15-6
 AuxIN1Open 23-31
 AuxOut1Closed 23-32
 AuxOut2Closed 23-32
 AuxOut2Open 23-32

B

Basic DNS configuration suggestions 15-19
 Bearer Incompatible 24-21
 bearerIncompatibility(75) 24-21, 24-22
 Bits in each nibble 4-5
 Blacklist Reject 24-28
 blacklisted 24-28
 BLER error 14-10
 blerCountAtThreshold(18) 11-6, 14-17, 23-5
 Blocking
 Block All Calls on DS0 Slot 25-16
 Block All Calls on Ds1 Slot 25-15
 Block Analog Calls on DS0 Slot 25-16
 Block Analog Calls on DS1 Slot 25-15
 Block Digital Calls on DS0 Slot 25-16
 Block Digital Calls on Ds1 Slot 25-15
 Block DS1 All Calls 25-8
 Block DS1 Analog Calls 25-8
 Block DS1 Digital Calls 25-8
 Block DS1 No Calls 25-9
 Block Error Count at Threshold 25-10
 blockAllCalls(7) 6-19
 blockAllCalls(9) 6-13, 25-8, 25-15
 blockAnalogCalls(5) 6-18
 blockAnalogCalls(7) 6-13, 25-8, 25-15
 blockDigitalCalls(6) 6-18
 blockDigitalCalls(8) 6-13, 25-8, 25-15
 blockNoCalls(10) 6-13, 25-9, 25-15
 blockNoCalls(8) 6-19
 Block No Calls on DS0 Slot 25-17

Block No Calls on Ds1 Slot 25-15
 Break Timeout 24-10
 bringUpDChannel(13) 6-14
 bulk access 6-16
 dsOBulkAccessTable 6-17
 idsOBulkAccessTable 6-19
 Bulk File
 bulkFileDownload(10) 13-3
 bulkFileDownload(11) 15-10, 17-2
 bulkFileUpload 10-3
 bulkFileUpload(10) 15-10, 17-2
 bulkFileUpload(9) 13-3
 Busy 24-9
 Busy out
 Busy Out DSO - T1 Slot 25-14
 Busy Out Phone Line 25-7
 Busy-Out Module's Analog Phone Lines 25-14
 byte 4-2, 4-3, 4-7, 25-14, 25-16

C

Call categories (hdr2CatMapTable) 9-3
 Call control 11-5, 12-1, 14-2, 14-8, 23-19, 23-20, 23-25, 23-28, 23-29, 23-30, 23-31
 Call Routing
 idt1CrTable 6-9
 t1hCrTable 9-15
 Call statistics 11-5, 14-2, 14-9, 15-17
 Call statistics groups 15-17
 Call Teardown 24-12
 callArriveEvent (76) 6-7, 9-10, 23-20
 callArriveEventHdsp (118) 9-10, 22-6, 23-29
 callBlacklisted(111) 24-28
 callConnectEvent(77) 6-7, 23-20
 callEvent(74) 6-7, 23-20
 callIgnore(3) 9-6
 callIgnore(6) 6-17
 Calling Party Category 9-3
 callTermFailedEvent(79) 6-7, 23-22
 callTerminateEventHdsp (119) 9-10, 22-6, 23-29
 callTermNormalEvent(78) 6-7, 9-10, 23-21
 Carrier Detect 15-3, 22-3, 23-7
 carrierLoss(4) 24-5
 CDMA/IWF traps 23-31, 23-32
 cdma42bisViolation 23-29
 cdmaAtViolation 23-29
 cdmaFrSrvViolation 23-29
 Cellular Support 14-2, 14-14, 14-16, 17-9
 Cellular V.42ETC 14-16
 CFM 15-10, 17-2
 CH1 17-8
 Change Password Message 19-2
 changedtoMainSrvsStat (90) 9-9, 23-25
 Changing a channel's configuration 11-3

channel 4-9, 11-2, 11-6, 12-1, 14-11, 14-12, 14-16, 16-4, 22-9, 23-19, 23-22, 23-30, 23-31, 23-32, 25-6, 25-14, 25-16, 25-17
 Channelized E1 6-3
 Channelized E1/R2 6-3
 Channelized T1 6-3, 6-4, 25-15, 25-16
 channelized T1 configuration 6-5
 chasAwarenessNotAvailable(102) 24-25
 Chassis
 chassis auto response scripts 4-8
 chassis auto response timers 4-8
 Chassis Awareness Unavailable from NMC 24-25
 Chassis command function 4-11
 chassis commands 4-8, 4-11
 chassis entities 4-8, 4-9, 4-10
 chassis environment 4-8, 4-11
 Chassis Environment, monitoring 4-11
 chassis management 1-4, 4-17, 17-7
 Chassis NACs 4-8
 Chassis Name 17-4
 Chassis NICs 4-8
 chassis slots 4-1, 4-2, 4-8
 Chassis status 4-2
 Chassis Trap MIB 22-6, 22-7
 Chassis-level events 25-2
 Chassis-level responses 25-5
 CHS MIB 25, 3-2, 4-1 (*see also*, *uchas*)
 CHS_TRAP MIB 3-2, 5-1, 22-7, 22-11
 Class 2 Fax Hangup Command 24-19
 class2FaxHangupCmd(66) 24-19
 closeAuxOutputPort1(14) 15-11, 17-2, 17-3, 23-32
 closeAuxOutputPort2(15) 15-11, 17-3, 23-32
 Code 3-5, 4-8, 4-16, 23-14, 23-22
 Cold Start 22-7
 coldStart(0) 22-7
 Command function 6-6, 6-13, 6-16, 6-18, 9-6, 9-8, 9-16, 13-2, 14-11
 Command Tables 3-5
 Commands ii, iv, 1-6, 4-8, 4-11, 6-16, 8-2, 13-2, 14-2, 14-4, 14-11, 15-1, 15-2, 17-1, 23-9
 Commands (dsOCmdTable) 6-16
 Commands (idsOCmdTable) 6-18
 Commands (t1hCmdTable) 9-15
 Commands (uds1CmdTable) 6-13
 Commands (usrds1CmdTable) 9-5, 9-8
 Commands, SNMP 1-6
 Communicating with SNMP 1-2
 Community string 1-7, 13-2, 15-7, 15-20, 17-9, 22-5, 22-7
 Community, snmp 1-8
 Comparing the HDM MIB to the MDM MIB 11-1

Configuration 3-7, 6-8, 6-16, 12-1, 13-2, 14-1, 14-4, 14-12, 14-16, 15-1, 15-8, 16-1, 16-2, 17-1, 23-6, 23-7, 23-14, 23-16, 23-17, 23-22, 23-23, 23-24, 23-26, 23-27
 ds0CfgTable 6-16
 ds1ConfigTable 6-10
 hdr2CfgTable 9-2
 ids0CfgTable 6-18
 t1hCfgTable 9-11
 uds1ConfigTable 6-12
 usrds1ConfigTable 9-5, 9-7
 Configuration Data 4-1, 6-5, 6-8, 6-10, 6-11, 6-16, 6-18, 9-2, 9-5, 9-6, 9-11, 10-1, 11-2, 12-1, 13-2, 14-1, 15-1
 Configuration Group 4-2, 15-2, 15-12
 Configuration notes 17-5
 Configuration options 22-4
 Configuring
 Configure Module From Modem Factory
 Default 25-13
 Configure Module From NMC Factory Default 25-6
 Configure Module from NMC NVRAM 25-6
 Configure Module from NVRAM 25-13
 Configuring a Template 11-2
 Configuring added cost features 17-8
 Configuring and saving settings on the NMC 15-7
 Configuring AutoResponse 25-17
 Configuring Hub Security Traps 19-4
 Configuring Logging Servers 18-3
 Configuring NMC Accounting and Event Logging 18-1
 Configuring NMC Dial-out 21-1
 Configuring NMC Hub Security 19-1
 Configuring NMC NTP Servers 20-1
 Configuring RADIUS DNS Servers 18-4
 Configuring Traps 22-4
 Connect Success Message 19-2
 connectAttemptFailure(13) 14-16, 23-4
 Connection 12-1, 14-8, 15-3, 15-15, 17-7, 23-3, 23-4, 23-11, 23-14, 23-23
 Connection Attempt Failed 25-10
 Connection Failure Limit 21-2
 Connection Time Limit Expired 23-4, 25-10
 connectTimerExpired(14) 11-6, 14-16, 23-4
 Constructor types 2-19
 Contacting 3Com v
 contCrcAlarm(59) 6-15, 9-9, 23-16
 contCrcAlarmClear(60) 6-15, 9-9, 23-16
 Continue If Test Passes 25-6, 25-13
 Counter 2-17, 2-20
 CPC 9-3
 ctConnectAttemptFailure 23-15
 ctIncomingConnectionEstablished 22-6, 23-15
 ctIncomingConnectionTerminated 22-6, 22-8, 23-15
 ctOutgoingConnectionEstablished 23-15

ctOutgoingConnectionTerminated 23-15
 Current interval (ds1CurrentTable) 6-10
 Current interval (uds1CurrentTable) 6-12
 Current table 10-2
 ds1CurrentTable 6-10
 uds1CurrentTable 6-12

D

Data compression 11-5, 14-2, 14-3
 Data Processing Generic Error 24-25
 dataProcessingGenericErr(99) 24-25
 Daylight Savings Time 15-2, 17-4
 dChanInService(70) 6-15, 9-10, 23-19
 dChanOutOfService(71) 6-15, 9-10, 23-19
 Default gateway IP address 17-3, 17-6
 Defined data types 2-16
 Delay Script Execution 25-5, 25-12
 Description 22-5, 22-8
 Destination IP 15-7, 22-5
 Dial Back
 Dial Back Attempt Limit 19-2
 Dial Back Delay 19-2
 Dial Back Name Prompt 19-1
 Dial Back Number Prompt 19-2
 Dial Back Password Prompt 19-1
 Dial Back Pending Prompt 19-2
 Dial Back Restrict Number Trap 19-4
 Dial Back Restricted Number 23-12, 25-10
 Dial Back Security 24-13
 Dial Back Using Restricted Modem 25-10
 Dial In Login Failure 23-11, 25-10
 Dial Security 24-11
 dialBackLink(43) 24-13
 dialBackRestrictedNum(40) 15-14, 19-4, 23-12
 Dialed Number Identification Service 6-8
 dialInCallDuration(46) 14-17, 23-13
 dialInLoginFail(38) 15-14, 23-11
 Dial out 14-5, 15-3, 15-4, 23-13
 dialOutCallDuration(45) 14-17, 23-13
 dialOutLoginFail(37) 15-14, 23-11
 dialOutRestrictedNum(39) 15-14, 23-12
 Dial Out Login Failure 23-11, 25-10
 Dial Out Restricted Number 25-10
 dialSecurity(34) 24-11
 disableAll(2) 4-13, 6-7, 6-14, 8-2, 9-3, 9-9, 9-19, 13-3, 14-14, 15-11, 22-4
 DISC Received 24-10
 disconnect(2) 6-18, 9-6
 disconnect(5) 6-17, 9-8
 disruptSelfTest(6) 6-6, 13-3
 DNIS 6-8
 DNIS-based resource access 6-8

DNS

- DNS RADIUS Security/Accounting Host Name 15-17
- DNS Resolver Feature 15-18
- DNS Server Select 18-4
- DNS Server's UDP Port 18-4
- dnsSrvrDegraded(95) 15-14, 18-4, 23-25
- dnsSrvrLoss(80) 15-14, 18-4, 23-23
- dnsSrvrRestore(94) 15-14, 18-4, 23-25
- DSO 3-3, 23-20, 23-21, 23-22, 23-24, 23-30
- DSO MIB 3-2, 6-3, 6-15
 - dsOBulkAccessTable 6-16, 6-17
 - dsOCfgTable 6-16
 - dsOCmdFunction 6-16
 - dsOCmdHardBusyOut(6) 9-6
 - dsOCmdInService(4) 9-6
 - dsOCmdSoftBusyOut(5) 9-6
 - dsOCmdTable 6-16
 - dsOsInService(72) 6-15, 9-9, 23-20
 - dsOsOutOfService(73) 6-15, 9-9, 23-20
 - dsOStatModem 6-16
 - dsOStatTable 6-16
 - dsOTeardown(37) 24-12, 24-21, 24-22, 24-23
- DS1 3-1, 23-6, 23-7, 23-8, 23-16, 23-22, 23-28
- DS1 MIB 3-1, 3-4, 6-3, 6-9
 - DS1 Near End Group 6-9
 - ds1ConfigTable 6-4, 6-10
 - ds1CSUIndex 6-12
 - ds1CurrentTable 6-4, 6-10, 9-2
 - ds1IntervalTable 6-4, 6-10, 9-2
 - ds1LineType 6-10
 - ds1TotalTable 6-4, 6-10, 9-2
- DSP
 - DSP Did not respond to disconnect request 24-25
 - DSP Did not respond to Keep Alive message 24-25
 - DSP Head Pointer Invalid 24-25
 - DSP Interrupt Timeout 24-18
 - DSP rebooted 24-24
 - DSP Tail Pointer Invalid 24-25
 - dspHeadPtrInvalid(98) 24-25
 - dspInterruptTimeout(64) 24-18
 - dspReboot(94) 24-24
 - dspReset 23-25
 - dspTailPtrInvalid(97) 24-25
- DSX1 MIB 9-1
 - dsx1ConfigTable 9-1
- DT1 3-3
- DT1 MIB 6-3, 6-4
 - dt1CfgTable 6-5
 - dt1CmdFunction 6-6
 - dt1CmdTable 6-5
 - dt1IdTable 6-5
 - dt1StatTable 6-5
 - dt1TrapEnaCallArriveEvent 6-7, 23-20
 - dt1TrapEnaCallConnEvent 6-7
 - dt1TrapEnaCallEvent 6-7
 - dt1TrapEnaCallFailEvent 6-7
 - dt1TrapEnaCallTermEvent 6-7, 23-21
 - dt1TrapEnaTable 6-5
 - dt1TrapEnaTxTmgSrcSwitch 6-7, 23-9

DTE 24-3

- DTE idle 14-10
- DTE interface 11-5, 14-2, 14-7
- DTE Interface (hdmDiTable) 11-5
- DTE Interface(mdmDiTable) 14-7
- DTE Ring No Answer 25-11
- DTE Transmit Idle 25-10
- dteRingNoAnswer(29) 11-6, 14-17, 23-10
- dteTransmitDataIdle(15) 11-6, 14-17, 23-4
- DTMF detection 14-9
- DTR 24-3
 - DTR dropped 24-3
 - DTR False 14-10, 25-11
 - DTR True 14-10, 25-11
 - dtrDrop(1) 24-3, 24-18
 - dtrFalse(17) 14-17, 23-4
 - dtrTrue(16) 14-10, 14-17, 22-8, 23-4
- Dual T1 identification 6-5
- Dual trunk card 6-3, 25-15, 25-16
- Dual Trunk Card MIBs 25, 6-3, 9-1

E

- E1/PRI 3-3, 6-4, 23-6, 23-7, 23-8, 23-9, 23-14, 23-15, 23-16, 23-19, 23-20, 23-21, 23-22, 23-28, 23-29
- E1/R2 6-3, 6-4
- EDGE MIB 3-3
- EdgeServer iv
- EdgeServer and CDMA/IWF traps 23-31
- EEPROM 15-5, 15-9, 17-2, 17-6
- egp 3-7
- Eighth Backup Logging Server 18-2, 18-3
- Eighth RADIUS Security Backup Server 19-3
- Enable
 - enable all 22-4
 - enable log 22-4
 - enable trap/disable all 22-4
 - enable traps 6-3, 16-1, 17-3
 - enableAll (4) 4-13, 6-7, 6-14, 8-2, 9-3, 9-9, 9-19, 13-3, 14-14, 15-12, 22-4
 - enableLog (3) 4-13, 6-7, 6-14, 8-2, 9-3, 9-9, 9-19, 13-3, 14-14, 15-12, 22-4
 - enableTrap (1) 4-13, 6-7, 6-14, 8-2, 9-3, 9-9, 9-19, 13-3, 14-14, 15-11, 22-4
- Enabling traps 11-4
- endTest(10) 14-12
- Enter blue alarm maintenance mode 23-32
- enterBlueAlmMaintMode(8) 9-8
- enterDChaDisConnMaintMode(6) 9-8

enterLoopback(3) 6-13
 Enterprise 22-10
 enterpriseSpecific(6) 22-7
 enterSpanToSpanLoopback(10) 6-6
 entityMgtBusFailure 4-13, 23-3
 entityWatchdogTimeout 4-13, 23-3
 Error Correction 11-5, 14-2, 14-8
 Errors during AutoResponse execution 25-19
 Escape code 24-4
 escapeSequence(2) 24-4
 Event configuration (usrds1EventCfgTable) 9-9
 Event counter 14-9
 Event logging server 18-2, 18-3
 Event Messages 22-10
 Event number 22-11
 event threshold 11-5, 14-2
 events 1-7, 3-2, 3-3, 4-13, 14-2, 14-9, 14-10,
 14-14, 14-15, 15-15, 18-1, 22-1, 22-2, 22-4, 25-3,
 25-17
 Events and responses 25-2
 Exit blue alarm maintenance mode 23-32
 exitBlueAlmMaintMode(9) 9-8
 exitDChaDisConnMaintMode(7) 9-8
 exitLoopback(4) 6-13
 exitSpanToSpanLoopback(11) 6-6
 Experimental MIBs 3-4
 Extra Stepup 24-10

F

failToTrain 24-12
 Fallback 14-10
 Fallback Count at Threshold 25-10
 fallbackCountAtThreshold(19) 11-6, 14-17, 23-5
 Fan
 Fan Failed 23-3, 25-3
 Fan Tray iv
 fanFailure(6) 4-13, 23-3
 Fatal Error 24-28
 Feature enable string 17-8
 Fifth Backup Logging Server 18-2, 18-3
 Fifth RADIUS Security Backup Server 19-3
 FILE MIB 7-1
 forceReceiverReframe(2) 6-13, 9-8
 forceTdmBusMastership(9) 6-6
 Formats 2-8
 Fourth Backup Logging Server 18-2, 18-3
 Fourth RADIUS Security Backup Server 19-3

G

Gateway 1-4, 3-3, 3-4, 3-7, 4-9, 8-1, 13-1, 14-15,
 16-1, 16-5, 23-10, 23-11, 23-13, 23-18, 23-22,
 23-24, 23-28, 23-29, 25-4, 25-17

Gateway Disconnect Command Received 24-18
 Gateway Network Failed 25-17
 Gateway Network Restored 25-17
 Gateway to reserved resource pool mapping
 assignment (idt1GwyRPA) 6-9
 Gauge 2-17, 2-21
 General span status 6-4, 9-1
 General Switch Telephone Network 24-11
 Generic trap ID 22-10
 Global Timer 1 Expired 25-3, 25-11
 Global Timer 2 Expired 25-3, 25-11
 Global Timer 3 Expired 25-3, 25-11
 Global Timer 4 Expired 25-3, 25-11
 GMT *see Greenwich Mean Time*
 GMTTimeNotSet(101) 24-25
 Greenwich Mean Time (GMT) 10-2, 15-2, 17-3
 Groups within a MIB 2-4
 GSTN 24-11
 GSTN Cleardown 24-11
 GW MIB 3-3, 8-1
 Gateway
 gwNetworkFailed 8-2, 23-18, 25-17
 gwNetworkRestored 8-2, 23-18, 25-17
 gwTeArNetFailed 8-2, 23-18, 25-17
 gwTeArNetRestored 8-2, 23-18, 25-17
 gwTegwlpAddress 8-2
 gwTegwNetworkFailed 8-2, 23-18
 gwTegwNetworkRestored 8-2, 23-18
 gwTeTable 8-1
 gwWanPortLinkActive 16-6, 23-11
 gwWanPortOutOfService 16-6, 23-11
 gwyTdmClockDown 23-28
 gwyTdmClockError 23-29
 gwyTdmClockUp 23-28
 Generate AutoResponse SNMP TRAP ID 25-5, 25-12,
 25-18

H

hardBusyOut(2) 6-16
 hardwareReset(4) 4-12
 HDM MIB 3-3, 11-1
 HDM MIB Tables 11-5
 hdmArTable 11-5
 hdmCcTable 11-5
 hdmDcTable 11-5
 hdmDiTable 11-5
 hdmDteNvramLock 11-5
 hdmEcTable 11-5
 hdmEtTable 11-5
 hdmIccTable 11-5
 hdmLiDialPause 11-4
 hdmLiTable 11-5
 hdmScLinkRateMax 11-4
 hdmScLinkRateMin 11-4

hdmScTable 11-5
 hdmTeBlerCountAtThresh 11-6, 23-5
 hdmTeConnLimitExpired 11-6
 hdmTeDteRingNoAns 11-6
 hdmTeDteXmitDataIdle 11-6
 hdmTeFallbkCountAtThresh 11-6, 23-5
 hdmTeInConnAttemptFail 11-6
 hdmTeInConnEstablished 11-6, 23-15
 hdmTeInConnTerminated 11-6, 23-15
 hdmTeOutConnAttemptFail 11-6
 hdmTeOutConnEstablished 23-15
 hdmTeOutConnEstablished 11-6
 hdmTeOutConnTerminated 11-6, 23-15
 hdmTePbActive 11-6, 23-10
 hdmTePBClockLost 11-6
 hdmTePBClockRestored 11-6
 hdmTePbLost 11-6, 23-11
 hdmTeTable 11-5
 HDR2 MIB 3-3, 9-1, 9-2
 hdr2CatMapTable 9-3
 hdr2CfgBlkToBlk 24-26
 hdr2CfgBNumNFn 24-27
 hdr2CfgBNumNFnd 24-27, 24-28
 hdr2CfgLineDirection 24-25, 24-27
 hdr2CfgOutCatAnalog 9-3
 hdr2CfgOutCatMaintenance 9-3
 hdr2CfgOutCatTest 9-3
 hdr2CfgTable 9-2
 hdr2CfgWrongNumber 24-27
 hdr2InCatMapCallType 24-27
 hdr2TeMultiFrame 9-4, 23-28
 hdr2TeMultiFrameClr 9-4, 23-28
 hdr2TeRemMultiFrame 9-4, 23-28
 hdr2TeRemMultiFrameClr 9-4, 23-28
 hdr2TeTable 9-3
 HDVI MIB 3-3
 Head pointer out of range 24-25
 High bit information 4-4
 HiPer ARC iv
 HiPer DSP iii, 3-1, 3-3, 4-9, 10-1, 11-1, 12-1, 14-1,
 14-16, 17-5, 23-5, 23-9, 23-10, 23-24, 23-25,
 23-29, 23-30, 23-31, 23-32, 25-3, 25-5, 25-16
 HiPer DSP Channelized T1/PRI 9-1
 HiPer DSP E1/R2 9-1
 HiPer TRAX traps 23-31
 HIST MIB 3-3, 10-1
 HST Speed Switch Timeout 24-19
 hstSpeedSwitchTimeout(67) 24-19
 Hub Security 14-2, 14-15, 15-1, 15-11, 17-9, 18-1,
 19-1
 Hub status LED 15-6
 HUB Temperature Out Of Range 25-3

I
 icmp 3-7, 15-13
 Identifying each object in a MIB 2-8
 idlePhoneLine(22) 14-14
 IDS0 MIB 3-3, 6-3, 6-17
 ids0BulkAccessTable 6-19
 ids0CfgTable 6-18
 ids0CmdFunction 6-18
 ids0CmdTable 6-18
 ids0StatTable 6-18
 IDT1 MIB 3-3, 6-3, 6-8
 idt1CfgTable 6-8, 6-9
 idt1CrTable 6-8, 6-9
 idt1GwyRPA 6-8, 6-9
 idt1mdmRPATable 6-8, 6-9
 idt1PITable 6-8, 6-9
 Illegal Command Code 24-10
 IMDM MIB 3-3, 12-1
 imdmCcTable 11-5, 12-1
 imdmLITable 12-1
 inactivityTimeout(5) 24-5
 Incoming Call
 Incoming Call - Call Blocked 24-24
 Incoming Call - Invalid bearer capability 24-23
 Incoming Call - Invalid Called Party 24-24
 Incoming Call - Invalid Calling Party 24-24
 Incoming Call - Invalid Channel ID 24-23
 Incoming Call - Invalid Progress Indication 24-24
 Incoming Call - Loop Start No Ring Off 24-24
 Incoming Call - Modem not available 24-23
 Incoming Calls 24-1
 incomingCallBlock(88) 24-24
 Incoming Connection
 Incoming Connection Established 25-10
 Incoming Connection Terminated 22-9, 25-10
 incomingConnectionEstablished(9) 11-6, 14-16,
 23-3, 23-15
 incomingConnectionTerminated 11-6, 14-16,
 23-3, 23-15
 incomingInvalidBearerCap(83) 24-23
 incomingInvalidCalledPty(87) 24-24
 incomingInvalidCallingPty(86) 24-24
 incomingInvalidChannelID(84) 24-23
 incomingInvalidProgInd(85) 24-24
 incomingLoopStNoRingOff(89) 24-24
 incomingModemNotAvailable(82) 24-23
 inconnectAttemptFailure 11-6, 14-17, 23-24
 Indexing 11-4
 Industry standards 1-6
 inService(3) 6-18, 9-8
 inService(5) 6-13, 25-8, 25-15
 INTEGER 2-15
 interfaces 3-4, 3-5, 3-7, 4-9, 23-6, 23-7
 Interval
 interval 3-1, 6-4, 10-1, 10-2, 15-13, 18-2, 18-3

Interval (uds1IntervalTable) 6-12
 Interval status 6-4, 9-2
 Interval tables 10-2
 Intervals (ds1IntervalTable) 6-10

Invalid
 Invalid Codeword 24-10
 Invalid Modem Select Message 19-2
 Invalid Speed 24-8
 invalidCauseValue 24-22
 invalidCauseValue(78) 24-22
 invalidSpeed(13) 24-8

IP
 IP 1-1, 1-5, 3-3, 3-7, 23-24
 IP Address Table 3-7
 IP address validation 15-13
 IP Route Table 3-7, 17-10, 18-3, 19-2
 IpAddress 2-17, 2-20

IPGW MIB 3-3, 13-1
 IPGW configuration (ipgwCfgTable) 13-2
 ipgwCfgTable 13-2
 ipgwCmdFunction 13-2
 ipgwCmdTable 13-2
 ipgwlinkDown 23-24
 ipgwLinkUp 23-24
 ipgwTrapEnaTable 13-2, 13-3
 ipgwTrapEnUiReset 13-3
 ipgwTrapEnUiResetx25gwTrapEnaUiReset
 (X.25) 23-11

ipInDiscards 15-13
 ISDN call control 11-5
 ISDN PRI/Dual trunk card 6-3
 IWFG MIB 3-3

K
 Keep alive 24-25
 keyAbort(17) 24-9
 Keypress Abort 24-9
 Known types 4-17

L
 LAN 17-7
 LAN IP Address 17-3, 17-6
 LAN Subnet Mask 17-3, 17-6
 lanLoopBack(8) 13-3

Last In First Out 10-2
 lclAnlgLpbk(13) 14-12
 lclDgtlLpbk(14) 14-12
 LD Received 24-7

LED 4-1, 4-2, 4-3, 4-5, 4-7, 4-10, 4-11, 15-6, 23-3,
 23-17
 LED byte strings 4-3, 4-7
 LED color 4-6
 LED state 4-5

Line 12-2
 Line Interface 11-5, 12-1, 12-2, 14-2, 14-3
 lineBusy(18) 24-9

Link
 Link Down 22-7
 Link Security 14-2, 14-15
 Link Start Received 24-16
 Link Up 22-7
 linkAbort(44) 24-13
 linkDisconnect 24-22
 linkDisconnectMsgReceived(11) 24-7
 linkDown(2) 22-7
 linkPassword(9) 24-6
 linkTrapDown 23-24
 linkTrapUp 23-24
 linkUp(3) 22-7

Load
 loadHwFlowDflt(23) 14-14
 loadMnp10CllulrDflt(25) 14-14
 loadSwFlowDflt(24) 14-14
 loadV42CllulrFxdDflt(27) 14-14
 loadV42CllulrMblDflt(26) 14-14

localOutOfService(4) 9-8
 localOutOfService(4) 6-18
 localOutOfService(6) 6-13, 25-7, 25-15

Log
 Log Group Selection 18-2, 18-3
 Log records 15-16
 Log Server's UDP Port 18-2, 18-3
 Logging Client TX Retry 18-2, 18-3
 Logging Server DNS Enable 18-2, 18-3
 Logging Server's Name 18-2, 18-3

Login
 Login Attempt Limit Exceeded 19-4, 23-13, 25-10
 Login Failed Message 19-2
 loginAttemptByBlacklistedUser 15-14, 23-12
 loginAttemptByBlacklistedUser(42) 19-4

Loop loss disconnect 24-7

Loopback
 loopback 13-3
 loopbackcleared (91) 9-9, 23-25
 loopbackTrap (92) 9-9, 23-25

loopLoss(36) 24-11, 24-23

Loss of Carrier 24-5

lossOfSignal (24) 6-14, 9-9, 22-5, 23-8, 23-14

lossOfSignalClear (52) 6-14, 9-9, 22-5, 23-8, 23-14

M
 Macros 2-14
 Managed networks, overview of 1-1
 Managed objects 1-5, 2-2
 Management Bus Failure 22-2, 25-11
 Management Information Base 1-5, 2-1
 Management, SNMP-based network 1-1

- managementCommand(15) 24-9
- Managers, snmp network 1-2
- Mapping 11-3, 14-2, 14-16
- mcCfgLog5SrvrAddr 15-16
- mcCfgLog6SrvrAddr 15-16
- mcCfgLogSecSrvrAddr 15-16
- MD5 Calculation 15-17, 18-2, 18-3
- MDM MIB 3-3, 11-4, 14-1
 - mdm105ResponderTest(121) 14-17, 23-31
 - mdmArAttmpLoginByBlistUsr 25-11
 - mdmArBlersAtThresh 25-10
 - mdmArConnectAttemptFail 25-10
 - mdmArConnectTimeExpire 25-10
 - mdmArDialBackRestModem 25-10
 - mdmArDialBackRestrNum 25-10
 - mdmArDialInLoginFail 25-10
 - mdmArDialOutLoginFail 25-10
 - mdmArDialOutRestrNum 25-10
 - mdmArDteRingNoAnswer 25-11
 - mdmArDteXmitIdle 25-10
 - mdmArDtrFalse 25-11
 - mdmArDtrTrue 25-11
 - mdmArFbacksAtThresh 14-11, 25-10
 - mdmArIncomConnectEstab 25-10
 - mdmArIncomConnectTerm 25-10
 - mdmArLoginAttemptsExceed 25-10
 - mdmArmdmRingNoAnswer 25-11
 - mdmArMgtBusFailure 25-11
 - mdmArNoDialTone 25-11
 - mdmArNoLoopCurrent 25-11
 - mdmArOutgoConnectEstab 25-10
 - mdmArOutgoConnectTerm 25-10
 - mdmArPacketBusActive 25-11
 - mdmArPacketBusLost 25-11
 - mdmArResetByDte 25-10
 - mdmArResetByDteReset by DTE 25-10
 - mdmArRspAttemptLimExceed 25-11
 - mdmArTable 14-2
 - mdmArTimer1 25-11
 - mdmArTimer2 25-11
 - mdmArTimer3 25-11
 - mdmArTimer4 25-11
 - mdmArUserBlacklisted 25-11
 - mdmArWatchdog 25-11
 - mdmCc2100AnswerTone 14-9
 - mdmCcDataOverVoice 14-9
 - mdmCcDtmTerminationTone 14-8
 - mdmCcTable 14-2, 14-8
 - mdmCdFunction 9-17, 14-11
 - mdmCdTable 14-2
 - mdmCeTable 14-2, 14-16
 - mdmCsCallDuration 22-9
 - mdmCsCallRefNum 22-9
 - mdmCsCollectedDtmfDigits 14-9
 - mdmCsConnectFailReason 14-9
 - mdmCsDisconnectReason 14-9, 22-9
 - mdmCsTable 14-2, 14-9
 - mdmDcDataCompression 14-3
 - mdmDcTable 14-2
 - mdmDiAtString 14-7
 - mdmDiTable 14-2, 14-7
 - mdmEcTable 14-2
 - mdmEtBlerThresh 14-10, 14-17
 - mdmEtConnTimeLimit 14-10
 - mdmEtDteldleThresh 14-10, 14-17
 - mdmEtDtrFalseThresh 14-10, 14-17
 - mdmEtDtrTrueThresh 14-10, 14-17
 - mdmEtFallbackThresh 14-10, 14-11, 14-17
 - mdmEtTable 14-2, 14-10
 - mdmEvTable 14-2
 - mdmHistCurrentTable 10-2
 - mdmHistIntervalTable 10-2
 - mdmHsTable 14-2, 14-15
 - mdmIDTable 14-2
 - mdmLiDialPause 11-4
 - mdmLiTable 14-2, 14-3
 - mdmLoginAttemptLimExceeded 15-14, 23-13
 - mdmLoginAttemptLimExceeded(44) 19-4
 - mdmLsTable 14-2, 14-15
 - mdmMaChangeIndicator 9-16, 11-3, 11-4, 14-16
 - mdmMaChannelConfig 11-4, 14-16
 - mdmMaTable 14-2, 14-16
 - mdmNacHistCurrentTable 10-2
 - mdmNacHistIntervalTable 10-2
 - mdmScLinkRateAmpU 11-4
 - mdmScLinkRateSelect 11-4
 - mdmScTable 14-2, 14-8
 - mdmStsPbClock 14-16
 - mdmStsTable 14-2, 14-16
 - mdmTe105ResponderTest 14-17, 23-31
 - mdmTeBlerCountAtThresh 14-17
 - mdmTeBlerCountAtThreshhdmTeBlerCountAtThresh (DSP) 23-5
 - mdmTeConnAttemptFailure 14-16
 - mdmTeConnLimitExpired 14-16
 - mdmTeConnTimeLimithdmTeConnTimeLimit (DSP) 23-4
 - mdmTeDialInCallDur 14-17, 23-13
 - mdmTeDialOutCallDur 14-17, 23-13
 - mdmTeDteRingNoAns 14-17
 - mdmTeDteRingNoAnsweranicCfgDteRingNATrapEna (analog) 23-10
 - mdmTeDteXmitDataIdle 14-17
 - mdmTeDteXmitDataIdlehdmTeDteXmitDataIdle (DSP) 23-4
 - mdmTeDtrFalse 14-17, 23-4
 - mdmTeDtrTrue 14-17, 23-4
 - mdmTeFallbkCountAtThresh 14-17
 - mdmTeFallbkCountAtThreshhdmTeFallbkCountAtThresh (DSP) 23-5

- mdmTelnConnAttemptFail 14-17, 23-24
- mdmTelnConnEstablished 14-16
- mdmTelnConnEstablishedhdmTelnConnEstablished (DSP) 23-15
- mdmTelnConnTerminated 14-16
- mdmTelnConnTerminatedhdmTelnConnTerminated (DSP) 23-15
- mdmTeNoDialTone 14-17, 23-5
- mdmTeNoLoopCurrent 14-17, 23-6
- mdmTeOutConnAttemptFail 14-17, 23-24
- mdmTeOutConnEstablished 14-16
- mdmTeOutConnEstablishedHdmTeOutConnEstablished (DSP) 23-15
- mdmTeOutConnTerminated 14-16
- mdmTeOutConnTerminatedhdmTeOutConnTerminated (DSP) 23-15
- mdmTePbActive 14-17
- mdmTePbClockLossEvent 14-17
- mdmTePbClockRestoreEvent 14-17
- mdmTePbLost 14-17
- mdmTeResetByDTE 14-17, 23-9
- mdmTeTable 14-2, 14-14
- mdmTfOdB1004FarNearLvl 14-7
- mdmTfOdB1004FarNearSts 14-7
- mdmTfOdB1004NearFarLvl 14-7
- mdmTfOdB1004NearFarSts 14-7
- mdmTf1004FarNearLvl 14-5
- mdmTf1004FarNearSts 14-6
- mdmTf1004NearFarLvl 14-5
- mdmTf1004NearFarSts 14-6
- mdmTf2804FarNearLvl 14-5
- mdmTf2804FarNearSts 14-6
- mdmTf2804NearFarLvl 14-6
- mdmTf2804NearFarSts 14-7
- mdmTf404FarNearLvl 14-5
- mdmTf404FarNearSts 14-6
- mdmTf404NearFarLvl 14-5
- mdmTf404NearFarSts 14-6
- mdmTfATG 14-4
- mdmTfCmsgFarNearLvl 14-6
- mdmTfCmsgFarNearSts 14-7
- mdmTfCmsgNearFarLvl 14-6
- mdmTfCmsgNearFarSts 14-7
- mdmTfCnotchFarNearLvl 14-6
- mdmTfCnotchFarNearSts 14-7
- mdmTfCnotchNearFarLvl 14-6
- mdmTfCnotchNearFarSts 14-7
- mdmTfRspndrTable 14-5
- mdmTfRxFreq 14-5
- mdmTfSigNoiseFarNearLvl 14-6
- mdmTfSigNoiseFarNearSts 14-7
- mdmTfSigNoiseNearFarLvl 14-6
- mdmTfSigNoiseNearFarSts 14-7
- mdmTfTable 14-2, 14-3
- mdmTfTest 14-3
- mdmTfTestTime 14-3
- mdmTfToneTable 14-4
- mdmTfTxAmpl 14-4
- mdmTfTxFreq 14-4
- mdmTfV54 14-3
- mdmTfV54Errors 14-4
- mdmDTR True 14-10
- MIB
 - MIB compatibility 15-6
 - MIB object, identifying an instance 2-22
 - MIB object, structure of 2-10
 - MIB objects, syntax of 2-12
 - MIB overview 7, 2-1
 - MIB Tables 9-2, 9-5, 9-7, 9-11, 11-4, 14-2, 15-14
 - MIB trees 2-2
- MIB2 3-1, 23-24
 - MIB2, supported groups 3-6
- MNP
 - MNP Incompatibility 24-6
 - MNP Protocol Violation 24-18
 - MNP10 14-16
 - mnpIncompatible(6) 24-6
 - mnpProtocolViolation(65) 24-18
- Mode Incompatible 24-13
- Modem
 - modelIncompatible(41) 24-13
 - Modem Attempt Limit 19-3
 - Modem commands 14-11
 - Modem Disconnect and Connect Fail Reasons 24-1
 - Modem Events 25-10
 - Modem identification 14-2
 - Modem Mapping 14-16
 - Modem Ring No Answer 23-9, 25-11
 - Modem Select Prompt 19-2
 - Modem Software Reset 25-14
 - Modem Status 14-16
 - Modem traps 22-6
 - Modem Watchdog Reset 25-11
 - modemResetByDte 9-19, 14-17, 23-9
 - modemRingNoAnswer 23-9, 23-10
 - ModemTests 14-3
- Module
 - Module Inserted 25-4, 25-18
 - Module Non operational 25-4
 - Module Reinitialized 25-4
 - Module Removed 25-4
 - Module Watchdog Timeout 25-5
 - moduleInserted(1) 4-13, 23-2
 - moduleRemoved 4-13, 23-2
- Monitoring the power supply 4-10
- Monitoring with SNMP, network 1-2
- Multiple objects 6-8
- Multiple Traps 22-6

N

- NAC-level commands (dt1CmdTable) 6-5
- nacUserInterfaceReset (34) 13-3, 16-5, 23-11
- NETServer iv
- NETServer syslog 24-3
- Network Management Card iv
- Network management, SNMP-based 1-1
- Network managers, SNMP 1-2
- Network monitoring with SNMP 1-2
- Network time protocol 15-1, 15-13, 20-1
- Networks, overview of managed 1-1
- New Password Message 19-2
- NFAS 6-14
- nibble 4-2, 4-3, 4-4
- NIC byte strings 4-7
- NMC MIB
 - NMC authorized access group 15-12
 - NMC Command Group 15-7
 - NMC configuration commands 15-8
 - NMC Configuration Group 15-2
 - NMC Dial-out 21-1
 - NMC Host Names 15-19
 - NMC Hub Security Group 15-11
 - NMC LED Display 17-4
 - NMC MIB 26, 15-1
 - NMC network time protocol (NTP) group 15-13
 - NMC Network Time Protocol group 15-13
 - NMC NIC LAN Interface 17-6
 - NMC Status Group 15-5
 - NMC Status group 15-5
 - NMC Trap Enable Group 15-11
 - NMC Trap Group 15-6
 - NMC traps 22-6
 - NMC User Interface Configuration Group 15-12
 - NMC.MIB 3-3
 - nmcArCustomTrap 23-13, 25-2
 - nmcAuthAccDescr 15-12, 17-10
 - nmcAuthACclpAddr 15-12, 15-13, 17-10
 - nmcAuthAccNetMask 15-12, 15-13, 17-10
 - nmcAuthAccTable 15-12, 15-14
 - nmcAuxIn1Sts 15-6
 - nmcAuxIn2Sts 15-6
 - nmcAuxOut1Sts 15-6
 - nmcAuxOut2Sts 15-6
 - nmcCfg 15-1
 - nmcCfgAtString 15-3, 15-4, 21-1
 - nmcCfgAuthFailTrapEnable 15-14, 17-3, 18-4
 - nmcCfgLog3SrvrAddr 15-16, 18-2
 - nmcCfgLog4SrvrAddr 15-16, 18-2
 - nmcCfgLog5SrvrAddr 18-2
 - nmcCfgLog6SrvrAddr 18-2
 - nmcCfgLog7SrvrAddr 15-16, 18-2
 - nmcCfgLog8SrvrAddr 15-16, 18-2
 - nmcCfgLogCallStatGrpSel 15-17, 18-2
 - nmcCfgLogDnsEna 15-16, 15-17, 18-2, 18-3
 - nmcCfgLogMD5Calc 15-17, 18-2, 18-3
 - nmcCfgLogPriSrvrAddr 15-16, 18-2
 - nmcCfgLogRetryCnt 18-2, 18-3
 - nmcCfgLogSecSrvrAddr 18-2
 - nmcCfgLogSrvrSelect 15-17
 - nmcCfgLogSrvrName 15-16, 18-2, 18-3
 - nmcCfgLogSrvrSelect 18-2
 - nmcCfgLogStatusInterval 15-18, 18-2, 18-3
 - nmcCfgLogUdpPortNum 18-2
 - nmcCfgNumFailBefSuspend 21-2
 - nmcCfgNumFailBeSuspend 15-3
 - nmcCfgNumWanRetries 15-3, 21-2
 - nmcCfgSessionIDNewFmt 17-5
 - nmcCfgSystemDate 15-2, 17-3
 - nmcCfgSystemTime 15-2, 17-3
 - nmcCfgTFTPTimeout 15-4, 17-4
 - nmcCfgWanDialOutPhoneNum 15-3, 15-4, 21-1
 - nmcCfgWanRetryPause 15-3, 21-2
 - nmcCfgWanRetrySuspendTime 15-3, 21-2
- nmcCmd 15-1
 - nmcCmdCode 15-6, 15-11
 - nmcCmdFunction 15-8, 15-12, 17-1, 17-6, 23-31
 - nmcCmdFunction
 - closeAuxOutputPort1(14) 23-32
 - closeAuxOutputPort2(15) 23-32
 - nmcCmdFunction
 - openAuxOutputPort1(12) 23-32
 - openAuxOutputPort2(13) 23-32
 - nmcCmdmgtStationId 15-6
 - nmcCmdReqlId 15-6
 - nmcCmdResult 15-11
- nmcDaySavingTime 15-2, 17-4
- nmcGmtime 10-2, 15-2, 17-3, 22-9
- nmcHs 15-1
 - nmcHsChangePasswordMsg 19-2
 - nmcHsConnectSuccessMsg 19-2
 - nmcHsDialBackAttempts 19-2
 - nmcHsDialBackDelay 19-2
 - nmcHsDialBackNamePrompt 19-1
 - nmcHsDialBackPendPrompt 19-2
 - nmcHsDialBackPhonePrompt 19-2
 - nmcHsDialBackPsswdPrompt 19-1
 - nmcHsDialInOutNamePrompt 19-1
 - nmcHsDialInOutPsswdPrompt 19-1
 - nmcHsDiPasswdEnaDis 19-3
 - nmcHsInvalidmdmSelecMsg 19-2
 - nmcHsLoginFailedMsg 19-2
 - nmcHsmdmAttemptLimit 19-3
 - nmcHsmdmSelectPrompt 19-2
 - nmcHsNewPasswordPrompt 19-2
 - nmcHsNoMdnsAvailMsg 19-2

- nmcHsPhoneRestrictPrompt 19-2
- nmcHsPromptRspAttempts 19-2
- nmcHsPromptRspEchoEna 19-2
- nmcHsPromptRspTimeout 19-2
- nmcHsSecondarySrvrAddr 15-16, 19-3
- nmcHsSecurity3SrvrAddr 15-16, 19-3
- nmcHsSecurity4SrvrAddr 15-17, 19-3
- nmcHsSecurity5SrvrAddr 15-17, 19-3
- nmcHsSecurity6SrvrAddr 15-17, 19-3
- nmcHsSecurity7SrvrAddr 15-17, 19-3
- nmcHsSecurity8SrvrAddr 15-17, 19-3
- nmcHsSecuritySrvrAddr 15-16, 19-2
- nmcHsSecuritySrvrDnsEna 15-17, 19-3
- nmcHsSecuritySrvrName 15-17, 19-3
- nmcHsSecuritySrvrPort 19-2
- nmcHsSecuritySrvrRetries 19-3
- nmcHsSecurityStatusInt 15-18, 19-3
- nmcHsServerSelect 15-17
- nmcHsServerUnavailable 19-3
- nmcNtp 15-1
 - nmcNtpOperationalMode 15-14, 20-1
 - nmcNtpSrvrPrimAddr 20-1
 - nmcNtpSrvrSecdAddr 20-1
 - nmcNtpSyncInterval 20-1
- nmcPowerUpAutoCfgEnable 4-13, 6-7, 9-16, 9-17, 14-15, 15-7, 15-12, 17-4, 17-5
- nmcStat 15-1
 - nmcStatCompSwVer 15-6
 - nmcStatEventId 22-11
 - nmcStatPowerUpTstFailBMap 15-5
 - nmcStatRedLed 15-6
 - nmcStatTestResult 15-5, 15-6, 15-9
- nmcTe 15-1
 - nmcTeDialBackRestrictNum 15-14, 19-4, 23-12
 - nmcTeDialInLogFail 15-14
 - nmcTeDialInLoginFail 23-11
 - nmcTeDialOutLogFail 23-11
 - nmcTeDialOutLoginFail 15-14
 - nmcTeDialOutRestrictNum 15-14, 23-12
 - nmcTeDnsSrvrDegraded 15-14, 18-4, 23-25
 - nmcTeDnsSrvrLoss 15-14, 18-4, 23-23
 - nmcTeDnsSrvrRestore 15-14, 18-4, 23-25
 - nmcTeLoginAttemptLimExceed 23-13
 - nmcTeLoginAttmpLimExceed 15-14, 19-4
 - nmcTeLogSrvrGroupDegr 15-15, 18-4, 23-26
 - nmcTeLogSrvrGroupNonOp 15-15, 18-5, 23-26
 - nmcTeLogSrvrGroupOper 15-15, 18-4, 23-25
 - nmcTeLogSrvrLoss 15-14, 18-4, 23-14
 - nmcTeLogSrvrRestore 15-14, 18-4, 23-25
 - nmcTeNtpSrvrDegraded 15-14, 18-4, 20-2, 23-24
 - nmcTeNtpSrvrLoss 15-14, 18-4, 20-2, 23-23
 - nmcTeNtpSrvrRestore 15-14, 18-4, 20-2, 23-23
 - nmcTePbClockFail 15-14
 - nmcTePbClockSwitch 15-14
 - nmcTeRespAttemptLimExceeded 15-14, 19-4, 23-12
 - nmcTeSecSrvrGroupDegr 15-15, 18-5, 23-27
 - nmcTeSecSrvrGroupNonOp 15-15, 18-5
 - nmcTeSecSrvrGroupOper 15-15, 18-5, 23-27
 - nmcTeSecSrvrGrpNonOp 23-28
 - nmcTeSecSrvrLoss 15-14, 19-4, 23-17
 - nmcTeSecSrvrRestore 15-15, 18-5, 23-26
 - nmcTeSinglePbClockFail 15-14
 - nmcTeSinglePbClockFailure 23-17
 - nmcTeUserBlacklist 15-14, 19-4, 23-12
 - nmcTeUserBlacklistLogIn 15-14, 19-4, 23-12
- nmcTimezone 15-2, 17-3
- nmcTrap 15-1
 - nmcTrapDestCommunity 15-7
 - nmcTrapDestDescr 15-7
 - nmcTrapDestIP 15-7
 - nmcTrapDestTable 15-7, 15-14
 - nmcTrapSequenceNumber 22-9, 22-10
- nmcUICfg 15-1, 17-2
 - nmcUICfgDefaultGwylP 17-3, 17-6
 - nmcUICfgInactiveTime 15-3, 17-7, 21-1
 - nmcUICfgLanIfEnable 17-6
 - nmcUICfgLanIPAddr 17-3, 17-6
 - nmcUICfgLanSubnetMask 17-3, 17-6
 - nmcUICfgPassword 17-7, 17-9
 - nmcUICfgPrivateString 17-9
 - nmcUICfgPublicString 17-9
 - nmcUICfgRouteEnable 17-7
 - nmcUICfgUISlipCfg 17-8
 - nmcUICfgWan2IpAddr 17-8
 - nmcUICfgWan2SubnetMask 17-8
 - nmcUICfgWanIPAddr 17-3, 17-6
 - nmcUICfgWanSubnetMask 17-3, 17-6
- No Answer 23-9, 24-9
- No Carrier 22-3, 24-9
- No cause value received or invalid cause value 24-22
- No Connection Online 24-11
- No Dialtone 24-9, 25-11
- No Fail 24-11
- No Loop Current Detected 25-11
- No Modems Available Message 19-2
- No Prompting in Non-ARQ 24-13
- No Prompting in Sync 24-13
- noAnswer(19) 24-9
- noAnswerTone(21) 24-9
- noCarrier(22) 24-9, 24-23
- noCommand(1) 4-12, 6-6, 6-13, 6-16, 6-18, 9-6, 9-8, 9-16, 13-2, 14-11, 15-8, 17-1
- noDialTone(16) 14-17, 23-5, 24-9
- noDSPRespToDisc(96) 24-25

noDSPRespToKA(95) 24-25
 noLoopCurrent(12) 23-6, 24-7
 noLoopCurrent(21) 14-17
 nonArgMode(40) 24-13
 nonDisruptSelfTest(5) 6-6, 13-3, 15-9, 17-2
 none(32) 24-11
 noPromptInSync(39) 24-13
 noPromptInNonARQ(42) 24-13
 Normal
 Normal User Call Clear 24-20
 Normal, Unspecified 24-20
 normalUnspecified(74) 24-20, 24-22
 normalUserCallClear(73) 24-20, 24-22
 ntp *see also Network Time Protocol.*
 ntpSrvrDgrConn(85) 15-14, 18-4, 20-2, 23-24
 ntpSrvrLossConn(81) 15-14, 18-4, 20-2
 ntpSrvrRestConn(82) 15-14, 20-2, 23-23
 Null 2-16
 NVRAM 11-3, 11-4, 13-2, 14-12, 15-2, 15-7, 17-5,
 17-11, 25-1, 25-2, 25-6

O

OBJECT IDENTIFIER 2-16
 Object Identifier 2-20
 Objects, managed 1-5
 Obsolete 3-7, 23-3, 23-4, 23-15, 23-20, 23-22
 OCTET STRING 2-15
 Octet String 2-20
 offHook(6) 14-12, 25-17
 OID 22-10
 On
 On Authentication Failure 18-4
 On DNS Server Lost 18-4
 On DNS Server Restored 18-4
 On Logging Server Group Degraded 18-4
 On Logging Server Group Non-Operational 18-5
 On Logging Server Group Operational 18-4
 On Logging Server Lost 18-4
 On Logging Server Restored 18-4
 On NTP Server Lost 18-4, 20-2
 On NTP Server Restored 18-4, 20-2
 On Primary DNS Server Failed 18-4
 On Primary NTP Server Failed 18-4, 20-2
 On RADIUS Server Group Degraded 18-5
 On RADIUS Server Group Non-Operational 18-5
 On RADIUS Server Group Operational 18-5
 onHook(7) 14-12
 OOF 6-13
 OOS 6-18
 Opaque 2-18
 openAuxOutputPort1(12) 15-11, 17-2, 23-32
 openAuxOutputPort2(13) 15-11, 17-2, 23-32
 Operational Mode 20-1
 Out of Frame 6-13

outconnectAttemptFailure(87) 11-6, 14-17, 23-24
 Outgoing
 Outgoing Call - E&M Wink Timeout 24-24
 Outgoing Call - E&M Wink Too Short 24-24
 Outgoing Call - No Channel Available 24-24
 Outgoing Call - Telco Disconnect 24-24
 Outgoing Calls 24-1
 outgoingConnectionEstablished(10) 11-6, 14-16,
 23-3, 23-15, 25-10
 outgoingConnectionTerminated(12) 11-6, 14-16,
 23-4, 23-15
 outgoingEMWinkTimeout(91) 24-24
 outgoingEMWinkTooShort(92) 24-24
 outgoingNoChannelAvail(93) 24-24
 outgoingTelcoDisconnect(90) 24-24
 Out-of-service 6-18
 Overview of managed networks 1-1

P

Packet Bus
 PACKET BUS - ACK Wait Timeout 24-17
 PACKET BUS - Bad Frame 24-17
 PACKET BUS - Clock Missing 24-16
 PACKET BUS - Generic Error 24-14
 PACKET BUS - Out of Sequence Frame 24-16
 PACKET BUS - Receive Bus Timeout 24-15
 PACKET BUS - Receive Message Buffer
 Overflow 24-17
 PACKET BUS - Receive Overflow, RNR Failed 24-17
 PACKET BUS - Received ACK Sequence
 Error 24-17
 PACKET BUS - Received LS while Link Up 24-16
 PACKET BUS - Transmit Bus Timeout 24-15
 PACKET BUS - Transmit Master Timeout 24-16
 Packet Bus Active 25-11
 Packet Bus Datagram MIB 16-2
 PACKET BUS LINK ERROR 24-14, 24-15
 Packet Bus Lost 25-11
 Password 17-7, 17-9
 Password Prompt 15-11, 19-1, 19-3
 Pause between Retries(sec) 21-2
 PB MIB 3-3, 16-1
 pbAckWaitTimeout(58) 24-17
 pbBadFrame(57) 24-17
 pbCfgTable 16-1
 pbClockMissing(54) 24-16
 PBDG MIB 16-2
 PBDG.MIB 3-3
 pbdgCfgClockLossEvent 23-19
 pbdgCfgClockRestoreEvent 23-19
 pbdgCfgTable 16-2
 pbdgDatagramTable 16-2
 pbGenericError(46) 24-14
 pbLinkErrRxTAL 24-15

- pbLinkErrRxTAL(52) 24-15
- pbLinkErrTxPreAck(47) 24-14
- pbLinkErrTxTAL(51) 24-15
- pbLinkErrTxTardyACK(48) 24-14
- pbOutOfSequenceFrame(56) 24-16
- pbReceiveBusTimeout(50) 24-15
- pbReceivedAckSequenceErr(59) 24-17
- pbReceivedLsWhileLinkUp(55) 24-16
- pbReceiveMsgBufOvrflw(61) 24-17
- pbRecieveOvrflwRNRFailed(60) 24-17
- pbSessionTable 16-1
- pbTransmitBusTimeout(49) 24-15, 24-16
- pbTransmitMasterTimeout(53) 24-16
- pbTrapEnaPktBusCongest 16-2, 23-10
- pbTrapEnaPktBusSessLost 16-2, 23-11
- pbTrapEnaSessActive 16-2, 23-10
- pbTrapEnaSessionError 16-2, 23-13
- pbTrapEnaSessionInactive 16-2, 23-11
- pbTrapEnaTable 16-1
- PDU 22-9
 - PDU type 22-10
- phyStateChng (61) 6-15, 9-9, 23-16
- PIAFs 17-9
- pktBusClkLost 14-17, 23-19
- pktBusClkRetsore(69) 14-17, 23-19
- pktBusClockFailure(65) 15-14, 23-17
- pktBusClockLost(68) 11-6
- pktBusClockRestore(69) 11-6
- pktBusClockSwitch 15-14, 23-17
- pktBusSessActive (30) 11-6, 14-17, 16-2, 23-10
- pktBusSessCongestion 16-2, 23-10
- pktBusSessError 16-2, 23-13
- pktBusSessInactive (33) 16-2, 23-11
- pktBusSessLost(32) 11-6, 14-17, 16-2, 23-11
- Portions of the OID 2-9
- Power Supply 4-10
 - power supply performance 4-8
 - power supply slots 4-8
- PRI Dialout Request Timeout 24-20
- priDialoutRqTimeout(71) 24-20
- Primary
 - Primary DNS Server Retries 18-4
 - Primary DNS Server's IP Address 18-4
 - Primary Log Server IP Address 18-2, 18-3
 - Primary NTP Server's IP Address 20-1
 - Primary Security Server IP Address 19-2
- Primitive data types (simple types) 2-15
- Private community string 17-9
- Prompting Not Enabled 24-12
- promptNotEnabled(38) 24-12
- Proprietary MIBs 3-2
- Protocol Error 24-22
- protocolErrorEvent(76) 24-22
- Proxy agents 1-4
- PSU

- PSU Failed 25-3
- PSU Voltage Out of Range 25-3
- psuFailure 4-13, 23-2
- psuIncompatible(75) 4-13, 23-20
- psuWarning(3) 4-13, 23-2
- Public community string 17-9

R

- R2
 - R2 Channel Blocked 24-26
 - R2 Congestion 24-27
 - R2 DNIS not Found 24-27
 - R2 Glare 24-26
 - R2 Line Direction 24-25
 - R2 Outcall Call Blocked 24-26
 - R2ChannelBlockedByNetwork(104) 24-26
 - R2DNISNotFound(107) 24-27
 - R2DSPFatalError(110) 24-28
 - R2Glare(105) 24-26
 - R2InvalidChannelDirection(103) 24-25
 - R2OutgoingCallBlocked(106) 24-26
 - R2SigCauseCongestion(108) 24-27
 - R2SigCauseUnallocNumber(109) 24-28
- RADIUS 4-13, 8-2, 13-3, 14-14, 15-4, 15-11, 15-15, 15-16, 17-5, 17-8, 18-1, 18-4, 18-5, 19-1, 22-1, 23-14, 23-18, 23-25, 23-26, 23-27, 23-28, 24-13, 24-14
 - RADIUS Security & Accounting Host Name 15-18
 - RADIUS Security Server DNS 19-3
 - RADIUS Security Server Host Name 19-3
- rcvdGatewayDiscCmd(62) 24-18
- rcvTone(9) 14-12
- RDL Timer Expired 24-19
- RDS0 MIB 3-3, 9-1
- RDS1 MIB 3-3, 6-11, 9-1, 9-6
 - rds1EnterMaintMode Enter D-channel disconnect maintenance mode 23-32
 - rds1EvDchSwitchOverEnd 23-30
 - rds1EvDchSwitchOverEnd (123) 9-10
 - rds1EvDchSwitchOverFailure 23-30, 23-31
 - rds1EvDchSwitchOverFailure (124) 9-10
 - rds1EvDchSwitchOverStart 23-30, 23-31
 - rds1EvDchSwitchOverStart (122) 9-10
 - rds1ExitMaintMode Exit D-channel disconnect maintenance mode 23-32
- Receive TAL 24-15
- Receiver gain objects 9-7
- Recovery Mechanism uses Status-Server Request 15-18
- redAlarm (23) 6-14, 9-9, 22-5, 23-7, 23-14
- redAlarmClear (51) 6-14, 9-9, 22-5, 23-7, 23-14
- redAlarmOverride(11) 6-14
- Refresh DNS Cache Suppression 15-19

refreshCfg1Chans(25) 9-18
 refreshCfg2Chans(26) 9-18
 refreshCfg3Chans(27) 9-19
 refreshCfg4Chans(28) 9-19
 Registration ID 4-1, 5-1, 6-5, 6-8, 6-10, 6-11, 6-16,
 6-17, 7-2, 8-1, 9-2, 9-5, 9-6, 9-11, 10-1, 11-2,
 12-1, 13-1, 14-1, 15-1, 16-1, 16-2, 16-3, 16-4,
 16-5, 16-6
 Remove DS0 Slot 25-16
 Remove DS1 Slot 25-15
 Remote On Hook Timeout 24-23
 remoteAccessDenied(35) 24-11
 remotePassword(8) 24-6
 remotHungUpDuringTraining(80) 24-23
 Remove DS1 from Service 25-7
 Remove Module from Service 25-6
 removeFromService(2) 4-12
 Reserved resource pool (idt1PITable) 6-9
 Reset by DTE 23-9, 25-10
 Reset Module 25-7
 resetToHiPrioTimingSrc(8) 6-6
 Resource pool mapping assignment
 (idt1mdmRPATable) 6-9
 Resources Unavailable 24-23
 resourceUnavailable(79) 24-23
 Response Attempt Limit 19-2, 19-4, 23-12, 25-11
 Response Echo Enable 19-2
 Response Timeout 19-2
 responseAttemptLimExceeded(43) 15-14, 19-4, 23-12
 Restore
 Restore Analog Phone Line 25-14
 Restore DS1 to Service 25-8
 Restore Module to Service 25-6
 Restore Phone Line 25-7
 restore(4) 6-17
 Restore DS0 - T1 Slot 25-14
 Restore DS0 Slot 25-16
 Restore DS1 Slot 25-15
 restoreCfg1FromNvram(21) 9-18
 restorecfg1todeflt(13) 9-17
 restoreCfg2FromNvram(22) 9-18
 restorecfg2todeflt(14) 9-18
 restoreCfg3FromNvram(23) 9-18
 restorecfg3todeflt(15) 9-18
 restoreCfg4FromNvram(24) 9-18
 restorecfg4todeflt(16) 9-18
 restoreDefaultUIPassword(12) 6-6, 9-17
 restoreFromDefault(4) 6-6, 13-2
 restoreFromDefaults(3) 15-9, 17-2
 restoreFromDflt(4) 14-11
 restoreFromNVRAM(3) 6-6, 13-2
 restoreFromNvram(4) 15-9, 17-2
 restoreFromNvram(5) 14-12
 restoremmdmFromDefault(4) 9-16
 restoremmdmNvram(10) 9-17

restoreNmcFromDefaults(8) 15-10, 17-2
 restoreNmcFromNvram(9) 15-10, 17-2
 restoreT1E1AndmdmDefaults(2) 9-16
 restoreT1E1AndmdmNvram(8) 9-17
 restoreT1E1FromDefault(3) 9-16
 restoreT1E1Nvram(9) 9-17
 restoreToService(3) 4-12
 Restricted Number Prompt 19-2
 Retransmit Limit 24-6
 retransmitLimit(10) 24-6
 Retries Suspension Interval(sec) 21-2
 RFC 1232 6-3, 6-4
 RFC 1406 3-1, 6-3, 6-4, 6-9, 9-1
 RMDM MIB 3-3
 rmdmTeRetrainEv 23-29
 rmdmTeSpeedShiftEv 23-29
 rmmieRetrainEvent 23-29
 rmmieSpeedShiftEvent 23-29
 rmtDgtLpbk(15) 14-13
 Route Traffic between LAN & WAN 17-7
 Routing 17-7
 rspndrTest102(12) 14-12
 rspndrTest105(11) 14-12

S

SABME 24-10
 SABME Timeout 24-10
 Save
 saveCfg1toNvram(17) 9-18
 saveCfg2toNvram(18) 9-18
 saveCfg3toNvram(19) 9-18
 saveCfg4toNvram(20) 9-18
 savemdmNvram(7) 9-17
 saveT1E1AndmdmNvram(5) 9-16
 saveT1E1Nvram(6) 9-17
 saveToNvram(2) 6-6, 13-2, 15-8, 17-1
 saveUiParamsToEEPROM(7) 15-10, 17-2
 Scalar objects 2-24
 Scheme, authentication 1-8
 ScLinkRateAmpU 14-8
 Script errors 25-19
 Second SLIP port 17-8
 Second SLIP Port IP Address 17-8
 Second SLIP Port Subnet Mask 17-8
 Secondary server
 Secondary DNS Server's IP Address 18-4
 Secondary Log Server IP Address 18-2, 18-3
 Secondary NTP Server's IP Address 20-1
 Secondary Security Server IP Address 19-3
 Seconds 25-5, 25-12
 Security iii, 1-7, 14-2, 14-15, 15-1, 15-4, 15-15,
 17-1, 22-2, 23-11, 23-12, 23-13, 23-14, 23-17,
 23-26, 23-27, 23-28, 25-17
 Security Abort 24-13

- Security and Accounting iv
 - Security Server 15-16, 19-2, 19-3
 - Security Server Lost 19-4
 - Security Server Retries 19-3
 - Security Server UDP Port 19-2
 - Security Server Unavailable 19-3
 - securityServerLoss 23-17
 - securityServerRestore 23-26
 - securitySrvrGrpDegr 23-27
 - securitySrvrGrpDegr(102) 15-15, 18-5
 - securitySrvrGrpNonOp 23-28
 - securitySrvrGrpNonOp(103) 15-15, 18-5
 - securitySrvrGrpOper 23-27
 - securitySrvrGrpOper(101) 15-15, 18-5
 - securitySrvrLoss(62) 15-14, 19-4
 - securitySrvrRestore(100) 15-15, 18-5
 - selfTest(16) 14-13
 - SEQUENCE 2-19, 2-21
 - Sequence number 22-10
 - SEQUENCE OF 2-19, 2-21
 - serial number 17-8
 - Server traps 18-4
 - Server trouble clearing 15-18
 - Session ID/Call reference number for RADIUS logging 15-4
 - Set Asynchronous Balance Mode Extended 24-10
 - Setting server traps 20-2
 - Seventh Backup Logging Server 18-2, 18-3
 - Seventh RADIUS Security Backup Server 19-3
 - SEVERITY 22-9
 - Signal Converter 11-5, 14-2, 14-8
 - Simultaneous event occurrences 25-18
 - singlePktBusClockFailure(63) 15-14, 23-17
 - Sixth Backup Logging Server 18-2, 18-3
 - Sixth RADIUS Security Backup Server 19-3
 - SLIP port 21-1
 - Slots and Entities 3-4
 - sndTone(8) 14-12
 - SNMP i, 1-4, 3-2, 3-5, 3-8, 4-9, 5-1, 5-2, 13-1, 13-2, 14-14, 15-2, 15-15, 17-1, 17-3, 17-9, 19-1, 20-1, 22-1, 23-13
 - SNMP agents 1-4
 - SNMP browser 25-1
 - SNMP commands 1-6
 - SNMP community 1-8
 - SNMP forwarding 15-19
 - SNMP Generic Traps 22-7, 22-10
 - SNMP network managers 1-2
 - SNMP overview 1-1
 - SNMP variables 2-22
 - SNMP, communicating with 1-2
 - SNMP, network monitoring with 1-2
 - SNMP-based network management 1-1
 - SNTP 15-13, 20-1
 - softBusyOut(3) 6-16
 - Software 23-3, 23-4
 - Software Download iv, 4-12, 4-14, 15-4, 17-4, 25-4, 25-18
 - Software Download (SDL) errors 25-18
 - software version iii, 3-2
 - softwareDownload(5) 4-12, 4-14, 4-16
 - softwareDownload2(6) 4-12
 - Software Reset
 - softwareReset(11) 9-17
 - softwareReset(2) 14-11
 - softwareReset(6) 15-10, 17-2
 - softwareReset(7) 6-6, 13-3 25-16, 25-17
 - Span 25-14, 25-15, 25-16
 - Span Status 6-4, 9-1
 - Specific trap ID 22-10
 - Standard MIBs 3-1
 - Standards, industry 1-6
 - Status 3-1, 3-3, 4-2, 4-10, 6-5, 6-16, 14-2, 14-6, 14-12, 14-16, 15-1, 15-5, 15-6, 15-11, 22-1, 23-3, 23-5, 23-8, 23-14, 23-17, 23-19
 - dsOStatTable 6-16
 - dt1StatTable 6-5
 - idsOStatTable 6-18
 - uds1StatTable 6-13
 - usrds1StatTable 9-5, 9-7
 - status, chassis 4-2, 4-7
 - Status-Server Poll Interval 15-18
 - Status-Server Request Interval 18-2, 18-3, 19-3
 - storeToNvram(3) 14-11
 - Subnet mask 15-12, 17-3, 17-6, 17-8, 17-10
 - SubTypes 2-18
 - SUMMARY 22-9
 - Synchronization Interval (sec) 20-1
 - sysObjectId 5-2, 22-10
 - System 15-13, 23-13, 23-29, 23-31, 23-32
 - System Time 15-2, 17-3
 - sysUpTime 22-10
-
- ## T
- T1 Card iv
 - T1 Glare 24-20
 - T1, E1, and PRI-ISDN traps 22-6
 - T1/PRI 3-4, 6-4
 - t1Glare(70) 24-20
 - T1H 17-5
 - T1H MIB 3-3, 9-1, 9-10, 11-4
 - t1hCfgLogCallStatGrpSel 9-11, 9-12
 - t1hCfgmdmRoutingMethod 9-11
 - t1hCfgTable 9-11
 - t1hCmdTable 9-15
 - t1hCrTable 9-15
 - t1hTeResetByDTE 9-19, 23-9
 - t1hTeTable 9-19

- Table objects 2-23
 - Tables 4-8, 5-1, 6-5, 6-8, 6-10, 6-11, 6-16, 6-18, 7-2, 8-1, 10-2, 11-4, 12-1, 13-2, 16-1, 16-2, 16-3, 16-4, 16-5, 16-6
 - Tables within a group 2-4
 - Tail pointer out of range 24-25
 - takeDownDChannel(12) 6-14
 - TCP/IP 1-1, 3-6, 3-8
 - telcoAbnormalResp (93) 9-9, 23-25
 - Template 10-3, 11-1, 11-3, 11-4, 11-6, 12-1, 14-14, 14-16, 17-5, 23-5
 - tempWarning 4-13, 23-2
 - Terminate Connection 25-14
 - Terminate Script Execution 25-6, 25-12
 - Test
 - Test Analog NIC 25-13
 - Test Analog Phone Line 25-13
 - test function 14-2
 - Test Modem 25-13
 - Test Module 25-7, 25-18
 - Test result bitmaps 15-5
 - testNVRAM(19) 14-13
 - testRam(17) 14-13
 - testRom(18) 14-13
 - TFTP timeout 15-4, 17-4
 - Third Backup Logging Server 18-2, 18-3
 - Third RADIUS Security Backup Server 19-3
 - Time Delay 25-12
 - Time stamp 22-10
 - Time To Live 15-19
 - Time Zone 15-2, 17-3
 - Timer 14-3, 15-4, 16-4, 23-10, 25-2
 - timerExpired(69) 24-19
 - Timeslot Un-available 24-25
 - timeslotUnavailable(100) 24-25
 - TimeTicks 2-18, 2-21
 - Token Passing Timeout 24-18
 - tokenPassingTimeout(63) 24-18
 - tooManyUnacked(68) 24-19
 - TOTAL CONTROL 24-9
 - Total Control Documentation Library CD-ROM iv
 - Total Control Manager 17-7, 23-2
 - Total Control MIBs 3-1
 - Total intervals
 - ds1TotalTable 6-10
 - uds1TotalTable 6-12
 - Training Timeout 24-23
 - trainingTimeout(81) 24-23
 - Transient Events 22-5
 - Transmission 1-2, 3-7, 3-8, 23-8
 - Transmit
 - Transmit Pre ACK 24-14
 - Transmit TAL 24-15
 - Transmit Tardy ACK 24-14
 - transmitTimingSourceSwitch 23-9
 - transmitTimingSourceSwitch(26) 6-7
 - transparentTest(7) 6-17
 - Trap destination table 15-6, 22-5
 - Trap destinations 15-1, 22-5
 - Trap Enable and Disable 22-4
 - Trap enables 4-13, 6-4, 6-5, 8-1, 8-2, 9-2, 10-3, 11-5, 13-2, 13-3, 14-2, 14-14, 14-15, 15-1, 16-2, 16-3, 16-5, 16-6
 - Trap enables (dt1TrapEnaTable) 6-7
 - Trap enables (hdr2TeTable) 9-3
 - Trap enables (t1hTeTable) 9-19
 - Trap enables (uds1TrapEnaTable) 6-14
 - Trap Listing 23-1
 - Trap Reference 23-1
 - Trap Table 23-1
 - Traps 4-13, 5-1, 6-7, 6-9, 6-11, 6-14, 6-17, 6-19, 7-2, 8-2, 9-4, 9-9, 9-19, 10-3, 11-6, 12-2, 13-3, 14-16, 15-14, 16-2, 16-3, 16-5, 16-6, 22-1, 22-2
 - Trouble clearing trap packets 22-11
 - Trouble clearing AutoResponse 25-18
 - TYPE 22-8
 - Types 2-13, 2-15
 - Types of Chassis Traps 22-6
-
- ## U
- UCHAS (CHS MIB)
 - uchasArFanFailed 25-3
 - uchasArHubTempOutOfRange 25-3
 - uchasArModuleInserted 25-4
 - uchasArModuleNonoper 25-4
 - uchasArModuleReinit 25-4
 - uchasArModuleRemoved 25-4
 - uchasArModuleWatchdog 25-5
 - uchasArPsuFailed 25-3
 - uchasArPsuVoltOutOfRange 25-3
 - uchasArSlotTable 4-8
 - uchasArTimer1 25-3
 - uchasArTimer2 25-3
 - uchasArTimer3 25-3
 - uchasArTimer4 25-3
 - uchasArTimerTable 4-8
 - uchasCmdTable 4-8, 4-11, 4-14
 - uchasDescr 17-4
 - uchasDisplayName 17-4
 - uchasEntityIndex 22-9
 - uchasEntityMgtBusFailTrapEna 4-13, 23-3
 - uchasEntityTable 4-8, 4-9, 4-10
 - uchasEntityWatchdogTrapEna 4-13, 23-3
 - uchasEnvironTable 4-11
 - uchasFanFailureTrapEna 4-13, 23-3
 - uchasFrontPanelLedColor 4-1, 4-2, 4-5, 4-7
 - uchasFrontPanelLedColor2 4-2, 4-7
 - uchasFrontPanelLedStates 4-2
 - uchasFrontPanelLedStates2 4-7

- uchasModuleInserted TrapEna 4-13
- uchasModuleInsertedTrapEna 23-2
- uchasModuleRemovedTrapEna 4-13, 23-2
- uchasNicStates 4-2, 4-7
- uchasPowerSupplyOutTable 4-8, 4-10, 4-11
- uchasPowerSupplyTable 4-8, 4-10, 4-11
- uchasPSUFailureTrapEna 4-13, 23-2
- uchasPsuIncompatible 4-13, 23-20
- uchasPSUWarningTrapEna 4-13, 23-2
- uchasSlotIndex 22-9
- uchasSlotStatFeEna 17-9
- uchasSlotTable 4-8, 4-9, 4-10, 25-4
- uchasTempWarningTrapEna 4-13, 23-2
- udp 1-1, 3-8, 18-3, 18-4
- UDS1 MIB 3-3, 6-3, 6-11
 - uds1CmdFunction 6-13, 25-8, 25-9, 25-15
 - uds1CmdTable 6-13
 - uds1ConfigTable 6-12
 - uds1CSUIndex 6-12
 - uds1CurrentTable 6-12
 - uds1EvtelcoAbnormalResp 23-25
 - uds1IntervalTable 6-12
 - uds1MultiFrame 23-28
 - uds1MultiFrame(104) 6-15, 9-4
 - uds1MultiFrameClr 23-28
 - uds1MultiFrameClr(106) 6-15, 9-4
 - uds1RemoteMultiFrame 23-28
 - uds1RemoteMultiFrame(105) 6-15, 9-4
 - uds1RemoteMultiFrameClr 23-28
 - uds1RemoteMultiFrameClr(107) 6-15, 9-4
 - uds1StatTable 6-4, 6-13
 - uds1TotalTable 6-12
 - uds1TrapEnaAlarmIndSignal
 - uds1TrapEnaAlrmIndSgnlClr
 - uds1TrapEnaContCrcAlrm
 - uds1TrapEnaContCrcAlrmClr
 - uds1TrapEnaDchanInSrvc 23-19
 - uds1TrapEnaDchanOutOfSrvc 23-19
 - uds1TrapEnaDs0InSrvc 6-15, 23-20
 - uds1TrapEnaDs0OutOfSrvc 6-15, 23-20
 - uds1TrapEnaLossOfSgnlClr
 - uds1TrapEnaLossOfSignal
 - uds1TrapEnaMultiFrame 6-15, 23-28
 - uds1TrapEnaMultiFrmClr 6-15, 23-28
 - uds1TrapEnaPhysStateChng
 - uds1TrapEnaRedAlarm
 - uds1TrapEnaRedAlarmClr
 - uds1TrapEnaRemMultiFrame 6-15, 23-28
 - uds1TrapEnaRemMultiFrmClr 6-15, 23-28
 - uds1TrapEnaTable 6-4, 6-14
- UI password protection 17-9
- UI Port Inactivity Time (minutes) 17-7
- ULPB MIB 3-4
 - ulpbAdmnTable 16-3
 - ulpbOperTable 16-3
 - ulpbStatTable 16-3
- Unable to Retrain 24-8
- unableToRetrain(14) 24-8
- Unallocated Number 24-28
- undefined(7) 24-6
- Understanding the Structure of a SNMP Trap PDU 22-9
- Unique Call Reference Number 17-5
- User
 - User Blacklist Login Trap 19-4
 - User Blacklist Trap 19-4
 - User Blacklisted 23-12, 25-11
 - User Interface 1-3, 4-17, 15-1, 15-12, 15-13, 16-5, 17-1, 17-3, 17-6, 17-7, 17-11, 23-11, 25-1
 - User interface configuration 15-1, 15-12, 17-6
 - User Name Prompt 19-1
 - User Password Prompt 19-1
 - User Server Selection Disabled 15-17
 - userBlacklisted(41) 15-14, 19-4, 23-12
- USR Enterprise 3-1
- USR node 5-2
- USR/3Com MIBs 25
- USRDS0 (RDS0 MIB)
 - usrds0CmdFunction 9-6
 - usrds0CmdTable 9-5
 - usrds0ConfigTable 9-5
 - usrds0StatTable 9-5
- USRDS1 (RDS1 MIB)
 - usrds1CfgNFASSpanType 23-31
 - usrds1CfgRcvGain 9-7
 - usrds1CfgSigGroupType 23-31
 - usrds1CmdFunction 9-8
 - usrds1CmdFunctionenterBlueAlmMaintMode(8) 23-32
 - usrds1CmdFunctionenterDChaDisConnMaintMode(6) 23-32
 - usrds1CmdFunctionexitBlueAlmMaintMode(9) 23-32
 - usrds1CmdFunctionexitDChaDisConnMaintMode(7) 23-32
 - usrds1CmdTable 9-8
 - usrds1ConfigTable 9-7
 - usrds1DchanOutOfSrvc 6-15, 9-10, 23-19
 - usrds1dsplnCallFailedEvent (120) 9-10
 - usrDs1dspOutCallFailedEvent (121) 9-10
 - usrds1EventAlarmIndSignal 6-14, 9-9, 23-8
 - usrds1EventAlrmIndSgnlClr 6-15, 9-9, 23-15
 - usrds1EventCallArrive 9-10, 23-20, 23-29
 - usrds1EventCallTerm 9-10, 23-21, 23-29
 - usrds1EventCfgTable 9-2
 - usrds1EventContCrcAlrm 6-15, 9-9, 23-16
 - usrds1EventContCrcAlrmClr 6-15, 9-9, 23-16
 - usrds1EventDchanInSrvc 9-10
 - usrds1EventDchanInSrvc 6-15, 23-19
 - usrds1EventDs0InConnFail 9-10, 23-29, 23-30

usrds1EventDs0InSrv 9-9, 23-20
 usrds1EventDs0OutConnFail 9-10, 23-29, 23-30
 usrds1EventDs0OutOfSrv 9-9, 23-20
 usrds1EventDs0ServStateMt 9-9, 23-25
 usrds1EventloopBack 9-9, 23-25
 usrds1EventloopBackClear 9-9, 23-25
 usrds1EventLossOfSgnlClr 6-14, 9-9, 23-14
 usrds1EventLossOfSignal 6-14, 9-9, 23-8
 usrds1EventNfasDchSwEnd 9-10, 23-30
 usrds1EventNfasDchSwfail 9-10, 23-31
 usrds1EventNfasDchSwStart 9-10, 23-30
 usrds1EventPhysStateChng 6-15, 9-9, 23-16
 usrds1EventRedAlarm 6-14, 9-9, 23-7
 usrds1EventRedAlarmClr 6-14, 9-9, 23-14
 usrds1EventYellowAlarm 6-14, 9-9, 23-6
 usrds1EventYellowAlarmClr 6-14, 9-9, 23-14
 usrds1EvtntelcoAbnormalRsp 9-9
 usrDs1HdsplnCallFailedEvent 23-30
 usrDs1HdspOutCallFailedEvent 23-30
 usrDs1InCallFailedEvent 23-29
 usrds1InCallFailedEvent (114) 9-10
 usrDs1OutCallFailedEvent 23-29
 usrds1OutCallFailedEvent (115) 9-10
 usrds1StatReceiverGain 9-7
 usrds1StatTable 9-1, 9-7
 UX25 MIB 3-4, 16-3
 ux25AdmnChannelTable 16-4
 ux25AdmnClassTable 16-4
 ux25AdmnPacketTable 16-4
 ux25AdmnSubscriberTable 16-4
 ux25AdmnTimerTable 16-4
 ux25OperChannelTable 16-4
 ux25OperClassTable 16-4
 ux25OperPacketTable 16-4
 ux25OperSubscriberTable 16-4
 ux25OperTimerTable 16-4
 ux25StatTable 16-5

V

v32Cleardown(33) 24-11
 V.42
 v42BadSetup(28) 24-10
 v42BreakTimeout(25) 24-10
 v42DisconnectCmd(26) 24-10
 v42ExtraStepup 24-10
 v42IdExchangeFail(27) 24-10
 v42InvalidCodeWord(29) 24-10
 v42InvalidCommand(31) 24-10
 v42SabmeTimeout(24) 24-10
 v42StringToLong(30) 24-10
 V.54
 v54LclAnlgLpbk(20) 14-13
 v54RmtDgtlLpbk(21) 14-13
 Values 2-14

Variable bindings 22-10
 voice(20) 24-9

W

WAN 15-2, 16-6
 WAN Connect Number 21-1
 WAN Dial Out Attempt Limit 21-2
 WAN IP Address 17-3, 17-6
 WAN port 21-1
 WAN Subnet Mask 17-3, 17-6
 WAN2 17-8
 Warm Start 22-7
 warmStart(1) 22-7
 Website v

X

X.25 Card v
 X.25 Gateway MIB 16-5
 X.25 Interface MIB 16-3
 X.25 WAN MIB(X25w MIB) 16-6
 x2/V.90 17-9
 X25g MIB 3-4, 16-5
 x25gwldTable 16-5
 x25gwTrapEnaUiReset 16-5, 23-11
 X25w MIB 3-4, 16-6
 x25wanAdmnTable 16-6
 x25wanOperTable 16-6
 x25wanStatsTable 16-6
 x25wanTrapEnaLinkActive 16-6, 23-11
 x25wanTrapEnaOutOfSvc 16-6, 23-11
 x25wanTrapEnaTable 16-6
 XID Timeout 24-10

Y

yellowAlarm (22) 6-14, 9-9, 22-5, 23-6, 23-14
 yellowAlarmClear (50) 6-14, 9-9, 22-5, 23-6, 23-14



3Com Corporation
5400 Bayfront Plaza
P.O. Box 58145
Santa Clara, CA
95052-8145

©1999
3Com Corporation
All rights reserved
Printed in the U.S.A.

Part No. 1.024.1661-00