



AREA CODE CHANGE

Please note that the area code for Paradyne Corporation in Largo, Florida has changed from 813 to 727.

For any Paradyne telephone number that appears in this manual with an 813 area code, dial 727 instead.

**COMSPHERE
6700 SERIES
NETWORK MANAGEMENT
SYSTEM
SNMP PROXY AGENT
FEATURE
USER'S GUIDE**

Document No. 6700-A2-GB20-10

COMSPHERE

6700 Series

Network Management System

SNMP Proxy Agent Feature

User's Guide

6700-A2-GB20-10

Issue 2 (December 1996)

Changes and enhancements to the product and to the information herein will be documented and issued as a new release.

Warranty, Sales, and Service Information

Contact your sales or service representative directly for any help needed. For additional information concerning warranty, sales, service, repair, installation, documentation or training, use one of the following methods:

- **Via the Internet:** Visit the Paradyne World Wide Web site at <http://www.paradyne.com>
- **Via Telephone:** Call our automated call system to receive current information via fax or to speak with a company representative.
 - Within the U.S.A., call 1-800-870-2221
 - International, call 813-530-2340

Trademarks

All products and services mentioned herein are the trademarks, service marks, registered trademarks or registered service marks of their respective owners.

COPYRIGHT © 1996 Paradyne Corporation.

All Rights Reserved

Printed in U.S.A.

This publication is protected by federal copyright law. No part of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language in any form or by any means, electronic, mechanical, magnetic, manual or otherwise, or disclosed to third parties without the express written permission of Paradyne Corporation, 8545 126th Avenue North, P.O. Box 2826, Largo, Florida 33770-2826.

Paradyne Corporation makes no representation or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. Further, Paradyne Corporation reserves the right to revise this publication and to make changes from time to time in the contents hereof without obligation of Paradyne Corporation to notify any person of such revision or changes.

Contents

Preface

- Related Documents v
- Ordering Information vi

1 Overview

- What is SNMP? 1-1
- What is the SNMP Proxy Agent Feature? 1-1
- Software Description 1-2
- Feature List 1-2
- Sample SNMP Network Topology 1-3
- SNMP Proxy Agent Feature Package Contents ... 1-4

2 Installation

- Hardware and Software Requirements 2-1
- Supported Network Adapters 2-1
- Preparing for Installation and Configuration 2-3
- Installing the SNMP Proxy Agent Feature 2-5
- Configuring the SNMP Proxy Agent Feature for
TCP/IP Networks 2-8

Contents

3 Management Information Bases

- MIB-II Support 3-1
 - MIB-II System Group Overview 3-2
 - MIB-II Interface Group Overview 3-2
- COMSPHERE 6700 Series NMS
 - Enterprise MIBs 3-3
 - COMSPHERE 6700 NMS MIB Overview 3-5
 - COMSPHERE 6700 Device MIB Overview .. 3-5
 - Front Panel MIB Overview 3-6
 - Call Directory MIB Overview 3-6
- DOS, Windows, and Workstation MIBs 3-6
- SNMP Traps Overview 3-7
- Device Traps 3-7
- SNMP Community Views 3-11
 - Community Name Example 3-11

A MIB Definition

B Device Enterprise MIB Definition

C Enterprise Trap Definitions

D Enterprise Common MIB Definitions

Glossary

- Refer to Document No. 6700-A2-GB22-00.

Index

Figures

1 Overview

- 1-1 Sample SNMP Network Topology 1-3

2 Installation

- 2-1 Welcome Window 2-5
- 2-2 Product Information Window 2-6
- 2-3 Destination Directory Window 2-6
- 2-4 Confirmation Window 2-7
- 2-5 Completion Status Window 2-7
- 2-6 Installation Completed! Window 2-8

3 Management Information Bases

- 3-1 SMI Defined Object Identifier Prefix Tree 3-4

Tables

2 Installation

- 2-1 Supported Network Adapter Boards 2-2

3 Management Information Bases

- 3-1 MIB Browser Output Table 3-2
- 3-2 Primary Alert Traps 3-8
- 3-3 System Security Event Traps 3-9
- 3-4 Primary Alert Clear Notifications 3-10

Preface

This guide describes how to install and use the COMSPHERE® 6700 Series Network Management System (NMS) SNMP Proxy Agent feature.

This manual assumes you have a basic understanding of COMSPHERE modems and data service units (DSUs) and their operation, are knowledgeable about data communications, and are familiar with **Windows**™ terminology and conventions.

Related Documents

Contact your sales representative for additional product documentation.

- | | |
|--------------|---|
| 3510-A2-GA31 | <i>COMSPHERE 3000 Series Carrier, Installation Manual</i> |
| 3610-A2-GB41 | <i>COMSPHERE 3600 Series Data Service Units, Models 3610 and 3611, Time Division Multiplexer, Multichannel Multipoint, and Digital Bridge Options, Applications Guide</i> |
| 3610-A2-GB91 | <i>COMSPHERE 3600 Series Data Service Units, Models 3610 and 3611, Operator's Guide (with Reference Card insert)</i> |
| 3610-A2-GN32 | <i>COMSPHERE 3600 Series Data Service Units, Models 3610 and 3611, Dial Backup Module and SNA Diagnostic Interface Options, Applications Guide</i> |
| 3810-A2-GB91 | <i>COMSPHERE 3800 Series Modems, Models 3810, 3811, and 3820, User's Guide</i> |

3910-A2-GN32	<i>COMSPHERE 3900 Series Modems, Models 3910 and 3911, Point-to-Point/Multipoint, Installation and Operation Manual</i>
6700-A2-GB21	<i>COMSPHERE 6700 Series Network Management System Multiuser Feature User's Guide</i>
6700-A2-GB22	<i>COMSPHERE 6700 Series Network Management System Network Configuration Guide</i>
6700-A2-GB41	<i>COMSPHERE 6700 Series Network Management System Security Manager Feature Supplement</i>
6700-A2-GY31	<i>COMSPHERE 6700 Series Network Management System User's Guide</i>
1001-40-1940	<i>NEWT TCP/IP for Windows Installation and User's Guide</i>

Overview

1

What is SNMP?

Simple Network Management Protocol (SNMP) is an Internet standard protocol for managing TCP/IP devices.

Using SNMP, a network administrator can address queries and commands to network nodes and devices. You can use SNMP to monitor network performance and status; control operational parameters; and report, analyze, and isolate faults.

Agents, managers, and Management Information Bases (MIBs) combine to control network devices.

Agents collect management information and store it in a database called the MIB. The agent provides management information to an SNMP manager upon request.

Non-TCP/IP devices can be managed with SNMP proxy agents.

What is the SNMP Proxy Agent Feature?

The COMSPHERE® 6700 Series Network Management System SNMP Proxy Agent feature provides the capability for any device managed by a COMSPHERE 6700 Series Network Management System (NMS) to also be monitored and controlled by an SNMP network management system, such as HP OpenView™ or SunNet™ Manager.

Software Description

The SNMP Proxy Agent feature includes NetManage Enhanced Windows TCP/IP (NEWT™) software, which supports transmission of SNMP messages across Ethernet®, Token Ring, FDDI, SLIP, and PPP interfaces.

The SNMP Proxy Agent feature is a software package that can be installed on top of the COMSPHERE 6700 Series NMS Release 4.0.0 or higher at any time.



NOTE:

COMSPHERE 6700 Series NMS Release 4.0.0 or higher is required to execute the SNMP Proxy Agent feature.

You can install the SNMP Proxy Agent feature in any of the following COMSPHERE 6700 Series NMS installations:

- Single user
- Multiuser server
- Multiuser client

Feature List

The SNMP Proxy Agent feature provides the following features:

- SNMP Proxy Agent for COMSPHERE devices using Enterprise MIBs
- SNMP Agent for COMSPHERE 6700 Series NMS using Enterprise MIBs
- SNMP Agent for MIB-II
- SNMP Agent for DOS, Windows™, and workstations using NetManage Enterprise MIBs
- COMSPHERE 6700 Series NMS Alerts exported as SNMP Traps
- One IP address per COMSPHERE 6700 Series NMS
- Specific Community View Access for each managed device

- Integrated TCP/IP, UDP/IP, and SNMP Protocol support
- Concurrent Ethernet, Token Ring, FDDI, SLIP, and PPP interface support

Sample SNMP Network Topology

Figure 1-1 shows a sample SNMP network topology.

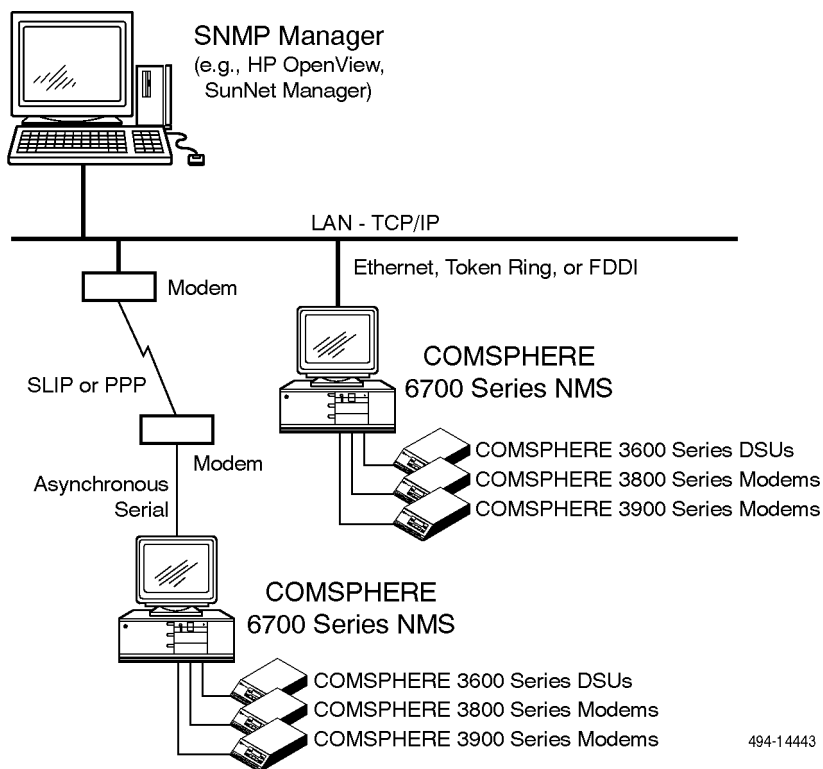


Figure 1-1. Sample SNMP Network Topology

SNMP Proxy Agent Feature Package Contents

The SNMP Proxy Agent feature package includes the following:

- SNMP feature software on one 3 1/2" disk
- COMSPHERE 6700 MIBs 3 1/2" disk
- One *COMSPHERE 6700 Series Network Management System SNMP Proxy Agent Feature User's Guide*
- One *COMSPHERE 6700 Series Network Management System Network Configuration Guide*

Hardware and Software Requirements

The SNMP Proxy Agent feature has the same hardware and software requirements as the basic single-user NMS. Refer to the *Hardware Description* and *Software Description* sections of the *COMSPHERE 6700 Series Network Management System User's Guide* for details.

Supported Network Adapters

To use the SNMP feature on Ethernet, Token Ring, or FDDI, you must install a supported network adapter board. Supported boards are listed in Table 2-1.

Table 2-1. Supported Network Adapter Boards (1 of 2)

Interface	Vendor	Model
Ethernet	3COM	3C501 Ether Link
		3C503 Ether Link II
		3C505 Ether Link Plus
		3C507 Ether Link 16
		3C509 Ether Link III
		3C523 Ether Link/MC TP
		3C527 Ether Link/MC 32
	Allied Telesis	AT-1500 Network Adapter
	Hewlett Packard®	HP27247/HP27252 Plus
	Intel®	EtherExpress 16 and 16T
	Novell/Excelan	EXOS105T
		NE1000
		NE2000
	Racal InterLan	NI5210
		NI6510
	SMC	SMC 8000
		SMC Elite 16
	Ungermann-Bass	Networth EtherNext 16-Bit UTP
		NIUpc/EOTP
		NIUps
	Western Digital	EtherCard Plus
		EtherCard Plus Elite 16
	Xircom	All PE Models
		All PE2 Models

Table 2-1. Supported Network Adapter Boards (2 of 2)

Interface	Vendor	Model
Ethernet	Xircom (Continued)	All PE3 Models
	Other	(NDIS Driver Required)
Token Ring	IBM®	16/4 Token Ring Adapter
	Xircom	All PT Models
	Madge	Smart 16/4 RingNode adapter
	Proteon®	p139X
		p189X
		p1990
	Raycone	16/4 Token Ring Adapter
	Other	(NDIS Driver Required)
FDDI	Any	(NDIS Driver Required)

Preparing for Installation and Configuration

Before installing and configuring the SNMP Proxy Agent feature, you need some system and network information. Use the following form to collect the necessary information before you start the installation process. Retain this form as a record of this information. Examples are shown in parentheses.

SNMP Feature Installation and Configuration Information
General Where to install the software (c:\pcnms): _____ Network Interface Name (Ethernet0): _____ Network Interface Type (Ethernet): _____
Workstation Internet Address of Workstation (192.0.2.2): _____ Unique Node Name (Largo Bld. G): _____
SNMP Manager Internet Address of Manager (192.0.2.8): _____ Unique Manager Name (Help Desk1): _____
LAN/Hardware Interface Adapter Vendor Name (Western Digital): _____ Board Type (Ether Card Plus): _____ Interrupt Level (5): _____ I/O Base Address (0x300): _____
SLIP Interface Baud Rate (9600): _____ Flow Control (Hardware): _____ Modem Type (Hayes® compatible): _____ Port (COM 1): _____ Telephone Number (9,1,813-555-2671): _____
Optional Subnet Mask (255.255.255.0): _____ Default Gateway (192.0.2.254): _____

Installing the SNMP Proxy Agent Feature

To install the SNMP Proxy Agent feature software, start from the Program Manager window and perform the following steps:

1. Insert Disk #1 into Drive A.
2. From the Program Manager window, choose **File**.
3. From the File menu, choose **Run**.
4. In the Command Line field, type **A:\INSTALL**.
5. Choose **OK**. The Welcome window appears, as shown in Figure 2-1.

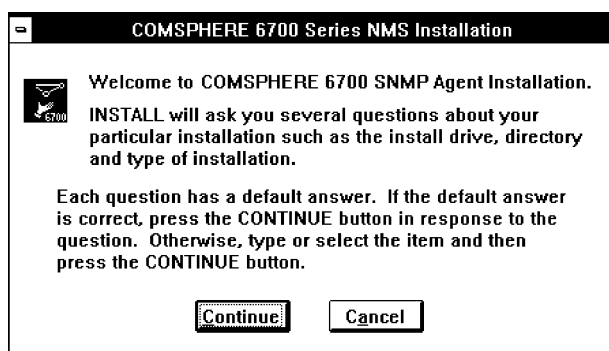


Figure 2-1. Welcome Window

6. Choose **Continue**. The Product Information window appears, as shown in Figure 2-2.

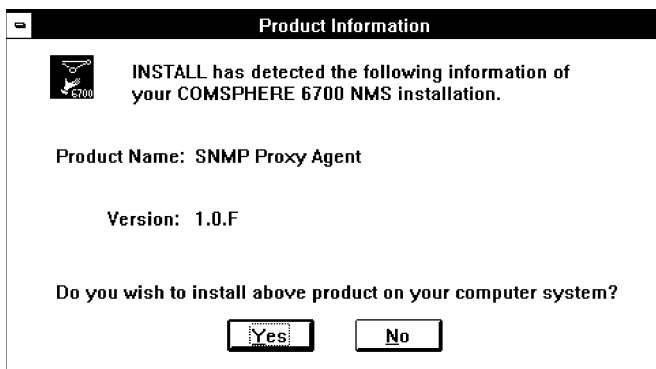


Figure 2-2. Product Information Window

7. Choose **Yes**. The Destination Directory window appears, as shown in Figure 2-3.

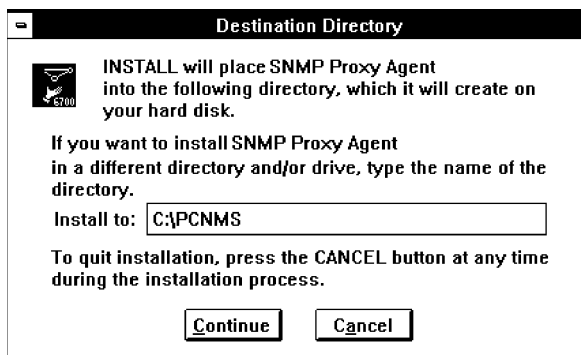


Figure 2-3. Destination Directory Window

The Destination Directory window allows you to place the NMS software into a specific directory. A single directory within a single partition is required.

8. Enter a subdirectory location or choose the default setting **C:\PCNMS**.

9. Choose **Continue**. The Confirmation window appears, as shown in Figure 2-4.

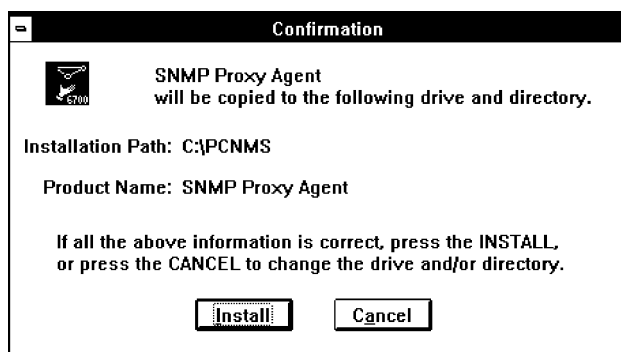


Figure 2-4. Confirmation Window

10. Choose **Install** to confirm the installation of the NMS software into the specified directory. The Completion status window appears, as shown in Figure 2-5.

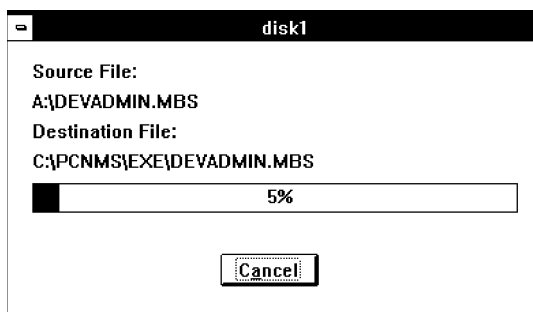


Figure 2-5. Completion Status Window

This window displays a bar indicating the percentage of completion for the current installation. In addition, the names of the files being installed appear above the bar until the installation is complete. Then the Installation Completed! window appears, as shown in Figure 2-6.

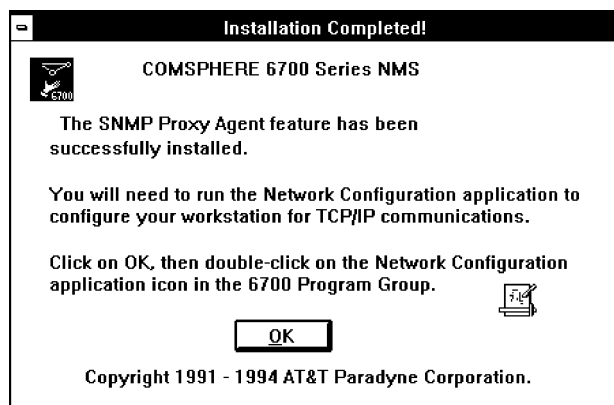


Figure 2-6. Installation Completed! Window

11. Choose **OK**.

Configuring the SNMP Proxy Agent Feature for TCP/IP Networks

After installation, use the Network Configuration application to customize your configuration for TCP/IP networks. For details on doing so, refer to the *COMSPHERE 6700 Series Network Management System Network Configuration Guide*.

Management Information Bases

3

MIB-II Support

The SNMP Proxy Agent feature includes support for the standard Management Information Base (MIB-II) used with network management protocols in TCP/IP-based internets. This support is provided by the NEWT component. The following MIB-II groups are supported as specified in RFC 1213.

- System Group
- Interface Group
- Address Translation Table Group
- IP Group
- ICMP Group
- TCP Group
- UDP Group

Specific characteristics of the system and interface groups are noted in the following paragraphs.

NOTE:

Because this product is a *proxy agent*, the MIB-II information reflects the state of the TCP/IP interface in the workstation where the proxy is running. If a MIB-II variable is accessed for a proxied device, the MIB-II value for the proxy workstation is returned.

MIB-II System Group Overview

The MIB-II system group contains system description information. You may customize the **sysContact**, **sysName**, and **sysLocation** text. For further details, refer to the *NEWT TCP/IP for Windows Installation and User's Guide*.

MIB browser output examples for the system group are presented in Table 3-1.

Table 3-1. MIB Browser Output Table

Object Descriptor	Example
sysDescr	80486 DOS 6.20 Windows 3.10 Enhanced Mode NetManage SNMP 4.00
sysObjectID	NetManage, Inc.
sysUpTime	00:00:24.99
sysContact	COMSPHERE 6700 NMS Administrator
sysName	Paradyne COMSPHERE 6700
sysLocation	COMSPHERE 6700 NMS Workstation
sysServices	76

MIB-II Interface Group Overview

The MIB-II interface group contains information about TCP/IP interfaces configured within NEWT.

COMSPHERE 6700 Series NMS Enterprise MIBs

Four new enterprise Management Information Base (MIB) definitions are introduced with SNMP Proxy Agent Release 1.0:

- COMSPHERE 6700 NMS MIB
- COMSPHERE 6700 Device MIB
- Front Panel MIB
- Call Directory MIB

NOTE:

For additional installation information, refer to the readme.txt file on the COMSPHERE 6700 SNMP MIBs diskette, Part No. 869-2745-0011.

Figure 3-1 shows the Structure of Management Information (SMI) defined object identifier prefix tree that includes the COMSPHERE 6700 Enterprise MIBs.



The COMSPHERE 6700 NMS and Device MIBs are defined in the appendices.

COMSPHERE 6700 NMS MIB

Overview

The NMS MIB administers the 6700 Series NMS. The following groups are defined in the NMS MIB.

- NMS Administration Group
- NMS System Group
- NMS Device Group
- NMS Manager Group
- NMS Test Group

Refer to [Appendix A](#) for a full description of the COMSPHERE 6700 NMS MIBs.

COMSPHERE 6700 Device MIB

Overview

The Device MIB supports management of COMSPHERE 3600, 3800, and 3900 model-type devices attached to a COMSPHERE 6700 Series NMS.

The device attributes and commands are defined in a device-independent manner that supports both modems and DSUs. The following groups are defined in the Device MIB.

- Device Administration Group
- Device Identity Group
- Device Status Group
- Device Circuit Quality Group
- Device EIA Status Group
- Device External Leads Group
- Device Command Group

The SNMP Proxy Agent automatically provides a specific community name for each managed device. An SNMP management application must use the device specific community name to select which device to act upon.

Refer to [Appendix B](#) for a full description of the COMSPHERE 6700 Device MIB.

Front Panel MIB Overview

The Front Panel MIB provides remote access to the front panel of a device and is accessed using the community procedures used for the 6700 Device MIB.

Front panel keypress codes can be sent to the device in Set requests. The current front panel display can be retrieved with a Get request to the front panel display table.

Refer to [Appendix D](#) for a full description of the Front Panel MIB.

Call Directory MIB Overview

The Call Directory MIB contains call directory data for a device and is accessed using the community procedures used for the 6700 Device MIB. The number of entries in a device's call directory table is dependent on the device type.

Refer to [Appendix D](#) for a full description of the Call Directory MIB.

DOS, Windows, and Workstation MIBs

Additional MIBs supported by the SNMP Proxy Agent feature include NetManage Enterprise MIBs for DOS, Windows, and Workstations.

Support for these MIBs is provided by the NEWT software and can be turned on/off through the COMSPHERE 6700 Series NMS Network Configuration feature. To do so, refer to the *NEWT TCP/IP for Windows Installation and User's Guide*.

SNMP Traps Overview

COMSPHERE 6700 NMS alerts may be exported to authorized SNMP managers as SNMP traps. Traps are forwarded for any event that makes it through the COMSPHERE 6700 Series NMS's alert filter.

All traps are enterprise specific, with an OBJECT IDENTIFIER specifying the COMSPHERE 6700 Enterprise Device product.

Trap forwarding can be turned on/off for each SNMP manager in the authorized manager list through the COMSPHERE 6700 Series NMS Network Configuration feature or via SNMP. To do so, refer to the *COMSPHERE 6700 Series Network Management System Network Configuration Guide*.

Refer to [Appendix C](#) for a full description of the COMSPHERE 6700 Enterprise Traps.

Device Traps

Device traps are generated for device-related alert conditions detected by the COMSPHERE 6700 Series NMS.

The Enterprise OBJECT IDENTIFIER used in device traps is the following:

```
iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) att-2(74) att-products(1)
paradyneNMS-products(13) nms6700-products(2) devOid(2)
```

The community name used in device product traps is the Device Read Community. This is formed by concatenating base read community with the device name.

The following COMSPHERE 6700 Device MIB variable bindings are included in each trap message.

- Device Name (**devAdminName**)
- Device Type (**devAdminModelType**)
- System Time (**nmsSystemTime**)

[Table 3-2](#) identifies which primary alert traps apply to the specific devices. Trap numbers 1–32 are reserved for primary alerts.

**NOTE:**

The state of a device's primary alerts can also be read from **devStatusAlert**.

Table 3-2. Primary Alert Traps (1 of 2)

Trap	Primary Alert Traps	3600	3800	3900
1	Device Failure	•	•	•
2	Configuration Change Notify	•	•	•
3	Test Mode	•	•	•
4	Disabled	•	•	•
5	Out of Threshold/VF Threshold Exceeded	•	•	•
6	Facility Alarm	•	•	•
7	External Alarm	•	•	•
8	Streaming Terminal	•	•	•
9	Access Security		•	•
10	Dial Backup Active for APL	•	•	•
11	DTE Alarm	•	•	•
12	Sub-normal Operating Speed	•	•	•
13	Primary Channel Interrupted		•	•
14	Firmware Downloading		•	•
15	Make Busy Mode		•	•
16	Service Line		•	•
17	Non-answering Modem		•	•
18	Short Holding Time Modem		•	•
19	Sub Tree Truncation	•		•
20	TDM Failure/ Good APL Line	•		•

Table 3-2. Primary Alert Traps (2 of 2)

Trap	Primary Alert Traps	3600	3800	3900
21	Tributary Timeout	•		
22	Dial Tone	•		
23	Redundant Power Supply	•		
32	No Response	•	•	•

Table 3-3 identifies which system security event traps apply to the specific devices.

Table 3-3. System Security Event Traps (1 of 2)

Trap	System Security Event Traps	3600	3800	3900
101	VF Login OK		•	•
102	User Login OK		•	•
103	User Login Rejected - Retry Failed		•	•
104	User Login Rejected - Password Timeout		•	•
105	User Login Aborted - Line Disconnected		•	•
106	VF Login Rejected - Password Invalid		•	•
107	VF Login Rejected - Password Timeout		•	•
108	VF Login Aborted - Line Disconnected		•	•
109	User Login OK - Multiple Password Retries		•	•
110	User ID and Password Combination Invalid		•	•
111	Invalid User ID - Password Valid for Device		•	•

Table 3-3. System Security Event Traps (2 of 2)

Trap	System Security Event Traps	3600	3800	3900
112	Invalid Access Time		●	●
113	User Login Hack - Multiple Sequential Password Retries		●	●
114	Device Security Table Invalid		●	●
116	Security Download Failed		●	●
117	Front Panel Modification		●	●

Table 3-4 lists the Primary Alert Clear Notifications.

Table 3-4. Primary Alert Clear Notifications (1 of 2)

Trap	Primary Alert Traps	3600	3800	3900
201	Device Failure Cleared	●	●	●
202	Configuration Change Notify Cleared	●	●	●
203	Test Mode Cleared	●	●	●
204	Disabled Cleared	●	●	●
205	Out of Threshold/VF Threshold Exceeded Cleared	●	●	●
206	Facility Alarm Cleared	●	●	●
207	External Alarm Cleared	●	●	●
208	Streaming Terminal Cleared	●	●	●
209	Access Security Cleared		●	●
210	Dial Backup Active for APL Cleared	●	●	●
211	DTE Alarm Cleared	●	●	●
212	Sub-normal Operating Speed Cleared	●	●	●

Table 3-4. Primary Alert Clear Notifications (2 of 2)

Trap	Primary Alert Traps	3600	3800	3900
213	Primary Channel Interrupted Cleared		●	●
214	Firmware Downloading Cleared		●	●
215	Make Busy Mode Cleared		●	●
216	Service Line Cleared		●	●
217	Non-answering Modem Cleared		●	●
218	Short Holding Time Modem Cleared		●	●
219	Sub Tree Truncation Cleared	●		●
220	TDM Failure Cleared/ Good APL Line Cleared	●		●
221	Tributary Timeout Cleared	●		
222	Dial Tone Cleared	●		
223	Redundant Power Supply Cleared	●		
232	No Response Cleared	●	●	●

SNMP Community Views

Different read and write community views are used for the COMSPHERE 6700 NMS MIB and for each managed device under the COMSPHERE 6700 Device MIB. Set up the community names for community views using the COMSPHERE 6700 Series NMS Network Configuration feature. To do so, refer to the *COMSPHERE 6700 Series Network Management System Network Configuration Guide*.

The community names for the Device MIB are generated by appending the desired device name to the appropriate base name. Consider the following rules when creating community names:

- Community names should not be the prefix of a device name.
- Write community names can be used for both reads and writes.
- Call Directory MIB and Front Panel MIB both use the same communities as the Device MIB.

Community Name Example

To better understand the use of community names, consider the following example. Using community name settings as follows:

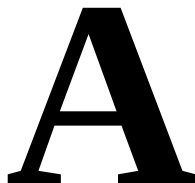
- **nmsAdminNmsReadCommunity** = clear
- **nmsAdminNmsWriteCommunity** = orange
- **nmsAdminBaseReadCommunity** = teal
- **nmsAdminBaseWriteCommunity** = pink

The community name for **Get** requests to the NMS MIB can be either “clear” or “orange.” The community name for **Set** requests to the NMS MIB must be “orange.”

The Device MIB community name for **Get** requests to a device named “*modem1*” in the Device MIB would be either “*tealmodem1*” or “*pinkmodem1*.” The community name for **Set** requests to the device would be “*pinkmodem1*.”

To access a Dial Backup Module (DBM) within a COMSPHERE 3600 Series DSU, the suffix “.dbm” is appended to the device community name. For example, the DBM for a device named “*DSU5*” would have a write community name of “*pinkDSU5.dbm*.”

MIB Definition



NMS6700-MIB DEFINITIONS ::= BEGIN

— Title: COMSPHERE 6700 NMS MIB for Customer Network Management

— Copyright (C) 1996, Paradyne. All rights reserved.

— This file may be freely copied and distributed as long as no changes are made to it.

— The NMS MIB contains attributes of the 6700 NMS entity.

— The user can inspect and control the 6700 NMS via this MIB.

IMPORTS

IpAddress

FROM RFC1155-SMI

DisplayString

FROM RFC1213-MIB

OBJECT-TYPE

FROM RFC-1212

DateAndTime

FROM HOST-RESOURCES-MIB

nms-6700

FROM ATTP-ENTERPRISES;

— Enterprise Identification for COMSPHERE 6700 NMS MIB

nms OBJECT IDENTIFIER ::= { nms-6700 1 }— NMS MIB

—
— NMS MIB Groups
—

nmsAdmin	OBJECT IDENTIFIER ::= { nms 1 }— administration
nmsSystem	OBJECT IDENTIFIER ::= { nms 2 }— system identity
nmsDevice	OBJECT IDENTIFIER ::= { nms 3 }— 6700 devices
nmsManager	OBJECT IDENTIFIER ::= { nms 4 }— SNMP managers
nmsTest	OBJECT IDENTIFIER ::= { nms 5 }— Tests

— NMS Administration Group (nms 1)

— NMS Information

nmsAdminName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Name of the 6700 NMS.”

::= { nmsAdmin 1 }

nmsAdminLocation OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Location of the 6700 NMS.”

::= { nmsAdmin 2 }

nmsAdminContact1 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Contact 1 for the 6700 NMS.”

::= { nmsAdmin 3 }

nmsAdminContact2 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Contact 2 for the 6700 NMS.”

::= { nmsAdmin 4 }

— NMS Community Names

nmsAdminNmsReadCommunity OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..32))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”NMS read community name. This community has read-only

access to all objects in the COMSPHERE 6700 NMS MIB.

An exception applies with the nmsAdminNmsWriteCommunity object for which the read community provides no access.”

::= { nmsAdmin 5 }

nmsAdminNmsWriteCommunity OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..32))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”NMS write community name. This community provides full access to all objects in the COMSPHERE 6700 NMS MIB.”

::= { nmsAdmin 6 }

— Device Community Names

nmsAdminBaseReadCommunity OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..16))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Base read community name string used with Device MIB. When string length is zero, the device read community name is equal to the device name.”

::= { nmsAdmin 7 }

nmsAdminBaseWriteCommunity OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..16))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Base write community name string used with Device MIB. When string length is zero, the device write community name is equal to the device name.”

::= { nmsAdmin 8 }

— NMS System Group (nms 2)

— Time of Day

nmsSystemTime OBJECT-TYPE

SYNTAX DateAndTime

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Specifies the date and time in RFC 1514 DateAndTime format.

The date and time are obtained from the system clock of
the workstation running the proxy agent software.”

::= { nmsSystem 1 }

— Features

nmsSystemFeatures OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Number of COMSPHERE 6700 NMS product or feature entries
in nmsSystemFeatureTable.”

::= { nmsSystem 2 }

nmsSystemFeatureTable OBJECT-TYPE

SYNTAX SEQUENCE OF NmsSystemFeatureEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

”The nms feature table contains a list of software
products/features installed in the COMSPHERE 6700 NMS.”

::= { nmsSystem 3 }

nmsSystemFeatureEntry OBJECT-TYPE

SYNTAX NmsSystemFeatureEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

”An entry in the software feature table.”

INDEX { nmsSystemFeatureIndex }

::= { nmsSystemFeatureTable 1 }

— Layout of one entry in nmsSystemFeatureTable.

```
NmsSystemFeatureEntry ::=
    SEQUENCE {
        nmsSystemFeatureIndex
            INTEGER,
        nmsSystemFeatureName
            DisplayString,
        nmsSystemFeatureVersion
            DisplayString,
        nmsSystemFeatureSerial
            DisplayString
    }
```

— Index

```
nmsSystemFeatureIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The index of the entry."
    ::= { nmsSystemFeatureEntry 1 }
```

— Name

```
nmsSystemFeatureName OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..25))
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Name of system product or feature."
    ::= { nmsSystemFeatureEntry 2 }
```

— Version

```
nmsSystemFeatureVersion OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..25))
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Version of system product or feature."
```

::= { nmsSystemFeatureEntry 3 }

— Serial Number

nmsSystemFeatureSerial OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..25))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Serial number of system product or feature. If feature has no serial number, this object contains a zero length string.”

::= { nmsSystemFeatureEntry 4 }

— NMS Device Group (nms 3)

nmsDevices OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

 "The number of devices configured in the 6700 nms network.

 This is also the number of entries in nmsDeviceTable."

::= { nmsDevice 1 }

nmsDeviceTable OBJECT-TYPE

SYNTAX SEQUENCE OF NmsDeviceEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

 "The nms device table contains an alpha-sorted list of names

 of devices configured for management in the COMSPHERE 6700
 NMS."

::= { nmsDevice 2 }

nmsDeviceEntry OBJECT-TYPE

SYNTAX NmsDeviceEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

 "An entry in the nms device table."

INDEX { nmsDeviceIndex }

::= { nmsDeviceTable 1 }

— Layout of one entry in nmsDeviceTable.

NmsDeviceEntry ::=

SEQUENCE {

 nmsDeviceIndex

 INTEGER,

 nmsDeviceName

 DisplayString

}

nmsDeviceIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object is the row index of the nmsDevice table.
Note that a device's index in the table may change as
devices are added or removed from the 6700 nms network,
thus this index should not be used as a device ID."

::= { nmsDeviceEntry 1 }

nmsDeviceName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object is the device name, as assigned within the
COMSPHERE 6700 NMS."

::= { nmsDeviceEntry 2 }

— NMS Manager Group (nms 4)

— Number of Authorized SNMP Managers

nmsManagers OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object contains the number of entries in
nmsManagerTable."

::= { nmsManager 1 }

— SNMP Manager Table

nmsManagerTable OBJECT-TYPE

SYNTAX SEQUENCE OF NmsManagerEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The manager table contains a list of SNMP managers
authorized access to the COMSPHERE 6700 MIBs."

::= { nmsManager 2 }

nmsManagerEntry OBJECT-TYPE

SYNTAX NmsManagerEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

”An entry in the manager table. Entries are indexed by IP Address.”

INDEX { nmsManagerAddress }

::= { nmsManagerTable 1 }

— Layout of one entry in nmsManagerTable.

NmsManagerEntry ::=

```
SEQUENCE {  
    nmsManagerAddress  
        IpAddress,  
    nmsManagerName  
        DisplayString,  
    nmsManagerAccess  
        INTEGER,  
    nmsManagerTraps  
        INTEGER,  
    nmsManagerLocation  
        DisplayString,  
    nmsManagerContact1  
        DisplayString,  
    nmsManagerContact2  
        DisplayString,  
    nmsManagerEntryStatus  
        INTEGER  
}
```

— Table Index (Manager IP Address)

nmsManagerAddress OBJECT-TYPE

SYNTAX IpAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

”This object is the row index in the manager table.”

::= { nmsManagerEntry 1 }

— Manager Name

nmsManagerName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..16))

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is the NMS manager name. The manager name must be unique in the table.

If a set operation is performed for a non-existing instance (i.e., index is a new IP address), a new table entry for the specified IP address will be created. The new entry's nmsManagerAccess will be defaulted to read-only(2), the nmsManagerTraps to disabled(2), the nmsManagerLocation, nmsManagerContact1, and nmsManagerContact2 to zero length octet strings, and nmsManagerStatus to valid(1)."

::= { nmsManagerEntry 2 }

— Access

nmsManagerAccess OBJECT-TYPE

SYNTAX INTEGER {

no-access(1),

read-only-access(2),

read-write-access(3)

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Maximum manager access level setting can be none(1), read-only(2), or read-write(3)."

::= { nmsManagerEntry 3 }

— Traps

nmsManagerTraps OBJECT-TYPE

SYNTAX INTEGER {

enabled(1),

disabled(2)

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Specifies if trap forwarding is enabled for this manager."

::= { nmsManagerEntry 4 }

— Location

nmsManagerLocation OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Text string describing manager location.”

::= { nmsManagerEntry 5 }

— Contact1

nmsManagerContact1 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Text string describing manager contact 1.”

::= { nmsManagerEntry 6 }

— Contact2

nmsManagerContact2 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Text string describing manager contact 2.”

::= { nmsManagerEntry 7 }

— Entry Status (Used to Delete an Entry)

nmsManagerEntryStatus OBJECT-TYPE

SYNTAX INTEGER {

valid(1),

invalid(2) — deletes entry from table

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

”This object is used to delete an entry from the table.

Only the invalid(2) value is allowed on write.”

::= { nmsManagerEntry 8 }

— NMS Test Group (nms 5)

— Textual Convention for Test Result Validity Indicator

TestResultValidity ::=

```

    INTEGER {
        valid-final(1),
        valid-intermediate(2),
        not-available-yet(3),
        count-overflow(4),
        not-used(5)
    }

```

— Number of Test Table Entries

nmsTestEntries OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”The number of NMS Test table entries that are configured.
This number is controlled by the following lines in
the 6700 profile (default value is shown).

[SNMP]

NmsTestEntries=8”

::= { nmsTest 1 }

— Test Table

nmsTestTable OBJECT-TYPE

SYNTAX SEQUENCE OF NmsTestEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

”The NMS Test table uses RMON-style arbitration (RFC 1271)
to allocate device test slots. There are 8 (by default)
simultaneous test slots available.

Each test slot controls testing of a single device (or a
control-tributary pair, depending on the test type).”

::= { nmsTest 2 }

nmsTestEntry OBJECT-TYPE

SYNTAX NmsTestEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

 "An entry in the nms test table."

INDEX { nmsTestIndex }

::= { nmsTestTable 1 }

— Layout of one entry in nmsTestTable.

```
NmsTestEntry ::=
SEQUENCE {
    nmsTestIndex
        INTEGER,
    nmsTestDeviceName
        DisplayString,
    nmsTestType
        INTEGER,
    nmsTestDuration
        INTEGER,
    nmsTestPortNumber
        INTEGER,
    nmsTestRemoteName
        DisplayString,
    nmsTestStatus
        INTEGER,
    nmsTestLocalTotalSecondsValidity
        TestResultValidity,
    nmsTestLocalTotalSeconds
        INTEGER,
    nmsTestLocalErrorSecondsValidity
        TestResultValidity,
    nmsTestLocalErrorSeconds
        INTEGER,
    nmsTestLocalTotalBlocksValidity
        TestResultValidity,
    nmsTestLocalTotalBlocks
        INTEGER,
    nmsTestLocalErrorBlocksValidity
        TestResultValidity,
    nmsTestLocalErrorBlocks
        INTEGER,
    nmsTestRemoteTotalSecondsValidity
        TestResultValidity,
    nmsTestRemoteTotalSeconds
        INTEGER,
    nmsTestRemoteErrorSecondsValidity
        TestResultValidity,
    nmsTestRemoteErrorSeconds
        INTEGER,
    nmsTestRemoteTotalBlocksValidity
        TestResultValidity,
```

```
nmsTestRemoteTotalBlocks
    INTEGER,
nmsTestRemoteErrorBlocksValidity
    TestResultValidity,
nmsTestRemoteErrorBlocks
    INTEGER,
nmsTestTimeOutsValidity
    TestResultValidity,
nmsTestTimeOuts
    INTEGER,
nmsTestOwner
    DisplayString,
nmsTestEntryStatus
    INTEGER
}
```

— Index

```
nmsTestIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "This object is the test table entry index."
    ::= { nmsTestEntry 1 }
```

— Device Name

```
nmsTestDeviceName OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..15))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The name of the device to test."
    ::= { nmsTestEntry 2 }
```

— Test Type

```
nmsTestType OBJECT-TYPE
```

```
SYNTAX INTEGER {  
    self-test(1),  
    local-loop(2),  
    rem-digital-loop(3),  
    loc-digital-loop(4),  
    dte-loop(5),  
    bert(6),  
    local-loop-bert(7),  
    rdl-bert(8),  
    digital-test(9),  
    end-to-end-test(10),  
    abort(11)  
}
```

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is the test type. Test type self-test(1) performs an internal self-test of the device. Test type local-loop(2) places the device into local loopback (for modems, this is a CCITT V.54 Loop 3). Test type rem-digital-loop(3) places the device in a remote digital loopback (for modems, this is a CCITT V.54 Loop 2). Test type loc-digital-loop(4) forces a local device to loopback any data received from the remote device (this is useful if the remote device is incapable of initiating a remote digital loopback from its location). Test type dte-loop(5) loops a DSU's data port back to the DTE/DCE interface on a per-port basis without affecting the operation of the remaining ports. Test type bert(6) initiates a pattern test. Test type local-loop-bert(7) places the device into local loopback and initiates a pattern test. Test type rdl-bert(8) places the device into remote digital loopback and initiates a pattern test. Test type digital-test(9) initiates a digital test of a pair of DSUs or DBMs and the data circuit between them. Test type end-to-end-test(10) initiates an end to end test of two devices. Test type abort aborts the current test."

::= { nmsTestEntry 3 }

—
— Test Duration
—

nmsTestDuration OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Test duration in seconds (or blocks). Only applicable for the following test types: bert, local-loop-bert, rdl-bert, digital-test, end-to-end-test.”

::= { nmsTestEntry 4 }

—
— Port Number
—

nmsTestPortNumber OBJECT-TYPE

SYNTAX INTEGER {

aggregate(1),

port1(2),

port2(3),

port3(4),

port4(5),

port5(6),

port6(7)

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Port number. Only applicable for multi-port devices.

Also only applicable for the following test types:

rem-digital-loop, loc-digital-loop, dte-loop, bert,

local-loop-bert, rdl-bert, digital-test, end-to-end-test.”

::= { nmsTestEntry 5 }

—
— Remote Device Name
—

nmsTestRemoteName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Remote device name. Only applicable for the following test types: digital-test, end-to-end-test.”

```
::= { nmsTestEntry 6 }
```

```
—  
— Test Status  
—
```

nmsTestStatus OBJECT-TYPE

```
SYNTAX INTEGER {  
    idle(1),  
    testMode(2),  
    testRunning(3),  
    testAborted(4),  
    testFailed(5)  
}
```

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Test status.

Status of idle(1) indicates that the device is not in test mode (from this control point); this is the initial value after createRequest. Status of testMode(2) indicates that the device is in a test mode (such as loopback); no further test results are pending. Status of testRunning(3) means the device is in test mode (such as bert test) and test results are pending. Status of testAborted(4) indicates that the device test was aborted and the device is no longer in test mode. Status of testFailed(5) indicates that the test has failed. Reason for test failure may be: invalid test parameter; device locked by another user; conflict with device environment, features, or configuration; device failure.”

```
::= { nmsTestEntry 7 }
```

```
— Test Results
```

```
— Local Total Seconds
```

nmsTestLocalTotalSecondsValidity OBJECT-TYPE

```
SYNTAX TestResultValidity
```

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the nmsTestLocalTotalSeconds test result object.”

```
::= { nmsTestEntry 8 }
```

nmsTestLocalTotalSeconds OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Total seconds for local device. (For 3600 model type devices,
this variable is also called elapsed seconds.)”

::= { nmsTestEntry 9 }

— Local Error Seconds

nmsTestLocalErrorSecondsValidity OBJECT-TYPE

SYNTAX TestResultValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the nmsTestLocalErrorSeconds test result object.”

::= { nmsTestEntry 10 }

nmsTestLocalErrorSeconds OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Errored seconds elapsed for local device. (The value is not
used for 3600 model type devices.)”

::= { nmsTestEntry 11 }

— Local Total Blocks (Bits for 3600 DSU)

nmsTestLocalTotalBlocksValidity OBJECT-TYPE

SYNTAX TestResultValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the nmsTestLocalTotalBlocks test result object.”

::= { nmsTestEntry 12 }

nmsTestLocalTotalBlocks OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Total blocks for local device.”

::= { nmsTestEntry 13 }

— Local Error Blocks (Bits for 3600 DSU)

nmsTestLocalErrorBlocksValidity OBJECT-TYPE

SYNTAX TestResultValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the nmsTestLocalErrorBlocks test result object.”

::= { nmsTestEntry 14 }

nmsTestLocalErrorBlocks OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Error Blocks for 3800 and 3900 modems, or Local Error Blocks (actually Bits) for 3600 DSUs; ”

::= { nmsTestEntry 15 }

— Remote Total Seconds

nmsTestRemoteTotalSecondsValidity OBJECT-TYPE

SYNTAX TestResultValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the nmsTestRemoteTotalSeconds test result object.”

::= { nmsTestEntry 16 }

nmsTestRemoteTotalSeconds OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Total seconds for remote device. (For 3600 model type devices, this variable is also called elapsed seconds.)”

::= { nmsTestEntry 17 }

— Remote Error Seconds

nmsTestRemoteErrorSecondsValidity OBJECT-TYPE

SYNTAX TestResultValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Validity of the nmsTestRemoteErrorSeconds test result object."

::= { nmsTestEntry 18 }

nmsTestRemoteErrorSeconds OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Errored seconds elapsed for the remote device. (The value is not used for 3600 model type devices.)"

::= { nmsTestEntry 19 }

— Remote Total Blocks (Bits for 3600 DSU)

nmsTestRemoteTotalBlocksValidity OBJECT-TYPE

SYNTAX TestResultValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Validity of the nmsTestRemoteTotalBlocks test result object."

::= { nmsTestEntry 20 }

nmsTestRemoteTotalBlocks OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Total blocks for remote device."

::= { nmsTestEntry 21 }

— Remote Error Blocks (Bits for 3600 DSU)

nmsTestRemoteErrorBlocksValidity OBJECT-TYPE

SYNTAX TestResultValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Validity of the nmsTestRemoteErrorBlocks test result object."

::= { nmsTestEntry 22 }

nmsTestRemoteErrorBlocks OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Remote Error blocks (actually Bits). Not used for 3800 and 3900 model type devices.”

::= { nmsTestEntry 23 }

— Time Outs

nmsTestTimeOutsValidity OBJECT-TYPE

SYNTAX TestResultValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the nmsTestTimeOuts test result object.”

::= { nmsTestEntry 24 }

nmsTestTimeOuts OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Time outs. Not used for 3800 and 3900 model type devices.”

::= { nmsTestEntry 25 }

— Test Owner

nmsTestOwner OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”The entity that configured this entry and is therefore using the resources assigned to it.

It is suggested that this string contain one or more of the following:

IP address, management station name, network manager’s name, location, or phone number.”

::= { nmsTestEntry 26 }

— Entry Status

nmsTestEntryStatus OBJECT-TYPE

```
SYNTAX INTEGER {  
    valid(1),  
    createRequest(2),  
    underCreation(3),  
    invalid(4)  
}
```

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The status of this nmsTestTable entry.

Setting this object to the value invalid(4) has the effect of invalidating the corresponding entry. If there is a test active when the entry is invalidated, the test will be aborted.

An existing instance of this object cannot be set to createRequest(2). This object may only be set to createRequest(2) when a new test instance is created. Immediately after completing the create operation, the proxy agent will set this object to underCreation(3).

Entries shall exist in the underCreation(3) state until the management station is finished configuring the entry and sets this object to valid(1) or aborts, setting this object to invalid(4). If the proxy agent determines that an entry has been in the underCreation(3) state for an abnormally long time, it may decide that the management station has crashed. If the proxy agent makes this decision, it will set the object to invalid(4)."

::= { nmsTestEntry 27 }

END

Device Enterprise MIB

Definition

B

NMS6700-DEV-MIB DEFINITIONS ::= BEGIN

— Title: COMSPHERE 6700 Device MIB for Customer Network
Management

—

— Copyright (C) 1996, Paradyne. All rights reserved.

—

— This file may be freely copied and distributed as
— long as no changes are made to it.

— The Device MIB contains the common device attributes.

— There is an instance of the Device MIB for each device in

— the 6700 network. Each device is identified by a unique

— community string. SNMP Management applications must use

— the community name to specify the device to access.

IMPORTS

TimeTicks

FROM RFC1155-SMI

DisplayString

FROM RFC1213-MIB

OBJECT-TYPE

FROM RFC-1212

nms-6700

FROM ATTP-ENTERPRISES;

UInteger32 ::=

INTEGER

—
— Enterprise Identification for COMSPHERE 6700 Device MIB
—

dev OBJECT IDENTIFIER ::= { nms-6700 2 } — Device MIB

— Device MIB Groups

devAdmin	OBJECT IDENTIFIER ::= { dev 1 } — administration
devIdentity	OBJECT IDENTIFIER ::= { dev 2 } — device identity
devStatus	OBJECT IDENTIFIER ::= { dev 3 } — status
devCirQual	OBJECT IDENTIFIER ::= { dev 4 } — circuit quality
devEIAStatus	OBJECT IDENTIFIER ::= { dev 5 } — EIA status
devExtLeads	OBJECT IDENTIFIER ::= { dev 6 } — external leads
devCommand	OBJECT IDENTIFIER ::= { dev 7 } — command

— Device Administration Group (dev 1)

devAdminName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Device name. This is the name used by the COMSPHERE 6700 NMS to identify the device.”

::= { devAdmin 1 }

devAdminAdpAddress OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Address of device in the ADP network (the network used by the COMSPHERE 6700 NMS to access the device).”

::= { devAdmin 2 }

devAdminModelType OBJECT-TYPE

SYNTAX INTEGER {

dial(1),

apl(2),

dsu(3),

dbm(4)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Device model type is dial(1) for COMSPHERE 3800 dial modems, apl(2) for COMSPHERE 3900 leased line modems, dsu(3) for COMSPHERE 3600 DSUs, and dbm(4) for

COMSPHERE

3600 Dial Backup Modules.”

::= { devAdmin 3 }

devAdminSite OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Device site name used by COMSPHERE 6700 NMS.”

::= { devAdmin 4 }

devAdminCabinet OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Device cabinet name used by COMSPHERE 6700 NMS."

::= { devAdmin 5 }

devAdminCarrier OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Device carrier name used by COMSPHERE 6700 NMS."

::= { devAdmin 6 }

devAdminCarrierSlot OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Device carrier slot number."

::= { devAdmin 7 }

devAdminCircuitName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..25))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Device circuit name."

::= { devAdmin 8 }

devAdminContact1 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..25))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Device contact string #1."

::= { devAdmin 9 }

devAdminContact2 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..25))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Device contact string #2."
 ::= { devAdmin 10 }

devAdminComment OBJECT-TYPE
 SYNTAX DisplayString (SIZE (0..40))
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Device comment string."
 ::= { devAdmin 11 }

devAdminDbmOption OBJECT-TYPE
 SYNTAX INTEGER {
 dbm-installed(1),
 dbm-not-installed(2)
 }
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "A value of dbm-installed(1) indicates that a dial backup
 module option is installed."
 ::= { devAdmin 12 }

devAdminMsdOption OBJECT-TYPE
 SYNTAX INTEGER {
 msd-installed(1),
 msd-not-installed(2)
 }
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "A value of msd-installed(1) indicates that a modem sharing
 device option is installed."
 ::= { devAdmin 13 }

devAdminMcmpOption OBJECT-TYPE
 SYNTAX INTEGER {
 mcmp-installed(1),
 mcmp-not-installed(2)
 }
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "A value of mcmp-installed(1) indicates that a multi control

multi point option is installed.”
::= { devAdmin 14 }

devAdminTdmOption OBJECT-TYPE

SYNTAX INTEGER {
 tdm-installed(1),
 tdm-not-installed(2)
}

ACCESS read-only

STATUS mandatory

DESCRIPTION

 “A value of tdm-installed(1) indicates that a time division
 multiplexer option is installed.”

::= { devAdmin 15 }

— Device Identity Group (dev 2)

devIdentityModel OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Model number.”

::= { devIdentity 1 }

devIdentityLineSpeed OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Line speed.”

::= { devIdentity 2 }

devIdentitySoftwareVersion OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Software version.”

::= { devIdentity 3 }

devIdentitySerialNumber OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Serial number.”

::= { devIdentity 4 }

devIdentityApplModuleID OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”APPL module (if applicable, otherwise length is zero).”

::= { devIdentity 5 }

devIdentityAccessModuleID OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

 "Access module (if applicable, otherwise length is zero)."

::= { devIdentity 6 }

devIdentityRestoralOption OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

 "Restoral option (if applicable, otherwise length is zero)."

::= { devIdentity 7 }

devIdentityConfiguration OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

 "Configuration (if applicable, otherwise length is zero)."

::= { devIdentity 8 }

devIdentityInternationalStrap OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

 "International strap."

::= { devIdentity 9 }

devIdentityHwPartNumber OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

 "Hardware part number."

::= { devIdentity 10 }

devIdentitySwPartNumber OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

 "Software part number."

::= { devIdentity 11 }

devIdentityOption1 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Option 1 (if applicable, otherwise length is zero).”

::= { devIdentity 12 }

devIdentityOption2 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Option 2 (if applicable, otherwise length is zero).”

::= { devIdentity 13 }

devIdentityOption3 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Option 3 (if applicable, otherwise length is zero).”

::= { devIdentity 14 }

devIdentityOption4 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Option 4 (if applicable, otherwise length is zero).”

::= { devIdentity 15 }

devIdentityOption5 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Option 5 (if applicable, otherwise length is zero).”

::= { devIdentity 16 }

devIdentityOption6 OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Option 6 (if applicable, otherwise length is zero).”
::= { devIdentity 17 }

— Device Status Group (dev 3)

devStatusConnectedDevice OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”The device name of the connected device. A zero length string will be returned if the information is unavailable.”

::= { devStatus 1 }

devStatusConnectTime OBJECT-TYPE

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Connect time (in seconds).”

::= { devStatus 2 }

devStatusAlert OBJECT-TYPE

SYNTAX UInteger32

ACCESS read-only

STATUS mandatory

DESCRIPTION

”The current alerts for the device. The bit position number starting with 0 and counting from the right (low order) bit will indicate each active Primary Alert. The Primary Alerts and their bit positions are defined as follows:

device-fail(0),
config-change-notify(1),
test-mode(2),
disabled(3),
out-of-threshold(4),
facility-alarm(5),
external-alarm(6),
streaming-terminal(7),
access-security(8),
dial-backup-active(9),
dte-alarm(10),
subnormal-speed(11),
primary-channel-interrupt(12),
firmware-downloading(13),
make-busy-mode(14),

```
    service-line(15),
    non-answering-modem(16),
    short-holding-time-modem(17),
    sub-tree-truncation(18),
    tdm-failure-good-apl(19),
    trib-timeout(20),
    dial-tone(21),
    redundant-power(22),
    no-response(31)"
 ::= { devStatus 3 }
```

devStatusAlertDesc OBJECT-TYPE

```
SYNTAX DisplayString (SIZE (0..255))
```

```
ACCESS read-only
```

```
STATUS mandatory
```

DESCRIPTION

"Contains current Primary Alerts for the device, in text format, delimited by semicolons (;). The following abbreviations are used for the primary alerts:
Fail, CfgChng, Test, Disab, Thresh, FacAlrm, ExtAlrm, StrTerm, AccSec, DialBkUp, DteAlrm, SubSpeed, PriChIr, FwDnLd, MkBusy, Service, NoAns, ShrtHld, SubTrunc, TdmFail(GoodApl), TribTO, DTone, Power, NoResp"

```
 ::= { devStatus 4 }
```

devStatusState OBJECT-TYPE

```
SYNTAX INTEGER {
```

```
    idle-or-leased(1),
    ring-indicate(2),
    answering(3),
    talk-mode(4),
    off-hook(5),
    dialing(6),
    remote-ringing(7),
    on-line(8),
    dial-backup(9),
    dial-standby(10)
}
```

```
ACCESS read-only
```

```
STATUS mandatory
```

DESCRIPTION

"The current state of the device."

```
 ::= { devStatus 5 }
```

devStatusSpeed OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”The current data rate of the device, in bits per second.

A value of zero indicates that the device is training.”

::= { devStatus 6 }

devStatusCtrlTrib OBJECT-TYPE

SYNTAX INTEGER {

local-control(1),

remote-control(2),

tributary(3),

other(4)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Indicates the position of the device in the network.”

::= { devStatus 7 }

devStatusConfigType OBJECT-TYPE

SYNTAX INTEGER {

leased(1),

dial(2)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Indicates the configured mode of the device.”

::= { devStatus 8 }

devStatusPollingState OBJECT-TYPE

SYNTAX INTEGER {

active(1),

inactive(2),

inventory(3),

suspended(4)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Indicates the polling state of the device in the network.”
::= { devStatus 9 }

— Circuit Quality Group (dev 4)

- Type Definition for Circuit Quality Validity indicators
- Validity is not-used(6) if not supported by device type.

```
CirQualValidity ::=
    INTEGER {
        valid(1),
        valid-greater-than(2),
        valid-less-than(3),
        not-valid-for-modulation-mode(4),
        not-valid-for-multipoint-mode(5),
        not-available-yet(6),
        count-overflow(7),
        not-used(8)
    }
```

— Bit Error Rate

```
devCirQualBitErrorRateValidity OBJECT-TYPE
    SYNTAX CirQualValidity
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Validity of the devCirQualBitErrorRate object."
    ::= { devCirQual 1 }
```

```
devCirQualBitErrorRate OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Bit error rate. Expressed as errors per billion (1E9)
        bits."
    ::= { devCirQual 2 }
```

— Mean Squared Error

```
devCirQualMeanSquareErrorValidity OBJECT-TYPE
    SYNTAX CirQualValidity
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
```

"Validity of the devCirQualMeanSquareError object."
 ::= { devCirQual 3 }

devCirQualMeanSquareError OBJECT-TYPE

 SYNTAX INTEGER
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Mean squared error."
 ::= { devCirQual 4 }

— Receive Level

devCirQualReceiveLevelValidity OBJECT-TYPE

 SYNTAX CirQualValidity
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Validity of the devCirQualReceiveLevel object."
 ::= { devCirQual 5 }

devCirQualReceiveLevel OBJECT-TYPE

 SYNTAX INTEGER
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Receive level. Units are (dBm *10). Possible range is
 -204.7 to +204.7 dBm."
 ::= { devCirQual 6 }

— 1004 Hz Loss

devCirQualHzLoss1004Validity OBJECT-TYPE

 SYNTAX CirQualValidity
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Validity of the devCirQualHzLoss1004 object."
 ::= { devCirQual 7 }

devCirQualHzLoss1004 OBJECT-TYPE

 SYNTAX INTEGER
 ACCESS read-only

STATUS mandatory
DESCRIPTION
 "1004 Hz loss."
::= { devCirQual 8 }

— Signal to Noise Ratio

devCirQualSignalToNoiseValidity OBJECT-TYPE
SYNTAX CirQualValidity
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Validity of the devCirQualSignalToNoise object."
::= { devCirQual 9 }

devCirQualSignalToNoise OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Signal to noise ratio. Units are (dB * 10). Possible
 range is 0.0 to 102.3 dB."
::= { devCirQual 10 }

— Phase Jitter 0–20 Hz

devCirQualPhaseJitter20Validity OBJECT-TYPE
SYNTAX CirQualValidity
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Validity of the devCirQualPhaseJitter20 object."
::= { devCirQual 11 }

devCirQualPhaseJitter20 OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Phase jitter 0–20 Hz. Units are (degrees * 10). Possible
 range 0.0 to 33 degrees."
::= { devCirQual 12 }

— Phase Jitter 20–300 Hz

devCirQualPhaseJitter300Validity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualPhaseJitter300 object.”

::= { devCirQual 13 }

devCirQualPhaseJitter300 OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Phase jitter 20–300 Hz. Units are (degrees * 10).
Possible range 0.0 to 33 degrees.”

::= { devCirQual 14 }

— Positive Attenuation

devCirQualPositiveAttenuationValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualPositiveAttenuation
object.”

::= { devCirQual 15 }

devCirQualPositiveAttenuation OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Positive attenuation.”

::= { devCirQual 16 }

— Negative Attenuation

devCirQualNegativeAttenuationValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualNegativeAttenuation object.”

::= { devCirQual 17 }

devCirQualNegativeAttenuation OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Negative attenuation.”

::= { devCirQual 18 }

— Envelope Delay

devCirQualEnvelopeDelayValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualEnvelopeDelay object.”

::= { devCirQual 19 }

devCirQualEnvelopeDelay OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Envelope delay.”

::= { devCirQual 20 }

— Frequency Offset

devCirQualFrequencyOffsetValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualFrequencyOffset object.”

::= { devCirQual 21 }

devCirQualFrequencyOffset OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Frequency offset. Units are (Hz * 10). Possible range
-51.1 to +51.1 Hz."

::= { devCirQual 22 }

— Non Linear Distortion

devCirQualNonLinearDistortionValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Validity of the devCirQualNonLinearDistortion
object."

::= { devCirQual 23 }

devCirQualNonLinearDistortion OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Non-linear distortion. Units are negative tenths of dB.
Possible range -0.0 to -102.3 dB. Example: -20 dB is
encoded as integer -200."

::= { devCirQual 24 }

— Retrains

devCirQualRetrainsValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Validity of the devCirQualRetrains object."

::= { devCirQual 25 }

devCirQualRetrains OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Retrains. Integer value represents the number of retrain events during the last 15 minutes. Possible range is 0 to 4095.”

::= { devCirQual 26 }

— Blown Startups

devCirQualBlownStartupsValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualBlownStartups object.”

::= { devCirQual 27 }

devCirQualBlownStartups OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Blown startups.”

::= { devCirQual 28 }

— Gain Hits

devCirQualGainHitsValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualGainHits object.”

::= { devCirQual 29 }

devCirQualGainHits OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Gain hits. Integer value represents the number of gain hit events during last 15 minutes. Possible range: 0 to 4095.”

::= { devCirQual 30 }

— Phase Hits

devCirQualPhaseHitsValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualPhaseHits object.”

::= { devCirQual 31 }

devCirQualPhaseHits OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Phase hits. Integer value represents the number of phase hit events during the last 15 minutes. Possible range is 0 to 4095.”

::= { devCirQual 32 }

— Impulse Noise

devCirQualImpulseNoiseValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualImpulseNoise object.”

::= { devCirQual 33 }

devCirQualImpulseNoise OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Impulse noise. Integer value represents the number of impulse noise events during the last 15 minutes. Possible range is 0 to 4095.”

::= { devCirQual 34 }

— Dropouts

devCirQualDropoutsValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualDropouts object.”

::= { devCirQual 35 }

devCirQualDropouts OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Dropouts. Integer value represents the number of dropout events during the last 15 minutes. Possible range is 0 to 4095.”

::= { devCirQual 36 }

— Line Quality

devCirQualLineQualityValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualLineQuality object.”

::= { devCirQual 37 }

devCirQualLineQuality OBJECT-TYPE

SYNTAX INTEGER {

 excellent(1),

 good(2),

 fair(3),

 poor(4),

 no-signal(5)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Line (signal) quality.”

::= { devCirQual 38 }

— Near End Echo

devCirQualNearEndEchoValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualNearEndEcho object.”

::= { devCirQual 39 }

devCirQualNearEndEcho OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Near end echo. Units are (dB *10). Possible range
0.0 to -102.4 dB.”

::= { devCirQual 40 }

— Far End Echo

devCirQualFarEndEchoValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualFarEndEcho object.”

::= { devCirQual 41 }

devCirQualFarEndEcho OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Far end echo. Units are (dB * 10). Possible range
0.0 to -102.4 dB.”

::= { devCirQual 42 }

— Far End Delay

devCirQualFarEndDelayValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Validity of the devCirQualFarEndDelay object.”

::= { devCirQual 43 }

devCirQualFarEndDelay OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Far end delay. Units are in milliseconds. Possible range
0 to 4095 ms."

::= { devCirQual 44 }

— Echo Frequency Offset

devCirQualEchoFreqOffsetValidity OBJECT-TYPE

SYNTAX CirQualValidity

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Validity of the devCirQualEchoFreqOffset object."

::= { devCirQual 45 }

devCirQualEchoFreqOffset OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Echo frequency offset. Units are tenths of Hertz.
Possible range -51.1 to 51.1 Hz."

::= { devCirQual 46 }

— EIA Status Group (dev 5)

— Textual Convention for EIA Status Integer

```
EIAStatus ::=
    INTEGER {
        off(1),
        on(2),
        unsupported-lead(3),
        lead-changing(4)
    }
```

— Number of Ports (Number of Entries in EIA Status Table)

```
devEIAStatusPorts OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Number of ports present in the device and the number
        of entries in the devEIAStatusTable."
    ::= { devEIAStatus 1 }
```

— EIA Status Table

```
devEIAStatusTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF DevEIAStatusEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "Table of EIA status values for each port."
    ::= { devEIAStatus 2 }
```

```
devEIAStatusEntry OBJECT-TYPE
    SYNTAX  DevEIAStatusEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A port entry in the EIA status table."
    INDEX   { devEIAStatusIndex }
    ::= { devEIAStatusTable 1 }
```

— Layout of one entry in devEIAStatusTable.

```
DevEIAStatusEntry ::=
    SEQUENCE {
        devEIAStatusIndex
            INTEGER,
        devEIAStatusCode
            OCTET STRING,
        devEIAStatusDTR
            EIAStatus,
        devEIAStatusTD
            EIAStatus,
        devEIAStatusRD
            EIAStatus,
        devEIAStatusDSR
            EIAStatus,
        devEIAStatusRTS
            EIAStatus,
        devEIAStatusCTS
            EIAStatus,
        devEIAStatusDCD
            EIAStatus,
        devEIAStatusTM
            EIAStatus,
        devEIAStatusDRI
            EIAStatus,
        devEIAStatusPIN13
            EIAStatus,
        devEIAStatusPIN19
            EIAStatus,
        devEIAStatusDRS
            EIAStatus,
        devEIAStatusLL
            EIAStatus,
        devEIAStatusRL
            EIAStatus,
        devEIAStatusRI
            EIAStatus,
        devEIAStatusDPR
            EIAStatus,
        devEIAStatusDLO
            EIAStatus,
        devEIAStatusCRQ
            EIAStatus,
```

```
    devEIAStatusACR
        EIAStatus,
    devEIAStatusDSC
        EIAStatus,
    devEIAStatusPND
        EIAStatus
}
```

devEIAStatusIndex OBJECT-TYPE

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
```

”The index of the entry in the devEIAStatusTable.
Note that the device hardware port number is used
as the index. Port numbers start with 1 and increment
monotonically up to number of ports in devEIAStatusPorts.”

```
::= { devEIAStatusEntry 1 }
```

devEIAStatusCode OBJECT-TYPE

```
SYNTAX  OCTET STRING (SIZE (6))
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
```

”This object contains the combined status of all 21 EIA
leads, in a compressed format which uses 2 bits for each
EIA lead, ordered left to right, as occurring below.
The status encoding is specified in binary as follows:
off=00; on=01; unsupported-lead=10; lead-changing=11.”

```
::= { devEIAStatusEntry 2 }
```

— 21 EIA status objects

devEIAStatusDTR OBJECT-TYPE

```
SYNTAX  EIAStatus
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
```

”DTR”

```
::= { devEIAStatusEntry 3 }
```

devEIAStatusTD OBJECT-TYPE

```
SYNTAX  EIAStatus
ACCESS  read-only
```

STATUS mandatory

DESCRIPTION

"TD."

::= { devEIASStatusEntry 4 }

devEIASStatusRD OBJECT-TYPE

SYNTAX EIASStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"RD."

::= { devEIASStatusEntry 5 }

devEIASStatusDSR OBJECT-TYPE

SYNTAX EIASStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"DSR."

::= { devEIASStatusEntry 6 }

devEIASStatusRTS OBJECT-TYPE

SYNTAX EIASStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"RTS."

::= { devEIASStatusEntry 7 }

devEIASStatusCTS OBJECT-TYPE

SYNTAX EIASStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"CTS."

::= { devEIASStatusEntry 8 }

devEIASStatusDCD OBJECT-TYPE

SYNTAX EIASStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"DCD."

::= { devEIASStatusEntry 9 }

devEIAStatusTM OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"TM."

::= { devEIAStatusEntry 10 }

devEIAStatusDRI OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"DRI."

::= { devEIAStatusEntry 11 }

devEIAStatusPIN13 OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Pin13."

::= { devEIAStatusEntry 12 }

devEIAStatusPIN19 OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Pin19."

::= { devEIAStatusEntry 13 }

devEIAStatusDRS OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"DRS."

::= { devEIAStatusEntry 14 }

devEIAStatusLL OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"LL."

::= { devEIAStatusEntry 15 }

devEIAStatusRL OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"RL."

::= { devEIAStatusEntry 16 }

devEIAStatusRI OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"RI."

::= { devEIAStatusEntry 17 }

devEIAStatusDPR OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"DPR."

::= { devEIAStatusEntry 18 }

devEIAStatusDLO OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"DLO."

::= { devEIAStatusEntry 19 }

devEIAStatusCRQ OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only

STATUS mandatory

DESCRIPTION

"CRQ."

::= { devEIAStatusEntry 20 }

devEIAStatusACR OBJECT-TYPE

SYNTAX EIAStatus

ACCESS read-only
STATUS mandatory
DESCRIPTION
 "ACR."
::= { devEIASStatusEntry 21 }

devEIASStatusDSC OBJECT-TYPE
 SYNTAX EIASStatus
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "DSC."
::= { devEIASStatusEntry 22 }

devEIASStatusPND OBJECT-TYPE
 SYNTAX EIASStatus
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "PND."
::= { devEIASStatusEntry 23 }

— External Leads Group (dev 6)

— Textual Convention for External Lead objects

ExternalLead ::=

```
    INTEGER {  
        lead-off(1),  
        lead-on(2)  
    }
```

devExtLeadsOutPin12 OBJECT-TYPE

SYNTAX ExternalLead

ACCESS read-write

STATUS mandatory

DESCRIPTION

”RS-232 output pin 12 (port 1).”

::= { devExtLeads 1 }

devExtLeadsOutPin13 OBJECT-TYPE

SYNTAX ExternalLead

ACCESS read-write

STATUS mandatory

DESCRIPTION

”RS-232 output pin 13 (port 1).”

::= { devExtLeads 2 }

devExtLeadsInPin19 OBJECT-TYPE

SYNTAX ExternalLead

ACCESS read-only

STATUS mandatory

DESCRIPTION

”RS-232 input pin 19 (port 1).”

::= { devExtLeads 3 }

devExtLeadsInPin23 OBJECT-TYPE

SYNTAX ExternalLead

ACCESS read-only

STATUS mandatory

DESCRIPTION

”RS-232 input pin 23 (port 1).”

::= { devExtLeads 4 }

— Command Group (dev 7)

— Dial an Entered or Stored Number

devCommandDial OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Setting this attribute will cause the modem to dial the entered number. To dial a stored number, the string must begin with ‘d’, followed by the call directory entry number. This variable will always return a zero-length string.”

::= { devCommand 1 }

— Disconnect Call or Service

devCommandDisconnect OBJECT-TYPE

SYNTAX INTEGER {
 disconnect-call(1),
 disconnect-service(2),
 unknown(3)
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Setting this attribute to disconnect-call(1) will cause the current call to be disconnected. Setting this attribute to disconnect-service(2) will cause service disconnect. This object will always return a value of unknown(3).”

::= { devCommand 2 }

— Change Busy State

devCommandBusy OBJECT-TYPE

SYNTAX INTEGER {
 set-busy(1),
 clear-busy(2),
 unknown(3)
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Setting this attribute to busy(1) will cause the device to busy out the attached analog phone line. Setting this attribute to clear(2) will cause the device to unbusy the analog phone line. This object will always return a value of unknown(3). To test the current busy state, refer to the devStatusAlert object.”

::= { devCommand 3 }

— Switch To Dial-Backup/Dial-Standby/Service-Line

devCommandSwitchTo OBJECT-TYPE

SYNTAX INTEGER {
 dial-backup(1),
 dial-standby(2),
 leased(3),
 service-line(4), — switch for mgmt mode
 unknown(5)
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Switch service mode to dial-backup(1), dial-standby(2), leased (3), or service-line(4). This object always returns a value of unknown(5).”

::= { devCommand 4 }

— Enable/Disable Device

devCommandEnable OBJECT-TYPE

SYNTAX INTEGER {
 enable-device(1),
 disable-device(2),
 unknown(3)
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Write to enable(1) or disable(2) the device.

This object always returns a value of unknown(3).”

::= { devCommand 5 }

— Reset Device

devCommandReset OBJECT-TYPE

SYNTAX INTEGER {
 software(1),
 hardware(2),
 unknown(3)
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

 "Invoke a software(1) or hardware(2) reset of the device.

 This object always returns a value of unknown(3)."

::= { devCommand 6 }

— Send Message to Front Panel

devCommandMessage OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..16))

ACCESS read-write

STATUS mandatory

DESCRIPTION

 "Write to send a message string to the device.

 This object always returns a null string."

::= { devCommand 7 }

END

Enterprise Trap Definitions

C

NMS6700-TRAP DEFINITIONS ::= BEGIN

— Title: COMSPHERE 6700 Trap Definitions

— Copyright (C) 1996, Paradyne. All rights reserved.

— This file may be freely copied and distributed as
— long as no changes are made to it.

IMPORTS

TRAP-TYPE

FROM RFC-1215

nms-6700-products

FROM ATTP-ENTERPRISES;

—
— Enterprise Identification for COMSPHERE 6700 SNMP Proxy Agent
Traps

devOid OBJECT IDENTIFIER ::= { nms-6700-products 2 } —
Device OID

—
— NOTE: All of the traps are subject to filtering that
— is controlled from a 6700 NMS workstation.

— Primary Alerts (1 – 32)

deviceFailure TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Device failure (major).”

::= 1

configChangeNotify TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Configuration change notify.”

::= 2

testMode TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Test mode.”

::= 3

deviceDisabled TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Device has been disabled.”

::= 4

vfThresholdExceeded TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”VF parameter threshold exceeded (minor).”

::= 5

facilityAlarm TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Facility alarm (major).”

::= 6

externalAlarm TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”External alarm.”

::= 7

streamingTerminal TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Streaming terminal (major)."

::= 8

accessSecurity TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Access security alarm (major)."

::= 9

dialBackup TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Dial backup active for APL."

::= 10

dteAlarm TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"DTE alarm (minor)."

::= 11

subNormalSpeed TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Sub-normal operating speed (minor)."

::= 12

primaryChannelInterrupted TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Primary channel interrupted. Reason for interruption is remote access using the primary channel. Source of interruption is front panel or network management."

::= 13

firmwareDownloading TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Firmware downloading."

::= 14

makeBusyMode TRAP-TYPE

ENTERPRISE devOid
DESCRIPTION
 "Make busy mode."
::= 15

serviceLine TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Service line."
::= 16

nonAnsweringModem TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Non-answering modem."
::= 17

shortHoldingTimeModem TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Short holding time modem."
::= 18

subTreeTruncation TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Sub-tree truncation."
::= 19

tdmFailure-goodAplLine TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "For 3800 and 3900 model types: good APL line detected
 while on backup (dial or dual lease line).
 For 3600 model types: TDM failure."
::= 20

tributaryTimeout TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Tributary timeout."
::= 21

dialTone TRAP-TYPE
 ENTERPRISE devOid

DESCRIPTION

"Dial tone."

::= 22

redundantPowerSupply TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Redundant power supply."

::= 23

noResponse TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"No response from the device."

::= 32

— System Events (101 – 117)

vfLoginOk TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"VF login OK. The calling party entered a correct VF (DTMF) password."

::= 101

userLoginOk TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"User Login OK. The calling party entered a correct user password"

::= 102

userLoginRejected–RetryFailed TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"User Login Rejected – Retry Failed. The calling party entered an excessive number of wrong passwords."

::= 103

userLoginRejected–PasswordTimeout TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"User Login Rejected – Password Timeout. The calling party failed to enter a user password within the allotted time."

::= 104

userLoginAborted–LineDisconnected TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"User Login Aborted – Line Disconnected. The calling party disconnected before user login completed."

::= 105

vfLoginRejected–PasswordInvalid TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"VF Login Rejected – Password Invalid. The calling party entered an incorrect VF (DTMF) password."

::= 106

vfLoginRejected–PasswordTimeout TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”VF Login Rejected – Password Timeout. The calling party failed to enter a VF (DTMF) password within the allotted time.”

::= 107

vfLoginAborted–LineDisconnected TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”VF Login Aborted – Line Disconnected. The calling party disconnected before VF (DTMF) login completed.”

::= 108

userLoginOk–MultiplePasswordRetries TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”User Login OK – Multiple Password Retries. The calling party has successfully completed user login, after multiple attempts.”

::= 109

userLoginRejected–InvalidCombination TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”User Login Rejected – Invalid Combination. The calling party entered an invalid user-id and password combination.”

::= 110

userLoginRejected–InvalidUserId TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”User Login Rejected – Invalid User ID. The calling party entered an invalid user-id (however the password was valid for the device).”

::= 111

userLoginRejected–InvalidTime TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”User Login Rejected – Invalid Access Time. The calling party entered a user-id for which access is not permitted at the current time.”

::= 112

userLoginHack TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"User Login Hack – Multiple Sequential Password Retries.
The password retry pattern indicates a hacker is attempting
to defeat login security."

::= 113

securityTableInvalid TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Device Security Table Invalid."

::= 114

securityDownloadFailed TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Security Download Failed."

::= 116

securityFrontPanelModification TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Security Front Panel Modification. Security configuration
has been modified from the device front panel."

::= 117

— Primary Alert Clear Notifications (201 – 232)

deviceFailureClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Device failure cleared.”

::= 201

configChangeNotifyClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Configuration change notify cleared.”

::= 202

testModeClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Test mode cleared.”

::= 203

deviceDisabledClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Device disabled cleared.”

::= 204

vfThresholdExceededClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”VF parameter threshold exceeded cleared.”

::= 205

facilityAlarmClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”Facility alarm cleared.”

::= 206

externalAlarmClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

”External alarm cleared.”

::= 207

streamingTerminalClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Streaming terminal cleared."

::= 208

accessSecurityClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Access security alarm cleared."

::= 209

dialBackupClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Dial backup active for APL cleared."

::= 210

dteAlarmClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"DTE alarm (minor)."

::= 211

subNormalSpeedClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Sub-normal operating speed cleared."

::= 212

primaryChannelInterruptedClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Primary channel interrupted cleared."

::= 213

firmwareDownloadingClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Firmware downloading cleared."

::= 214

makeBusyModeClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

"Make busy mode cleared."
::= 215

serviceLineClear TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Service line mode cleared."
::= 216

nonAnsweringModemClear TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Non-answering modem cleared."
::= 217

shortHoldingTimeModemClear TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Short holding time modem cleared."
::= 218

subTreeTruncationClear TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Sub-tree truncation cleared."
::= 219

tmdFailureClear-goodAplLineClear TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "For 3800 and 3900 model types: good APL line cleared.
 For 3600 model types: TDM failure cleared."
::= 220

tributaryTimeoutClear TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Tributary timeout cleared."
::= 221

dialToneClear TRAP-TYPE
 ENTERPRISE devOid
 DESCRIPTION
 "Dial tone cleared."
::= 222

redundantPowerSupplyClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

 "Redundant power supply cleared."

::= 223

noResponseClear TRAP-TYPE

ENTERPRISE devOid

DESCRIPTION

 "No response from the device cleared."

::= 232

END

Enterprise Common MIB

Definitions

D

ATTP-ENTERPRISES DEFINITIONS ::= BEGIN

IMPORTS

enterprises, IpAddress FROM RFC1155-SMI
OBJECT-TYPE FROM RFC-1212
DisplayString FROM RFC1213-MIB;

—
— Copyright (C) 1996, Paradyne. All rights reserved.
—
— Title: Paradyne Top Level Enterprise Definitions
—
— This file may be freely copied and distributed as
— long as no changes are made to it.
—

att-2 OBJECT IDENTIFIER ::= { enterprises 74 }
att-products OBJECT IDENTIFIER ::= { att-2 1 }
att-mgmt OBJECT IDENTIFIER ::= { att-2 2 }

—
— The NMS products subtrees are used for enterprise OIDs
—

— NMS products
paradyneNMS-products OBJECT IDENTIFIER ::= { att-products 13 }
nms-6700-products OBJECT IDENTIFIER ::= {
paradyneNMS-products 2 }

— products

paradyne-products OBJECT IDENTIFIER ::= { att-products 14 }
attp-cellRelay-products OBJECT IDENTIFIER ::= { paradyne-products 1 }
attp-31XX-products OBJECT IDENTIFIER ::= { paradyne-products 2 }
attp-36XX-products OBJECT IDENTIFIER ::= { paradyne-products 3 }

—
— The NMS mgmt subtrees are used for MIB objects
—

— NMS mgmt

paradyneNMS OBJECT IDENTIFIER ::= { att-mgmt 23 }
nms-6700 OBJECT IDENTIFIER ::= { paradyneNMS 5 }

— mgmt

paradyne OBJECT IDENTIFIER ::= { att-mgmt 24 }

attp-cellRelay OBJECT IDENTIFIER ::= { paradyne 1 }

attp-common OBJECT IDENTIFIER ::= { paradyne 2 }
attp-frontPanel OBJECT IDENTIFIER ::= { attp-common 1 }
attp-chassis OBJECT IDENTIFIER ::= { attp-common 2 }
attp-callDir OBJECT IDENTIFIER ::= { attp-common 3 }
attp-devStatus OBJECT IDENTIFIER ::= { attp-common 4 }
attp-devID OBJECT IDENTIFIER ::= { attp-common 5 }

attp-31XX OBJECT IDENTIFIER ::= { paradyne 3 }
attp-36XX OBJECT IDENTIFIER ::= { paradyne 4 }

—
— Device Identity
—

devIdentity OBJECT IDENTIFIER ::= { attp-devID 1 }

devIdentityModel OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Model Number.”

::= { devIdentity 1 }

devIdentityCustAssignedID OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Device identifier assigned by the customer.

Used for display in ASCII interfaces.”

::= { devIdentity 2 }

devIdentitySlotID OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Slot identifier for this device. NoSuchName is returned to the user if the slot identifier is not-applicable.”

::= { devIdentity 3 }

devIdentitySerialNumber OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..16))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Serial Number.”

::= { devIdentity 4 }

—
— Hardware Revision Table
—

devIdentityHWRevEntries OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Number of hardware revision entries in table.”

::= { devIdentity 5 }

devIdentityHWRevTable OBJECT-TYPE

SYNTAX SEQUENCE OF DevIdentityHWRevEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

”The hardware revision table contains the hardware revision for each CCA in the device.”

::= { devIdentity 6 }

devIdentityHWRevEntry OBJECT-TYPE

SYNTAX DevIdentityHWRevEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"An entry in the hardware revision table."

INDEX { devIdentityHWRevIndex }

::= { devIdentityHWRevTable 1 }

—
— Layout of one entry in the devIdentityHWRevTable
—

DevIdentityHWRevEntry ::=

SEQUENCE {

devIdentityHWRevIndex

INTEGER,

devIdentityHWRevCCA

DisplayString

}

devIdentityHWRevIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Hardware Revision table index."

::= { devIdentityHWRevEntry 1 }

devIdentityHWRevCCA OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"CCA revision number."

::= { devIdentityHWRevEntry 2 }

—
— Software Revision Table – Note, the 31xx and 36xx
— products may have only one entry in the table.
—

devIdentitySWRevEntries OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Number of software revision entries in table.”

::= { devIdentity 7 }

devIdentitySWRevTable OBJECT-TYPE

SYNTAX SEQUENCE OF DevIdentitySWRevEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

”The software revision table contains the software revision for every CCA in the device with a unique software module.”

::= { devIdentity 8 }

devIdentitySWRevEntry OBJECT-TYPE

SYNTAX DevIdentitySWRevEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

”An entry in the software revision table.”

INDEX { devIdentitySWRevIndex }

::= { devIdentitySWRevTable 1 }

—
— Layout of one entry in the devIdentitySWRevTable
—

DevIdentitySWRevEntry ::=

SEQUENCE {

devIdentitySWRevIndex

INTEGER,

devIdentitySWRevNumber

DisplayString,

devIdentitySWRevHWIndex

INTEGER

}

devIdentitySWRevIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Software Revision table index."

::= { devIdentitySWRevEntry 1 }

devIdentitySWRevNumber OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..16))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Software revision number for the CCA software module."

::= { devIdentitySWRevEntry 2 }

devIdentitySWRevHWIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Hardware Revision Table Index for this software revision."

::= { devIdentitySWRevEntry 3 }

—
— Device Health and Status
—

devStatus OBJECT IDENTIFIER ::= { attp-devStatus 1 }

devHealthAndStatus OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Health and Status of the device. Health & Status is represented as a concatenation of health and status message strings, where each message string is separated by a semi-colon. Refer to device-specific user documentation for a complete description of the Health and Status codes."

::= { devStatus 1 }

devSelfTestResults OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Self-test results. Self-test (or power-up test) results summarizes the test results of each CCA, where each CCA test result is separated by a semi-colon. Refer to device-specific user documentation for a complete description of the self test codes and messages.”

::= { devStatus 2 }

—

— Device Front Panel

—

— A device agent, if it supports the Front Panel MIB, must

— always support this group.

devFrontPanel OBJECT IDENTIFIER ::= { attp-frontPanel 1 }

devFrontPanelKeyPress OBJECT-TYPE

SYNTAX INTEGER {

doubleup(1),

up(2),

left(3),

right(4),

f1(5),

f2(6),

f3(7)

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Front panel keypress. Any other

value causes rejection of the Set request.

On read, the value returned is undefined.”

::= { devFrontPanel 1 }

devFrontPanelCursorDisplay OBJECT-TYPE

SYNTAX INTEGER {

hide(1),

show(2)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Front panel cursor display. CursorColumn and

CursorRow valid only when CursorDisplay = show.”

::= { devFrontPanel 2 }

devFrontPanelCursorRow OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Front panel cursor row (zero origin).”

::= { devFrontPanel 3 }

devFrontPanelCursorColumn OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Front panel cursor column (zero origin).”

::= { devFrontPanel 4 }

devFrontPanelDisplayRows OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Number of front panel display rows in the
devFrontPanelDisplayTable (default of 2 rows).”

::= { devFrontPanel 5 }

devFrontPanelDisplayColumns OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Number of front panel display columns in the
devFrontPanelDisplayTable (default of 16 columns).”

::= { devFrontPanel 6 }

devFrontPanelDisplayTable OBJECT-TYPE

SYNTAX SEQUENCE OF DevFrontPanelDisplayEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

”Contents of front panel display.”

::= { devFrontPanel 7 }

devFrontPanelDisplayEntry OBJECT-TYPE

SYNTAX DevFrontPanelDisplayEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

”A row entry in front panel display table.”

INDEX { devFrontPanelDisplayRowIndex }

::= { devFrontPanelDisplayTable 1 }

DevFrontPanelDisplayEntry ::=

SEQUENCE {

devFrontPanelDisplayRowIndex

INTEGER,

devFrontPanelDisplayRow

DisplayString

}

devFrontPanelDisplayRowIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Index of entry in front panel display table (zero origin).”

::= { devFrontPanelDisplayEntry 1 }

devFrontPanelDisplayRow OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..80))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Contents of a front panel display row.”

::= { devFrontPanelDisplayEntry 2 }

— Administration of Front Panel Group (attp-frontPanel 2)

—

— A device agent may choose not to support this group.

adminFrontPanel OBJECT IDENTIFIER ::= { attp-frontPanel 2 }

adminFrontPanelStatus OBJECT-TYPE

SYNTAX INTEGER {

free(1),

nms(2),

physicalFP(3),

emulationFP(4),

terminalFP(5)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Indicates who owns the front panel at this time.

If the status is free, then a keyPress set will successfully update the front panel and the status will change to nms. The agent is allowed to set the status back to free after a specified inter-keyPress timeout."

::= { adminFrontPanel 1 }

adminFrontPanelNMSOwner OBJECT-TYPE

SYNTAX IpAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

"If status is nms, then this is its IP address; otherwise it is undefined.

Note: It is possible for multiple NMS users to have the same IP address. Use adminFrontPanelNMSComment to further differentiate the user."

::= { adminFrontPanel 2 }

adminFrontPanelNMSComment OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..40))

ACCESS read-write

STATUS mandatory

DESCRIPTION

"It is suggested that this be maintained by a NMS application and that it contain information such as the owners name, address, phone number, etc."

::= { adminFrontPanel 3 }

adminFrontPanelCommand OBJECT-TYPE

SYNTAX INTEGER {
 release(1)
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Release changes the status to free. The agent is allowed to reject this value if the IpAddress of the set does not match adminFrontPanelNMSOwner.

The value returned on a GET is undefined.”
::= { adminFrontPanel 4 }

—
— Call Directory
—

devCallDir OBJECT IDENTIFIER ::= { attp-callDir 1 }

devCallDirEntries OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Number of directories in the call directory table."
::= { devCallDir 1 }

— The AT&T Paradyne device call directory table.

devCallDirTable OBJECT-TYPE
SYNTAX SEQUENCE OF DevCallDirEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "The call directory table contains telephone numbers
 stored in a device's call directory. The number of
 entries is given by the value of devCallDirEntries."
::= { devCallDir 2 }

devCallDirEntry OBJECT-TYPE
SYNTAX DevCallDirEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "An entry in the call directory table."
INDEX { devCallDirIndex }
::= { devCallDirTable 1 }

— Layout of one entry in devCallDirTable.

DevCallDirEntry ::=
SEQUENCE {
 devCallDirIndex
 INTEGER,

```
devCallDirPurpose
    INTEGER,
devCallDirId
    DisplayString,
devCallDirPhoneNumber
    DisplayString
}
```

devCallDirIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

”A unique value for each call directory. Its value ranges between 1 and the value of devCallDirEntries.”

::= { devCallDirEntry 1 }

devCallDirPurpose OBJECT-TYPE

SYNTAX INTEGER {

other (1), — none of the following

alarm-trap (2),

dial-back-up (3),

unknown (4) — 6700 default

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

”Call directory purpose.”

::= { devCallDirEntry 2 }

devCallDirId OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..15))

ACCESS read-only

STATUS mandatory

DESCRIPTION

”A textual string containing the call directory identifier. If this identifier is a number, then it may have the same value as devCallDirIndex.”

::= { devCallDirEntry 3 }

devCallDirPhoneNumber OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..40))

ACCESS read-write

STATUS mandatory

DESCRIPTION

”Call directory phone number.”

::= { devCallDirEntry 4 }

END

Index

C

- Call Directory MIB, 3-6
- COMSPHERE 6700 Series NMS Enterprise MIBs, 3-3

D

- Device MIB, 3-5
- device traps, 3-7
- DOS MIB, 3-6

F

- Front Panel MIB, 3-6

H

- hardware requirements, 2-1

I

- installation
 - preparation, 2-3
 - procedure, 2-5

M

- Management Information Bases
 - Call Directory MIB, 3-6
 - COMSPHERE 6700 Device MIB, 3-5
 - COMSPHERE 6700 NMS MIB, 3-5
 - COMSPHERE 6700 Series NMS Enterprise MIBs, 3-3
 - DOS MIB, 3-6
 - Front Panel MIB, 3-6
 - MIB-II Interface Group, 3-2
 - MIB-II System Group, 3-2
 - Windows MIB, 3-6
 - Workstation MIB, 3-6
- MIB-II, 3-1
 - System Group, 3-2
- MIB-II, Interface Group, 3-2

N

- NMS MIB, 3-5

P

- primary alert
 - clear notifications, 3-10
 - traps, 3-8

S

SNMP

- community views, 3-11
- overview, 1-1
- sample network topology, 1-3

SNMP traps, 3-7

SNMP Proxy Agent

- feature list, 1-2
- overview, 1-1
- package contents, 1-4
- software description, 1-2
- software requirements, 2-1
- supported network adapters, 2-2
- system event traps, 3-9

T

topology, sample, 1-3

traps

- device, 3-7
- primary alert, 3-8
- SNMP, 3-7
- system event, 3-9

W

Windows, MIB, 3-6

Workstation MIB, 3-6