



Norand[®] Management Information Bases
REFERENCE MANUAL



P/N 977-051-002
Revision B
December 2000

► NOTICE

The information contained herein is proprietary and is provided solely for the purpose of allowing customers to operate and service Intermecc manufactured equipment and is not to be released, reproduced, or used for any other purpose without written permission of Intermecc.

Disclaimer of Warranties. The sample source code included in this document is presented for reference only. The code does not necessarily represent complete, tested programs. The code is provided **“AS IS WITH ALL FAULTS.” ALL WARRANTIES ARE EXPRESSLY DISCLAIMED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.**

We welcome your comments concerning this publication. Although every effort has been made to keep it free of errors, some may occur. When reporting a specific problem, please describe it briefly and include the book title and part number, as well as the paragraph or figure number and the page number.

Send your comments to:
Intermecc Technologies Corporation
Publications Department
550 Second Street SE
Cedar Rapids, IA 52401

ANTARES, INTERMECC, NORAND, NOR*WARE, PEN*KEY, ROUTEPOWER, TRAKKER, and TRAKKER ANTARES are registered trademarks and ENTERPRISE WIRELESS LAN, INCA, Mobile Framework, TE 2000, UAP, and UNIVERSAL ACCESS POINT are trademarks of Intermecc Technologies Corporation.

© 1996 Intermecc Technologies Corporation. All rights reserved.

Acknowledgments

Portions of this product contain software which is licensed from and is copyright by Epilogue Technology Corporation, 1988-1995, all rights reserved.

ActiveX, *Microsoft*, *MS*, and *MS-DOS*, *Windows*, and *Windows NT* are registered trademarks and *MSDN*, *Visual Basic*, *Visual C++*, and *Windows for Pen* are trademarks of Microsoft Corporation.

AS/400, *IBM*, *IBM PC*, *Micro Channel*, *PS/2*, and *OS/2* are registered trademarks and *AIX* and *Presentation Manager* are trademarks of International Business Machines Corporation.

DEC, *Digital*, *VAX*, *VMS*, *VT*, and *VT220* are registered trademarks of Digital Equipment Corporation.

HP is a registered trademark of Hewlett-Packard Company

Intel is a registered trademark of Intel Corporation.

Novell and *UNIX* are registered trademarks of Novell Inc.

XEROX is a registered trademark of Xerox Corporation.

This software and documentation are based in part on HP OpenView under license from the Hewlett-Packard Company.

CONTENTS



Quick Reference

Access to Norand Management Information on 6710 Access Point and RC4030E Gateway	1
MIB II Notes for Release 1.x and 1.2x	2

SECTION 1

Introduction

Before You Begin	1-1
Product Contents	1-1
About this Product	1-2
Getting Started	1-2
Open Wireless LAN Management Overview	1-3
Customer Support	1-4
Customer Support Center	1-4
Web Site	1-4
Bulletin Board Service	1-4

SECTION 2

Network Management Basics

Network Management Functions	2-1
Network Management Components	2-1
Simple Network Management Protocol	2-2
Management Information Base	2-2
MIB-II Information	2-4
ASN.1 Definition — System.sysDescr of MIB-II	2-4
SNMP Access Control	2-5
Traps	2-5

SECTION 3

Supported Management Information

MIB-II Information	3-1
NORAND MIB Information	3-2

SECTION 4

Norand Open Wireless LAN MIB 6710 Access Point

Access Point MIB Outline	4-2
Product OIDs	4-2
System Information	4-2
Interface Information	4-4
SNMP Version 1 Configuration Group	4-8
Bridging Parameters	4-9
Control Groups	4-12
Access Point MIB Definitions	4-13

SECTION 5**Norand Open Wireless LAN MIB RC4030E Gateway**

Gateway MIB Outline	5-2
Product OIDs	5-2
System Information	5-2
SNMP Version 1 Configuration Group	5-4
Transport Groups	5-5
Application Layer Group	5-8
Control Group	5-11
Gateway MIB Definitions	5-12

SECTION 6**Norand Open Wireless LAN MIB 6910 Integrated Gateway/Access Point**

Integrated Gateway/Access Point MIB Outline	6-2
Product OIDs	6-2
System Information	6-3
Interface Information	6-5
SNMP Version 1 Configuration Group	6-9
Bridging Parameters	6-10
Transport Groups	6-13
Application Layer Groups	6-16
Control Groups	6-19
Integrated Gateway/Access Point MIB Definitions	6-21

FIGURES

Figure 2-1 SNMP Network Management Architecture	2-2
---	-----

TABLES

Table 2-1 Major Groups of MIB-II	2-4
Table 3-1 Supported MIB-II Information	3-1
Table 3-2 6710 Access Point MIB Information	3-2
Table 3-3 RC4030E Gateway MIB Information	3-2
Table 3-4 6910 Integrated Gateway/Access Point MIB Information	3-3
Table 4-1 AP MIB Directory	4-1
Table 4-2 products GROUP	4-2
Table 4-3 hw GROUP	4-2
Table 4-4 fsinfo GROUP	4-3
Table 4-5 segment GROUP	4-3
Table 4-6 dir GROUP	4-3
Table 4-7 criticalErrors GROUP	4-4
Table 4-8 nifx GROUP	4-4
Table 4-9 portState GROUP	4-5
Table 4-10 portStats GROUP	4-5
Table 4-11 ptxq GROUP	4-6
Table 4-12 pmsg GROUP	4-7
Table 4-13 community TABLE	4-8
Table 4-14 trapTarget TABLE	4-8
Table 4-15 rt GROUP	4-9
Table 4-16 brg GROUP	4-10
Table 4-17 addr GROUP	4-10
Table 4-18 brgState GROUP	4-10
Table 4-19 bridgeStats GROUP	4-11
Table 4-20 powerUp GROUP	4-12
Table 4-21 softwareDownload GROUP	4-12
Table 5-1 GW MIB Directory	5-1

Table 5-2 products GROUP	5-2
Table 5-3 hw GROUP	5-2
Table 5-4 dir GROUP	5-3
Table 5-5 criticalErrors GROUP	5-3
Table 5-6 community TABLE	5-4
Table 5-7 trapTarget TABLE	5-4
Table 5-8 wst GROUP	5-5
Table 5-9 hlit GROUP	5-6
Table 5-10 wcbst GROUP	5-6
Table 5-11 gs GROUP	5-8
Table 5-12 rcbs GROUP	5-9
Table 5-13 hd GROUP	5-10
Table 5-14 td GROUP	5-10
Table 5-15 ahost GROUP	5-11
Table 5-16 powerUp GROUP	5-11
Table 6-1 IGAP MIB Directory	6-1
Table 6-2 products GROUP	6-2
Table 6-3 hw GROUP	6-3
Table 6-4 fsinfo GROUP	6-3
Table 6-5 segment GROUP	6-4
Table 6-6 dir GROUP	6-4
Table 6-7 criticalErrors GROUP	6-4
Table 6-8 nifx GROUP	6-5
Table 6-9 portState GROUP	6-6
Table 6-10 portStats GROUP	6-6
Table 6-11 ptxq GROUP	6-7
Table 6-12 pmsg GROUP	6-8
Table 6-13 community TABLE	6-9
Table 6-14 trapTarget TABLE	6-9
Table 6-15 rt GROUP	6-10
Table 6-16 brg GROUP	6-11
Table 6-17 addr GROUP	6-11
Table 6-18 brgState GROUP	6-11
Table 6-19 bridgeStats GROUP	6-12
Table 6-20 wst GROUP	6-13
Table 6-21 hlit GROUP	6-14
Table 6-22 wcbst GROUP	6-14
Table 6-23 gs GROUP	6-16
Table 6-24 rcbs GROUP	6-17
Table 6-25 hd GROUP	6-18
Table 6-26 td GROUP	6-18
Table 6-27 ahost GROUP	6-19
Table 6-28 powerUp GROUP	6-19
Table 6-29 softwareDownload GROUP	6-20

GLOSSARY**INDEX**

Quick Reference



Access to Norand Management Information on 6710 Access Point and RC4030E Gateway

Access to Norand[®] Management Information is obtained with the proper COMMUNITY name. Intermecc Technologies Corporation provides three levels of access. This table outlines the levels with the required community name.

► **NOTE:** *Community Strings are case-sensitive.*

Community String	Access Type	Description of Access Type
public	READ-ONLY	May read MIB objects, but not write or change values. EXCLUSIONS: Will not be able to read or write the Norand Community Table.
CR52401	READ-WRITE	May read MIB objects. May write to MIB objects that have read-write access. EXCLUSIONS: Will not be able to read or write the Norand Community Table.
secret	SUPER-USER	May read MIB objects. May write to MIB objects with read-write access. Can read and write the Norand Community Table.

The Norand Community Table

(`norand.manage.norandNet.nSNMP.v1Config.communityTable`) stores the names of the community strings for each community or access group. These three records may be viewed and modified if used with the SUPER-USER community. There is a maximum, allowing for three levels of access. Records may be added or deleted via setting the `communityStatus` object to enable, disable, or delete. The first row in the `CommunityTable` is reserved for the SUPER-USER community definition. This record is not removable. This is a fixed record to ensure read-write access to the MIBs on the device. Note the `communityName` for the first record can be changed to ensure end-user control of security for the device. See *Norand MIB* for `communityTable` object descriptions.

MIB II Notes for Release 1.x and 1.2x

System Group: Three fields in the MIB II system group are writeable. Those fields are: `sysContact`, `sysName`, and `sysLocation`. It is important that these values be preserved in case the device is powered (off and on) or rebooted. The following lists the number of characters for each field that will be preserved in the event of a device power (off and on) or reboot.

- ▶ `sysContact`: 31 characters
- ▶ `sysName`: 31 characters
- ▶ `sysLocation`: 39 characters

Interfaces Group: The `ifTable.ifAdminStatus` object is read-write accessible. For this initial release of MIB II for the 6710 Access Point and RC4030E Gateway, this functionality has not been enabled.

IP Forwarding: IP Forwarding is disabled for this release of the 6710 Access Point and RC4030E Gateway. Therefore, the MIB II `ipForwarding` object is not changeable.

Section 1

Introduction

Before You Begin

This publication contains Management Information Bases (MIBs) as they existed in 1996. This manual will not be updated to include more recent information.

Additional MIBs can be found in the following product-related publications:

- ▶ *6710 Access Point User's Guide, Revision C* P/N: 961-047-081
- ▶ *RC4030E Gateway User's Guide, Revision B* P/N: 961-047-087
- ▶ *6910 Integrated and 6910 Telnet Gateway / Access Point User's Guide, Revision A* P/N: 961-047-122

Product Contents

If you purchased the HP OpenView for Windows and NORAND[®] MIBs, you should have the following:

- ▶ HP OpenView for Windows V7.2 (3 diskettes)
- ▶ FTP Software's PC/TCP TCP/IP stack (3 diskettes)
- ▶ *Norand Open Wireless LAN with HP OpenView for Windows User's Guide* (P/N: 961-051-009)
- ▶ *Norand Management Information Bases Reference Manual* (P/N: 977-051-002)

If you purchased the NORAND MIBs without the HP OpenView for Windows, you should have the following:

- ▶ Norand Open Wireless LAN Management Information Bases and Icons (1 diskette)
- ▶ *Norand Management Information Bases Reference Manual* (P/N: 977-051-002)

About this Product

Intermec Technologies Corporation packaged the NORAND Open Wireless LAN MIBs and device icons to provide basic network management capability for the Open Wireless LAN. When installed onto your management platform, NORAND Open Wireless LAN device icons will be visible from your management platform.

The NORAND Open Wireless LAN MIBs are found on the OWLMIBS diskette. You need to load these MIBs into your management platform to query NORAND Open Wireless LAN devices for these management objects.

If you have purchased the HP OpenView for Windows from Intermec Technologies Corporation (P/N: 215-575-001 or 215-576-001), then the NORAND MIBs are found on the OpenView diskette.

Getting Started

- **NOTE:** *If you have purchased HP OpenView for Windows (Open Wireless LAN version) from Intermec Technologies Corporation, skip this section. See the Norand Open Wireless LAN with HP OpenView for Windows User's Guide P/N: 961-051-009.*

Install the NORAND MIBs into your management platform. These files are found on the OWLMIBS distribution diskette, P/N: 210-739-001. There are three subdirectories: MIBS, SYMBOLS, and DEVICES:

- **MIBS:** ASN.1 MIB files for the NORAND Open Wireless LAN, that can load onto an SNMP Network Management Station. Once loaded, you can query the Open Wireless LAN devices for Open Wireless LAN MIB objects. Install the MIBs *in this order*:
 - a. **RFC1213.MIB:** Contains MIB II.
 - b. **RFC1398.MIB:** The standard Ethernet MIB (dot3).
 - c. Any of the following based on what device you are employing:
 - **NOROWL.MIB:** The NORAND proprietary Open Wireless LAN specific MIB definition for the 6910 Integrated Gateway and Access Point.
 - **OWLAP.MIB:** A subset of NOROWL.MIB, with MIB objects supported by the 6710 Access Point.
 - **OWLGW.MIB:** A subset of NOROWL.MIB, with MIB objects supported by the RC4030E Gateway.

- **NOTE:** *If you are using HP OpenView for Windows, use the OpenView Control/SNMP Manager/Manage Database menu items to add the above listed MIBs to the HP OpenView MIB database.*
*If you are **not** using HP OpenView for Windows, consult your Network Management Station user's guide for instructions on adding these MIBs.*

- ▶ **SYMBOLS:** Icons for the Open Wireless LAN devices. These icons can be incorporated into a Network Management Station for the NORAND icon, representing Intermecc[®] devices, to appear when the network is discovered and drawn.
 - ▶ If you are using HP OpenView for Windows, copy the .ICO files from this directory into the HP OpenView SYMBOLS directory:
`COPY *.ICO C:\OV\SYMBOLS`
 - ▶ If you are *not* using HP OpenView for Windows, consult your Network Management Station user's guide for instructions on how to incorporate new devices and new icons into the management station.

▶ **NOTE:**

If you purchased HP OpenView for Windows (Open Wireless LAN version) from Intermecc Technologies Corporation, these icons are incorporated into HP OpenView.

- ▶ **DEVICES:** Data that *must* get added to the HP OpenView DEVICES file includes the OID of the Intermecc device and OpenView device type. The HP OpenView DEVICES file is found in the C:\OV\OVFILES directory. Edit the DEVICES file and copy (cut and paste) the DEVICES.TXT contents into C:\OV\OVFILES\DEVICES. The DEVICES file enables the NORAND symbols to appear on the discovery map.

▶ **NOTE:**

If you purchased HP OpenView for Windows (Open Wireless LAN version) from Intermecc Technologies Corporation, the DEVICES file is already augmented to incorporate the Intermecc devices.

Open Wireless LAN Management Overview

The NORAND Open Wireless LAN is manageable using SNMP (Simple Network Management Protocol). The Open Wireless LAN Fixed-End devices (Access Points and Gateway devices) are equipped with an SNMP agent which makes the devices accessible for SNMP managers, such as HP OpenView for Windows.

Each fixed-end Open Wireless LAN device maintains a set of management objects specific to the operation of the device. Descriptions of these objects can be found in this reference manual.

- ▶ **NOROWL.MIB** contains all of the management objects found on the 6910 Integrated Gateway/Access Point fixed-end device. It also contains objects supported by the RC4030E Gateway and the 6710 Access Point.
- ▶ **OWLAP.MIB** contains only those management objects supported by the Open Wireless LAN 6710 Access Point.
- ▶ **OWLGW.MIB** contains only those management objects supported by the Open Wireless LAN RC4030E Gateway.
- ▶ **RFC1398.MIB** is the standard Ethernet MIB.
- ▶ **RFC1213.MIB** is the standard MIB II.

Each Open Wireless LAN fixed-end device (Access Point and Gateway) supports MIB II (RFC1213.MIB) and the Ethernet MIB (RFC1398.MIB).

Customer Support

Customer Support Center

The Intermec Customer Support Center (technical support) telephone number is 800-755-5505 (U.S.A. or Canada) or 425-356-1799. The facsimile number is 425-356-1688. Email is support@intermec.com.

If you email or fax a problem or question include the following information in your message: your name, your company name and address, phone number and email to respond to, and specific problem description or question.

Web Site

The Customer Support File Libraries, including Hot Tips and Product Awareness Bulletins, are available via the Intermec Product Support page at this URL: <http://norbbs.norand.com/index.htm>. New users can sign up for a new account on this page.

PDF versions of Intermec manuals can be found at this URL: <http://corp.intermec.com/manuals/english.htm>.

Bulletin Board Service

The Customer Support Bulletin Board (BBS), by Intermec Technologies Corporation, has software and documentation:

- ▶ **Phone number:** 319-369-3515 (14.4 Kbps modem)
319-369-3516 (28.8 Kbps modem)
- ▶ **Protocol:** Full duplex, ANSI or ANSI-BBS; 300 to 28,800 bps; v.32bis; 8 bits, no parity, 1 stop bit. *For high-speed modems, disable XON/XOFF and enable RTS/CTS.*

This is the same location available via the web site. If your web access uses high-speed phone lines, the web interface provides a faster response.

Network Management Basics



A goal of network management is to detect and correct network problems before the user is aware there is a problem.

Network Management Functions

- ▶ Fault: Enables the detection, isolation, and correction of abnormal operation of the network environment.
- ▶ Configuration: Identifies and exercises control over managed devices to assist in the continuous operation of network services.
- ▶ Performance: Evaluates managed device behavior and the effectiveness of communication activities.
- ▶ Accounting: Establishes costs and tracks the use of network resources.
- ▶ Security: Monitors and controls access to networks and network devices.

Network Management Components

- ▶ Management Station: Provides an interface for monitoring and controlling the network.
- ▶ Management Agent: Resides on the managed device and responds to management station information requests. May provide unsolicited information about extraordinary events to the station.
- ▶ Management Information Base: A collection of management devices used as access points at the management agent for the management station.
- ▶ Network Management Protocol: Allows management stations and management agents to exchange information.

Simple Network Management Protocol

Simple Network Management Protocol (SNMP) provides a set of rules which cover how information can be collected from each managed device using a common protocol.

SNMP is designed for Transmission Control Protocol over Internet Protocol (TCP/IP) networks.

- ▶ Uses User Datagram Protocol (UDP) as its transport layer.
- ▶ Uses Internet Protocol (IP) as the network layer.

There are three basic operations in SNMP:

- ▶ GET Retrieves management information
- ▶ SET Manipulates management information
- ▶ TRAP Reports extraordinary events

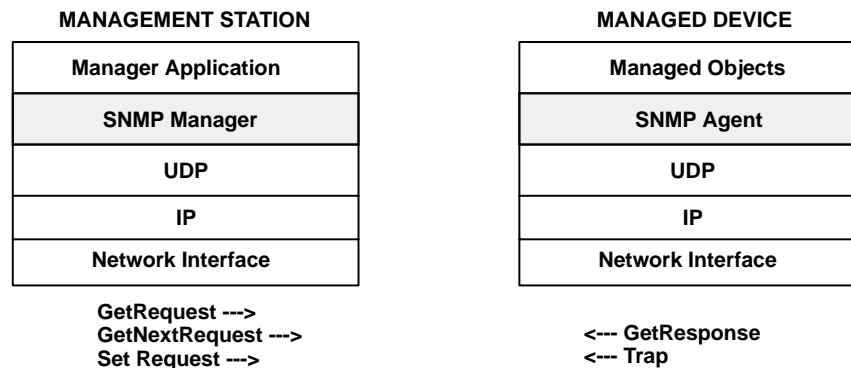


Figure 2-1

SNMP Network Management Architecture

Management Information Base

The Management Information Base (MIB) is a database that contains information about the elements to be managed. The information identifies the management element and specifies its type and access mode (Read-Only, Read-Write)

- ▶ **NOTE:** *Elements to be managed are represented by objects. The MIB is a structured collection of such objects.*

Each object has a unique identifier called an Object Identifier (OID). OIDs consist of a sequence of integer values represented in dot notation.

- EXAMPLE:** The internet OID = 1.3.6.1.

Objects are stored in a tree structure. OIDs are assigned based on the position of the object in the tree.

MIBs are written in ASN.1 (Abstract Syntax Notation.1) – a machine independent data definition language.

EXAMPLE: MIB Tree
iso (1)
 org (3)
 dod (6)
 internet (1)
 directory (1)
 mgmt (2)
 mib-2 (1)
 system (1)
 interfaces (2)
 at (3)
 ip (4)
 icmp (5)
 tcp (6)
 udp (7)
 egp (8)
 cmot (9)
 transmission (10)
 snmp (11)
 experimental (3)
 private (4)
 enterprises (1)
 proteon (1)
 ibm (2)
 cisco (9)
 novell (23)
 rockwell/cmc (44)
 norand (469)
 mci (515)
 tek logix (752)
 .
 .
 .

MIB-II Information

MIB-II is the second version of the Management Information Base (MIB). It is for use with network management protocols in TCP/IP-based internets. MIB-II contains 171 manageable objects, as a super set of MIB. Table 2-1 contains the major groups of MIB-II.

Table 2-1
Major Groups of MIB-II

MIB Family	OID	Purpose	Groups
System	1.3.6.1.2.1.1	Model and device type	dot3 (ethernet)
Interfaces	1.3.6.1.2.1.2	I/O Ports	
AT	1.3.6.1.2.1.3	Table of IP to MAC/DLC address	
IP	1.3.6.1.2.1.4	IP process	
ICMP	1.3.6.1.2.1.5	ICMP process	
TCP	1.3.6.1.2.1.6	TCP process	
UDP	1.3.6.1.2.1.7	UDP process	
EGP	1.3.6.1.2.1.8	EGP process	
CMOT	1.3.6.1.2.1.9	Historical inclusion for OSI support	
Transmission	1.3.6.1.2.1.10	Allows for data based on I/O port type	
SNMP	1.3.6.1.2.1.11	Allows data to be collected about SNMP statistics	

ASN.1 Definition — System.sysDescr of MIB-II

ASN.1 is a machine-independent data definition language used by ISO protocols. Its full name is Abstract Syntax Notation One and it is defined in ISO documents 8824.2 and 8825.2. A notable use is for SNMP and MIBs.

EXAMPLE:

```
Object Name ..... sysDescr OBJECT-TYPE
Object Data Type ..... SYNTAX DisplayString (SIZE (0..255))
Object Access Rights ... ACCESS read-only
Object Status ..... STATUS mandatory
Object Description ..... DESCRIPTION
    "A textual description of the entity. This value should
    include the full name and version identification of the
    system's hardware type, software operating system, and
    networking software. It is mandatory that this only
    contain printable ASCII characters."
Object Identifier ..... ::= { system 1 }
```

SNMP Access Control

- ▶ **MIB Access Mode:** A MIB element is assigned an access mode in the MIB definition (Read-Only, Read-Write). The SNMP management station's access is limited by the MIB access mode.
- ▶ **Community:** Defines a level of authentication and access control at the SNMP agent. The SNMP agent established one community for each desired combination of authentication and access control characteristics. Each community is given a unique name within the SNMP agent. SNMP management stations are members of a community and must employ the community name in all GET and SET operations.

Traps

Traps are unsolicited messages sent by the SNMP agent to the SNMP management station indicating that an extraordinary event has occurred.

- ▶ The SNMP agent keeps a list of SNMP managers to which it will send traps.
- ▶ **Generic Traps:**
One of a few extraordinary events (part of the SNMP standard)
 - cold Start
 - warmStart
 - linkDown
 - linkUp
 - authenticationFailure
 - egpNeighborLoss
- ▶ **Enterprise Specific Traps:**
Defined by the private enterprise as part of that enterprise's MIB and controlled by that enterprise's agent.

Supported Management Information



MIB-II Information

NORAND® Open Wireless LAN 6710 Access Point, RC4030E Gateway, and 6910 Integrated Gateway/Access Point devices support MIB-II. Table 3-1 shows which Management Information Base Two (MIB-II) information Intermecc Technologies Corporation supports.

Table 3-1
Supported MIB-II Information

MIB Family	OID	Purpose	Groups
System	1.3.6.1.2.1.1	Model & device type	
Interfaces	1.3.6.1.2.1.2	I/O Ports	
AT	1.3.6.1.2.1.3	Table of IP to MAC/DLC address	
IP	1.3.6.1.2.1.4	IP process	
ICMP	1.3.6.1.2.1.5	ICMP process	
TCP	1.3.6.1.2.1.6	TCP process	
UDP	1.3.6.1.2.1.7	UDP process	
Transmission	1.3.6.1.2.1.10	Allows for data based on I/O port type	dot3 (ethernet)
SNMP	1.3.6.1.2.1.11	Allows data to be collected about SNMP devices	

NORAND MIB Information

Intermec Technologies Corporation has structured its proprietary management information similar to that of MIB-II. In addition to MIB II, NORAND devices support additional information that is specific to the NORAND devices.

EXAMPLE: Device system information is found under `nSystem`, similar to MIB-II System. The OID for the `nSystem` group ends in "1," just as the OID for MIB-II system ends in "1."

Table 3-2 shows the NORAND MIB information for the 6710 Access Point MIB. See Section 4 for the 6710 Access Point MIB.

Table 3-2
6710 Access Point MIB Information

MIB Family	OID	Purpose	MIB Groups
nSystem	1.3.6.1.4.1.469.1000.2.1	Model, device type, software, file system	hw, file, fsinfo, segment, dir, sysErrors, criticalErrors
nInterfaces	1.3.6.1.4.1.469.1000.2.2	I/O Ports	nifx, portState, portStats, ptxq, pmsg
nSNMP	1.3.6.1.4.1.469.1000.2.11	SNMP	community, trapTarget
nBridge	1.3.6.1.4.1.469.1000.2.17	Bridging	rt, brg, addr, brgState, bridgeStats
nControl	1.3.6.1.4.1.469.1000.2.105	Device Control	powerUp, softwareDownLoad

Table 3-3 shows the RC4030E Gateway MIB, which is detailed in Section 5.

Table 3-3
RC4030E Gateway MIB Information

MIB Family	OID	Purpose	MIB Groups
nSystem	1.3.6.1.4.1.469.1000.2.1	Model, device type, software, file system	hw, file, dir, sysErrors, criticalErrors
nSNMP	1.3.6.1.4.1.469.1000.2.11	SNMP	community, trapTarget
nTransport	1.3.6.1.4.1.469.1000.2.102	Transport	wst, hlit, wcbst
nApplication	1.3.6.1.4.1.469.1000.2.104	Device applications	gwSession (gs, rcbs, hd, td, ahost)
nControl	1.3.6.1.4.1.469.1000.2.105	Device Control	powerUp

Table 3-4 shows the 6910 Integrated Gateway/Access Point MIB, which is detailed in Section 6.

Table 3-4
6910 Integrated Gateway/Access Point MIB Information

MIB Family	OID	Purpose	MIB Groups
nSystem	1.3.6.1.4.1.469.1000.2.1	Model, device type, software, file system	hw, file, fsinfo, segment, dir, sysErrors, criticalErrors
nInterfaces	1.3.6.1.4.1.469.1000.2.2	I/O Ports	nifx, portState, portStats, ptxq, pmsg
nSNMP	1.3.6.1.4.1.469.1000.2.11	SNMP	community, trapTarget
nBridge	1.3.6.1.4.1.469.1000.2.17	Bridging	rt, brg, addr, brgState, bridgeStats
nTransport	1.3.6.1.4.1.469.1000.2.102	Transport	wst, hlit, webst
nApplication	1.3.6.1.4.1.469.1000.2.104	Device applications	gwSession (gs, rcbs, hd, td, ahost)
nControl	1.3.6.1.4.1.469.1000.2.105	Device Control	powerUp, softwareDownload

Section 4

Norand Open Wireless LAN MIB 6710 Access Point



This section describes various groups Intermecc Technologies Corporation supports for the 6710 Access Point (AP). Table 4-1 lists groups, their meaning, and page numbers for each group's table summary and actual definitions.

Table 4-1
AP MIB Directory

Group	Meaning	Group Summary	MIB Definition
Product OIDs			
products	Intermec® Products	4-2	4-13
System Information			
hw	Hardware Information	4-2	4-13
fsinfo	Filesystem Information	4-3	4-14
segment	Filesegment Information	4-3	4-15
dir	Software Directory Listing	4-3	4-16
criticalErrors	Critical Errors Information	4-4	4-18
Interface Information			
nifx	Norand Extensions to Interfaces Table	4-4	4-19
portState	Port State Information	4-5	4-22
portStats	Port Statistics	4-5	4-26
ptxq	Port Transmit Queue	4-6	4-29
pmsg	Pending Message Services	4-7	4-32
SNMP Version 1 Configuration			
community	Community Table	4-8	4-34
trapTarget	Trap Target Table	4-8	4-35
Bridging Parameters			
rt	Route Table	4-9	4-36
brg	Bridge Table	4-10	4-40
addr	Address Table	4-10	4-41
brgState	Bridge State Information	4-10	4-43
bridgeStats	Bridge Statistics	4-11	4-46
Control Groups			
powerUp	Power Up Objects	4-12	4-48
softwareDownload	Software Download	4-12	4-49

Access Point MIB Outline

This outline summarizes the various MIB groups Intermecc Technologies Corporation supports for the 6710 Access Point.

Product OIDs

This group contains an Object IDentification (OID) for each Intermecc device.

Table 4-2
products GROUP

Device Products
norand.manage.products.x
(1.3.6.1.4.1.469.1000.1.x)

Object ID	Object Name	Object Type	Access
1	ap6710	OBJECT ID	Not Applicable (N/A)
2	gw4030	OBJECT ID	(N/A)
3	wnas	OBJECT ID	(N/A)
4	ts6950	OBJECT ID	(N/A)
5	gwap6910	OBJECT ID	(N/A)

System Information

The following groups contain system level objects describing hardware and file-system configuration properties. The groups also contain information about critical errors.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- hw Hardware Information (page 4-13)
- fsinfo Filesystem Information (page 4-14)
- segment Filesegment Information (page 4-15)
- dir Software Directory Listing (page 4-16)
- criticalErrors Critical Errors Information (page 4-18)

Table 4-3
hw GROUP

Device Hardware Information
norand.manage.norandNet.nSystem.hw.x
(1.3.6.1.4.1.469.1000.2.1.1.x)

Object ID	Object Name	Object Type	Access
1	hwPartNo	INTEGER	read
2	hwDescription	DisplayString	read
3	hwRevision	INTEGER	read
4	hwSerialNo	INTEGER	read
5	hwID	INTEGER	read

Table 4-4
fsinfo GROUP

Device Filesystem Information
norand.manage.norandNet.nSystem.file.fsinfo.x
(1.3.6.1.4.1.469.1000.2.1.3.1.x)

Object ID	Object Name	Object Type	Access
1	fsEnabled	INTEGER	read
2	fsMaxSectors	INTEGER	read
3	fsSectorSize	INTEGER	read
4	fsNumSegments	INTEGER	read
5	fsNumFiles	Gauge	read
6	fsBootSegment	INTEGER	read
7	fsDataSegment	INTEGER	read

Table 4-5
segment GROUP

Device Filesegment Information
norand.manage.norandNet.nSystem.file.segment.x
(1.3.6.1.4.1.469.1000.2.1.3.2.x)

Object ID	Object Name	Object Type	Access
2.1.1	segID	INTEGER	read
2.1.2	segFirstSector	INTEGER	read
2.1.3	segLastSector	INTEGER	read
2.1.4	segStatus	INTEGER	read
2.1.5	segSize	INTEGER	read
2.1.6	segFree	INTEGER	read

Table 4-6
dir GROUP

Device Software Directory Listing
norand.manage.norandNet.nSystem.file.dir.x
(1.3.6.1.4.1.469.1000.2.1.3.3.x)

Object ID	Object Name	Object Type	Access
2.1.1	dirIndex	INTEGER	read
2.1.2	dirName	DisplayString	read
2.1.3	dirSegment	INTEGER	read
2.1.4	dirType	INTEGER	read
2.1.5	dirSize	INTEGER	read
2.1.6	dirDate	DisplayString	read
2.1.7	dirTime	DisplayString	read
2.1.8	dirVersion	DisplayString	read

Table 4-7
criticalErrors GROUP

Device Critical Errors Information
norand.manage.norandNet.nSystem.sysErrors.criticalErrors.x
(1.3.6.1.4.1.469.1000.2.1.4.1.x)

Object ID	Object Name	Object Type	Access
1	ceEnabled	INTEGER	read
2	ceOverflow	INTEGER	read
3	ceReset	INTEGER	write
4.1.1	ceLogErrorCode	INTEGER	read
4.1.2	ceLogErrorCount	Counter	read

Interface Information

The following groups relate information about Intermec interfaces, port state, port statistics, port transmit queue, and pending message services.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- nifx NORAND Extensions to Interfaces Table (page 4-19)
- portState Port State Information (page 4-22)
- portStats Port Statistics (page 4-26)
- ptxq Port Transmit Queue (page 4-29)
- pmsg Pending Message Services (page 4-32)

Table 4-8
nifx GROUP

NORAND Extensions to MIB-II Interfaces Table
norand.manage.norandNet.nInterfaces.nifx.x
(1.3.6.1.4.1.469.1000.2.2.2.x)

Object ID	Object Name	Object Type	Access
4.1.1	nifxIndex	INTEGER	read
4.1.2	nifxType	INTEGER	read
4.1.3	nifxInDisabledDiscards	Counter	read
4.1.4	nifxInOverruns	Counter	read
4.1.5	nifxInHWOverruns	Counter	read
4.1.6	nifxInUcastDPkts	Counter	read
4.1.7	nifxInNUcastDPkts	Counter	read
4.1.8	nifxInLenErrors	Counter	read
4.1.9	nifxExcessiveDeferrals	Counter	read
4.1.10	nifxInNetIDDiscards	Counter	read
4.1.11	nifxInFragDiscards	Counter	read
4.1.12	nifxInUFilterDiscards	Counter	read
4.1.13	nifxInNUFilterDiscards	Counter	read
4.1.14	nifxInQFullDiscards	Counter	read

Table 4-9
portState GROUP

Device Port State Information
norand.manage.norandNet.nInterfaces.portState.x
(1.3.6.1.4.1.469.1000.2.2.3.x)

Object ID	Object Name	Object Type	Access
4.1.1	psPort	INTEGER	read
4.1.2	psIfIndex	INTEGER	read
4.1.3	psAddress	PhysAddress	read
4.1.4	psType	INTEGER	read
4.1.5	psState	INTEGER	read
4.1.6	psCost	INTEGER	read
4.1.7	psHelloPeriod	INTEGER	read
4.1.8	psHelloCount	Counter	read
4.1.9	psMacdWindow	INTEGER	read
4.1.10	psMacdQSize	Gauge	read
4.1.11	psMacdTimeouts	Counter	read
4.1.12	psIsPrimary	INTEGER	read
4.1.13	psIsSecondary	INTEGER	read
4.1.14	psIsSecondaryCandidate	INTEGER	read
4.1.15	psSecondaryUniFlooding	INTEGER	read
4.1.16	psSecondaryMultiFlooding	INTEGER	read
4.1.17	psIsRadio	INTEGER	read
4.1.18	psPendEnabled	INTEGER	read

Table 4-10
portStats GROUP

Device Port Statistics
norand.manage.norandNet.nInterfaces.portStats.x
(1.3.6.1.4.1.469.1000.2.2.4.x)

Object ID	Object Name	Object Type	Access
4.1.1	pstcPort	INTEGER	read
4.1.2	pstcInOWLPkts	Counter	read
4.1.3	pstcInUcastOWLDataPkts	Counter	read
4.1.4	pstcInNUcastOWLDataPkts	Counter	read
4.1.5	pstcInOWLErrors	Counter	read
4.1.6	pstcOutOWLPkts	Counter	read
4.1.7	pstcOutUcastOWLDataPkts	Counter	read
4.1.8	pstcOutNUcastOWLDataPkts	Counter	read
4.1.9	pstcOutOWLErrors	Counter	read
4.1.10	pstcParentLinkErrors	Counter	read

Table 4-10 (Continued)
portStats GROUP

Device Port Statistics
norand.manage.norandNet.nInterfaces.portStats.x
(1.3.6.1.4.1.469.1000.2.2.4.x)

Object ID	Object Name	Object Type	Access
4.1.11	pstcAlertLinkErrors	Counter	read
4.1.12	pstcInUcastRelayPkts	Counter	read
4.1.13	pstcInNUcastRelayPkts	Counter	read
4.1.14	pstcOutUcastRelayPkts	Counter	read
4.1.15	pstcOutNUcastRelayPkts	Counter	read
4.1.16	pstcInUcastInbound	Counter	read
4.1.17	pstcInUcastOutbound	Counter	read
4.1.18	pstcInUcastSec	Counter	read
4.1.19	pstcInUcastFlood	Counter	read
4.1.20	pstcUcastDiscards	Counter	read
4.1.21	pstcInNUcastDiscards	Counter	read
4.1.22	pstcInUcastToIFC	Counter	read
4.1.23	pstcInNUcastToIFC	Counter	read
4.1.24	pstcOutDelayDiscards	Counter	read

Table 4-11
ptxq GROUP

Device Port Transmit Queue
norand.manage.norandNet.nInterfaces.ptxq.x
(1.3.6.1.4.1.469.1000.2.2.5.x)

Object ID	Object Name	Object Type	Access
1.1.1	ptxqPort	INTEGER	read
1.1.2	ptxqRegQSize	Gauge	read
1.1.3	ptxqRegQMax	INTEGER	read
1.1.4	ptxqExpQSize	Gauge	read
1.1.5	ptxqExpQMax	INTEGER	read
1.1.6	ptxqQHpCount	Counter	read
1.1.7	ptxqQExpCount	Counter	read
1.1.8	ptxqQRegCount	Counter	read
1.1.9	ptxqQHpDiscards	Counter	read
1.1.10	ptxqQExpDiscards	Counter	read
1.1.11	ptxqQRegDiscards	Counter	read
1.1.12	ptxqMultiQSize	Gauge	read
1.1.13	ptxqMultiQMax	INTEGER	read
1.1.14	ptxqMultiQDiscards	Counter	read

Table 4-12
pmsg GROUP

Device Pending Message Service
 norand.manage.norandNet.nInterfaces.pmsg.x
 (1.3.6.1.4.1.469.1000.2.2.6.x)

Object ID	Object Name	Object Type	Access
1.1.1	pmsgPort	INTEGER	read
1.1.2	pmsgPendRecCurrent	Gauge	read
1.1.3	pmsgPendRecMax	INTEGER	read
1.1.4	pmsgPendMsgCurrent	Gauge	read
1.1.5	pmsgPendMsgMax	INTEGER	read
1.1.6	pmsgPendMsgTotal	Counter	read
1.1.7	pmsgPendMsgDiscards	Counter	read
1.1.8	pmsgPendRecOverflowErrors	Counter	read
1.1.9	pmsgPendMsgOverflowErrors	Counter	read
1.1.10	pmsgPendAgedRecCount	Counter	read
1.1.11	pmsgPendAgedMsgCount	Counter	read

SNMP Version 1 Configuration Group

This group contains objects that configure the version 1 Simple Network Management Protocol (SNMP) agent.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- community Community Table (page 4-34)
- trapTarget Trap Target Table (page 4-35)

Table 4-13

community TABLE

Device SNMP v1 Configurations
norand.manage.norandNet.nSNMP.v1Config.x
(1.3.6.1.4.1.469.1000.2.11.1.x)

Object ID	Object Name	Object Type	Access
2.1.1	communityIndex	INTEGER	read
2.1.2	communityStatus	INTEGER	write
2.1.3	communityName	DisplayString	write
2.1.4	communityPrivileges	INTEGER	write

Table 4-14

trapTarget TABLE

Device SNMP v1 Configurations (page 4-34)
norand.manage.norandNet.nSNMP.v1Config.x
(1.3.6.1.4.1.469.1000.2.11.1.x)

Object ID	Object Name	Object Type	Access
3.1.1	trapTargetIndex	INTEGER	read
3.1.2	trapTargetStatus	INTEGER	write
3.1.3	trapTargetName	DisplayString	write
3.1.4	trapTargetIpAddress	IpAddress	write

Bridging Parameters

The following group contains objects relating to the wireless transparent bridging operation.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- rt Route Table (page 4-36)
- brg Bridge Table (page 4-40)
- addr Address Table (page 4-41)
- brgState Bridge State Information (page 4-43)
- bridgeStats Bridge Statistics (page 4-46)

Table 4-15
rt GROUP

Device Route Table
norand.manage.norandNet.nBridge.rt.x
(1.3.6.1.4.1.469.1000.2.17.2.x)

Object ID	Object Name	Object Type	Access
2.1.1	rtDestination	PhysAddress	read
2.1.2	rtPort	INTEGER	read
2.1.3	rtAge	INTEGER	read
2.1.4	rtNodeId	INTEGER	read
2.1.5	rtAttachId	INTEGER	read
2.1.6	rtAttachTime	TimeTicks	read
2.1.7	rtApEaddr	PhysAddress	read
2.1.8	rtHopAddrLen	INTEGER	read
2.1.9	rtHopAddr16	INTEGER	read
2.1.10	rtHopEaddr	PhysAddress	read
2.1.11	rtIsBound	INTEGER	read
2.1.12	rtIsRemote	INTEGER	read
2.1.13	rtIsChild	INTEGER	read
2.1.14	rtIsAp	INTEGER	read
2.1.15	rtIsDistributed	INTEGER	read
2.1.16	rtIsRemoteLan	INTEGER	read
2.1.17	rtNS	INTEGER	read
2.1.18	rtNR	INTEGER	read

Table 4-16
brg GROUP

Device Bridge Table
norand.manage.norandNet.nBridge.brg.x
(1.3.6.1.4.1.469.1000.2.17.3.x)

Object ID	Object Name	Object Type	Access
2.1.1	brgDestination	PhysAddress	read
2.1.2	brgPort	INTEGER	read
2.1.3	brgAge	INTEGER	read
2.1.4	brgType	INTEGER	read
2.1.5	brgIsPermanent	INTEGER	read
2.1.6	brgTimestamp	TimeTicks	read

Table 4-17
addr GROUP

Address Table
norand.manage.norandNet.nBridge.addr.x
(1.3.6.1.4.1.469.1000.2.17.4.x)

Object ID	Object Name	Object Type	Access
2.1.1	addrDestination	PhysAddress	read
2.1.2	addrAge	INTEGER	read
2.1.3	addrNodeId	INTEGER	read
2.1.4	addrAlias	DisplayString	read
2.1.5	addrDeviceId	INTEGER	read
2.1.6	addrIpAddress	IPAddress	read

Table 4-18
brgState GROUP

Bridge State Information
norand.manage.norandNet.nBridge.brgState.x
(1.3.6.1.4.1.469.1000.2.17.6.x)

Object ID	Object Name	Object Type	Access
3	bsAddress	PhysAddress	read
4	bsLanId	INTEGER	read
5	bsCostToRoot	INTEGER	read
6	bsIsRoot	INTEGER	read
7	bsIsAttached	INTEGER	read
8	bsAttachId	INTEGER	read
9	bsMyRootPriority	INTEGER	read
10	bsRootPort	INTEGER	read
11	bsDesignatedRootAddress	PhysAddress	read
12	bsDesignatedRootPriority	INTEGER	read
13	bsDesignatedRootSequence	INTEGER	read

Table 4-18 (Continued)
brgState GROUP

Bridge State Information
norand.manage.norandNet.nBridge.brgState.x
(1.3.6.1.4.1.469.1000.2.17.6.x)

Object ID	Object Name	Object Type	Access
14	bsParentAddress	PhysAddress	read
15	bsPortCount	INTEGER	read
16	bsNodeId	INTEGER	read
17	bsRootChangedCount	Counter	read
18	bsRootCount	Counter	read
19	bsAttachCount	Counter	read
20	bsDetachReason	INTEGER	read
21	bsNetworkTime	TimeTicks	read
22	bsUniFloodLevel	INTEGER	read
23	bsMultiFloodLevel	INTEGER	read
24	bsIsPrimaryBridge	INTEGER	read
25	bsIsSecondaryBridge	INTEGER	read
26	bsUniFilterExpr	INTEGER	read
27	bsMuliFilterExpr	INTEGER	read

Table 4-19
bridgeStats GROUP

Bridge Statistics
norand.manage.norandNet.nBridge.bridgeStats.x
(1.3.6.1.4.1.469.1000.2.17.7.x)

Object ID	Object Name	Object Type	Access
3	bstcRouteCount	Gauge	read
4	bstcChildCount	Gauge	read
5	bstcChildApCount	Gauge	read
6	bstcRemoteCount	Gauge	read
7	bstcPrimaryCount	Gauge	read
8	bstcInboundCount	Gauge	read
9	bstcSecondaryCount	Gauge	read
10	bstcRemoteLanCount	Gauge	read
11	bstcRouteGetErrors	Counter	read
12	bstcEntryGetErrors	Counter	read
13	bstcRmtLanGetErrors	Counter	read
14	bstcRouteSeqErrors	Counter	read
15	bstcDeleteSeqErrors	Counter	read
16	bstcEntrySeqErrors	Counter	read
17	bstcInvalidUpdateErrors	Counter	read

Control Groups

The objects in the following groups exert control over Intermecc Access Points. Present functions include rebooting and scheduling software downloads.

► **NOTE:**

The MIB definitions for each group starts on the page given below.

- powerUp Power Up Objects (page 4-48)
- softwareDownload Software Download (page 4-49)

Table 4-20
powerUp GROUP

Device Power Up Objects
norand.manage.norandNet.nControl.powerUp.x
(1.3.6.1.4.1.469.1000.2.105.1.x)

Object ID	Object Name	Object Type	Access
1	pwrPowerUpCount	Counter	read
2	pwrNextPowerUpTime	TimeTicks	write

Table 4-21
softwareDownload GROUP

Device Software Download
norand.manage.norandNet.nControl.softwareDownload.x
(1.3.6.1.4.1.469.1000.2.105.2.x)

Object ID	Object Name	Object Type	Access
1	sdStartTime	TimeTicks	write
2	sdServerIpAddress	IpAddress	write
3	sdScriptFilename	DisplayString	write
4	sdStatus	INTEGER	read
5	sdErrorString	DisplayString	read
6	sdCheckPoint	INTEGER	write
7	sdSetActivePointers	INTEGER	write
8	sdTerminate	INTEGER	write

Access Point MIB Definitions

Following is the 6710 Access Point MIB definition for the NORAND® Open Wireless LAN.

```
-- Norand Open Wireless LAN MIB - 6710 Access Point
-- Version 1.15
-- Version Date: 8/23/96
-- This MIB contains objects supported by V1.15 of the 6710 Access Point.

-- Subject to Change

OWLAP-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        enterprises, IpAddress, Counter, Gauge, TimeTicks
            FROM RFC1155-SMI
        PhysAddress, DisplayString
            FROM RFC1213-MIB
        OBJECT-TYPE
            FROM RFC-1212;

    -- This MIB module uses the extended OBJECT-TYPE macro as defined in
    -- RFC-1212;

norand
    manage
        products
            ap6710
            gw4030
            wnas
            ts6950
            gwap6910
        norandNet
        nSystem

        OBJECT IDENTIFIER ::= { enterprises 469 }
        OBJECT IDENTIFIER ::= { norand 1000 }
        OBJECT IDENTIFIER ::= { manage 1 }
        OBJECT IDENTIFIER ::= { products 1 }
        OBJECT IDENTIFIER ::= { products 2 }
        OBJECT IDENTIFIER ::= { products 3 }
        OBJECT IDENTIFIER ::= { products 4 }
        OBJECT IDENTIFIER ::= { products 5 }
        OBJECT IDENTIFIER ::= { manage 2 }
        OBJECT IDENTIFIER ::= { norandNet 1 }

        hw
            OBJECT IDENTIFIER ::= { nSystem 1 }

    -- The Hardware Parameters Group

    hwPartNo      OBJECT-TYPE
        SYNTAX INTEGER (0..2147483647)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The Norand part number of the hardware device."
        ::= { hw 1 }

    hwDescription OBJECT-TYPE
        SYNTAX DisplayString (SIZE (0..40))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The description of the hardware device."
        ::= { hw 2 }

    hwRevision    OBJECT-TYPE
        SYNTAX INTEGER (0..2147483647)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The revision level of the hardware device."
        ::= { hw 3 }
```

```

hwSerialNo    OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The serial number of the hardware device."
    ::= { hw 4 }

hwID          OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The device identifier of the hardware device. Values =
        3250, 4000, 4020, 4030, 4033, 3240, 1000, 1100, 1700, 5940,
        4650, 100 (ACE process), 200 (DOSNMS), 300 (Norand Proxy
        Agent), 6710 (Access Point)."
    ::= { hw 5 }

file          OBJECT IDENTIFIER ::= { nSystem 3 }

fsinfo      OBJECT IDENTIFIER ::= { file 1 }

-- The FileSystem Information Table

fsEnabled     OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the file system is enabled."
    ::= { fsinfo 1 }

fsMaxSectors  OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of physical sectors. A file segment consists of
        one or more adjacent physical sectors."
    ::= { fsinfo 2 }

fsSectorSize  OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The size of a physical sector in bytes."
    ::= { fsinfo 3 }

fsNumSegments OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of logical file segments (0-MAX_SECTORS)."
    ::= { fsinfo 4 }

fsNumFiles    OBJECT-TYPE
    SYNTAX Gauge (0..25)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of files (0-25)."
    ::= { fsinfo 5 }

```

```

fsBootSegment OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The index of the current boot segment.  If the index is
        non-zero and the first file in the associated segment is
        executable, then control is passed to that file during the
        power-up sequence."
    ::= { fsinfo 6 }

    fsDataSegment OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The index of the active data segment.  Files stored in
            this segment will be accessible to an executing
            application."
        ::= { fsinfo 7 }

segment    OBJECT IDENTIFIER ::= { file 2 }

-- The File Segment Table

-- Table Definition

segTable      OBJECT-TYPE
    SYNTAX SEQUENCE OF SEEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A catalog of memory segments and their utilization."
    ::= { segment 2 }

-- Row Definition

segEntry      OBJECT-TYPE
    SYNTAX SEEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { segID }
    ::= { segTable 1 }

-- Columnar Object Definitions

SEEntry ::=
    SEQUENCE {
        segID          INTEGER,
        segFirstSector INTEGER,
        segLastSector  INTEGER,
        segStatus      INTEGER,
        segSize        INTEGER,
        segFree        INTEGER
    }

segID        OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The segment ID (1 - (NUM_SEGMENTS+1)).  A non-zero number
        which uniquely identifies a segment."
    ::= { segEntry 1 }

```

```

segFirstSector OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The first physical sector in the segment
        (1 - (MAX_SECTORS + 1))."
    ::= { segEntry 2 }

segLastSector OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The last physical sector in the segment
        (FIRST_SECTOR - (MAX_SECTORS + 1))."
    ::= { segEntry 3 }

segStatus OBJECT-TYPE
    SYNTAX INTEGER { valid(1),
                    invalid(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The segment status:
         valid = 1,
         invalid = 2."
    ::= { segEntry 4 }

segSize OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The segment size in bytes."
    ::= { segEntry 5 }

segFree OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of available bytes in the segment which are not
        currently allocated to a file."
    ::= { segEntry 6 }

dir OBJECT IDENTIFIER ::= { file 3 }

-- The File Directory Table
-- Table Definition

dirTable OBJECT-TYPE
    SYNTAX SEQUENCE OF DIREntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The FileSystem Directory."
    ::= { dir 2 }

-- Row Definition

dirEntry OBJECT-TYPE
    SYNTAX DIREntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { dirIndex }
    ::= { dirTable 1 }

```

```

-- Columnar Object Definitions

DIREntry ::=
SEQUENCE {
    dirIndex    INTEGER,
    dirName     DisplayString,
    dirSegment  INTEGER,
    dirType     INTEGER,
    dirSize     INTEGER,
    dirDate     DisplayString,
    dirTime     DisplayString,
    dirVersion  DisplayString
}

dirIndex      OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Directory Index"
::= { dirEntry 1 }

dirName       OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..14))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"File name"
::= { dirEntry 2 }

dirSegment    OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"File segment (1 - (NUM_SEGMENTS + 1)).
The segment ID which identifies the segment containing the
file."
::= { dirEntry 3 }

dirType       OBJECT-TYPE
SYNTAX INTEGER { executable(1),
                data(2),
                invalid(3) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"File type:
    executable = 1,
    data       = 2,
    invalid    = 3."
::= { dirEntry 4 }

dirSize       OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The file size in bytes."
::= { dirEntry 5 }

```

```

dirDate          OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..12))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file date in MM-DD-YYYY display format."
    ::= { dirEntry 6 }

dirTime          OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..10))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file time in HH:MM:SS display format."
    ::= { dirEntry 7 }

dirVersion       OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..8))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file version in v99.99 display format."
    ::= { dirEntry 8 }

sysErrors        OBJECT IDENTIFIER ::= { nSystem 4 }

criticalErrors OBJECT IDENTIFIER ::= { sysErrors 1 }

    ceEnabled     OBJECT-TYPE
        SYNTAX INTEGER { true(1), false(2) }
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "A value of true(1) signifies that the critical error
            log was successfully initialized as part of the power-up
            sequence. Any errors in that initialization process
            result in a value of false(2)."
        ::= { criticalErrors 1 }

    ceOverflow    OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "Overflow error code. If the overflow code is non-zero,
            it indicates that the log has overflowed and the
            overflow code contains the last displaced value."
        ::= { criticalErrors 2 }

    ceReset       OBJECT-TYPE
        SYNTAX INTEGER { true(1), false(2) }
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
            "A user can reset the critical error log by setting
            ceReset to true(1). Valid values are true(1) or
            false(2)."
        ::= { criticalErrors 3 }

    ceLogTable    OBJECT-TYPE
        SYNTAX SEQUENCE OF CELogEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION
            "Critical Error Log Table"
        ::= { criticalErrors 4 }

```



```

ceLogEntry          OBJECT-TYPE
    SYNTAX CELogEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { ceLogErrorCode }
    ::= { ceLogTable 1 }

CELogEntry ::=
    SEQUENCE {
        ceLogErrorCode  INTEGER,
        ceLogErrorCount Counter
    }

ceLogErrorCode      OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Critical error code.  A 16-bit value which uniquely
        identifies a system software error.  The error codes are
        intended for internal Norand use."
    ::= { ceLogEntry 1 }

ceLogErrorCount     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Error count for the associated ceLogErrorCode"
    ::= { ceLogEntry 2 }

nInterfaces          OBJECT IDENTIFIER ::= { norandNet 2 }

    nifx          OBJECT IDENTIFIER ::= { nInterfaces 2 }

-- The Norand Extended Interfaces Table

nifxTable            OBJECT-TYPE
    SYNTAX SEQUENCE OF NIFXEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Norand Extended Interface Table"
    ::= { nifx 4 }

nifxEntry            OBJECT-TYPE
    SYNTAX NIFXEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { nifxIndex }
    ::= { nifxTable 1 }

```

```

NIFXEntry ::=
  SEQUENCE {
    nifxIndex          INTEGER,
    nifxType           INTEGER,
    nifxInDisabledDiscards Counter,
    nifxInOverruns    Counter,
    nifxInHWOverruns  Counter,
    nifxInUcastDPkts Counter,
    nifxInNUcastDPkts Counter,
    nifxInLenErrors   Counter,
    nifxExcessiveDeferrals Counter,
    nifxInNetIDDiscards Counter,
    nifxInFragDiscards Counter,
    nifxInUFilterDiscards Counter,
    nifxInNUFilterDiscards Counter,
    nifxInQFullDiscards Counter
  }

nifxIndex          OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Interface index"
  ::= { nifxEntry 1 }

nifxType           OBJECT-TYPE
  SYNTAX INTEGER {
    ether(4),
    proxim24(196),
    falcon902(197),
    uhf(198),
    nor24(199)
  }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Norand Interface Type"
  ::= { nifxEntry 2 }

nifxInDisabledDiscards OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The number of received unicast frames which do not require
    forwarding. Unicast ethernet frames are discarded if
    ether-to-radio flooding is disabled and the destination is
    unknown; otherwise, unicast frames are discarded if the bridge
    has learned that the destination port is the same as the
    source port."
  ::= { nifxEntry 3 }

nifxInOverruns    OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The number of received frames discarded because the frame could
    not be queued for the MAC-D task."
  ::= { nifxEntry 4 }

```

```
nifxInHWOverruns          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received frames discarded due to hardware
        overruns."
    ::= { nifxEntry 5 }

nifxInUcastDPkts          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received unicast frames successfully delivered to
        the MAC-D task."
    ::= { nifxEntry 6 }

nifxInNUcastDPkts         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received multicast frames successfully delivered
        to the MAC-D task."
    ::= { nifxEntry 7 }

nifxInLenErrors           OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number received frames with length errors."
    ::= { nifxEntry 8 }

nifxExcessiveDeferrals    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of aborted transmissions due to excessive deferrals."
    ::= { nifxEntry 9 }

nifxInNetIDDiscards       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received frames discarded because the LAN ID did
        not match."
    ::= { nifxEntry 10 }

nifxInFragDiscards        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received frame fragments discarded because a
        fragmented frame could not be re-assembled."
    ::= { nifxEntry 11 }
```

```

nifxInUFilterDiscards    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of enabled received unicast frames discarded due to a
        unicast filter expression."
    ::= { nifxEntry 12 }

nifxInNUFilterDiscards   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of enabled received multicast frames discarded due to
        a multicast filter expression."
    ::= { nifxEntry 13 }

nifxInQFullDiscards     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received frames discarded because the frame could
        not be queued for the MAC-R task."
    ::= { nifxEntry 14 }

portState                OBJECT IDENTIFIER ::= { nInterfaces 3 }

```

-- The Port State Table

```

psTable                  OBJECT-TYPE
    SYNTAX SEQUENCE OF PSEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "MAC-R port state variables"
    ::= { portState 4 }

psEntry                  OBJECT-TYPE
    SYNTAX PSEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { psPort }
    ::= { psTable 1 }

```

```

PSEntry ::=
  SEQUENCE {
    psPort          INTEGER,
    psIfIndex       INTEGER,
    psAddress       PhysAddress,
    psType          INTEGER,
    psState         INTEGER,
    psCost          INTEGER,
    psHelloPeriod  INTEGER,
    psHelloCount   Counter,
    psMacdWindow   INTEGER,
    psMacdQSize    Gauge,
    psMacdTimeouts Counter,
    psIsPrimary    INTEGER,
    psIsSecondary  INTEGER,
    psIsSecondaryCandidate INTEGER,
    psSecondaryUniFlooding INTEGER,
    psSecondaryMultiFlooding INTEGER,
    psIsRadio      INTEGER,
    psPendEnabled  INTEGER
  }

psPort          OBJECT-TYPE
  SYNTAX INTEGER (1..4)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "MAC-R port ID (1-4).  A number which uniquely identifies the
    port."
  ::= { psEntry 1 }

psIfIndex       OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "MAC-D interface index.  The index matches the interface index
    of the associated row in the mib-II interface table."
  ::= { psEntry 2 }

psAddress       OBJECT-TYPE
  SYNTAX PhysAddress
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "802 address of the port."
  ::= { psEntry 3 }

psType         OBJECT-TYPE
  SYNTAX INTEGER {
    ether(4),
    proxim24(196),
    falcon902(197),
    uhf(198),
    nor24(199)
  }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Norand port type:
    ether = 4,
    Proxim = 196,
    Falcon = 197,
    UHF = 198."
  ::= { psEntry 4 }

```

```

psState                OBJECT-TYPE
    SYNTAX INTEGER { disabled(0),
                    idle(1),
                    open(2),
                    receive(3),
                    transmit(4) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Port state:
    disabled = 0,
    idle     = 1,
    open     = 2,
    receive  = 3,
    transmit = 4."
    ::= { psEntry 5 }

psCost                 OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Incremental path cost of the port.
    Default values:
    ether    = 20,
    Falcon   = 100,
    UHF      = 255."
    ::= { psEntry 6 }

psHelloPeriod         OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Inter-HELLO time (.01 seconds)."
```

```

    ::= { psEntry 7 }

psHelloCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "HELLO transmit count"
    ::= { psEntry 8 }

psMacdWindow          OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Maximum number of active MAC-D transmit requests."
    ::= { psEntry 9 }

psMacdQSize           OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Current number of active MAC-D transmit requests."
    ::= { psEntry 10 }

```

```
psMacdTimeouts          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-D transmit timeout errors"
    ::= { psEntry 11 }

psIsPrimary              OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for primary bridge ports."
    ::= { psEntry 12 }

psIsSecondary           OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for secondary bridge ports."
    ::= { psEntry 13 }

psIsSecondaryCandidate OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if secondary bridge port candidates."
    ::= { psEntry 14 }

psSecondaryUniFlooding  OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for secondary bridge ports which require unicast
        flooding."
    ::= { psEntry 15 }

psSecondaryMultiFlooding OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for secondary bridge ports which require multicast
        flooding."
    ::= { psEntry 16 }

psIsRadio               OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for radio ports."
    ::= { psEntry 17 }

psPendEnabled           OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the port supports pending messages."
    ::= { psEntry 18 }
```

```

portStats      OBJECT IDENTIFIER ::= { nInterfaces 4 }

pstcTable      OBJECT-TYPE
    SYNTAX SEQUENCE OF PSTCEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "MAC-R port state variables"
    ::= { portStats 4 }

pstcEntry      OBJECT-TYPE
    SYNTAX PSTCEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { pstcPort }
    ::= { pstcTable 1 }

PSTCEntry ::=
    SEQUENCE {
        pstcPort                INTEGER,
        pstcInOWLPkts           Counter,
        pstcInUcastOWLDataPkts Counter,
        pstcInNUcastOWLDataPkts Counter,
        pstcInOWLErrors         Counter,
        pstcOutOWLPkts          Counter,
        pstcOutUcastOWLDataPkts Counter,
        pstcOutNUcastOWLDataPkts Counter,
        pstcOutOWLErrors        Counter,
        pstcParentLinkErrors     Counter,
        pstcAlertLinkErrors      Counter,
        pstcInUcastRelayPkts     Counter,
        pstcInNUcastRelayPkts    Counter,
        pstcOutUcastRelayPkts    Counter,
        pstcOutNUcastRelayPkts   Counter,
        pstcInUcastInbound       Counter,
        pstcInUcastOutbound      Counter,
        pstcInUcastSec           Counter,
        pstcInUcastFlood         Counter,
        pstcInUcastDiscards      Counter,
        pstcInNUcastDiscards     Counter,
        pstcInUcastToIFC         Counter,
        pstcInNUcastToIFC       Counter,
        pstcOutDelayDiscards     Counter
    }

pstcPort       OBJECT-TYPE
    SYNTAX INTEGER (1..4)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-R port ID (1-4). A number which uniquely identifies
        the port."
    ::= { pstcEntry 1 }

pstcInOWLPkts  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total received OWL packets"
    ::= { pstcEntry 2 }

```



```

pstcInUcastOWLDataPkts    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received OWL unicast data packets"
    ::= { pstcEntry 3 }

pstcInNUcastOWLDataPkts   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received multicast OWL data packets"
    ::= { pstcEntry 4 }

pstcInOWLErrors           OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received OWL packets with errors"
    ::= { pstcEntry 5 }

pstcOutOWLPkts            OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total OWL packets sent"
    ::= { pstcEntry 6 }

pstcOutUcastOWLDataPkts   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total unicast OWL data packets sent"
    ::= { pstcEntry 7 }

pstcOutNUcastOWLDataPkts  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total multicast OWL data packets sent"
    ::= { pstcEntry 8 }

pstcOutOWLErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "OWL packet send errors"
    ::= { pstcEntry 9 }

pstcParentLinkErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Root port send link errors"
    ::= { pstcEntry 10 }

```

```

pstcAlertLinkErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Link errors which generated an ALERT."
    ::= { pstcEntry 11 }

pstcInUcastRelayPkts     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast relay packets"
    ::= { pstcEntry 12 }

pstcInNUcastRelayPkts   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received multicast relay packets"
    ::= { pstcEntry 13 }

pstcOutUcastRelayPkts   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total unicast relay packets sent"
    ::= { pstcEntry 14 }

pstcOutNUcastRelayPkts  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total multicast relay packets sent"
    ::= { pstcEntry 15 }

pstcInUcastInbound      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets routed inbound or relayed
         onto the distribution LAN."
    ::= { pstcEntry 16 }

pstcInUcastOutbound     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets routed outbound"
    ::= { pstcEntry 17 }

pstcInUcastSec          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets relayed to a secondary LAN."
    ::= { pstcEntry 18 }

```

```

pstcInUcastFlood          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets with an unknown destination."
    ::= { pstcEntry 19 }

pstcInUcastDiscards      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets discarded"
    ::= { pstcEntry 20 }

pstcInNUcastDiscards    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received multicast data packets discarded"
    ::= { pstcEntry 21 }

pstcInUcastToIFC        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast packets passed to the data link
        interface."
    ::= { pstcEntry 22 }

pstcInNUcastToIFC      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received multicast packets passed to the data link
        interface."
    ::= { pstcEntry 23 }

pstcOutDelayDiscards    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Send packets discarded due to excessive delay."
    ::= { pstcEntry 24 }

ptxq                      OBJECT IDENTIFIER ::= { nInterfaces 5 }

ptxqTable                 OBJECT-TYPE
    SYNTAX SEQUENCE OF PTXQEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The Port Transmit Queue Table"
    ::= { ptxq 1 }

ptxqEntry                 OBJECT-TYPE
    SYNTAX PTXQEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { ptxqPort }
    ::= { ptxqTable 1 }

```

```

PTXQEntry ::=
  SEQUENCE {
    ptxqPort          INTEGER,
    ptxqRegQSize     Gauge,
    ptxqRegQMax      INTEGER,
    ptxqExpQSize     Gauge,
    ptxqExpQMax      INTEGER,
    ptxqQHpCount     Counter,
    ptxqQRegCount    Counter,
    ptxqQExpCount    Counter,
    ptxqQHpDiscards  Counter,
    ptxqQRegDiscards Counter,
    ptxqQExpDiscards Counter,
    ptxqMultiQSize   Gauge,
    ptxqMultiQMax    INTEGER,
    ptxqMultiQDiscards Counter
  }

ptxqPort          OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "MAC-R port ID (1-4).  A number which uniquely identifies
    the port."
  ::= { ptxqEntry 1 }

ptxqRegQSize      OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Current regular queue size (0-REG_Q_MAX).  The number of
    regular priority packets which are currently queued for
    transmission on the port."
  ::= { ptxqEntry 2 }

ptxqRegQMax       OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The maximum number of regular priority packets which can be
    queued for transmission on the port."
  ::= { ptxqEntry 3 }

ptxqExpQSize      OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Current expedited queue size (0..EXP_Q_MAX).  The number of
    expedited packets which are currently queued for
    transmission on the port."
  ::= { ptxqEntry 4 }

ptxqExpQMax       OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The maximum number of expedited packets which can be queued
    for transmission on the port."
  ::= { ptxqEntry 5 }

```

```

ptxqQHpCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of attempts to queue a high priority packet for
        transmission."
    ::= { ptxqEntry 6 }

ptxqQExpCount     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of attempts to queue an expedited priority
        packet for transmission."
    ::= { ptxqEntry 7 }

ptxqQRegCount     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of attempts to queue a regular priority packet
        for transmission."
    ::= { ptxqEntry 8 }

ptxqQHpDiscards  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of failed attempts to queue a high priority
        packet."
    ::= { ptxqEntry 9 }

ptxqQExpDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of failed attempts to queue an expedited
        priority packet."
    ::= { ptxqEntry 10 }

ptxqQRegDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of failed attempts to queue a regular priority
        packet."
    ::= { ptxqEntry 11 }

ptxqMultiQSize   OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Current multicast queue size. The number of multicast
        packets which are queued for transmission on the (radio)
        port. Multicast packets are transmitted after HELLO packets
        on OWL radio ports."
    ::= { ptxqEntry 12 }

```

```

ptxqMultiQMax      OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The maximum number of multicast packets which will be
        queued for transmission on the (radio) port."
    ::= { ptxqEntry 13 }

ptxqMultiQDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of failed attempts to queue a multicast packet."
    ::= { ptxqEntry 14 }

pmsg                OBJECT IDENTIFIER ::= { nInterfaces 6 }

pmsgTable           OBJECT-TYPE
    SYNTAX SEQUENCE OF PmsgEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Pending Message Table"
    ::= { pmsg 1 }

pmsgEntry           OBJECT-TYPE
    SYNTAX PmsgEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { pmsgPort }
    ::= { pmsgTable 1 }

PmsgEntry ::=
    SEQUENCE {
        pmsgPort                INTEGER,
        pmsgPendRecCurrent      Gauge,
        pmsgPendRecMax          INTEGER,
        pmsgPendMsgCurrent      Gauge,
        pmsgPendMsgMax          INTEGER,
        pmsgPendMsgTotal        Counter,
        pmsgPendMsgDiscards     Counter,
        pmsgPendRecOverflowErrors Counter,
        pmsgPendMsgOverflowErrors Counter,
        pmsgPendAgedRecCount     Counter,
        pmsgPendAgedMsgCount     Counter
    }

pmsgPort            OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-R port ID (1-4).  A number which uniquely identifies
        the port."
    ::= { pmsgEntry 1 }

pmsgPendRecCurrent  OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Current terminal record count"
    ::= { pmsgEntry 2 }

```

```

pmsgPendRecMax          OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Maximum terminal record count"
    ::= { pmsgEntry 3 }

pmsgPendMsgCurrent      OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Current pending message count"
    ::= { pmsgEntry 4 }

pmsgPendMsgMax          OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Maximum pending message count"
    ::= { pmsgEntry 5 }

pmsgPendMsgTotal        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total pending message count"
    ::= { pmsgEntry 6 }

pmsgPendMsgDiscards     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of pending messages in-queue which were
        discarded before they could be delivered because the
        terminal's queue was full."
    ::= { pmsgEntry 7 }

pmsgPendRecOverflowErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times that a terminal requested pending
        message services when no pending message records were
        available."
    ::= { pmsgEntry 8 }

pmsgPendMsgOverflowErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times when the maximum number of stored
        messages, per platform, was exceeded."
    ::= { pmsgEntry 9 }

```

```

pmsgPendAgedRecCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of terminal records discarded due to maximum age
        (12 minutes)."
```

```

 ::= { pmsgEntry 10 }

pmsgPendAgedMsgCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of pending messages which were discarded due to
        maximum age. (default = 5 seconds)."
```

```

 ::= { pmsgEntry 11 }

nSNMP                      OBJECT IDENTIFIER ::= { norandNet 11 }

vlConfig                    OBJECT IDENTIFIER ::= { nSNMP 1 }

--Norand Community table defines the accepted community strings
--and their access privileges

-- The Community Table

communityTable              OBJECT-TYPE
    SYNTAX SEQUENCE OF CommunityEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The community table is used to define communities and their
        access privileges. Norand's implementation of the
        community table has some special considerations:

        1) GETs and SETs to the community table can only be
        accomplished using the SUPER-USER community string which
        Norand has defined;

        2) This SUPER-USER community string, or password, is
        defined in the first row of the community table. The
        communityName contained in the first row of the
        community table is always the SUPER-USER community
        string. This community string (communityName) may be
        modified.

        3) All rows of the community table are modifiable (SET)
        when using the SUPER-USER community string. However,
        for the first row of the community table, only the
        communityName object is modifiable. This ensures that
        the SUPER-USER will always have maximum access to the
        MIB data. All other rows in the community Table are
        accessible as defined in the MIB definition.

        4) The SUPER-USER and other default community string values
        can be found in Norand's User's Guide."
 ::= { vlConfig 2 }

-- Row Definition

communityEntry              OBJECT-TYPE
    SYNTAX CommunityEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Each entry relates to a specific community and associates
        to its access privileges."
    INDEX { communityIndex }
 ::= { communityTable 1 }

```



```

-- Columnar Object Definition

CommunityEntry ::=
    SEQUENCE {
        communityIndex      INTEGER,
        communityStatus     INTEGER,
        communityName       DisplayString,
        communityPrivileges INTEGER
        communityViewTblIndex  INTEGER
    }

-- Leaf Definition

communityIndex      OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Identifies the community row."
    ::= { communityEntry 1 }

communityStatus     OBJECT-TYPE
    SYNTAX  INTEGER { enabled(1),
                    disabled(2),
                    deleted(3) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "Status of a community record. Alterations to the table may
        only be performed by a manager using the SUPER-USER
        community name. Status types:
            Enabled = Community record active
            Disabled = Community record not active
            Deleted = Disables and nulls objects in record."
    ::= { communityEntry 2 }

communityName       OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..15))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The authoritative name for the community. Unless the
        Norand SUPER-USER community name is employed, a GET from
        this column yields an access violation."
    ::= { communityEntry 3 }

communityPrivileges OBJECT-TYPE
    SYNTAX  INTEGER { get-only(1),
                    set-and-get(3) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "SET and GET privileges of community."
    ::= { communityEntry 4 }

-- Norand trap table defines all trap target IP addresses

-- Table Definition

trapTargetTable     OBJECT-TYPE
    SYNTAX  SEQUENCE OF TrapTargetEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "The trap target table specifies the IP address of SNMPv1
        managers that expect trap notifications."
    ::= { v1Config 3 }

```

```

-- Row Definition

trapTargetEntry      OBJECT-TYPE
    SYNTAX  TrapTargetEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "Each entry relates to a specific named manager at a given
        IP address and belonging to given community."
    INDEX   { trapTargetIndex }
    ::= { trapTargetTable 1 }

-- Columnar Object Definition

TrapTargetEntry ::=
    SEQUENCE {
        trapTargetIndex      INTEGER,
        trapTargetStatus     INTEGER,
        trapTargetName       DisplayString,
        trapTargetIpAddress  IPAddress
    }

-- Leaf Definition

trapTargetIndex      OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Identifies the trapTarget row"
    ::= { trapTargetEntry 1 }

trapTargetStatus     OBJECT-TYPE
    SYNTAX  INTEGER { enabled(1),
                    disabled(2),
                    deleted(3) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "Status of a trapTarget record."

    ::= { trapTargetEntry 2 }

trapTargetName       OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..16))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The authoritative name for the trapTarget."
    ::= { trapTargetEntry 3 }

trapTargetIpAddress  OBJECT-TYPE
    SYNTAX  IPAddress
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "IP Address of manager (which is assumed to be bound to and
        listening on port 162)."
    ::= { trapTargetEntry 4 }

nBridge              OBJECT IDENTIFIER ::= { norandNet 17 }

rt                   OBJECT IDENTIFIER ::= { nBridge 2 }

-- The RT Table

```

```

-- Table Definition

rtTable          OBJECT-TYPE
SYNTAX SEQUENCE OF REntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Each entry in this table provides routing information for
child nodes which are reachable via a route."
::= { rt 2 }

-- Row Definition

rtEntry          OBJECT-TYPE
SYNTAX REntry
ACCESS not-accessible
STATUS mandatory
INDEX { rtDestination }
::= { rtTable 1 }

-- Columnar Object Definition

REntry ::=
SEQUENCE {
    rtDestination  PhysAddress,
    rtPort         INTEGER,
    rtAge          INTEGER,
    rtNodeId       INTEGER,
    rtAttachId     INTEGER,
    rtAttachTime   TimeTicks,
    rtApEaddr      PhysAddress,
    rtHopAddrLen   INTEGER,
    rtHopAddr16    INTEGER,
    rtHopEaddr     PhysAddress,
    rtIsBound      INTEGER,
    rtIsRemote     INTEGER,
    rtIsChild      INTEGER,
    rtIsAp         INTEGER,
    rtIsDistributed INTEGER,
    rtIsRemoteLan  INTEGER,
    rtNS           INTEGER,
    rtNR           INTEGER
}

-- Leaf Definition

rtDestination    OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The 802 address of the destination."
::= { rtEntry 1 }

rtPort           OBJECT-TYPE
SYNTAX INTEGER (1..4)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The MAC-R port ID (1-4). A number which uniquely identifies
the port."
::= { rtEntry 2 }

```

```

rtAge                OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The time (in minutes) since the route was updated."
 ::= { rtEntry 3 }

rtNodeId             OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"16-bit node ID of the destination. A 16-bit identifier which
uniquely identifies an OWL node in an OWL LAN."
 ::= { rtEntry 4 }

rtAttachId           OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Attach sequence number. The sequence number is copied from
an OWL ATTACH request PDU. The sequence number is not valid
for 'remote' nodes."
 ::= { rtEntry 5 }

rtAttachTime         OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Last attach time (.01 seconds)."
 ::= { rtEntry 6 }

rtApEaddr            OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"802 address of AP which is the first hop on the path to the
destination."
 ::= { rtEntry 7 }

rtHopAddrLen         OBJECT-TYPE
SYNTAX INTEGER { twoByte(2),
                 sixByte(6) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"MAC-D address length (2 or 6). A MAC-D entity may use either
16-bit locally assigned addresses or 48-bit 802 addresses."
 ::= { rtEntry 8 }

rtHopAddr16          OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"16-bit MAC-D address ( if rtHopAddrLen is twoByte(2) )."
 ::= { rtEntry 9 }

```

```
rtHopEaddr          OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"48-bit MAC-D address ( if rtHopAddrLen is sixByte(6) )."
 ::= { rtEntry 10 }

rtIsBound           OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is fully attached and the path can be
used to forward data."
 ::= { rtEntry 11 }

rtIsRemote          OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is a non-OWL node."
 ::= { rtEntry 12 }

rtIsChild           OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is a child node."
 ::= { rtEntry 13 }

rtIsAp              OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is an AP."
 ::= { rtEntry 14 }

rtIsDistributed     OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the path is through a distributed AP (root node
only)."
```

```
 ::= { rtEntry 15 }

rtIsRemoteLan      OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is a remote LAN."
 ::= { rtEntry 16 }
```

```

rtNS                OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"MAC-R send sequence number for terminal nodes. The 16-bit
(0-65535) sequence number of the last OWL data request PDU
sent to the destination."
 ::= { rtEntry 17 }

rtNR                OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"MAC-R receive sequence number for terminal nodes. The 16-bit
(0-65535) sequence number of the last OWL data request PDU
received from the destination."
 ::= { rtEntry 18 }

brg                OBJECT IDENTIFIER ::= { nBridge 3 }

-- The BRG Table

-- Table Definition

brgTable            OBJECT-TYPE
SYNTAX SEQUENCE OF BRGEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Each entry in this table provides bridge information for
child nodes which are reachable via a bridge."
 ::= { brg 2 }

-- Row Definition

brgEntry            OBJECT-TYPE
SYNTAX BRGEntry
ACCESS not-accessible
STATUS mandatory
INDEX { brgDestination }
 ::= { brgTable 1 }

-- Columnar Object Definition

BRGEntry ::=
SEQUENCE {
    brgDestination PhysAddress,
    brgPort         INTEGER,
    brgAge          INTEGER,
    brgType         INTEGER,
    brgIsPermanent INTEGER,
    brgTimestamp    TimeTicks
}

-- Leaf Definition

brgDestination     OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The 802 address of the destination."
 ::= { brgEntry 1 }

```

```

brgPort          OBJECT-TYPE
    SYNTAX INTEGER (1..4)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-R port ID (1-4).  A number which uniquely identifies the
        port."
    ::= { brgEntry 2 }

brgAge           OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Time (in minutes) since the entry was updated."
    ::= { brgEntry 3 }

brgType          OBJECT-TYPE
    SYNTAX INTEGER { primary(1),
                    secondary(2),
                    inbound(4) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Entry Type:
         primary   = 1,
         secondary = 2,
         inbound   = 4."
    ::= { brgEntry 4 }

brgIsPermanent  OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the entry is permanent."
    ::= { brgEntry 5 }

brgTimestamp     OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time when the primary or inbound entry was added or the
        time when the secondary entry was added or re-attached."
    ::= { brgEntry 6 }

addr          OBJECT IDENTIFIER ::= { nBridge 4 }

-- The Addr Table

-- Table Definition

addrTable        OBJECT-TYPE
    SYNTAX SEQUENCE OF AddrEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Each entry in this table provides address information for
        all OWL nodes in the network.  This table exists only at the
        root node."
    ::= { addr 2 }

-- Row Definition

```

```

addrEntry          OBJECT-TYPE
    SYNTAX AddrEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { addrDestination }
    ::= { addrTable 1 }

-- Columnar Object Definition

AddrEntry ::=
    SEQUENCE {
        addrDestination PhysAddress,
        addrAge          INTEGER,
        addrNodeId       INTEGER,
        addrAlias        DisplayString,
        addrDeviceId     INTEGER,
        addrIpAddress    IpAddress
    }

-- Leaf Definition

addrDestination    OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The 802 address of the registered port."
    ::= { addrEntry 1 }

addrAge            OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time (in minutes) since the entry was updated."
    ::= { addrEntry 2 }

addrNodeId         OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "16-bit (0-65535) node/port ID"
    ::= { addrEntry 3 }

addrAlias          OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..16))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "An alias for the 802 address."
    ::= { addrEntry 4 }

addrDeviceId       OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Device ID (0-65535).  An OWL node can, optionally, set a
        device ID in a registration request PDU."
    ::= { addrEntry 5 }

```



```

addrIpAddress      OBJECT-TYPE
    SYNTAX IpAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "32-bit IP address for IP nodes (e.g. APs)."
    ::= { addrEntry 6 }

brgState            OBJECT IDENTIFIER ::= { nBridge 6 }

-- The Bridge State Group

bsAddress           OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "802 address of the AP."
    ::= { brgState 3 }

bsLanId             OBJECT-TYPE
    SYNTAX INTEGER (0..254)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "OWL LAN ID (0-254)."
    ::= { brgState 4 }

bsCostToRoot       OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Path cost to the root."
    ::= { brgState 5 }

bsIsRoot            OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the AP is the root."
    ::= { brgState 6 }

bsIsAttached        OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the AP is attached."
    ::= { brgState 7 }

bsAttachId          OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "16-bit (0-65535) ATTACH sequence number. This number is
        incremented each time the AP sends an ATTACH request."
    ::= { brgState 8 }

```

```

bsMyRootPriority          OBJECT-TYPE
    SYNTAX INTEGER (0..7)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Root priority of the AP (0-7).  An AP with a root priority
        of 0 cannot become the root node.  The AP with the highest
        priority will become the root in an OWL LAN."
    ::= { brgState 9 }

bsRootPort                OBJECT-TYPE
    SYNTAX INTEGER (1..4)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-R root port number.  The port number (1-4) of the port
        used to communicate with the parent node."
    ::= { brgState 10 }

bsDesignatedRootAddress   OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "802 address of the current root."
    ::= { brgState 11 }

bsDesignatedRootPriority  OBJECT-TYPE
    SYNTAX INTEGER (1..7)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Root priority of the current root (1-7)."
    ::= { brgState 12 }

bsDesignatedRootSequence  OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Root sequence of the current root (0-255).  The sequence
        number identifies a single instance of the root."
    ::= { brgState 13 }

bsParentAddress           OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "802 address of the parent AP."
    ::= { brgState 14 }

bsPortCount               OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of MAC-R ports"
    ::= { brgState 15 }

```

```

bsNodeId OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "16-bit node ID (0-65535). The node ID uniquely identifies
        the node in an OWL LAN."
    ::= { brgState 16 }

bsRootChangedCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times that the root has changed."
    ::= { brgState 17 }

bsRootCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times that the AP became the root."
    ::= { brgState 18 }

bsAttachCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times that the AP has changed from an unattached
        state to an attached state."
    ::= { brgState 19 }

bsDetachReason OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Last detach reason code. The code indicates the reason
        that the AP became unattached for the last occurrence.
        0 - Initial Value,
        1 - A new root node was detected,
        2 - The network inactivity timer expired,
        4 - A better path to the root was detected,
        5 - The node's parent became unattached,
        7 - The node was in a detach list in a HELLO PDU,
        8 - The node was functioning as the root and relinquished
            the root status,
        9 - The maximum number of attache retries was exceeded
            without receiving an ATTACH response PDU,
        900-90F - A MAC-D link error occurred while sending a PDU to
            the parent node."
    ::= { brgState 20 }

bsNetworkTime OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Distributed network time (.01 second)."
    ::= { brgState 21 }

```

```

bsUniFloodLevel          OBJECT-TYPE
    SYNTAX INTEGER (1..2)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Flooding level for unicast frames (1-2)."
```

```

 ::= { brgState 22 }
```

```

bsMultiFloodLevel       OBJECT-TYPE
    SYNTAX INTEGER (0..3)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Flooding level for multicast frames (0-3)."
```

```

 ::= { brgState 23 }
```

```

bsIsPrimaryBridge       OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the AP bridges to the distribution LAN."
```

```

 ::= { brgState 24 }
```

```

bsIsSecondaryBridge     OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the AP is the designated bridge for a secondary
        LAN."
```

```

 ::= { brgState 25 }
```

```

bsUniFilterExpr         OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Unicast ethernet filter ID (0-255).  If the ID is non-zero,
        it points to a user defined expression which is used to
        filter unicast frames on the ethernet port."
```

```

 ::= { brgState 26 }
```

```

bsMultiFilterExpr       OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Multicast ethernet filter ID (0-255).  If the ID is
        non-zero, it points to a user defined expression which is
        used to filter multicast frames on the ethernet port."
```

```

 ::= { brgState 27 }
```

```

bridgeStats    OBJECT IDENTIFIER ::= { nBridge 7 }
```

```

bstcRouteCount          OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total route table entries"
```

```

 ::= { bridgeStats 3 }
```

```

bstcChildCount          OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of attached children"
    ::= { bridgeStats 4 }

bstcChildApCount        OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of attached AP children"
    ::= { bridgeStats 5 }

bstcRemoteCount         OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Non-OWL bridge table entries"
    ::= { bridgeStats 6 }

bstcPrimaryCount        OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Distribution LAN bridge table entries"
    ::= { bridgeStats 7 }

bstcInboundCount        OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Inbound bridge table entries"
    ::= { bridgeStats 8 }

bstcSecondaryCount      OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Secondary LAN bridge table entries"
    ::= { bridgeStats 9 }

bstcRemoteLanCount      OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Route entries for remote LANs"
    ::= { bridgeStats 10 }

bstcRouteGetErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Route table overflow errors"
    ::= { bridgeStats 11 }

```

```

bstcEntryGetErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Bridge table overflow errors"
    ::= { bridgeStats 12 }

bstcRmtLanGetErrors     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Remote LAN overflow errors"
    ::= { bridgeStats 13 }

bstcRouteSeqErrors     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Out-of-sequence route update errors"
    ::= { bridgeStats 14 }

bstcDeleteSeqErrors    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Out-of-sequence route delete errors"
    ::= { bridgeStats 15 }

bstcEntrySeqErrors     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Out-of-sequence bridge entry update errors"
    ::= { bridgeStats 16 }

bstcInvalidUpdateErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Invalid route update errors"
    ::= { bridgeStats 17 }

nControl                OBJECT IDENTIFIER ::= { norandNet 105 }

powerUp                 OBJECT IDENTIFIER ::= { nControl 1 }

pwrPowerUpCount        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Power-up count"
    ::= { powerUp 1 }

pwrNextPowerUpTime     OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Next power-up time (Used to reboot the device)."
```

```

softwareDownload OBJECT IDENTIFIER ::= { nControl 2 }

sdServerIpAddress OBJECT-TYPE
    SYNTAX IpAddress
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "TFTP server IP address"
    ::= { softwareDownload 2 }

sdScriptFilename OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..80))
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Download script filename (May include path)."
    ::= { softwareDownload 3 }

sdStartTime OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The amount of time to delay before beginning the software
        download."
    ::= { softwareDownload 1 }

sdStatus OBJECT-TYPE
    SYNTAX INTEGER {
        sdPending(1),
        sdSTStopped(2),
        sdInProgress(3),
        sdTerminated(4),
        sdSuccess(5),
        sdError(6),
        pwrNPUT(7),
        tftpError(8)
    }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Status of the current software download."
    ::= { softwareDownload 4 }

sdErrorString OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..40))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Description of sdStatus field"
    ::= { softwareDownload 5 }

sdCheckPoint OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "An application variable intended to contain a number
        relating the progress of the current software download."
    ::= { softwareDownload 6 }

```

```
sdSetActivePointers OBJECT-TYPE
  SYNTAX INTEGER {
      none(1),
      boot(2),
      data(3),
      both(4)
  }
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
  "If the device reboots due to the expiration of the
  pwrNextPwrUpTime timer, this value specifies which active
  pointers will be toggled prior to rebooting."
  ::= { softwareDownload 7 }

sdTerminate          OBJECT-TYPE
  SYNTAX INTEGER {
      true(1),
      false(2)
  }
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
  "Terminate the current software download."
  ::= { softwareDownload 8 }
```

END

Norand Open Wireless LAN MIB RC4030E Gateway



This section describes various groups Intermecc Technologies Corporation supports for the RC4030E Gateway (GW). Table 5-1 lists groups, their meaning, and page numbers where each group's table summary and each group's actual definitions appear in this section.

*Table 5-1
GW MIB Directory*

Group	Meaning	Group Summary	MIB Definition
Product OIDs			
products	Intermec [®] Products	5-2	5-12
System Information			
hw	Hardware Information	5-2	5-12
dir	Software Directory Listing	5-3	5-13
criticalErrors	Critical Errors Information	5-3	5-14
SNMP Version 1 Configuration			
community	Community Table	5-4	5-16
trapTarget	Trap Target Table	5-4	5-17
Transport Groups			
wst	Wireless Transport Protocol (WTP) Status Table	5-5	5-18
hlit	High Level Interface Table	5-6	5-23
webst	WTP Control Block Status Table	5-6	5-24
Application Layer Groups			
gs	Gateway Statistics	5-8	5-28
rcbs	Radio Terminal Control Block Statistics	5-9	5-33
hd	Host Delay Statistics	5-10	5-39
td	Terminal Transaction Delay Statistics	5-10	5-40
ahost	Asynchronous Host application	5-11	5-41
Control Group			
powerUp	Power Up Objects	5-11	5-44

Gateway MIB Outline

This outline summarizes the various MIB groups Intermecc Technologies supports for the RC4030E Gateway.

Product OIDs

This group contains an Object IDentification (OID) for each Intermecc[®] device.

Table 5-2
products GROUP

Device Products
norand.manage.products.x
(1.3.6.1.4.1.469.1000.1.x)

Object ID	Object Name	Object Type	Access
1	ap6710	OBJECT ID	Not Applicable (N/A)
2	gw4030	OBJECT ID	(N/A)
3	wnas	OBJECT ID	(N/A)
4	ts6950	OBJECT ID	(N/A)
5	gwap6910	OBJECT ID	(N/A)

System Information

The following groups contain system level objects describing hardware and file-system configuration properties. The groups also contain information about critical errors.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- hw Hardware Information (page 5-12)
- dir Software Directory Listing (page 5-13)
- criticalErrors Critical Errors Information (page 5-14)

Table 5-3
hw GROUP

Device Hardware Information
norand.manage.norandNet.nSystem.hw.x
(1.3.6.1.4.1.469.1000.2.1.1.x)

Object ID	Object Name	Object Type	Access
1	hwPartNo	INTEGER	read
2	hwDescription	DisplayString	read
3	hwRevision	INTEGER	read
4	hwSerialNo	INTEGER	read
5	hwID	INTEGER	read

Table 5-4
dir GROUP

Device Software Directory Listing
norand.manage.norandNet.nSystem.file.dir.x
(1.3.6.1.4.1.469.1000.2.1.3.3.x)

Object ID	Object Name	Object Type	Access
2.1.1	dirIndex	INTEGER	read
2.1.2	dirName	DisplayString	read
2.1.3	dirSegment	INTEGER	read
2.1.4	dirType	INTEGER	read
2.1.5	dirSize	INTEGER	read
2.1.6	dirDate	DisplayString	read
2.1.7	dirTime	DisplayString	read
2.1.8	dirVersion	DisplayString	read

Table 5-5
criticalErrors GROUP

Device Critical Errors Information
norand.manage.norandNet.nSystem.sysErrors.criticalErrors.x
(1.3.6.1.4.1.469.1000.2.1.4.1.x)

Object ID	Object Name	Object Type	Access
1	ceEnabled	INTEGER	read
2	ceOverflow	INTEGER	read
3	ceReset	INTEGER	write
4.1.1	ceLogErrorCode	INTEGER	read
4.1.2	ceLogErrorCount	Counter	read

SNMP Version 1 Configuration Group

This group contains objects that configure the version 1 Simple Network Management Protocol (SNMP) agent.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- community Community Table (page 5-16)
- trapTarget Trap Target Table (page 5-17)

Table 5-6

community TABLE

Device SNMP v1 Configurations
norand.manage.norandNet.nSNMP.v1Config.x
(1.3.6.1.4.1.469.1000.2.11.1.x)

Object ID	Object Name	Object Type	Access
2.1.1	communityIndex	INTEGER	read
2.1.2	communityStatus	INTEGER	write
2.1.3	communityName	DisplayString	write
2.1.4	communityPrivileges	INTEGER	write

Table 5-7

trapTarget TABLE

Device SNMP v1 Configurations
norand.manage.norandNet.nSNMP.v1Config.x
(1.3.6.1.4.1.469.1000.2.11.1.x)

Object ID	Object Name	Object Type	Access
3.1.1	trapTargetIndex	INTEGER	read
3.1.2	trapTargetStatus	INTEGER	write
3.1.3	trapTargetName	DisplayString	write
3.1.4	trapTargetIpAddress	IpAddress	write

Transport Groups

These groups contain objects that relate to active transport layers information.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- wst Wireless Transport Protocol (WTP) Status Table (page 5-18)
- hlit High Level Interface Table (page 5-23)
- wcbst WTP Control Block Status Table (page 5-24)

Table 5-8
wst GROUP

WTP Status Table
norand.manage.norandNet.nTransport.wst.x
(1.3.6.1.4.1.469.1000.2.102.2.x)

Object ID	Object Name	Object Type	Access
1	wstReset	INTEGER	read
2	wstLastReset	TimeTicks	read
3	wstClockTicks	TimeTicks	read
4	wstInternalErrorCount	Counter	read
5	wstSessionsLost	Counter	read
6	wstFrmrCount	Counter	read
7	wstDataBytesXmit	Counter	read
8	wstDataBytesRecv	Counter	read
9	wstIframesXmit	Counter	read
10	wstIframesRecv	Counter	read
11	wstIframesRetransmitted	Counter	read
12	wstChecksumErrors	Counter	read
13	wstHIDiscardCount	Counter	read
14	wstState	INTEGER	read
15	wstBrgBusy	INTEGER	read
16	wstT1Timeouts	Counter	read
17	wstT1Bucket1	Counter	read
18	wstT1Bucket2	Counter	read
19	wstT1Bucket3	Counter	read
20	wstT1Bucket4	Counter	read
21	wstSABMCount	Counter	read
22	wstDiscCount	Counter	read
23	wstDMCount	Counter	read
24	wstRNRCCount	Counter	read
25	wstRejectCount	Counter	read
26	wstSessionsReset	Counter	read
27	wstBrgSrvTime	TimeTicks	read

Table 5-8 (Continued)
wst GROUP

WTP Status Table
norand.manage.norandNet.nTransport.wst.x
(1.3.6.1.4.1.469.1000.2.102.2.x)

Object ID	Object Name	Object Type	Access
28	wstBrgSrvCount	Counter	read
29	wstBrgSrvThreshold	TimeTicks	read
30	wstBrgSrvThreshCount	Counter	read
31	wstBrgSrvLongest	TimeTicks	read
32	wstBrgTxErrors	Counter	read
33	wstFatalBrgErrors	Counter	read

Table 5-9
hlit GROUP

High Level Interface Table
norand.manage.norandNet.nTransport.hlit.x
(1.3.6.1.4.1.469.1000.2.102.3.x)

Object ID	Object Name	Object Type	Access
1	hlitReset	INTEGER	write
2	hlitLastReset	TimeTicks	read
3	hlitTime	TimeTicks	read
4	hlitCount	Counter	read
5	hlitThreshold	TimeTicks	write
6	hlitThreshCount	Counter	read
7	hlitLongest	Counter	read

Table 5-10
wcbst GROUP

WTP Control Block Status Table
norand.manage.norandNet.nTransport.wcbst.x
(1.3.6.1.4.1.469.1000.2.102.4.x)

Object ID	Object Name	Object Type	Access
1	wcbstReset	INTEGER	write
2	wcbstLastReset	TimeTicks	read
3.1.1	wcbstIndex	INTEGER	read
3.1.2	wcbstInUse	INTEGER	read
3.1.3	wcbstH1State	INTEGER	read
3.1.4	wcbstDsap	INTEGER	read
3.1.5	wcbstSsap	INTEGER	read
3.1.6	wcbstNet16ADDR	INTEGER	read
3.1.7	wcbstT1Value	INTEGER	read
3.1.8	wcbstT1Average	INTEGER	read

Table 5-10 (Continued)

wcbst GROUP

WTP Control Block Status Table
 norand.manage.norandNet.nTransport.wcbst.x
 (1.3.6.1.4.1.469.1000.2.102.4.x)

Object ID	Object Name	Object Type	Access
3.1.9	wcbstT1Deviation	INTEGER	read
3.1.10	wcbstT1Timeouts	Counter	write
3.1.11	wcbstT2Timeouts	Counter	write
3.1.12	wcbstTiTimeouts	Counter	write
3.1.13	wcbstFramesRecv	Counter	write
3.1.14	wcbstIframesRecv	Counter	write
3.1.15	wcbstIframesDiscarded	Counter	write
3.1.16	wcbstDataBytesRecv	Counter	write
3.1.17	wcbstFramesXmit	Counter	write
3.1.18	wcbstIframesXmit	Counter	write
3.1.19	wcbstDataBytesXmit	Counter	write
3.1.20	wcbstIframesRetransmitted	Counter	write
3.1.21	wcbstState	INTEGER	read

Application Layer Group

These groups relate resident applications information.

► **NOTE:**

The MIB definition for each group starts on the page given below.

Gateway application:

- gs Gateway Statistics (page 5-28)
- rcbs Radio Terminal Control Block Statistics (page 5-33)
- hd Host Delay Statistics (page 5-39)
- td Terminal Transaction Delay Statistics (page 5-40)

Asynchronous host:

- ahost Asynchronous Host application (page 5-41)

Table 5-11
gs GROUP

Gateway Statistics
norand.manage.norandNet.nApplication.gwSession.gs.x
(1.3.6.1.4.1.469.1000.2.104.1.1.x)

Object ID	Object Name	Object Type	Access
1	gsReset	INTEGER	write
2	gsLastReset	TimeTicks	read
3	gsClockTicks	TimeTicks	read
4	gsNetworkErrors	Counter	read
5	gsHostDataCount	Counter	read
6	gsHostHaltCount	Counter	read
7	gsHostInvalidCount	Counter	read
8	gsHostDiscardCount	Counter	read
9	gsConnectCount	Counter	read
10	gsCloseCount	Counter	read
11	gsBlocksXmit	Counter	read
12	gsDataXmit	Counter	read
13	gsBlocksRecv	Counter	read
14	gsDataRecv	Counter	read
15	gsTermResetCount	Counter	read
16	gsTermContinueCount	Counter	read
17	gsTermInvalidCount	Counter	read
18	gsConnectionErrors	Counter	read
19	gsConnectionResets	Counter	read
20	gsRetransmissions	Counter	read
21	gsRecvSeqErrors	Counter	read
22	gsXmitErrors	Counter	read
23	gsTermDiscardCount	Counter	read

Table 5-11 (Continued)
gs GROUP

Gateway Statistics
norand.manage.norandNet.nApplication.gwSession.gs.x
(1.3.6.1.4.1.469.1000.2.104.1.1.x)

Object ID	Object Name	Object Type	Access
24	gsHostDelayMax	INTEGER	write
25	gsHostDelayTotal	Counter	read
26	gsHostTransactions	Counter	read
27	gsRecvErrors	Counter	read
28	gsTermDelayMax	INTEGER	write
29	gsTermDelayThreshold	INTEGER	write
30	gsTermDelayTotal	Counter	read
31	gsTermTransactions	Counter	read
32	gsTermThresholdCount	Counter	read
33	gsTermMaxCount	Counter	read
34	gsHostMaxCount	Counter	read
35	gsTermDelayTraceOn	INTEGER	write
36	gsHostActive	INTEGER	read

Table 5-12
rcbs GROUP

Gateway (Terminal Emulation) Session Table
(each record is a terminal session)
norand.manage.norandNet.nApplication.gwSession.rcbs.x
(1.3.6.1.4.1.469.1000.2.104.1.2.x)

Object ID	Object Name	Object Type	Access
4.1.1	rcbsIndex	INTEGER	read
4.1.2	rcbsTerminal	INTEGER	read
4.1.3	rcbsType	INTEGER	read
4.1.4	rcbsState	INTEGER	read
4.1.5	rcbsLLCIndex	INTEGER	read
4.1.6	rcbsHostDataCount	Counter	read
4.1.7	rcbsBlocksXmit	Counter	read
4.1.8	rcbsDataXmit	Counter	read
4.1.9	rcbsBlocksRecv	Counter	read
4.1.10	rcbsDataRecv	Counter	read
4.1.11	rcbsTermResetCount	Counter	read
4.1.12	rcbsTermContinueCount	Counter	read
4.1.13	rcbsCloseCount	Counter	read
4.1.14	rcbsHostTransactions	Counter	read
4.1.15	rcbsHostDelayTotal	Counter	read

Table 5-12 (Continued)
rcbs GROUP

Gateway (Terminal Emulation) Session Table
(each record is a terminal session)
norand.manage.norandNet.nApplication.gwSession.rcbs.x
(1.3.6.1.4.1.469.1000.2.104.1.2.x)

Object ID	Object Name	Object Type	Access
4.1.16	rcbsHostDelayLast	INTEGER	read
4.1.17	rcbsTermResetTime	INTEGER	read
4.1.18	rcbsTermInactTime	INTEGER	read
4.1.19	rcbsHostInactTime	INTEGER	read
4.1.20	rcbsConnectionErrors	Counter	read
4.1.21	rcbsNetAddress	PhysAddress	read
4.1.22	rcbsSeqErrors	Counter	read
4.1.23	rcbsTermTransactions	Counter	read
4.1.24	rcbsTermDelayTotal	Counter	read
4.1.25	rcbsTermDelayLast	INTEGER	read
4.1.26	rcbsTermDelayLongest	INTEGER	read
4.1.27	rcbsTermThresholdCount	Counter	read
4.1.28	rcbsTermMaxCount	Counter	read
4.1.29	rcbsTermDelayTraceOn	INTEGER	write
4.1.30	rcbsHostAlias	DisplayS- tring	read

Table 5-13
hd GROUP

Host Delay Statistics
norand.manage.norandNet.nApplication.gwSession.hd.x
(1.3.6.1.4.1.469.1000.2.104.1.3.x)

Object ID	Object Name	Object Type	Access
4.1.1	hdHostDelay	INTEGER	write
4.1.2	hdCount	Counter	read
4.1.3	hdBucket	INTEGER	read

Table 5-14
td GROUP

Terminal Transaction Delay Statistics
norand.manage.norandNet.nApplication.gwSession.td.x
(1.3.6.1.4.1.469.1000.2.104.1.4.x)

Object ID	Object Name	Object Type	Access
4.1.1	tdTrxnTime	INTEGER	write
4.1.2	tdCount	Counter	read
4.1.3	tdBucket	INTEGER	read

Table 5-15
ahost GROUP

Asynchronous Host
norand.manage.norandNet.nApplication.ahost.x
(1.3.6.1.4.1.469.1000.2.104.2.x)

Object ID	Object Name	Object Type	Access
1	ahstReset	INTEGER	write
2	ahstLastReset	TimeTicks	read
3	ahstFramesOut	Counter	read
4	ahstFramesIn	Counter	read
5	ahstIdleDetected	Counter	read
6	ahstNoErrors	Counter	read
7	ahstSyntaxErrors	Counter	read
8	ahstLengthErrors	Counter	read
9	ahstDisabledErrors	Counter	read
10	ahstGapErrors	Counter	read
11	ahstCheckErrors	Counter	read
12	ahstNoBuffersErrors	Counter	read
13	ahstSequenceErrors	Counter	read
15	ahstParityErrors	Counter	read
16	ahstFrameErrors	Counter	read
17	ahstNoiseErrors	Counter	read
18	ahstBreakErrors	Counter	read

Control Group

The objects in the following group exerts control over Intermecc Access Points. Present functions include rebooting.

► **NOTE:**

The MIB definition for this group starts on the page given.

- powerUp Power Up Objects (page 5-44)

Table 5-16
powerUp GROUP

Device Power Up Objects
norand.manage.norandNet.nControl.powerUp.x
(1.3.6.1.4.1.469.1000.2.105.1.x)

Object ID	Object Name	Object Type	Access
1	pwrPowerUpCount	Counter	read
2	pwrNextPowerUpTime	TimeTicks	write

Gateway MIB Definitions

Following is the RC4030E Gateway MIB definition for the NORAND® Open Wireless LAN.

```
-- Norand Open Wireless LAN MIB - RC4030 Gateway
-- Version 1.15
-- Version Date: 8/23/96
-- This MIB contains objects supported by V1.15 of the RC4030 Gateway.

-- Subject to Change

OWL DEFINITIONS ::= BEGIN

    IMPORTS
        enterprises, IpAddress, Counter, Gauge, TimeTicks
            FROM RFC1155-SMI
        PhysAddress, DisplayString
            FROM RFC1213-MIB
        OBJECT-TYPE
            FROM RFC-1212;

    -- This MIB module uses the extended OBJECT-TYPE macro as defined in
    -- RFC-1212;

    norand
        OBJECT IDENTIFIER ::= { enterprises 469 }
    manage
        OBJECT IDENTIFIER ::= { norand 1000 }
    products
        OBJECT IDENTIFIER ::= { manage 1 }
        ap6710
            OBJECT IDENTIFIER ::= { products 1 }
        gw4030
            OBJECT IDENTIFIER ::= { products 2 }
        wnas
            OBJECT IDENTIFIER ::= { products 3 }
        ts6950
            OBJECT IDENTIFIER ::= { products 4 }
        gwap6910
            OBJECT IDENTIFIER ::= { products 5 }
    norandNet
        OBJECT IDENTIFIER ::= { manage 2 }
    nSystem
        OBJECT IDENTIFIER ::= { norandNet 1 }

    hw
        OBJECT IDENTIFIER ::= { nSystem 1 }

    -- The Hardware Parameters Group

    hwPartNo      OBJECT-TYPE
        SYNTAX INTEGER (0..2147483647)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The Norand part number of the hardware device."
        ::= { hw 1 }

    hwDescription OBJECT-TYPE
        SYNTAX DisplayString (SIZE (0..40))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The description of the hardware device."
        ::= { hw 2 }

    hwRevision    OBJECT-TYPE
        SYNTAX INTEGER (0..2147483647)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The revision level of the hardware device."
        ::= { hw 3 }
```

```

hwSerialNo    OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The serial number of the hardware device."
    ::= { hw 4 }

hwID          OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The device identifier of the hardware device. Values =
        3250, 4000, 4020, 4030, 4033, 3240, 1000, 1100, 1700, 5940,
        4650, 100 (ACE process), 200 (DOSNMS), 300 (Norand Proxy
        Agent), 6710 (Access Point)."
    ::= { hw 5 }

file          OBJECT IDENTIFIER ::= { nSystem 3 }

dir          OBJECT IDENTIFIER ::= { file 3 }

-- The File Directory Table

-- Table Definition

dirTable     OBJECT-TYPE
    SYNTAX SEQUENCE OF DIREntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The FileSystem Directory"
    ::= { dir 2 }

-- Row Definition

dirEntry     OBJECT-TYPE
    SYNTAX DIREntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { dirIndex }
    ::= { dirTable 1 }

-- Columnar Object Definitions

DIREntry ::=
    SEQUENCE {
        dirIndex    INTEGER,
        dirName     DisplayString,
        dirSegment  INTEGER,
        dirType     INTEGER,
        dirSize     INTEGER,
        dirDate     DisplayString,
        dirTime     DisplayString,
        dirVersion  DisplayString
    }

dirIndex     OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Directory Index"
    ::= { dirEntry 1 }

```

```

dirName          OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..14))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "File name"
    ::= { dirEntry 2 }

dirSegment       OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "File segment (1 - (NUM_SEGMENTS + 1)). The segment ID
        which identifies the segment containing the file."
    ::= { dirEntry 3 }

dirType          OBJECT-TYPE
    SYNTAX INTEGER { executable(1),
                    data(2),
                    invalid(3) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "File type:
         executable = 1,
         data       = 2,
         invalid    = 3."
    ::= { dirEntry 4 }

dirSize          OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file size in bytes."
    ::= { dirEntry 5 }

dirDate          OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..12))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file date in MM-DD-YYYY display format."
    ::= { dirEntry 6 }

dirTime          OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..10))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file time in HH:MM:SS display format."
    ::= { dirEntry 7 }

dirVersion       OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..8))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file version in v99.99 display format."
    ::= { dirEntry 8 }

sysErrors        OBJECT IDENTIFIER ::= { nSystem 4 }

criticalErrors   OBJECT IDENTIFIER ::= { sysErrors 1 }

```

```

ceEnabled          OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A value of true(1) signifies that the critical error
        log was successfully initialized as part of the power-up
        sequence. Any errors in that initialization process
        result in a value of false(2)."
```

::= { criticalErrors 1 }

```

ceOverflow         OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Overflow error code. If the overflow code is non-zero,
        it indicates that the log has overflowed and the
        overflow code contains the last displaced value."
```

::= { criticalErrors 2 }

```

ceReset           OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "A user can reset the critical error log by setting
        ceReset to true(1). Valid values are true(1) or
        false(2)."
```

::= { criticalErrors 3 }

```

ceLogTable        OBJECT-TYPE
    SYNTAX SEQUENCE OF CELogEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Critical Error Log Table"
```

::= { criticalErrors 4 }

```

ceLogEntry        OBJECT-TYPE
    SYNTAX CELogEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { ceLogErrorCode }
    ::= { ceLogTable 1 }
```

```

CELogEntry ::=
    SEQUENCE {
        ceLogErrorCode INTEGER,
        ceLogErrorCount Counter
    }
```

```

ceLogErrorCode    OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Critical error code. A 16-bit value which uniquely
        identifies a system software error. The error codes are
        intended for internal Norand use."
```

::= { ceLogEntry 1 }

```

ceLogErrorCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Error count for the associated ceLogErrorCode"
    ::= { ceLogEntry 2 }

nSNMP          OBJECT IDENTIFIER ::= { norandNet 11 }

vlConfig      OBJECT IDENTIFIER ::= { nSNMP 1 }

--Norand Community table defines the accepted community strings
--and their access privileges.

-- The Community Table

communityTable OBJECT-TYPE
    SYNTAX SEQUENCE OF CommunityEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The community table is used to define communities and their
        access privileges. Norand's implementation of the
        community table has some special considerations:

        1) GETs and SETs to the community table can only be
            accomplished using the SUPER-USER community string which
            Norand has defined;

        2) This SUPER-USER community string, or password, is
            defined in the first row of the community table. The
            communityName contained in the first row of the
            community table is always the SUPER-USER community
            string. This community string (communityName) may be
            modified.

        3) All rows of the community table are modifiable (SET)
            when using the SUPER-USER community string. However,
            for the first row of the community table, only the
            communityName object is modifiable. This ensures that
            the SUPER-USER will always have maximum access to the
            MIB data. All other rows in the community Table are
            accessible as defined in the MIB definition.

        4) The SUPER-USER and other default community string values
            can be found in Norand's User's Guide."
    ::= { vlConfig 2 }

-- Row Definition

communityEntry OBJECT-TYPE
    SYNTAX CommunityEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Each entry relates to a specific community and associates
        to its access privileges."
    INDEX { communityIndex }
    ::= { communityTable 1 }

-- Columnar Object Definition

CommunityEntry ::=
    SEQUENCE {
        communityIndex      INTEGER,
        communityStatus     INTEGER,
        communityName       DisplayString,
        communityPrivileges INTEGER
    }

```



```

-- Leaf Definition

communityIndex      OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Identifies the community row."
    ::= { communityEntry 1 }

communityStatus      OBJECT-TYPE
    SYNTAX  INTEGER { enabled(1),
                    disabled(2),
                    deleted(3) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "Status of a community record. Alterations to the table may
        only be performed by a manager using the SUPER-USER
        community name. Status types:
         Enabled = Community record active
         Disabled = Community record not active
         Deleted = Disables and nulls objects in record."
    ::= { communityEntry 2 }

communityName        OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..15))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The authoritative name for the community. Unless the
        Norand SUPER-USER community name is employed, a GET from
        this column yields an access violation."
    ::= { communityEntry 3 }

communityPrivileges OBJECT-TYPE
    SYNTAX  INTEGER { get-only(1),
                    set-and-get(3) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "SET and GET privileges of community."
    ::= { communityEntry 4 }
-- Norand trap table defines all trap target IP addresses

-- Table Definition

trapTargetTable      OBJECT-TYPE
    SYNTAX  SEQUENCE OF TrapTargetEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "The trap target table specifies the IP address of SNMPv1
        managers that expect trap notifications."
    ::= { v1Config 3 }

-- Row Definition

trapTargetEntry      OBJECT-TYPE
    SYNTAX  TrapTargetEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "Each entry relates to a specific named manager at a given
        IP address and belonging to given community."
    INDEX   { trapTargetIndex }
    ::= { trapTargetTable 1 }

```

```

-- Columnar Object Definition

TrapTargetEntry ::=
    SEQUENCE {
        trapTargetIndex      INTEGER,
        trapTargetStatus     INTEGER,
        trapTargetName       DisplayString,
        trapTargetIpAddress  IpAddress
    }

-- Leaf Definition

trapTargetIndex      OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Identifies the trapTarget row"
    ::= { trapTargetEntry 1 }

trapTargetStatus     OBJECT-TYPE
    SYNTAX  INTEGER { enabled(1),
                    disabled(2),
                    deleted(3) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "Status of a trapTarget record."
    ::= { trapTargetEntry 2 }

trapTargetName       OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..16))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The authoritative name for the trapTarget."
    ::= { trapTargetEntry 3 }

trapTargetIpAddress OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "IP Address of manager (which is assumed to be bound to and
        listening on port 162)."
    ::= { trapTargetEntry 4 }

nTransport            OBJECT IDENTIFIER ::= { norandNET 102 }

wst                   OBJECT IDENTIFIER ::= { nTransport 2 }

wstReset              OBJECT-TYPE
    SYNTAX  INTEGER { true(1), false(2) }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Reset variable for the wst group."
    ::= { wst 1 }

wstLastReset          OBJECT-TYPE
    SYNTAX  TimeTicks
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Elapsed time since the wst group was last reset."
    ::= { wst 2 }

```

```

wstClockTicks          OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "WTP clock ticks in hundredths of seconds."
    ::= { wst 3 }

wstInternalErrorCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of internal WTP errors."
    ::= { wst 4 }

wstSessionsLost        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of active WTP connections which were terminated."
    ::= { wst 5 }

wstFrmrCount           OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of WTP Frame Rejects which were transmitted."
    ::= { wst 6 }

wstDataBytesXmit       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of bytes transmitted."
    ::= { wst 7 }

wstDataBytesRecv       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of bytes received."
    ::= { wst 8 }

wstIframesXmit         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames transmitted."
    ::= { wst 9 }

wstIframesRecv         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames received."
    ::= { wst 10 }

```

```
wstIframesRetransmitted OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames which were lost and
        retransmitted."
    ::= { wst 11 }

wstCheckSumErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of WTP information frames which were discarded
        to WTP check sum errors."
    ::= { wst 12 }

wstHlDiscardCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received WTP information frames which were
        discarded without being read by a higher layer application.
        Received information frames are queued per connection and
        should normally not be discarded."
    ::= { wst 13 }

wstState                   OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The internal software state of the WTP module."
    ::= { wst 14 }

wstBrgBusy                  OBJECT-TYPE
    SYNTAX INTEGER (0..1)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The value of wstBrgBusy is 1 if the WTP currently has an
        outstanding network transmit request. The value is
        otherwise 0."
    ::= { wst 15 }

wstTlTimeouts              OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times the WTP Tl timer has expired
        (i.e. due to lost frames)."
    ::= { wst 16 }

wstTlBucket1               OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times that 2 consecutive Tl errors have
        occurred."
    ::= { wst 17 }
```

```
wstTlBucket2          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times that 3 consecutive Tl errors have
        occurred."
    ::= { wst 18 }

wstTlBucket3          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times that 5 consecutive Tl errors have
        occurred."
    ::= { wst 19 }

wstTlBucket4          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times that 6 or more consecutive Tl
        errors have occurred."
    ::= { wst 20 }

wstSABMCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total SABM frames received by the local WTP entity."
    ::= { wst 21 }

wstDiscCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of DISC frames received by the local WTP entity"
    ::= { wst 22 }

wstDMCount            OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of DM frames received by the local WTP entity."
    ::= { wst 23 }

wstRNRCount           OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of RNR frames received by the local WTP entity."
    ::= { wst 24 }
```

```

wstRejectCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of REJECT frames received by the local WTP
        entity."
    ::= { wst 25 }

wstSessionsReset        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of active WTP sessions which have been reset in
        the local WTP entity."
    ::= { wst 26 }

wstBrgSrvTime           OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total bridge service time for WTP transmissions in
        hundredths of seconds."
    ::= { wst 27 }

wstBrgSrvCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of WTP bridge transmissions.  The average
        service time is wstBrgSrvTime/wstBrgSrvCount."
    ::= { wst 28 }

wstBrgSrvThreshold      OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A threshold bridge service time for WTP transmissions.
        Service times which exceed the threshold are counted in
        wstBrgSrvThreshCount."
    ::= { wst 29 }

wstBrgSrvThreshCount    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A count of the number of times the bridge service time has
        exceeded the threshold time defined by wstBrgSrvThreshold."
    ::= { wst 30 }

wstBrgSrvLongest        OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The longest network service time measured for a WTP
        transmission."
    ::= { wst 31 }

```

```

wstBrgTxErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of transmission errors reported to the WTP layer
        by the network layer."
    ::= { wst 32 }

wstFatalBrgErrors       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of data errors reported to the WTP layer by the
        network layer.  A fatal error indicates that the network
        root node has changed."
    ::= { wst 33 }

hlit                    OBJECT IDENTIFIER ::= { nTransport 3 }

hlitReset               OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the HLIT group."
    ::= { hlit 1 }

hlitLastReset          OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Elapsed time since the HLIT group was reset."
    ::= { hlit 2 }

hlitTime               OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This is the cumulative time that the WTP took to service
        transmit requests.  Time is in hundredths of seconds."
    ::= { hlit 3 }

hlitCount              OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This is the number of times the WTP was sent a transmit
        request."
    ::= { hlit 4 }

hlitThreshold          OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "When the WTP exceeds this time while servicing a transmit
        request, the hlitThreshCount will be incremented.  Time is
        in hundredths of seconds."
    ::= { hlit 5 }

```

```

hlitThreshCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "When the WTP exceeds the time in hlitThreshold, this count
        is incremented. Time in hundreths of seconds."
    ::= { hlit 6 }

hlitLongest      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This is the maximum time that the WTP took to service
        transmit requests. Time is in hundreths of seconds."
    ::= { hlit 7 }

wcbst            OBJECT IDENTIFIER ::= { nTransport 4 }

wcbstReset      OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the wcbst table."
    ::= { wcbst 1 }

wcbstLastReset  OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Elapsed time since the wcbst table was reset."
    ::= { wcbst 2 }

wcbstTable      OBJECT-TYPE
    SYNTAX SEQUENCE OF WcbstEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "WTP connection control block statistics table. Each entry
        is for an WTP connection statistics object. Statistics are
        per WTP connection."
    ::= { wcbst 3 }

wcbstEntry      OBJECT-TYPE
    SYNTAX WcbstEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { wcbstIndex }
    ::= { wcbstTable 1 }

```



```

WcbsEntry ::=
  SEQUENCE {
    wcbstIndex          INTEGER,
    wcbstInUse         INTEGER,
    wcbstHlState       INTEGER,
    wcbstDsap          INTEGER,
    wcbstSsap          INTEGER,
    wcbstNet16Addr     INTEGER,
    wcbstT1Value       INTEGER,
    wcbstT1Average     INTEGER,
    wcbstT1Deviation   INTEGER,
    wcbstT1Timeouts    Counter,
    wcbstT2Timeouts    Counter,
    wcbstTiTimeouts    Counter,
    wcbstFramesRecv    Counter,
    wcbstIframesRecv   Counter,
    wcbstIframesDiscarded Counter,
    wcbstDataBytesRecv Counter,
    wcbstFramesXmit     Counter,
    wcbstIframesXmit    Counter,
    wcbstDataBytesXmit Counter,
    wcbstIframesRetransmitted Counter,
    wcbstState          INTEGER
  }

wcbstIndex          OBJECT-TYPE
  SYNTAX INTEGER (1..999)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Connection control block identifier"
  ::= { wcbsEntry 1 }

wcbstInUse          OBJECT-TYPE
  SYNTAX INTEGER (0..1)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "This variable is 1 if the control block is in use."
  ::= { wcbsEntry 2 }

wcbstHlState        OBJECT-TYPE
  SYNTAX INTEGER (0..9999)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The control block state. 0=closed, 1=busy, 2=connect,
    3=listen, 4=unitdata, 5=unitdata listen, 6=open, 7=send,
    8=closing, 9=disconnect."
  ::= { wcbsEntry 3 }

wcbstDsap           OBJECT-TYPE
  SYNTAX INTEGER (0..255)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Destination service access point."
  ::= { wcbsEntry 4 }

```

```

wcbstSsap                OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Source service access point."
    ::= { wcbstEntry 5 }

wcbstNet16Addr           OBJECT-TYPE
    SYNTAX INTEGER (2048..9192)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The 16-bit network address of the remote node which is
        active on the connection, if the control block is in an
        active state."
    ::= { wcbstEntry 6 }

wcbstT1Value             OBJECT-TYPE
    SYNTAX INTEGER (0..999)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The current response time-out value in tenths of seconds."
    ::= { wcbstEntry 7 }

wcbstT1Average           OBJECT-TYPE
    SYNTAX INTEGER (0..999)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The average response time-out value in tenths of seconds."
    ::= { wcbstEntry 8 }

wcbstT1Deviation         OBJECT-TYPE
    SYNTAX INTEGER (0..999)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The average deviation from the T1Average values in tenths
        of seconds."
    ::= { wcbstEntry 9 }

wcbstT1Timeouts          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Number of T1 timeouts. A T1 timeout occurs when an
        expected response is lost."
    ::= { wcbstEntry 10 }

wcbstT2Timeouts          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Number of T2 timeouts. A T2 timeout occurs if the higher
        layer does not have data ready to send before a WTP
        supervisory response is required."
    ::= { wcbstEntry 11 }

```

```

wcbstTiTimeouts          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Number of Ti timeouts. A Ti timeout occurs when a
        connection is idle for an INACTIVE TIMEOUT period."
    ::= { wcbEntry 12 }

wcbstFramesRecv          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP frames received."
    ::= { wcbEntry 13 }

wcbstIframesRecv        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames received."
    ::= { wcbEntry 14 }

wcbstIframesDiscarded    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames which were discarded as
        duplicates."
    ::= { wcbEntry 15 }

wcbstDataBytesRecv      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of data bytes received."
    ::= { wcbEntry 16 }

wcbstFramesXmit          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP frames transmitted."
    ::= { wcbEntry 17 }

wcbstIframesXmit        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames transmitted."
    ::= { wcbEntry 18 }

wcbstDataBytesXmit      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames transmitted."
    ::= { wcbEntry 19 }

```

```

wcbstIframesRetransmitted OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of data frames which were lost and had to be
        retransmitted."
    ::= { wcbsEntry 20 }

wcbstState OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The internal software state of the LLC control block."
    ::= { wcbsEntry 21 }

nApplication OBJECT IDENTIFIER ::= { norandNet 104 }
gwSession OBJECT IDENTIFIER ::= { nApplication 1 }

    gs OBJECT IDENTIFIER ::= { gwSession 1 }

        gsReset OBJECT-TYPE
            SYNTAX INTEGER { true(1), false(2) }
            ACCESS read-write
            STATUS mandatory
            DESCRIPTION
                "Reset variable for the gs group."
            ::= { gs 1 }

        gsLastReset OBJECT-TYPE
            SYNTAX TimeTicks
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                "The time elapsed since the last reset of gs group."
            ::= { gs 2 }

        -- The Gateway Statistics group

        gsClockTicks OBJECT-TYPE
            SYNTAX TimeTicks
            ACCESS read-only --read-reset
            STATUS mandatory
            DESCRIPTION
                "Number of .01 second timer ticks since the timer was
                reset."
            ::= { gs 3 }

        gsNetworkErrors OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only --read-reset
            STATUS mandatory
            DESCRIPTION
                "Number of times the network has been lost."
            ::= { gs 4 }

        gsHostDataCount OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only --read-reset
            STATUS mandatory
            DESCRIPTION
                "Number of data blocks received from the host
                computer."
            ::= { gs 5 }

```

```

gsHostHaltCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of HALT commands received from the host
        task."
    ::= { gs 6 }

gsHostInvalidCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of invalid messages received from the host
        task."
    ::= { gs 7 }

gsHostDiscardCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of blocks from the host computer which were
        discarded."
    ::= { gs 8 }

gsConnectCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of transport connections."
    ::= { gs 9 }

gsCloseCount        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times a transport connection was closed."
    ::= { gs 10 }

gsBlocksXmit        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of blocks passed to the network."
    ::= { gs 11 }

gsDataXmit          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks passed to the network."
    ::= { gs 12 }

gsBlocksRecv        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of blocks received from the network."
    ::= { gs 13 }

```

```
gsDataRecv          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks received from the network."
    ::= { gs 14 }

gsTermResetCount    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times a session was reset by a terminal."
    ::= { gs 15 }

gsTermContinueCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times a session was continued after a
        transport connection was lost."
    ::= { gs 16 }

gsTermInvalidCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of invalid blocks received from terminals."
    ::= { gs 17 }

gsConnectionErrors  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of transport connection errors."
    ::= { gs 18 }

gsConnectionResets  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times an active transport connection was
        reset."
    ::= { gs 19 }

gsRetransmissions   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks which were retransmitted."
    ::= { gs 20 }

gsRecvSeqErrors     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks which were discarded due to
        receive sequence errors."
    ::= { gs 21 }
```

```
gsXmitErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of transport transmission errors."
    ::= { gs 22 }

gsTermDiscardCount    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of discarded blocks received from terminals."
    ::= { gs 23 }

gsHostDelayMax        OBJECT-TYPE
    SYNTAX INTEGER (0..9999)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The maximum host delay time. Host delay is the time
        a message is received from a terminal until the host
        responds. Times less than the maximum host delay
        time are counted as transactions times. Host delay
        time is measured in hundredths of seconds."
    ::= { gs 24 }

gsHostDelayTotal      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total host delay time in hundredths of seconds,
        not including delay times which exceeded the maximum
        host delay."
    ::= { gs 25 }

gsHostTransactions    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of times the host responded to a
        terminal message in a time less than the maximum host
        delay time."
    ::= { gs 26 }

gsRecvErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of transport layer receive errors."
    ::= { gs 27 }
```

```
gsTermDelayMax          OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The maximum terminal transaction time in hundredths
        of seconds. Round-trip times which exceed the
        maximum delay are not included as terminal
        transactions."
    ::= { gs 28 }

gsTermDelayThreshold OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Terminal transactions which take longer than the
        value set in gsTermDelayThreshold are counted in
        gsTermThresholdCount. This threshold is specified
        in 1/100th seconds."
    ::= { gs 29 }

gsTermDelayTotal        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total time for all terminal transactions."
    ::= { gs 30 }

gsTermTransactions     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of terminal transactions. Terminal
        transaction information is sent to the controller
        from terminals."
    ::= { gs 31 }

gsTermThresholdCount   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of terminal transactions which exceed the
        threshold value set in gsTermDelayThreshold."
    ::= { gs 32 }

gsTermMaxCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of terminal transactions which exceed the
        maximum transaction time set in gsTermDelayMax."
    ::= { gs 33 }
```



```

gsHostMaxCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of transactions which exceed the maximum
        host delay time set in gsHostDelayMax."
    ::= { gs 34 }

gsTermDelayTraceOn     OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Terminal transaction delay diagnostics are enabled
        by setting gsTermDelayTraceOn to a non-zero value.
        The results of this trace are found in the tdTable."
    ::= { gs 35 }

gsHostActive           OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Indicates whether the host is active and
        communicating with the controller.  A value of 1 =
        active, a value of 0 = inactive.  The default is 0.
        Once the host has communicated with the controller,
        this value will always show 'active' (1) UNLESS the
        cpctPortInactTimeout value in the cpcTable is set to
        a non-zero value.  If the controller has not received
        a response from the host in the time (in seconds)
        specified by cpctPortInactTimeout, gsHostActive will
        get set to 0."
    ::= { gs 36 }

rcbs OBJECT IDENTIFIER ::= { gwSession 2 }

-- The RCBS Table

rcbsTableReset         OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the rcbsTable table."
    ::= { rcbs 2 }

rcbsTableLastReset     OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time elapsed since the last reset of rcbsTable
        table."
    ::= { rcbs 3 }

-- Table Definition

rcbsTable              OBJECT-TYPE
    SYNTAX SEQUENCE OF RcbsEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "SESSION control block status table.  Each entry
        corresponds to a terminal session."
    ::= { rcbs 4 }

```

```

-- Row Definition

rcbsEntry          OBJECT-TYPE
    SYNTAX RcbEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { rcbsIndex }
    ::= { rcbsTable 1 }

-- Columnar Object Definition

RcbEntry ::=
    SEQUENCE {
        rcbsIndex          INTEGER,
        rcbsTerminal       INTEGER,
        rcbsType           INTEGER,
        rcbsState          INTEGER,
        rcbsLLCIndex       INTEGER,
        rcbsHostDataCount  Counter,
        rcbsBlocksXmit     Counter,
        rcbsDataXmit       Counter,
        rcbsBlocksRecv     Counter,
        rcbsDataRecv       Counter,
        rcbsTermResetCount Counter,
        rcbsTermContinueCount Counter,
        rcbsCloseCount     Counter,
        rcbsHostTransactions Counter,
        rcbsHostDelayTotal Counter,
        rcbsHostDelayLast  INTEGER,
        rcbsTermResetTime  INTEGER,
        rcbsTermInactTime  INTEGER,
        rcbsHostInactTime  INTEGER,
        rcbsConnectionErrors Counter,
        rcbsNetAddress     PhysAddress,
        rcbsSeqErrors      Counter,
        rcbsTermTransactions Counter,
        rcbsTermDelayTotal Counter,
        rcbsTermDelayLast  INTEGER,
        rcbsTermDelayLongest INTEGER,
        rcbsTermThresholdCount Counter,
        rcbsTermMaxCount   Counter,
        rcbsTermDelayTraceOn INTEGER,
        rcbsHostAlias      DisplayString
    }

-- Leaf Definition

rcbsIndex          OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Non-zero table index."
    ::= { rcbsEntry 1 }

rcbsTerminal       OBJECT-TYPE
    SYNTAX INTEGER (0..127)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Terminal session identifier"
    ::= { rcbsEntry 2 }

```

```

rcbsType          OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The terminal power up type as passed by the terminal
        at the beginning of an RTC session. (Unused until
        V4.16 of terminal software). Values: 11, 21, 46, 47,
        61 indicate 3270 emulation; 62 = 5250 emulation; 63 =
        VT220; Any other value indicates Native mode."
    ::= { rcbsEntry 3 }

rcbsState          OBJECT-TYPE
    SYNTAX INTEGER (0..99)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Session state. 0=reset state, 1=active state."
    ::= { rcbsEntry 4 }

rcbsLLCIndex      OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Transport connection control block identifier.
        Corresponds to an lcbstIndex value."
    ::= { rcbsEntry 5 }

rcbsHostDataCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks received from the host."
    ::= { rcbsEntry 6 }

rcbsBlocksXmit    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Total number of blocks passed to the network."
    ::= { rcbsEntry 7 }

rcbsDataXmit      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks passed to the network."
    ::= { rcbsEntry 8 }

rcbsBlocksRecv    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Total number of blocks received from the terminal."
    ::= { rcbsEntry 9 }

```

```

rcbsDataRecv          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks received from the terminal."
    ::= { rcbsEntry 10 }

rcbsTermResetCount    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times the terminal has reset the session."
    ::= { rcbsEntry 11 }

rcbsTermContinueCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times a terminal has continued a session
        after the transport connection was lost and
        re-established."
    ::= { rcbsEntry 12 }

rcbsCloseCount        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times an active transport connection used
        by the terminal session was closed."
    ::= { rcbsEntry 13 }

rcbsHostTransactions  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of times the host responded to a
        message from the terminal in a time less than the
        maximum host delay time."
    ::= { rcbsEntry 14 }

rcbsHostDelayTotal    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total host delay for the terminal in hundredths
        of seconds, not including times which exceeded the
        maximum host delay time."
    ::= { rcbsEntry 15 }

rcbsHostDelayLast     OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The host response time, in hundredths of seconds,
        for the last message sent from the terminal."
    ::= { rcbsEntry 16 }

```

```
rcbsTermResetTime      OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time, in seconds, since the terminal reset its
        session."
    ::= { rcbsEntry 17 }

rcbsTermInactTime      OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time, in seconds, since a message was received
        from the terminal."
    ::= { rcbsEntry 18 }

rcbsHostInactTime      OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time, in seconds, since a message was received
        from the host for the terminal."
    ::= { rcbsEntry 19 }

rcbsConnectionErrors  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of times the terminal session lost a
        transport connection."
    ::= { rcbsEntry 20 }

rcbsNetAddress         OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The network address of the remote terminal defined
        by rcbsTerminal."
    ::= { rcbsEntry 21 }

rcbsSeqErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of messages discarded due to receive
        sequence errors."
    ::= { rcbsEntry 22 }

rcbsTermTransactions  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of terminal transactions for the
        terminal defined by rcbsTerminal."
    ::= { rcbsEntry 23 }
```

```
rcbsTermDelayTotal      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total time for all transactions for the terminal
        defined by rcbsTerminal."
    ::= { rcbsEntry 24 }

rcbsTermDelayLast       OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time for the last transaction for the terminal
        defined by rcbsTerminal, in 1/100th seconds."
    ::= { rcbsEntry 25 }

rcbsTermDelayLongest    OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time for the longest transaction for the
        terminal defined by rcbsTerminal, in 1/100th
        seconds."
    ::= { rcbsEntry 26 }

rcbsTermThresholdCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of transactions which exceeded the
        value specified by gsTermDelayThreshold."
    ::= { rcbsEntry 27 }

rcbsTermMaxCount        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of transactions which exceeded the
        value specified by gsTermDelayMax."
    ::= { rcbsEntry 28 }

rcbsTermDelayTraceOn    OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Terminal delay tracing is enabled for the terminal
        defined by rcbsTerminal if rcbsTermDelayTraceOn is
        set to a non-zero value. The results of the trace
        will be found in the tdTable. For this to work,
        global terminal delay trace must be turned off - to
        do this set gsTermDelayTraceOn to 0 (this value is
        found in the gsTable)."
    ::= { rcbsEntry 29 }
```

```

rcbsHostAlias          OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..16))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The alias of the host computer associated with the
        RCB."
    ::= { rcbsEntry 30 }

hd          OBJECT IDENTIFIER ::= { gwSession 3 }

-- The HD Table

hdTableReset          OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the host delay table."
    ::= { hd 2 }

hdTableLastReset      OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time elapsed since the last reset of the host
        delay table."
    ::= { hd 3 }

-- Table Definition

hdTable              OBJECT-TYPE
    SYNTAX SEQUENCE OF HdEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Host delay table.  Host delay is defined as the time
        from when a terminal message is received until the
        host replys.  Each entry represents a bucket in the
        host delay table."
    ::= { hd 4 }

-- Row Definition

hdEntry              OBJECT-TYPE
    SYNTAX HdEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { hdBucket }
    ::= { hdTable 1 }

-- Columnar Object Definition

HdEntry ::=
    SEQUENCE {
        hdHostDelay  INTEGER,
        hdCount      Counter,
        hdBucket     INTEGER
    }

-- Leaf Definition

```

```

hdHostDelay          OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The host delay time.  The value in each row defines
        a host delay bucket bounded by the value in the
        preceding row (or 0) and the value in the row.  Delay
        times are defined in 1/100th seconds."
    ::= { hdEntry 1 }

hdCount              OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of times the host responded to a
        message from the terminal in a time bounded by the
        associated host delay time."
    ::= { hdEntry 2 }

hdBucket             OBJECT-TYPE
    SYNTAX INTEGER (1..20)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The bucket number used to identify a bucket in the
        host delay table (hdTable)."
    ::= { hdEntry 3 }

td      OBJECT IDENTIFIER ::= { gwSession 4 }

-- The TD Table

tdTableReset        OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the tdTable"
    ::= { td 2 }

tdTableLastReset    OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time elapsed since last reset of tdTable."
    ::= { td 3 }

-- Table Definition

tdTable             OBJECT-TYPE
    SYNTAX SEQUENCE OF TdEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Terminal delay table.  Terminal delay is the time
        used to send a message from a terminal to a host plus
        the time used to send a response.  Host processing
        and queueing time is not included.  Each entry in the
        table contains a response time bucket and an
        associated count.  Set gsTermDelayTraceOn=1 to enable
        for all terminals.  Set rcbsTermDelayTraceOn=1 and
        gsTermDelayTraceOn=0 to enable for single terminal."
    ::= { td 4 }

```



```

-- Row Definition

tdEntry                OBJECT-TYPE
    SYNTAX TdEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { tdBucket }
    ::= { tdTable 1 }

-- Columnar Object Definition

TdEntry ::=
    SEQUENCE {
        tdTrxnTime    INTEGER,
        tdCount        Counter,
        tdBucket       INTEGER
    }

-- Leaf Definition

tdTrxnTime             OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The round trip transaction time.  The value in each
        row defines a terminal delay bucket bounded by the
        value in the preceeding row (or 0) and the value in
        the row.  Delay times are defined in 1/100th
        seconds."
    ::= { tdEntry 1 }

tdCount                OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of times the terminal delay fell
        within the time interval bounded by the associated
        tdTrxnTime time."
    ::= { tdEntry 2 }

tdBucket               OBJECT-TYPE
    SYNTAX INTEGER (1..20)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The bucket number used to identify a bucket in the
        terminal delay table (tdTable)."
    ::= { tdEntry 3 }

ahost                  OBJECT IDENTIFIER ::= { nApplication 2 }

ahstReset              OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the ahost group."
    ::= { ahost 1 }

ahstLastReset          OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time elapsed since the last reset of ahost group."
    ::= { ahost 2 }

```

```
ahstFramesOut      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of frames from the gateway sent to the host."
    ::= { ahost 3 }

ahstFramesIn      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of frames received from the host."
    ::= { ahost 4 }

ahstIdleDetected  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times that the host went inactive. The
        time out is set in the gateway from the user interface."
    ::= { ahost 5 }

ahstNoErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of host interface errors detected by the
        gateway."
    ::= { ahost 6 }

ahstSyntaxErrors  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of invalid commands received. Incremented
        whenever a ?1 error is sent to the host."
    ::= { ahost 7 }

ahstLengthErrors  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of host frames that are too long. Increments
        whenever a ?4 is issued by the gateway."
    ::= { ahost 8 }

ahstDisabledErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of commands sent to disabled terminals.
        Increments whenever the gateway issues a ?5 response."
    ::= { ahost 9 }
```

```

ahstGapErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times the Gap time out is exceeded. The time
        out value is set with the CMT5 command and is measured
        by the start of a command and the receipt of a carriage
        return <CR>. Increments whenever the gateway issues a
        ?6 response."
    ::= { ahost 10 }

ahstCheckErrors        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times the host command fails the LRC or
        CRC-16 check. LRC/CRC error checking is enabled by the
        CMT8 command. Increments whenever the controller sends a
        ?8 response."
    ::= { ahost 11 }

ahstNoBuffersErrors    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times the gateway cannot receive a command
        from the host because of the lack of local buffers.
        Increments when the gateway sends a ?10 response."
    ::= { ahost 12 }

ahstSequenceErrors    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times the host sends an R command before the
        host initialized the gateway with CMT commands.
        Increments when the gateway sends a ?11 response."
    ::= { ahost 13 }

-- ahstOtherErrors { ahost 14 } is obsolete and has been
-- deleted.

ahstParityErrors        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of async parity errors."
    ::= { ahost 15 }

ahstFrameErrors        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of async character framing errors."
    ::= { ahost 16 }

```

```
ahstNoiseErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of UART signal samples with ambiguous results."
    ::= { ahost 17 }

ahstBreakErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of async break characters received."
    ::= { ahost 18 }

nControl              OBJECT IDENTIFIER ::= { norandNet 105 }

    powerUp            OBJECT IDENTIFIER ::= { nControl 1 }

pwrPowerUpCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Power-up count"
    ::= { powerUp 1 }

pwrNextPowerUpTime   OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Next power-up time (Used to reboot the device)."
    ::= { powerUp 2 }
```

END

Norand Open Wireless LAN MIB 6910 Integrated Gateway/Access Point



This section describes various groups Intermec Technologies Corporation supports for the 6910 Integrated Gateway/Access Point (IGAP). Table 6-1 lists groups, their meaning, and page numbers where each group's table summary and each group's actual definitions appear in this section.

Table 6-1
IGAP MIB Directory

Group	Meaning	Group Summary	MIB Definition
Product OIDs			
products	Intermec [®] Products	6-2	6-21
System Information			
hw	Hardware Information	6-3	6-21
fsinfo	Filesystem Information	6-3	6-22
segment	Filesegment Information	6-4	6-23
dir	Software Directory Listing	6-4	6-24
criticalErrors	Critical Errors Information	6-4	6-26
Interface Information			
nifx	Norand Extensions to Interfaces Table	6-5	6-27
portState	Port State Information	6-6	6-30
portStats	Port Statistics	6-6	6-34
ptxq	Port Transmit Queue	6-7	6-37
pmsg	Pending Message Services	6-8	6-40
SNMP Version 1 Configuration			
community	Community Table	6-9	6-42
trapTarget	Trap Target Table	6-9	6-44
Bridging Parameters			
rt	Route Table	6-10	6-45
brg	Bridge Table	6-11	6-48
addr	Address Table	6-11	6-49
brgState	Bridge State Information	6-11	6-51
bridgeStats	Bridge Statistics	6-12	6-54

Table 6-1 (Continued)
IGAP MIB Directory

Group	Meaning	Group Summary	MIB Definition
Transport Groups			
wst	Wireless Transport Protocol (WTP) Status Table	6-13	6-56
hlit	High Level Interface Table	6-14	6-61
wcbst	WTP Control Block Status Table	6-14	6-62
Application Layer Groups			
gs	Gateway Statistics	6-16	6-66
rcbs	Radio Terminal Control Block Statistics	6-17	6-71
hd	Host Delay Statistics	6-18	6-77
td	Terminal Transaction Delay Statistics	6-18	6-78
ahost	Asynchronous Host application	6-19	6-79
Control Groups			
powerUp	Power Up Objects	6-19	6-82
softwareDownload	Software Download	6-20	6-82

Integrated Gateway/Access Point MIB Outline

This outline summarizes the various MIB groups Intermecc Technologies Corporation supports for the 6910 Integrated Gateway/Access Point.

Product OIDs

This group contains an Object Identification (OID) for each Intermecc device.

Table 6-2
products GROUP

Device Products
norand.manage.products.x
(1.3.6.1.4.1.469.1000.1.x)

Object ID	Object Name	Object Type	Access
1	ap6710	OBJECT ID	Not Applicable (N/A)
2	gw4030	OBJECT ID	(N/A)
3	wnas	OBJECT ID	(N/A)
4	ts6950	OBJECT ID	(N/A)
5	gwap6910	OBJECT ID	(N/A)

System Information

The following groups contain system level objects describing hardware and file-system configuration properties. The groups also contain information about critical errors.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- hw Hardware Information (page 6-21)
- fsinfo Filesystem Information (page 6-22)
- segment Filesegment Information (page 6-23)
- dir Software Directory Listing (page 6-24)
- criticalErrors Critical Errors Information (page 6-26)

Table 6-3
hw GROUP

Device Hardware Information
norand.manage.norandNet.nSystem.hw.x
(1.3.6.1.4.1.469.1000.2.1.1.x)

Object ID	Object Name	Object Type	Access
1	hwPartNo	INTEGER	read
2	hwDescription	DisplayString	read
3	hwRevision	INTEGER	read
4	hwSerialNo	INTEGER	read
5	hwID	INTEGER	read

Table 6-4
fsinfo GROUP

Device Filesystem Information
norand.manage.norandNet.nSystem.file.fsinfo.x
(1.3.6.1.4.1.469.1000.2.1.3.1.x)

Object ID	Object Name	Object Type	Access
1	fsEnabled	INTEGER	read
2	fsMaxSectors	INTEGER	read
3	fsSectorSize	INTEGER	read
4	fsNumSegments	INTEGER	read
5	fsNumFiles	Gauge	read
6	fsBootSegment	INTEGER	read
7	fsDataSegment	INTEGER	read

Table 6-5
segment GROUP

Device Filesegment Information
norand.manage.norandNet.nSystem.file.segment.x
(1.3.6.1.4.1.469.1000.2.1.3.2.x)

Object ID	Object Name	Object Type	Access
2.1.1	segID	INTEGER	read
2.1.2	segFirstSector	INTEGER	read
2.1.3	segLastSector	INTEGER	read
2.1.4	segStatus	INTEGER	read
2.1.5	segSize	INTEGER	read
2.1.6	segFree	INTEGER	read

Table 6-6
dir GROUP

Device Software Directory Listing
norand.manage.norandNet.nSystem.file.dir.x
(1.3.6.1.4.1.469.1000.2.1.3.3.x)

Object ID	Object Name	Object Type	Access
2.1.1	dirIndex	INTEGER	read
2.1.2	dirName	DisplayString	read
2.1.3	dirSegment	INTEGER	read
2.1.4	dirType	INTEGER	read
2.1.5	dirSize	INTEGER	read
2.1.6	dirDate	DisplayString	read
2.1.7	dirTime	DisplayString	read
2.1.8	dirVersion	DisplayString	read

Table 6-7
criticalErrors GROUP

Device Critical Errors Information
norand.manage.norandNet.nSystem.sysErrors.criticalErrors.x
(1.3.6.1.4.1.469.1000.2.1.4.1.x)

Object ID	Object Name	Object Type	Access
1	ceEnabled	INTEGER	read
2	ceOverflow	INTEGER	read
3	ceReset	INTEGER	write
4.1.1	ceLogErrorCode	INTEGER	read
4.1.2	ceLogErrorCount	Counter	read

Interface Information

The following groups relate information about NORAND® Interfaces, port state, port statistics, port transmit queue, and pending message services.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- nifx NORAND Extensions to Interfaces Table (page 6-27)
- portState Port State Information (page 6-30)
- portStats Port Statistics (page 6-34)
- ptxq Port Transmit Queue (page 6-37)
- pmsg Pending Message Services (page 6-40)

Table 6-8
nifx GROUP

NORAND Extensions to MIB-II Interfaces Table
norand.manage.norandNet.nInterfaces.nifx.x
(1.3.6.1.4.1.469.1000.2.2.2.x)

Object ID	Object Name	Object Type	Access
4.1.1	nifxIndex	INTEGER	read
4.1.2	nifxType	INTEGER	read
4.1.3	nifxInDisabledDiscards	Counter	read
4.1.4	nifxInOverruns	Counter	read
4.1.5	nifxInHWOverruns	Counter	read
4.1.6	nifxInUcastDPkts	Counter	read
4.1.7	nifxInNUcastDPkts	Counter	read
4.1.8	nifxInLenErrors	Counter	read
4.1.9	nifxExcessiveDeferrals	Counter	read
4.1.10	nifxInNetIDDiscards	Counter	read
4.1.11	nifxInFragDiscards	Counter	read
4.1.12	nifxInUFilterDiscards	Counter	read
4.1.13	nifxInNUFilterDiscards	Counter	read
4.1.14	nifxInQFullDiscards	Counter	read

Table 6-9
portState GROUP

Device Port State Information
norand.manage.norandNet.nInterfaces.portState.x
(1.3.6.1.4.1.469.1000.2.2.3.x)

Object ID	Object Name	Object Type	Access
4.1.1	psPort	INTEGER	read
4.1.2	psIfIndex	INTEGER	read
4.1.3	psAddress	PhysAddress	read
4.1.4	psType	INTEGER	read
4.1.5	psState	INTEGER	read
4.1.6	psCost	INTEGER	read
4.1.7	psHelloPeriod	INTEGER	read
4.1.8	psHelloCount	Counter	read
4.1.9	psMacdWindow	INTEGER	read
4.1.10	psMacdQSize	Gauge	read
4.1.11	psMacdTimeouts	Counter	read
4.1.12	psIsPrimary	INTEGER	read
4.1.13	psIsSecondary	INTEGER	read
4.1.14	psIsSecondaryCandidate	INTEGER	read
4.1.15	psSecondaryUniFlooding	INTEGER	read
4.1.16	psSecondaryMultiFlooding	INTEGER	read
4.1.17	psIsRadio	INTEGER	read
4.1.18	psPendEnabled	INTEGER	read

Table 6-10
portStats GROUP

Device Port Statistics
norand.manage.norandNet.nInterfaces.portStats.x
(1.3.6.1.4.1.469.1000.2.2.4.x)

Object ID	Object Name	Object Type	Access
4.1.1	pstcPort	INTEGER	read
4.1.2	pstcInOWLPkts	Counter	read
4.1.3	pstcInUcastOWLDataPkts	Counter	read
4.1.4	pstcInNUcastOWLDataPkts	Counter	read
4.1.5	pstcInOWLErrors	Counter	read
4.1.6	pstcOutOWLPkts	Counter	read
4.1.7	pstcOutUcastOWLDataPkts	Counter	read
4.1.8	pstcOutNUcastOWLDataPkts	Counter	read
4.1.9	pstcOutOWLErrors	Counter	read
4.1.10	pstcParentLinkErrors	Counter	read

Table 6-10 (Continued)
portStats GROUP

Device Port Statistics
norand.manage.norandNet.nInterfaces.portStats.x
(1.3.6.1.4.1.469.1000.2.2.4.x)

Object ID	Object Name	Object Type	Access
4.1.11	pstcAlertLinkErrors	Counter	read
4.1.12	pstcInUcastRelayPkts	Counter	read
4.1.13	pstcInNUcastRelayPkts	Counter	read
4.1.14	pstcOutUcastRelayPkts	Counter	read
4.1.15	pstcOutNUcastRelayPkts	Counter	read
4.1.16	pstcInUcastInbound	Counter	read
4.1.17	pstcInUcastOutbound	Counter	read
4.1.18	pstcInUcastSec	Counter	read
4.1.19	pstcInUcastFlood	Counter	read
4.1.20	pstcUcastDiscards	Counter	read
4.1.21	pstcInNUcastDiscards	Counter	read
4.1.22	pstcInUcastToIFC	Counter	read
4.1.23	pstcInNUcastToIFC	Counter	read
4.1.24	pstcOutDelayDiscards	Counter	read

Table 6-11
ptxq GROUP

Device Port Transmit Queue
norand.manage.norandNet.nInterfaces.ptxq.x
(1.3.6.1.4.1.469.1000.2.2.5.x)

Object ID	Object Name	Object Type	Access
1.1.1	ptxqPort	INTEGER	read
1.1.2	ptxqRegQSize	Gauge	read
1.1.3	ptxqRegQMax	INTEGER	read
1.1.4	ptxqExpQSize	Gauge	read
1.1.5	ptxqExpQMax	INTEGER	read
1.1.6	ptxqQHpCount	Counter	read
1.1.7	ptxqQExpCount	Counter	read
1.1.8	ptxqQRegCount	Counter	read
1.1.9	ptxqQHpDiscards	Counter	read
1.1.10	ptxqQExpDiscards	Counter	read
1.1.11	ptxqQRegDiscards	Counter	read
1.1.12	ptxqMultiQSize	Gauge	read
1.1.13	ptxqMultiQMax	INTEGER	read
1.1.14	ptxqMultiQDiscards	Counter	read

Table 6-12
pmsg GROUP

Device Pending Message Services
 norand.manage.norandNet.nInterfaces.pmsg.x
 (1.3.6.1.4.1.469.1000.2.2.6.x)

Object ID	Object Name	Object Type	Access
1.1.1	pmsgPort	INTEGER	read
1.1.2	pmsgPendRecCurrent	Gauge	read
1.1.3	pmsgPendRecMax	INTEGER	read
1.1.4	pmsgPendMsgCurrent	Gauge	read
1.1.5	pmsgPendMsgMax	INTEGER	read
1.1.6	pmsgPendMsgTotal	Counter	read
1.1.7	pmsgPendMsgDiscards	Counter	read
1.1.8	pmsgPendRecOverflowErrors	Counter	read
1.1.9	pmsgPendMsgOverflowErrors	Counter	read
1.1.10	pmsgPendAgedRecCount	Counter	read
1.1.11	pmsgPendAgedMsgCount	Counter	read

SNMP Version 1 Configuration Group

This group contains objects that configure the version 1 Simple Network Management Protocol (SNMP) agent.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- community Community Table (page 6-42)
- trapTarget Trap Target Table (page 6-44)

Table 6-13

community TABLE

Device SNMP v1 Configurations
norand.manage.norandNet.nSNMP.v1Config.x
(1.3.6.1.4.1.469.1000.2.11.1.x)

Object ID	Object Name	Object Type	Access
2.1.1	communityIndex	INTEGER	read
2.1.2	communityStatus	INTEGER	write
2.1.3	communityName	DisplayString	write
2.1.4	communityPrivileges	INTEGER	write

Table 6-14

trapTarget TABLE

Device SNMP v1 Configurations
norand.manage.norandNet.nSNMP.v1Config.x
(1.3.6.1.4.1.469.1000.2.11.1.x)

Object ID	Object Name	Object Type	Access
3.1.1	trapTargetIndex	INTEGER	read
3.1.2	trapTargetStatus	INTEGER	write
3.1.3	trapTargetName	DisplayString	write
3.1.4	trapTargetIpAddress	IpAddress	write

Bridging Parameters

The following group contains objects relating to the wireless transparent bridging operation.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- rt Route Table (page 6-45)
- brg Bridge Table (page 6-48)
- addr Address Table (page 6-49)
- brgState Bridge State Information (page 6-51)
- bridgeStats Bridge Statistics (page 6-54)

Table 6-15
rt GROUP

Device Route Table
norand.manage.norandNet.nBridge.rt.x
(1.3.6.1.4.1.469.1000.2.17.2.x)

Object ID	Object Name	Object Type	Access
2.1.1	rtDestination	PhysAddress	read
2.1.2	rtPort	INTEGER	read
2.1.3	rtAge	INTEGER	read
2.1.4	rtNodeId	INTEGER	read
2.1.5	rtAttachId	INTEGER	read
2.1.6	rtAttachTime	TimeTicks	read
2.1.7	rtApEaddr	PhysAddress	read
2.1.8	rtHopAddrLen	INTEGER	read
2.1.9	rtHopAddr16	INTEGER	read
2.1.10	rtHopEaddr	PhysAddress	read
2.1.11	rtIsBound	INTEGER	read
2.1.12	rtIsRemote	INTEGER	read
2.1.13	rtIsChild	INTEGER	read
2.1.14	rtIsAp	INTEGER	read
2.1.15	rtIsDistributed	INTEGER	read
2.1.16	rtIsRemoteLan	INTEGER	read
2.1.17	rtNS	INTEGER	read
2.1.18	rtNR	INTEGER	read

Table 6-16
brg GROUP

Device Bridge Table
norand.manage.norandNet.nBridge.brg.x
(1.3.6.1.4.1.469.1000.2.17.3.x)

Object ID	Object Name	Object Type	Access
2.1.1	brgDestination	PhysAddress	read
2.1.2	brgPort	INTEGER	read
2.1.3	brgAge	INTEGER	read
2.1.4	brgType	INTEGER	read
2.1.5	brgIsPermanent	INTEGER	read
2.1.6	brgTimestamp	TimeTicks	read

Table 6-17
addr GROUP

Address Table
norand.manage.norandNet.nBridge.addr.x
(1.3.6.1.4.1.469.1000.2.17.4.x)

Object ID	Object Name	Object Type	Access
2.1.1	addrDestination	PhysAddress	read
2.1.2	addrAge	INTEGER	read
2.1.3	addrNodeId	INTEGER	read
2.1.4	addrAlias	DisplayString	read
2.1.5	addrDeviceId	INTEGER	read
2.1.6	addrIpAddress	IPAddress	read

Table 6-18
brgState GROUP

Bridge State Information
norand.manage.norandNet.nBridge.brgState.x
(1.3.6.1.4.1.469.1000.2.17.6.x)

Object ID	Object Name	Object Type	Access
3	bsAddress	PhysAddress	read
4	bsLanId	INTEGER	read
5	bsCostToRoot	INTEGER	read
6	bsIsRoot	INTEGER	read
7	bsIsAttached	INTEGER	read
8	bsAttachId	INTEGER	read
9	bsMyRootPriority	INTEGER	read
10	bsRootPort	INTEGER	read
11	bsDesignatedRootAddress	PhysAddress	read
12	bsDesignatedRootPriority	INTEGER	read
13	bsDesignatedRootSequence	INTEGER	read

Table 6-18 (Continued)
brgState GROUP

Bridge State Information
norand.manage.norandNet.nBridge.brgState.x
(1.3.6.1.4.1.469.1000.2.17.6.x)

Object ID	Object Name	Object Type	Access
14	bsParentAddress	PhysAddress	read
15	bsPortCount	INTEGER	read
16	bsNodeId	INTEGER	read
17	bsRootChangedCount	Counter	read
18	bsRootCount	Counter	read
19	bsAttachCount	Counter	read
20	bsDetachReason	INTEGER	read
21	bsNetworkTime	TimeTicks	read
22	bsUniFloodLevel	INTEGER	read
23	bsMultiFloodLevel	INTEGER	read
24	bsIsPrimaryBridge	INTEGER	read
25	bsIsSecondaryBridge	INTEGER	read
26	bsUniFilterExpr	INTEGER	read
27	bsMuliFilterExpr	INTEGER	read

Table 6-19
bridgeStats GROUP

Bridge Statistics
norand.manage.norandNet.nBridge.bridgeStats.x
(1.3.6.1.4.1.469.1000.2.17.7.x)

Object ID	Object Name	Object Type	Access
3	bstcRouteCount	Gauge	read
4	bstcChildCount	Gauge	read
5	bstcChildApCount	Gauge	read
6	bstcRemoteCount	Gauge	read
7	bstcPrimaryCount	Gauge	read
8	bstcInboundCount	Gauge	read
9	bstcSecondaryCount	Gauge	read
10	bstcRemoteLanCount	Gauge	read
11	bstcRouteGetErrors	Counter	read
12	bstcEntryGetErrors	Counter	read
13	bstcRmtLanGetErrors	Counter	read
14	bstcRouteSeqErrors	Counter	read
15	bstcDeleteSeqErrors	Counter	read
16	bstcEntrySeqErrors	Counter	read
17	bstcInvalidUpdateErrors	Counter	read

Transport Groups

These groups contain objects that relate active transport layers information.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- wst Wireless Transport Protocol (WTP) Status Table (page 6-56)
- hlit High Level Interface Table (page 6-61)
- wcbst WTP Control Block Status Table (page 6-62)

Table 6-20

wst GROUP

WTP Status Table
norand.manage.norandNet.nTransport.wst.x
(1.3.6.1.4.1.469.1000.2.102.2.x)

Object ID	Object Name	Object Type	Access
1	wstReset	INTEGER	read
2	wstLastReset	TimeTicks	read
3	wstClockTicks	TimeTicks	read
4	wstInternalErrorCount	Counter	read
5	wstSessionsLost	Counter	read
6	wstFrmrCount	Counter	read
7	wstDataBytesXmit	Counter	read
8	wstDataBytesRecv	Counter	read
9	wstIframesXmit	Counter	read
10	wstIframesRecv	Counter	read
11	wstIframesRetransmitted	Counter	read
12	wstChecksumErrors	Counter	read
13	wstHIDiscardCount	Counter	read
14	wstState	INTEGER	read
15	wstBrgBusy	INTEGER	read
16	wstT1Timeouts	Counter	read
17	wstT1Bucket1	Counter	read
18	wstT1Bucket2	Counter	read
19	wstT1Bucket3	Counter	read
20	wstT1Bucket4	Counter	read
21	wstSABMCount	Counter	read
22	wstDiscCount	Counter	read
23	wstDMCount	Counter	read
24	wstRNRCCount	Counter	read
25	wstRejectCount	Counter	read
26	wstSessionsReset	Counter	read
27	wstBrgSrvTime	TimeTicks	read

Table 6-20 (Continued)

wst GROUP

WTP Status Table
 norand.manage.norandNet.nTransport.wst.x
 (1.3.6.1.4.1.469.1000.2.102.2.x)

Object ID	Object Name	Object Type	Access
28	wstBrgSrvCount	Counter	read
29	wstBrgSrvThreshold	TimeTicks	read
30	wstBrgSrvThreshCount	Counter	read
31	wstBrgSrvLongest	TimeTicks	read
32	wstBrgTxErrors	Counter	read
33	wstFatalBrgErrors	Counter	read

Table 6-21

hlit GROUP

High Level Interface Table
 norand.manage.norandNet.nTransport.hlit.x
 (1.3.6.1.4.1.469.1000.2.102.3.x)

Object ID	Object Name	Object Type	Access
1	hlitReset	INTEGER	write
2	hlitLastReset	TimeTicks	read
3	hlitTime	TimeTicks	read
4	hlitCount	Counter	read
5	hlitThreshold	TimeTicks	write
6	hlitThreshCount	Counter	read
7	hlitLongest	Counter	read

Table 6-22

wcbst GROUP

WTP Control Block Status Table
 norand.manage.norandNet.nTransport.wcbst.x
 (1.3.6.1.4.1.469.1000.2.102.4.x)

Object ID	Object Name	Object Type	Access
1	wcbstReset	INTEGER	write
2	wcbstLastReset	TimeTicks	read
3.1.1	wcbstIndex	INTEGER	read
3.1.2	wcbstInUse	INTEGER	read
3.1.3	wcbstH1State	INTEGER	read
3.1.4	wcbstDsap	INTEGER	read
3.1.5	wcbstSsap	INTEGER	read
3.1.6	wcbstNet16ADDR	INTEGER	read
3.1.7	wcbstT1Value	INTEGER	read
3.1.8	wcbstT1Average	INTEGER	read

Table 6-22 (Continued)

wcbst GROUP

WTP Control Block Status Table
 norand.manage.norandNet.nTransport.wcbst.x
 (1.3.6.1.4.1.469.1000.2.102.4.x)

Object ID	Object Name	Object Type	Access
3.1.9	wcbstT1Deviation	INTEGER	read
3.1.10	wcbstT1Timeouts	Counter	write
3.1.11	wcbstT2Timeouts	Counter	write
3.1.12	wcbstTiTimeouts	Counter	write
3.1.13	wcbstFramesRecv	Counter	write
3.1.14	wcbstIframesRecv	Counter	write
3.1.15	wcbstIframesDiscarded	Counter	write
3.1.16	wcbstDataBytesRecv	Counter	write
3.1.17	wcbstFramesXmit	Counter	write
3.1.18	wcbstIframesXmit	Counter	write
3.1.19	wcbstDataBytesXmit	Counter	write
3.1.20	wcbstIframesRetransmitted	Counter	write
3.1.21	wcbstState	INTEGER	read

Application Layer Groups

The following groups relate resident applications information.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- Gateway application:
 - gs Gateway Statistics (page 6-66)
 - rcbs Radio Terminal Control Block Statistics (page 6-71)
 - hd Host Delay Statistics (page 6-77)
 - td Terminal Transaction Delay Statistics (page 6-78)
- Asynchronous host:
 - ahost Asynchronous Host application (page 6-79)

Table 6-23

gs GROUP

Gateway Statistics

norand.manage.norandNet.nApplication.gwSession.gs.x
(1.3.6.1.4.1.469.1000.2.104.1.1.x)

Object ID	Object Name	Object Type	Access
1	gsReset	INTEGER	write
2	gsLastReset	TimeTicks	read
3	gsClockTicks	TimeTicks	read
4	gsNetworkErrors	Counter	read
5	gsHostDataCount	Counter	read
6	gsHostHaltCount	Counter	read
7	gsHostInvalidCount	Counter	read
8	gsHostDiscardCount	Counter	read
9	gsConnectCount	Counter	read
10	gsCloseCount	Counter	read
11	gsBlocksXmit	Counter	read
12	gsDataXmit	Counter	read
13	gsBlocksRecv	Counter	read
14	gsDataRecv	Counter	read
15	gsTermResetCount	Counter	read
16	gsTermContinueCount	Counter	read
17	gsTermInvalidCount	Counter	read
18	gsConnectionErrors	Counter	read
19	gsConnectionResets	Counter	read
20	gsRetransmissions	Counter	read
21	gsRecvSeqErrors	Counter	read
22	gsXmitErrors	Counter	read
23	gsTermDiscardCount	Counter	read
24	gsHostDelayMax	INTEGER	write

Table 6-23 (Continued)
gs GROUP

Gateway Statistics
norand.manage.norandNet.nApplication.gwSession.gs.x
(1.3.6.1.4.1.469.1000.2.104.1.1.x)

Object ID	Object Name	Object Type	Access
25	gsHostDelayTotal	Counter	read
26	gsHostTransactions	Counter	read
27	gsRecvErrors	Counter	read
28	gsTermDelayMax	INTEGER	write
29	gsTermDelayThreshold	INTEGER	write
30	gsTermDelayTotal	Counter	read
31	gsTermTransactions	Counter	read
32	gsTermThresholdCount	Counter	read
33	gsTermMaxCount	Counter	read
34	gsHostMaxCount	Counter	read
35	gsTermDelayTraceOn	INTEGER	write
36	gsHostActive	INTEGER	read

Table 6-24
rcbs GROUP

Gateway (Terminal Emulation) Session Table
(each record is a terminal session)
norand.manage.norandNet.nApplication.gwSession.rcbs.x
(1.3.6.1.4.1.469.1000.2.104.1.2.x)

Object ID	Object Name	Object Type	Access
4.1.1	rcbsIndex	INTEGER	read
4.1.2	rcbsTerminal	INTEGER	read
4.1.3	rcbsType	INTEGER	read
4.1.4	rcbsState	INTEGER	read
4.1.5	rcbsLLCIndex	INTEGER	read
4.1.6	rcbsHostDataCount	Counter	read
4.1.7	rcbsBlocksXmit	Counter	read
4.1.8	rcbsDataXmit	Counter	read
4.1.9	rcbsBlocksRecv	Counter	read
4.1.10	rcbsDataRecv	Counter	read
4.1.11	rcbsTermResetCount	Counter	read
4.1.12	rcbsTermContinueCount	Counter	read
4.1.13	rcbsCloseCount	Counter	read
4.1.14	rcbsHostTransactions	Counter	read
4.1.15	rcbsHostDelayTotal	Counter	read
4.1.16	rcbsHostDelayLast	INTEGER	read

Table 6-24 (Continued)
rcbs GROUP

Gateway (Terminal Emulation) Session Table
(each record is a terminal session)
norand.manage.norandNet.nApplication.gwSession.rcbs.x
(1.3.6.1.4.1.469.1000.2.104.1.2.x)

Object ID	Object Name	Object Type	Access
4.1.17	rcbsTermResetTime	INTEGER	read
4.1.18	rcbsTermInactTime	INTEGER	read
4.1.19	rcbsHostInactTime	INTEGER	read
4.1.20	rcbsConnectionErrors	Counter	read
4.1.21	rcbsNetAddress	PhysAddress	read
4.1.22	rcbsSeqErrors	Counter	read
4.1.23	rcbsTermTransactions	Counter	read
4.1.24	rcbsTermDelayTotal	Counter	read
4.1.25	rcbsTermDelayLast	INTEGER	read
4.1.26	rcbsTermDelayLongest	INTEGER	read
4.1.27	rcbsTermThresholdCount	Counter	read
4.1.28	rcbsTermMaxCount	Counter	read
4.1.29	rcbsTermDelayTraceOn	INTEGER	write
4.1.30	rcbsHostAlias	DisplayS- tring	read

Table 6-25
hd GROUP

Host Delay Statistics
norand.manage.norandNet.nApplication.gwSession.hd.x
(1.3.6.1.4.1.469.1000.2.104.1.3.x)

Object ID	Object Name	Object Type	Access
4.1.1	hdHostDelay	INTEGER	write
4.1.2	hdCount	Counter	read
4.1.3	hdBucket	INTEGER	read

Table 6-26
td GROUP

Terminal Transaction Delay Statistics
norand.manage.norandNet.nApplication.gwSession.td.x
(1.3.6.1.4.1.469.1000.2.104.1.4.x)

Object ID	Object Name	Object Type	Access
4.1.1	tdTrxnTime	INTEGER	write
4.1.2	tdCount	Counter	read
4.1.3	tdBucket	INTEGER	read

Table 6-27
ahost GROUP

Asynchronous Host
norand.manage.norandNet.nApplication.ahost.x
(1.3.6.1.4.1.469.1000.2.104.2.x)

Object ID	Object Name	Object Type	Access
1	ahstReset	INTEGER	write
2	ahstLastReset	TimeTicks	read
3	ahstFramesOut	Counter	read
4	ahstFramesIn	Counter	read
5	ahstIdleDetected	Counter	read
6	ahstNoErrors	Counter	read
7	ahstSyntaxErrors	Counter	read
8	ahstLengthErrors	Counter	read
9	ahstDisabledErrors	Counter	read
10	ahstGapErrors	Counter	read
11	ahstCheckErrors	Counter	read
12	ahstNoBuffersErrors	Counter	read
13	ahstSequenceErrors	Counter	read
15	ahstParityErrors	Counter	read
16	ahstFrameErrors	Counter	read
17	ahstNoiseErrors	Counter	read
18	ahstBreakErrors	Counter	read

Control Groups

The objects in the following groups exert control over Intermec Access Points. Present functions include rebooting and scheduling software downloads.

► **NOTE:**

The MIB definition for each group starts on the page given below.

- powerUp Power Up Objects (page 6-82)
- softwareDownload Software Download (page 6-82)

Table 6-28
powerUp GROUP

Device Power Up Objects
norand.manage.norandNet.nControl.powerUp.x
(1.3.6.1.4.1.469.1000.2.105.1.x)

Object ID	Object Name	Object Type	Access
1	pwrPowerUpCount	Counter	read
2	pwrNextPowerUpTime	TimeTicks	write

Table 6-29
softwareDownload GROUP

Device Software Download
 norand.manage.norandNet.nControl.softwareDownload.x
 (1.3.6.1.4.1.469.1000.2.105.2.x)

Object ID	Object Name	Object Type	Access
1	sdStartTime	TimeTicks	write
2	sdServerIpAddress	IpAddress	write
3	sdScriptFilename	DisplayString	write
4	sdStatus	INTEGER	read
5	sdErrorString	DisplayString	read
6	sdCheckPoint	INTEGER	write
7	sdSetActivePointers	INTEGER	write
8	sdTerminate	INTEGER	write

Integrated Gateway/Access Point MIB Definitions

Following is the 6910 Integrated Gateway/Access Point MIB definition for the NORAND® Open Wireless LAN.

```
-- Norand Open Wireless LAN MIB - Access Point and Gateway
-- Version 1.15
-- Version Date: 8/23/96
-- This MIB contains objects supported by V1.15 of the 6710 Access Point and V1.14
-- of the RC4030 Gateway.
-- This is also the MIB for the 6910 IGAP - Access Point / Gateway.

-- Subject to Change

OWL DEFINITIONS ::= BEGIN

    IMPORTS
        enterprises, IpAddress, Counter, Gauge, TimeTicks
            FROM RFC1155-SMI
        PhysAddress, DisplayString
            FROM RFC1213-MIB
        OBJECT-TYPE
            FROM RFC-1212;

    -- This MIB module uses the extended OBJECT-TYPE macro as defined in
    -- RFC-1212;

norand
    manage
        products
            ap6710
            gw4030
            wnas
            ts6950
            gwap6910
        norandNET
        nSystem

        OBJECT IDENTIFIER ::= { enterprises 469 }
        OBJECT IDENTIFIER ::= { norand 1000 }
        OBJECT IDENTIFIER ::= { manage 1 }
        OBJECT IDENTIFIER ::= { products 1 }
        OBJECT IDENTIFIER ::= { products 2 }
        OBJECT IDENTIFIER ::= { products 3 }
        OBJECT IDENTIFIER ::= { products 4 }
        OBJECT IDENTIFIER ::= { products 5 }
        OBJECT IDENTIFIER ::= { manage 2 }
        OBJECT IDENTIFIER ::= { norandNET 1 }

        hw
            OBJECT IDENTIFIER ::= { nSystem 1 }

    -- The Hardware Parameters Group

    hwPartNo OBJECT-TYPE
        SYNTAX INTEGER (0..2147483647)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The Norand part number of the hardware device."
        ::= { hw 1 }

    hwDescription OBJECT-TYPE
        SYNTAX DisplayString (SIZE (0..40))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The description of the hardware device."
        ::= { hw 2 }

    hwRevision OBJECT-TYPE
        SYNTAX INTEGER (0..2147483647)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The revision level of the hardware device."
        ::= { hw 3 }
```

```

hwSerialNo    OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The serial number of the hardware device."
    ::= { hw 4 }

hwID          OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The device identifier of the hardware device. Values = 3250,
        4000, 4020, 4030, 4033, 3240, 1000, 1100, 1700, 5940, 4650,
        100 (ACE process), 200 (DOSNMS), 300 (Norand Proxy Agent),
        6710 (Access Point)."
    ::= { hw 5 }

file          OBJECT IDENTIFIER ::= { nSystem 3 }

fsinfo      OBJECT IDENTIFIER ::= { file 1 }

    -- The FileSystem Information Table

fsEnabled     OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the file system is enabled."
    ::= { fsinfo 1 }

fsMaxSectors  OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of physical sectors. A file segment
        consists of one or more adjacent physical sectors."
    ::= { fsinfo 2 }

fsSectorSize  OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The size of a physical sector in bytes."
    ::= { fsinfo 3 }

fsNumSegments OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of logical file segments (0-MAX_SECTORS)."
    ::= { fsinfo 4 }

fsNumFiles    OBJECT-TYPE
    SYNTAX Gauge (0..25)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of files (0-25)."
    ::= { fsinfo 5 }

```

```

fsBootSegment OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The index of the current boot segment.  If the index is
        non-zero and the first file in the associated segment is
        executable, then control is passed to that file during
        the power-up sequence."
    ::= { fsinfo 6 }

fsDataSegment OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The index of the active data segment.  Files stored in
        this segment will be accessible to an executing
        application."
    ::= { fsinfo 7 }

segment          OBJECT IDENTIFIER ::= { file 2 }

-- The File Segment Table

-- Table Definition

segTable         OBJECT-TYPE
    SYNTAX SEQUENCE OF SEEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A catalog of memory segments and their utilization."
    ::= { segment 2 }

-- Row Definition

segEntry         OBJECT-TYPE
    SYNTAX SEEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { segID }
    ::= { segTable 1 }

-- Columnar Object Definitions

SEEntry ::=
    SEQUENCE {
        segID          INTEGER,
        segFirstSector INTEGER,
        segLastSector  INTEGER,
        segStatus      INTEGER,
        segSize        INTEGER,
        segFree        INTEGER
    }

segID           OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The segment ID (1 - (NUM_SEGMENTS+1)).  A non-zero
        number which uniquely identifies a segment."
    ::= { segEntry 1 }

```

```

segFirstSector OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The first physical sector in the segment
        (1 - (MAX_SECTORS + 1))."
    ::= { segEntry 2 }

segLastSector OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The last physical sector in the segment
        (FIRST_SECTOR - (MAX_SECTORS + 1))."
    ::= { segEntry 3 }

segStatus      OBJECT-TYPE
    SYNTAX INTEGER { valid(1),
                    invalid(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The segment status:
         valid   = 1,
         invalid = 2 "
    ::= { segEntry 4 }

segSize        OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The segment size in bytes."
    ::= { segEntry 5 }

segFree        OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of available bytes in the segment which are
        not currently allocated to a file."
    ::= { segEntry 6 }

dir            OBJECT IDENTIFIER ::= { file 3 }

-- The File Directory Table
-- Table Definition
dirTable      OBJECT-TYPE
    SYNTAX SEQUENCE OF DIREntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The FileSystem Directory."
    ::= { dir 2 }

-- Row Definition
dirEntry      OBJECT-TYPE
    SYNTAX DIREntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { dirIndex }
    ::= { dirTable 1 }

```

```

-- Columnar Object Definitions

DIREntry ::=
SEQUENCE {
    dirIndex    INTEGER,
    dirName     DisplayString,
    dirSegment  INTEGER,
    dirType     INTEGER,
    dirSize     INTEGER,
    dirDate     DisplayString,
    dirTime     DisplayString,
    dirVersion  DisplayString
}

dirIndex    OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Directory Index"
    ::= { dirEntry 1 }

dirName     OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..14))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "File name"
    ::= { dirEntry 2 }

dirSegment  OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "File segment (1 - (NUM_SEGMENTS + 1)). The segment ID
        which identifies the segment containing the file."
    ::= { dirEntry 3 }

dirType     OBJECT-TYPE
    SYNTAX INTEGER { executable(1),
                    data(2),
                    invalid(3) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "File type:
        executable = 1,
        data      = 2,
        invalid   = 3 "
    ::= { dirEntry 4 }

dirSize     OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file size in bytes."
    ::= { dirEntry 5 }

```

```

dirDate      OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..12))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file date in MM-DD-YYYY display format."
    ::= { dirEntry 6 }

dirTime      OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..10))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file time in HH:MM:SS display format."
    ::= { dirEntry 7 }

dirVersion   OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..8))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The file version in v99.99 display format."
    ::= { dirEntry 8 }

sysErrors    OBJECT IDENTIFIER ::= { nSystem 4 }

criticalErrors OBJECT IDENTIFIER ::= { sysErrors 1 }

ceEnabled    OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A value of true(1) signifies that the critical error
        log was successfully initialized as part of the power-up
        sequence. Any errors in that initialization process
        result in a value of false(2)."
    ::= { criticalErrors 1 }

ceOverflow   OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Overflow error code. If the overflow code is non-zero,
        it indicates that the log has overflowed and the
        overflow code contains the last displaced value."
    ::= { criticalErrors 2 }

ceReset      OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "A user can reset the critical error log by setting
        ceReset to true(1). Valid values are true(1) or
        false(2)."
    ::= { criticalErrors 3 }

ceLogTable   OBJECT-TYPE
    SYNTAX SEQUENCE OF CELogEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Critical Error Log Table"
    ::= { criticalErrors 4 }

```

```

ceLogEntry          OBJECT-TYPE
    SYNTAX CELogEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { ceLogErrorCode }
    ::= { ceLogTable 1 }

CELogEntry ::=
    SEQUENCE {
        ceLogErrorCode  INTEGER,
        ceLogErrorCount Counter
    }

ceLogErrorCode  OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Critical error code.  A 16-bit value which uniquely
        identifies a system software error.  The error codes are
        intended for internal Norand use."
    ::= { ceLogEntry 1 }

ceLogErrorCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Error count for the associated ceLogErrorCode"
    ::= { ceLogEntry 2 }

nInterfaces          OBJECT IDENTIFIER ::= { norandNET 2 }

nifx                OBJECT IDENTIFIER ::= { nInterfaces 2 }

-- The Norand Extended Interfaces Table

nifxTable           OBJECT-TYPE
    SYNTAX SEQUENCE OF NIFXEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Norand Extended Interface Table"
    ::= { nifx 4 }

nifxEntry           OBJECT-TYPE
    SYNTAX NIFXEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { nifxIndex }
    ::= { nifxTable 1 }

```

```

NIFXEntry ::=
  SEQUENCE {
    nifxIndex          INTEGER,
    nifxType           INTEGER,
    nifxInDisabledDiscards Counter,
    nifxInOverruns    Counter,
    nifxInHWOverruns Counter,
    nifxInUcastDPkts Counter,
    nifxInNUcastDPkts Counter,
    nifxInLenErrors   Counter,
    nifxExcessiveDeferrals Counter,
    nifxInNetIDDiscards Counter,
    nifxInFragDiscards Counter,
    nifxInUFilterDiscards Counter,
    nifxInNUFilterDiscards Counter,
    nifxInQFullDiscards Counter
  }

nifxIndex          OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Interface index"
  ::= { nifxEntry 1 }

nifxType           OBJECT-TYPE
  SYNTAX INTEGER {
    ether(4),
    proxim24(132),
    falcon902(197),
    uhf(198),
    nor24(199)
  }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Norand Interface Type"
  ::= { nifxEntry 2 }

nifxInDisabledDiscards OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of received unicast frames which do not require
  forwarding. Unicast ethernet frames are discarded if
  ether-to-radio flooding is disabled and the destination is
  unknown; otherwise, unicast frames are discarded if the
  bridge has learned that the destination port is the same as
  the source port."
  ::= { nifxEntry 3 }

nifxInOverruns    OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of received frames discarded because the frame
  could not be queued for the MAC-D task."
  ::= { nifxEntry 4 }

```



```
nifxInHWOverruns          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received frames discarded due to hardware
        overruns."
    ::= { nifxEntry 5 }

nifxInUcastDPkts          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received unicast frames successfully delivered
        to the MAC-D task."
    ::= { nifxEntry 6 }

nifxInNUcastDPkts         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received multicast frames successfully
        delivered to the MAC-D task."
    ::= { nifxEntry 7 }

nifxInLenErrors           OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number received frames with length errors."
    ::= { nifxEntry 8 }

nifxExcessiveDeferrals    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of aborted transmissions due to excessive
        deferrals."
    ::= { nifxEntry 9 }

nifxInNetIDDiscards       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received frames discarded because the LAN ID
        did not match."
    ::= { nifxEntry 10 }

nifxInFragDiscards        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received frame fragments discarded because a
        fragmented frame could not be re-assembled."
    ::= { nifxEntry 11 }
```

```

nifxInUFilterDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of enabled received unicast frames discarded due
        to a unicast filter expression."
    ::= { nifxEntry 12 }

nifxInNUFilterDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of enabled received multicast frames discarded
        due to a multicast filter expression."
    ::= { nifxEntry 13 }

nifxInQFullDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received frames discarded because the frame
        could not be queued for the MAC-R task."
    ::= { nifxEntry 14 }

portState OBJECT IDENTIFIER ::= { nInterfaces 3 }

-- The Port State Table

psTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PSEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "MAC-R port state variables."
    ::= { portState 4 }

psEntry OBJECT-TYPE
    SYNTAX PSEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { psPort }
    ::= { psTable 1 }

```

```

PSEntry ::=
  SEQUENCE {
    psPort                INTEGER,
    psIfIndex             INTEGER,
    psAddress             PhysAddress,
    psType                INTEGER,
    psState               INTEGER,
    psCost                INTEGER,
    psHelloPeriod        INTEGER,
    psHelloCount         Counter,
    psMacdWindow         INTEGER,
    psMacdQSize          Gauge,
    psMacdTimeouts      Counter,
    psIsPrimary          INTEGER,
    psIsSecondary        INTEGER,
    psIsSecondaryCandidate INTEGER,
    psSecondaryUniFlooding INTEGER,
    psSecondaryMultiFlooding INTEGER,
    psIsRadio            INTEGER,
    psPendEnabled        INTEGER
  }

psPort                OBJECT-TYPE
  SYNTAX INTEGER (1..4)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "MAC-R port ID (1-4).  A number which uniquely identifies
    the port."
  ::= { psEntry 1 }

psIfIndex             OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "MAC-D interface index.  The index matches the interface
    index of the associated row in the mib-II interface table."
  ::= { psEntry 2 }

psAddress             OBJECT-TYPE
  SYNTAX PhysAddress
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "802 address of the port."
  ::= { psEntry 3 }

psType                OBJECT-TYPE
  SYNTAX INTEGER {
    ether(4),
    proxim24(132),
    falcon902(197),
    uhf(198),
    nor24(199)
  }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Norand port type:
    ether = 4,
    Proxim = 196,
    Falcon = 197,
    UHF = 198"
  ::= { psEntry 4 }

```

```

psState                OBJECT-TYPE
    SYNTAX INTEGER { disabled(0),
                    idle(1),
                    open(2),
                    receive(3),
                    transmit(4) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Port state:
     disabled = 0,
     idle     = 1,
     open     = 2,
     receive  = 3,
     transmit = 4 "
    ::= { psEntry 5 }

psCost                 OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Incremental path cost of the port.
     Default values:
     ether    = 20,
     Falcon   = 100,
     UHF      = 255 "
    ::= { psEntry 6 }

psHelloPeriod         OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Inter-HELLO time (.01 secs.)"
    ::= { psEntry 7 }

psHelloCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "HELLO transmit count"
    ::= { psEntry 8 }

psMacdWindow          OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Maximum number of active MAC-D transmit requests."
    ::= { psEntry 9 }

psMacdQSize           OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Current number of active MAC-D transmit requests."
    ::= { psEntry 10 }

```

```

psMacdTimeouts          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-D transmit timeout errors."
    ::= { psEntry 11 }

psIsPrimary              OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for primary bridge ports."
    ::= { psEntry 12 }

psIsSecondary           OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for secondary bridge ports."
    ::= { psEntry 13 }

psIsSecondaryCandidate  OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if secondary bridge port candidates."
    ::= { psEntry 14 }

psSecondaryUniFlooding  OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for secondary bridge ports which require unicast
        flooding."
    ::= { psEntry 15 }

psSecondaryMultiFlooding OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for secondary bridge ports which require multicast
        flooding."
    ::= { psEntry 16 }

psIsRadio                OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, for radio ports."
    ::= { psEntry 17 }

psPendEnabled           OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the port supports pending messages."
    ::= { psEntry 18 }

```

```

portStats          OBJECT IDENTIFIER ::= { nInterfaces 4 }

pstcTable          OBJECT-TYPE
    SYNTAX SEQUENCE OF PSTCentry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "MAC-R port state variables."
    ::= { portStats 4 }

pstcEntry          OBJECT-TYPE
    SYNTAX PSTCentry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { pstcPort }
    ::= { pstcTable 1 }

PSTCentry ::=
    SEQUENCE {
        pstcPort          INTEGER,
        pstcInOWLPkts    Counter,
        pstcInUcastOWLDataPkts Counter,
        pstcInNUcastOWLDataPkts Counter,
        pstcInOWLErrors Counter,
        pstcOutOWLPkts   Counter,
        pstcOutUcastOWLDataPkts Counter,
        pstcOutNUcastOWLDataPkts Counter,
        pstcOutOWLErrors Counter,
        pstcParentLinkErrors Counter,
        pstcAlertLinkErrors Counter,
        pstcInUcastRelayPkts Counter,
        pstcInNUcastRelayPkts Counter,
        pstcOutUcastRelayPkts Counter,
        pstcOutNUcastRelayPkts Counter,
        pstcInUcastInbound Counter,
        pstcInUcastOutbound Counter,
        pstcInUcastSec Counter,
        pstcInUcastFlood Counter,
        pstcInUcastDiscards Counter,
        pstcInNUcastDiscards Counter,
        pstcInUcastToIFC Counter,
        pstcInNUcastToIFC Counter,
        pstcOutDelayDiscards Counter
    }

pstcPort          OBJECT-TYPE
    SYNTAX INTEGER (1..4)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-R port ID (1-4). A number which uniquely identifies
        the port."
    ::= { pstcEntry 1 }

pstcInOWLPkts     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total received OWL packets."
    ::= { pstcEntry 2 }

```

```

pstcInUcastOWLDataPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received OWL unicast data packets."
    ::= { pstcEntry 3 }

pstcInNUcastOWLDataPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received multicast OWL data packets."
    ::= { pstcEntry 4 }

pstcInOWLErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received OWL packets with errors."
    ::= { pstcEntry 5 }

pstcOutOWLPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total OWL packets sent."
    ::= { pstcEntry 6 }

pstcOutUcastOWLDataPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total unicast OWL data packets sent."
    ::= { pstcEntry 7 }

pstcOutNUcastOWLDataPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total multicast OWL data packets sent."
    ::= { pstcEntry 8 }

pstcOutOWLErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "OWL packet send errors."
    ::= { pstcEntry 9 }

pstcParentLinkErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Root port send link errors"
    ::= { pstcEntry 10 }

```

```

pstcAlertLinkErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Link errors which generated an ALERT."
    ::= { pstcEntry 11 }

pstcInUcastRelayPkts     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast relay packets"
    ::= { pstcEntry 12 }

pstcInNUcastRelayPkts    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received multicast relay packets"
    ::= { pstcEntry 13 }

pstcOutUcastRelayPkts    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total unicast relay packets sent"
    ::= { pstcEntry 14 }

pstcOutNUcastRelayPkts   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total multicast relay packets sent"
    ::= { pstcEntry 15 }

pstcInUcastInbound       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets routed inbound or relayed
         onto the distribution LAN."
    ::= { pstcEntry 16 }

pstcInUcastOutbound      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets routed outbound"
    ::= { pstcEntry 17 }

pstcInUcastSec           OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets relayed to a secondary LAN."
    ::= { pstcEntry 18 }

```



```

pstcInUcastFlood          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets with an unknown destination."
    ::= { pstcEntry 19 }

pstcInUcastDiscards       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast data packets discarded"
    ::= { pstcEntry 20 }

pstcInNUcastDiscards     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received multicast data packets discarded"
    ::= { pstcEntry 21 }

pstcInUcastToIFC         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received unicast packets passed to the data link interface"
    ::= { pstcEntry 22 }

pstcInNUcastToIFC        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Received multicast packets passed to the data link
        interface."
    ::= { pstcEntry 23 }

pstcOutDelayDiscards     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Send packets discarded due to excessive delay."
    ::= { pstcEntry 24 }

ptxq                      OBJECT IDENTIFIER ::= { nInterfaces 5 }

ptxqTable                  OBJECT-TYPE
    SYNTAX SEQUENCE OF PTXQEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The Port Transmit Queue Table"
    ::= { ptxq 1 }

ptxqEntry                  OBJECT-TYPE
    SYNTAX PTXQEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { ptxqPort }
    ::= { ptxqTable 1 }

```

```

PTXQEntry ::=
  SEQUENCE {
    ptxqPort          INTEGER,
    ptxqRegQSize     Gauge,
    ptxqRegQMax      INTEGER,
    ptxqExpQSize     Gauge,
    ptxqExpQMax      INTEGER,
    ptxqQHpCount     Counter,
    ptxqQRegCount    Counter,
    ptxqQExpCount    Counter,
    ptxqQHpDiscards  Counter,
    ptxqQRegDiscards Counter,
    ptxqQExpDiscards Counter,
    ptxqMultiQSize   Gauge,
    ptxqMultiQMax    INTEGER,
    ptxqMultiQDiscards Counter
  }

ptxqPort          OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "MAC-R port ID (1-4).  A number which uniquely identifies
    the port."
  ::= { ptxqEntry 1 }

ptxqRegQSize      OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Current regular queue size (0-REG_Q_MAX).  The number of
    regular priority packets which are currently queued for
    transmission on the port."
  ::= { ptxqEntry 2 }

ptxqRegQMax       OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The maximum number of regular priority packets which can be
    queued for transmission on the port."
  ::= { ptxqEntry 3 }

ptxqExpQSize      OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Current expedited queue size (0..EXP_Q_MAX).  The number of
    expedited packets which are currently queued for
    transmission on the port."
  ::= { ptxqEntry 4 }

ptxqExpQMax       OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The maximum number of expedited packets which can be queued
    for transmission on the port."
  ::= { ptxqEntry 5 }

```

```

ptxqQHpCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of attempts to queue a high priority packet for
        transmission."
    ::= { ptxqEntry 6 }

ptxqQExpCount     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of attempts to queue an expedited priority
        packet for transmission."
    ::= { ptxqEntry 7 }

ptxqQRegCount     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of attempts to queue a regular priority packet
        for transmission."
    ::= { ptxqEntry 8 }

ptxqQHpDiscards  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of failed attempts to queue a high priority
        packet."
    ::= { ptxqEntry 9 }

ptxqQExpDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of failed attempts to queue an expedited
        priority packet."
    ::= { ptxqEntry 10 }

ptxqQRegDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of failed attempts to queue a regular priority
        packet."
    ::= { ptxqEntry 11 }

ptxqMultiQSize   OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Current multicast queue size. The number of multicast
        packets which are queued for transmission on the (radio)
        port. Multicast packets are transmitted after HELLO packets
        on OWL radio ports."
    ::= { ptxqEntry 12 }

```

```

ptxqMultiQMax      OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The maximum number of multicast packets which will be
        queued for transmission on the (radio) port."
    ::= { ptxqEntry 13 }

ptxqMultiQDiscards OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of failed attempts to queue a multicast packet."
    ::= { ptxqEntry 14 }

pmsg                OBJECT IDENTIFIER ::= { nInterfaces 6 }

pmsgTable           OBJECT-TYPE
    SYNTAX SEQUENCE OF PmsgEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Pending Message Table"
    ::= { pmsg 1 }

pmsgEntry           OBJECT-TYPE
    SYNTAX PmsgEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { pmsgPort }
    ::= { pmsgTable 1 }

PmsgEntry ::=
    SEQUENCE {
        pmsgPort                INTEGER,
        pmsgPendRecCurrent      Gauge,
        pmsgPendRecMax          INTEGER,
        pmsgPendMsgCurrent      Gauge,
        pmsgPendMsgMax          INTEGER,
        pmsgPendMsgTotal        Counter,
        pmsgPendMsgDiscards     Counter,
        pmsgPendRecOverflowErrors Counter,
        pmsgPendMsgOverflowErrors Counter,
        pmsgPendAgedRecCount    Counter,
        pmsgPendAgedMsgCount    Counter
    }

pmsgPort            OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-R port ID (1-4).  A number which uniquely identifies
        the port."
    ::= { pmsgEntry 1 }

pmsgPendRecCurrent  OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Current terminal record count"
    ::= { pmsgEntry 2 }

```

```

pmsgPendRecMax          OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Maximum terminal record count"
    ::= { pmsgEntry 3 }

pmsgPendMsgCurrent      OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Current pending message count"
    ::= { pmsgEntry 4 }

pmsgPendMsgMax          OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Maximum pending message count"
    ::= { pmsgEntry 5 }

pmsgPendMsgTotal        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total pending message count"
    ::= { pmsgEntry 6 }

pmsgPendMsgDiscards     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of pending messages in-queue which were
        discarded before they could be delivered because the
        terminal's queue was full."
    ::= { pmsgEntry 7 }

pmsgPendRecOverflowErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times that a terminal requested pending
        message services when no pending message records were
        available."
    ::= { pmsgEntry 8 }

pmsgPendMsgOverflowErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times when the maximum number of stored
        messages, per platform, was exceeded."
    ::= { pmsgEntry 9 }

```

```

pmsgPendAgedRecCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of terminal records discarded due to maximum age
        (12 minutes)."
```

```

 ::= { pmsgEntry 10 }
```

```

pmsgPendAgedMsgCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of pending messages which were discarded due to
        maximum age. (default = 5 seconds)."
```

```

 ::= { pmsgEntry 11 }
```

```

nSNMP                      OBJECT IDENTIFIER ::= { norandNET 11 }
```

```

vlConfig                    OBJECT IDENTIFIER ::= { nSNMP 1 }
```

```

--Norand Community table defines the accepted community
--strings and their access privileges
```

```

-- The Community Table
```

```

communityTable            OBJECT-TYPE
    SYNTAX SEQUENCE OF CommunityEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The community table defines communities and their access
        privileges. Norand's implementation of the community table
        has some special considerations:
```

- 1) GETs and SETs to the community table can only be accomplished using the SUPER-USER community string which Norand has defined;
- 2) This SUPER-USER community string, or password, is defined in the first row of the community table. The communityName contained in the first row of the community table is always the SUPER-USER community string. This community string (communityName) may be modified.
- 3) All rows of the community table are modifiable (SET) when using the SUPER-USER community string. However, for the first row of the community table, only the communityName object is modifiable. This ensures that the SUPER-USER will always have maximum access to the MIB data. All other rows in the community Table are accessible as defined in the MIB definition.
- 4) The SUPER-USER and other default community string values can be found in Norand's User's Guide."

```

 ::= { vlConfig 2 }
```

```

-- Row Definition
```

```

communityEntry            OBJECT-TYPE
    SYNTAX CommunityEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Each entry relates to a specific community and associates
        to its access privileges."
```

```

    INDEX { communityIndex }
```

```

 ::= { communityTable 1 }
```

```

-- Columnar Object Definition

CommunityEntry ::=
    SEQUENCE {
        communityIndex      INTEGER,
        communityStatus     INTEGER,
        communityName       DisplayString,
        communityPrivileges INTEGER,
        communityViewTblIndex INTEGER
    }

-- Leaf Definition

communityIndex      OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Identifies the community row"
    ::= { communityEntry 1 }

communityStatus     OBJECT-TYPE
    SYNTAX  INTEGER { enabled(1),
                    disabled(2),
                    deleted(3) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "Status of a community record. Alterations to the table may
        only be performed by a manager using the SUPER-USER
        community name.
        Status types:
            Enabled = Community record active
            Disabled = Community record not active
            Deleted = Disables and nulls objects in record."
    ::= { communityEntry 2 }

communityName       OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..15))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The authoritative name for the community. Unless the
        Norand SUPER-USER community name is employed, a GET from
        this column yields an access violation."
    ::= { communityEntry 3 }

communityPrivileges OBJECT-TYPE
    SYNTAX  INTEGER { get-only(1),
                    set-and-get(3) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "SET and GET privileges of community."
    ::= { communityEntry 4 }

-- Norand trap table defines all trap target IP addresses

-- Table Definition

```

```

trapTargetTable      OBJECT-TYPE
    SYNTAX  SEQUENCE OF TrapTargetEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "The trap target table specifies the IP address of SNMPv1
        managers that expect trap notifications."
    ::= { v1Config 3 }

-- Row Definition

trapTargetEntry      OBJECT-TYPE
    SYNTAX  TrapTargetEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "Each entry relates to a specific named manager at a given
        IP address and belonging to given community."
    INDEX   { trapTargetIndex }
    ::= { trapTargetTable 1 }

-- Columnar Object Definition

TrapTargetEntry ::=
    SEQUENCE {
        trapTargetIndex      INTEGER,
        trapTargetStatus     INTEGER,
        trapTargetName       DisplayString,
        trapTargetIpAddress  IpAddress
    }

-- Leaf Definition

trapTargetIndex      OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Identifies the trapTarget row"
    ::= { trapTargetEntry 1 }

trapTargetStatus     OBJECT-TYPE
    SYNTAX  INTEGER { enabled(1),
                    disabled(2),
                    deleted(3) }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "Status of a trapTarget record."
    ::= { trapTargetEntry 2 }

trapTargetName       OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..16))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The authoritative name for the trapTarget."
    ::= { trapTargetEntry 3 }

trapTargetIpAddress OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "IP Address of manager (which is assumed to be bound to and
        listening on port 162)."
    ::= { trapTargetEntry 4 }

```



```

nBridge          OBJECT IDENTIFIER ::= { norandNET 17 }

  rt              OBJECT IDENTIFIER ::= { nBridge 2 }

-- The RT Table

-- Table Definition

rtTable          OBJECT-TYPE
    SYNTAX SEQUENCE OF RTEEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Each entry in this table provides routing information for
        child nodes which are reachable via a route."
    ::= { rt 2 }

-- Row Definition

rtEntry          OBJECT-TYPE
    SYNTAX RTEEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { rtDestination }
    ::= { rtTable 1 }

-- Columnar Object Definition

RTEEntry ::=
SEQUENCE {
    rtDestination  PhysAddress,
    rtPort         INTEGER,
    rtAge          INTEGER,
    rtNodeId       INTEGER,
    rtAttachId     INTEGER,
    rtAttachTime   TimeTicks,
    rtApEaddr      PhysAddress,
    rtHopAddrLen   INTEGER,
    rtHopAddr16    INTEGER,
    rtHopEaddr     PhysAddress,
    rtIsBound      INTEGER,
    rtIsRemote     INTEGER,
    rtIsChild      INTEGER,
    rtIsAp         INTEGER,
    rtIsDistributed INTEGER,
    rtIsRemoteLan  INTEGER,
    rtNS           INTEGER,
    rtNR           INTEGER
}

-- Leaf Definition

rtDestination    OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The 802 address of the destination."
    ::= { rtEntry 1 }

rtPort           OBJECT-TYPE
    SYNTAX INTEGER (1..4)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The MAC-R port ID (1-4).  A number which uniquely identifies
        the port."
    ::= { rtEntry 2 }

```

```

rtAge                OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The time (in minutes) since the route was updated."
 ::= { rtEntry 3 }

rtNodeId             OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"16-bit node ID of the destination. A 16-bit identifier which
uniquely identifies an OWL node in an OWL LAN."
 ::= { rtEntry 4 }

rtAttachId           OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Attach sequence number. The sequence number is copied from
an OWL ATTACH request PDU. The sequence number is not valid
for 'remote' nodes."
 ::= { rtEntry 5 }

rtAttachTime         OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Last attach time (.01 secs.)"
 ::= { rtEntry 6 }

rtApEaddr            OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"802 address of AP which is the first hop on the path to the
destination."
 ::= { rtEntry 7 }

rtHopAddrLen         OBJECT-TYPE
SYNTAX INTEGER { twoByte(2),
                 sixByte(6) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"MAC-D address length (2 or 6). A MAC-D entity may use either
16-bit locally assigned addresses or 48-bit 802 addresses."
 ::= { rtEntry 8 }

rtHopAddr16          OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"16-bit MAC-D address ( if rtHopAddrLen is twoByte(2) )"
 ::= { rtEntry 9 }

```

```
rtHopEaddr          OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"48-bit MAC-D address ( if rtHopAddrLen is sixByte(6) )"
 ::= { rtEntry 10 }

rtIsBound           OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is fully attached and the path can be
used to forward data."
 ::= { rtEntry 11 }

rtIsRemote          OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is a non-OWL node."
 ::= { rtEntry 12 }

rtIsChild           OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is a child node."
 ::= { rtEntry 13 }

rtIsAp              OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is an AP."
 ::= { rtEntry 14 }

rtIsDistributed     OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the path is through a distributed AP (root node
only)."
 ::= { rtEntry 15 }

rtIsRemoteLan       OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"True if the destination is a remote LAN."
 ::= { rtEntry 16 }
```

```

rtNS                OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"MAC-R send sequence number for terminal nodes. The 16-bit
(0-65535) sequence number of the last OWL data request PDU
sent to the destination."
 ::= { rtEntry 17 }

rtNR                OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"MAC-R receive sequence number for terminal nodes. The 16-bit
(0-65535) sequence number of the last OWL data request PDU
received from the destination."
 ::= { rtEntry 18 }

brg                OBJECT IDENTIFIER ::= { nBridge 3 }

-- The BRG Table

-- Table Definition

brgTable            OBJECT-TYPE
SYNTAX SEQUENCE OF BRGEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Each entry in this table provides bridge information for
child nodes which are reachable via a bridge."
 ::= { brg 2 }

-- Row Definition

brgEntry            OBJECT-TYPE
SYNTAX BRGEntry
ACCESS not-accessible
STATUS mandatory
INDEX { brgDestination }
 ::= { brgTable 1 }

-- Columnar Object Definition

BRGEntry ::=
SEQUENCE {
    brgDestination PhysAddress,
    brgPort          INTEGER,
    brgAge           INTEGER,
    brgType          INTEGER,
    brgIsPermanent  INTEGER,
    brgTimestamp     TimeTicks
}

-- Leaf Definition

brgDestination     OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The 802 address of the destination."
 ::= { brgEntry 1 }

```

```

brgPort          OBJECT-TYPE
    SYNTAX INTEGER (1..4)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-R port ID (1-4).  A number which uniquely identifies the
        port."
    ::= { brgEntry 2 }

brgAge           OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Time (in minutes) since the entry was updated."
    ::= { brgEntry 3 }

brgType          OBJECT-TYPE
    SYNTAX INTEGER { primary(1),
                    secondary(2),
                    inbound(4) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Entry Type:
         primary   = 1,
         secondary = 2,
         inbound   = 4 "
    ::= { brgEntry 4 }

brgIsPermanent  OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the entry is permanent."
    ::= { brgEntry 5 }

brgTimestamp     OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time when the primary or inbound entry was added or the
        time when the secondary entry was added or re-attached."
    ::= { brgEntry 6 }

addr           OBJECT IDENTIFIER ::= { nBridge 4 }

-- The Addr Table

-- Table Definition

addrTable        OBJECT-TYPE
    SYNTAX SEQUENCE OF AddrEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Each entry in this table provides address information for
        all OWL nodes in the network.  This table exists only at the
        root node."
    ::= { addr 2 }

-- Row Definition

```

```

addrEntry          OBJECT-TYPE
    SYNTAX AddrEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { addrDestination }
    ::= { addrTable 1 }

-- Columnar Object Definition

AddrEntry ::=
    SEQUENCE {
        addrDestination PhysAddress,
        addrAge          INTEGER,
        addrNodeId       INTEGER,
        addrAlias        DisplayString,
        addrDeviceId     INTEGER,
        addrIpAddress    IpAddress
    }

-- Leaf Definition

addrDestination    OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The 802 address of the registered port."
    ::= { addrEntry 1 }

addrAge            OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time (in minutes) since the entry was updated."
    ::= { addrEntry 2 }

addrNodeId         OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "16-bit (0-65535) node/port ID"
    ::= { addrEntry 3 }

addrAlias          OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..16))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "An alias for the 802 address."
    ::= { addrEntry 4 }

addrDeviceId      OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Device ID (0-65535).  An OWL node can, optionally, set a
        device ID in a registration request PDU."
    ::= { addrEntry 5 }

```

```

addrIpAddress          OBJECT-TYPE
    SYNTAX IpAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "32-bit IP address for IP nodes (e.g. APs)."
    ::= { addrEntry 6 }

brgState              OBJECT IDENTIFIER ::= { nBridge 6 }

-- The Bridge State Group

bsAddress            OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "802 address of the AP."
    ::= { brgState 3 }

bsLanId              OBJECT-TYPE
    SYNTAX INTEGER (0..254)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "OWL LAN ID (0-254)."
    ::= { brgState 4 }

bsCostToRoot         OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Path cost to the root."
    ::= { brgState 5 }

bsIsRoot             OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the AP is the root."
    ::= { brgState 6 }

bsIsAttached         OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the AP is attached."
    ::= { brgState 7 }

bsAttachId           OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "16-bit (0-65535) ATTACH sequence number. This number is
        incremented each time the AP sends an ATTACH request."
    ::= { brgState 8 }

```

```

bsMyRootPriority          OBJECT-TYPE
    SYNTAX INTEGER (0..7)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Root priority of the AP (0-7).  An AP with a root priority
        of 0 cannot become the root node.  The AP with the highest
        priority will become the root in an OWL LAN."
    ::= { brgState 9 }

bsRootPort                OBJECT-TYPE
    SYNTAX INTEGER (1..4)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "MAC-R root port number.  The port number (1-4) of the port
        used to communicate with the parent node."
    ::= { brgState 10 }

bsDesignatedRootAddress  OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "802 address of the current root."
    ::= { brgState 11 }

bsDesignatedRootPriority OBJECT-TYPE
    SYNTAX INTEGER (1..7)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Root priority of the current root (1-7)."
    ::= { brgState 12 }

bsDesignatedRootSequence OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Root sequence of the current root (0-255).  The sequence
        number identifies a single instance of the root."
    ::= { brgState 13 }

bsParentAddress          OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "802 address of the parent AP."
    ::= { brgState 14 }

bsPortCount              OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of MAC-R ports."
    ::= { brgState 15 }

```



```

bsNodeId                OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "16-bit node ID (0-65535).  The node ID uniquely identifies
        the node in an OWL LAN."
    ::= { brgState 16 }

bsRootChangedCount     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times that the root has changed."
    ::= { brgState 17 }

bsRootCount            OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times that the AP became the root."
    ::= { brgState 18 }

bsAttachCount         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times that the AP has changed from an unattached
        state to an attached state."
    ::= { brgState 19 }

bsDetachReason        OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Last detach reason code.  The code indicates the reason
        that the AP became unattached for the last occurrence.
        0 - Initial Value,
        1 - A new root node was detected,
        2 - The network inactivity timer expired,
        4 - A better path to the root was detected,
        5 - The node's parent became unattached,
        7 - The node was in a detach list in a HELLO PDU,
        8 - The node was functioning as the root and relinquished
            the root status,
        9 - The maximum number of attache retries was exceeded
            without receiving an ATTACH response PDU,
        900-90F - A MAC-D link error occurred while sending a PDU to
            the parent node."
    ::= { brgState 20 }

bsNetworkTime         OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Distributed network time (.01 sec)."
    ::= { brgState 21 }

```

```

bsUniFloodLevel          OBJECT-TYPE
    SYNTAX INTEGER (1..2)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Flooding level for unicast frames (1-2)."
```

```

 ::= { brgState 22 }
```

```

bsMultiFloodLevel       OBJECT-TYPE
    SYNTAX INTEGER (0..3)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Flooding level for multicast frames (0-3)."
```

```

 ::= { brgState 23 }
```

```

bsIsPrimaryBridge        OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the AP bridges to the distribution LAN."
```

```

 ::= { brgState 24 }
```

```

bsIsSecondaryBridge     OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "TRUE, if the AP is the designated bridge for a secondary
        LAN."
```

```

 ::= { brgState 25 }
```

```

bsUniFilterExpr          OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Unicast ethernet filter ID (0-255).  If the ID is non-zero,
        it points to a user defined expression which filters unicast
        frames on the ethernet port."
```

```

 ::= { brgState 26 }
```

```

bsMultiFilterExpr        OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Multicast ethernet filter ID (0-255).  If the ID is
        non-zero, it points to a user defined expression which
        filters multicast frames on the ethernet port."
```

```

 ::= { brgState 27 }
```

```

bridgeStats          OBJECT IDENTIFIER ::= { nBridge 7 }
```

```

bstcRouteCount           OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total route table entries"
```

```

 ::= { bridgeStats 3 }
```

```

bstcChildCount          OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of attached children"
    ::= { bridgeStats 4 }

bstcChildApCount        OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of attached AP children"
    ::= { bridgeStats 5 }

bstcRemoteCount         OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Non-OWL bridge table entries"
    ::= { bridgeStats 6 }

bstcPrimaryCount        OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Distribution LAN bridge table entries"
    ::= { bridgeStats 7 }

bstcInboundCount        OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Inbound bridge table entries"
    ::= { bridgeStats 8 }

bstcSecondaryCount      OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Secondary LAN bridge table entries"
    ::= { bridgeStats 9 }

bstcRemoteLanCount      OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Route entries for remote LANs"
    ::= { bridgeStats 10 }

bstcRouteGetErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Route table overflow errors"
    ::= { bridgeStats 11 }

```

```

bstcEntryGetErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Bridge table overflow errors"
    ::= { bridgeStats 12 }

bstcRmtLanGetErrors     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Remote LAN overflow errors"
    ::= { bridgeStats 13 }

bstcRouteSeqErrors     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Out-of-sequence route update errors"
    ::= { bridgeStats 14 }

bstcDeleteSeqErrors    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Out-of-sequence route delete errors"
    ::= { bridgeStats 15 }

bstcEntrySeqErrors     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Out-of-sequence bridge entry update errors"
    ::= { bridgeStats 16 }

bstcInvalidUpdateErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Invalid route update errors"
    ::= { bridgeStats 17 }

nTransport              OBJECT IDENTIFIER ::= { norandNET 102 }

wst                     OBJECT IDENTIFIER ::= { nTransport 2 }

wstReset                OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the wst group."
    ::= { wst 1 }

wstLastReset            OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Elapsed time since the wst group was last reset."
    ::= { wst 2 }

```

```

wstClockTicks          OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "WTP clock ticks in hundredths of seconds."
    ::= { wst 3 }

wstInternalErrorCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of internal WTP errors."
    ::= { wst 4 }

wstSessionsLost        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of active WTP connections which were terminated."
    ::= { wst 5 }

wstFrmrCount           OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of WTP Frame Rejects which were transmitted."
    ::= { wst 6 }

wstDataBytesXmit       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of bytes transmitted."
    ::= { wst 7 }

wstDataBytesRecv      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of bytes received."
    ::= { wst 8 }

wstIframesXmit        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames transmitted."
    ::= { wst 9 }

wstIframesRecv        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames received."
    ::= { wst 10 }

```

```

wstIframesRetransmitted OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames which were lost and
        retransmitted."
    ::= { wst 11 }

wstCheckSumErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of WTP information frames which were discarded
        to WTP check sum errors."
    ::= { wst 12 }

wstHlDiscardCount         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received WTP information frames which were
        discarded without being read by a higher layer application.
        Received information frames are queued per connection and
        should normally not be discarded."
    ::= { wst 13 }

wstState                  OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The internal software state of the WTP module."
    ::= { wst 14 }

wstBrgBusy                OBJECT-TYPE
    SYNTAX INTEGER (0..1)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The value of wstBrgBusy is 1 if the WTP currently has an
        outstanding network transmit request. The value is
        otherwise 0."
    ::= { wst 15 }

wstTlTimeouts            OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times the WTP Tl timer has expired
        (i.e. due to lost frames)."
    ::= { wst 16 }

wstTlBucket1             OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times that 2 consecutive Tl errors have
        occurred."
    ::= { wst 17 }

```

```
wstTlBucket2          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times that 3 consecutive Tl errors have
        occurred."
    ::= { wst 18 }

wstTlBucket3          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times that 5 consecutive Tl errors have
        occurred."
    ::= { wst 19 }

wstTlBucket4          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of times that 6 or more consecutive Tl
        errors have occurred."
    ::= { wst 20 }

wstSABMCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total SABM frames received by the local WTP entity."
    ::= { wst 21 }

wstDiscCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of DISC frames received by the local WTP entity"
    ::= { wst 22 }

wstDMCount            OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of DM frames received by the local WTP entity."
    ::= { wst 23 }

wstRNRCount           OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of RNR frames received by the local WTP entity."
    ::= { wst 24 }
```

```

wstRejectCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of REJECT frames received by the local WTP
        entity."
    ::= { wst 25 }

wstSessionsReset        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of active WTP sessions which have been reset in
        the local WTP entity."
    ::= { wst 26 }

wstBrgSrvTime           OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total bridge service time for WTP transmissions in
        hundredths of seconds."
    ::= { wst 27 }

wstBrgSrvCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of WTP bridge transmissions.  The average
        service time is wstBrgSrvTime/wstBrgSrvCount."
    ::= { wst 28 }

wstBrgSrvThreshold      OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A threshold bridge service time for WTP transmissions.
        Service times which exceed the threshold are counted in
        wstBrgSrvThreshCount."
    ::= { wst 29 }

wstBrgSrvThreshCount    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A count of the number of times the bridge service time has
        exceeded the threshold time defined by wstBrgSrvThreshold."
    ::= { wst 30 }

wstBrgSrvLongest        OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The longest network service time measured for a WTP
        transmission."
    ::= { wst 31 }

```



```

wstBrgTxErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of transmission errors reported to the WTP layer
        by the network layer."
    ::= { wst 32 }

wstFatalBrgErrors      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of data errors reported to the WTP layer by the
        network layer.  A fatal error indicates that the network
        root node has changed."
    ::= { wst 33 }

hlit                    OBJECT IDENTIFIER ::= { nTransport 3 }

hlitReset              OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the HLIT group."
    ::= { hlit 1 }

hlitLastReset         OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Elapsed time since the HLIT group was reset."
    ::= { hlit 2 }

hlitTime              OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This is the cumulative time that the WTP took to service
        transmit requests.  Time is in hundredths of seconds."
    ::= { hlit 3 }

hlitCount             OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This is the number of times the WTP was sent a transmit
        request."
    ::= { hlit 4 }

hlitThreshold         OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "When the WTP exceeds this time while servicing a transmit
        request, the hlitThreshCount will be incremented.  Time is
        in hundredths of seconds."
    ::= { hlit 5 }

```

```

hlitThreshCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "When the WTP exceeds the time in hlitThreshold, this count
        is incremented. Time in hundreths of seconds."
    ::= { hlit 6 }

hlitLongest      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This is the maximum time that the WTP took to service
        transmit requests. Time is in hundreths of seconds."
    ::= { hlit 7 }

wcbst            OBJECT IDENTIFIER ::= { nTransport 4 }

wcbstReset      OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the wcbst table."
    ::= { wcbst 1 }

wcbstLastReset  OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Elapsed time since the wcbst table was reset."
    ::= { wcbst 2 }

wcbstTable      OBJECT-TYPE
    SYNTAX SEQUENCE OF WcbstEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "WTP connection control block statistics table. Each entry
        is for an WTP connection statistics object. Statistics are
        per WTP connection."
    ::= { wcbst 3 }

wcbstEntry      OBJECT-TYPE
    SYNTAX WcbstEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { wcbstIndex }
    ::= { wcbstTable 1 }

```

```

WcbsEntry ::=
  SEQUENCE {
    wcbstIndex          INTEGER,
    wcbstInUse          INTEGER,
    wcbstHlState        INTEGER,
    wcbstDsap           INTEGER,
    wcbstSsap           INTEGER,
    wcbstNet16Addr      INTEGER,
    wcbstT1Value        INTEGER,
    wcbstT1Average      INTEGER,
    wcbstT1Deviation    INTEGER,
    wcbstT1Timeouts     Counter,
    wcbstT2Timeouts     Counter,
    wcbstTiTimeouts     Counter,
    wcbstFramesRecv     Counter,
    wcbstIframesRecv    Counter,
    wcbstIframesDiscarded Counter,
    wcbstDataBytesRecv  Counter,
    wcbstFramesXmit      Counter,
    wcbstIframesXmit     Counter,
    wcbstDataBytesXmit  Counter,
    wcbstIframesRetransmitted Counter,
    wcbstState          INTEGER
  }

```

```

wcbstIndex          OBJECT-TYPE
  SYNTAX INTEGER (1..999)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Connection control block identifier"
  ::= { wcbsEntry 1 }

```

```

wcbstInUse          OBJECT-TYPE
  SYNTAX INTEGER (0..1)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "This variable is 1 if the control block is in use."
  ::= { wcbsEntry 2 }

```

```

wcbstHlState        OBJECT-TYPE
  SYNTAX INTEGER (0..9999)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The control block state. 0=closed, 1=busy, 2=connect,
    3=listen, 4=unitdata, 5=unitdata listen, 6=open, 7=send,
    8=closing, 9=disconnect."
  ::= { wcbsEntry 3 }

```

```

wcbstDsap           OBJECT-TYPE
  SYNTAX INTEGER (0..255)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Destination service access point."
  ::= { wcbsEntry 4 }

```

```
wcbstSsap                OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Source service access point."
    ::= { wcbstEntry 5 }

wcbstNet16Addr           OBJECT-TYPE
    SYNTAX INTEGER (2048..9192)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The 16-bit network address of the remote node which is
        active on the connection, if the control block is in an
        active state."
    ::= { wcbstEntry 6 }

wcbstT1Value             OBJECT-TYPE
    SYNTAX INTEGER (0..999)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The current response time-out value in tenths of seconds."
    ::= { wcbstEntry 7 }

wcbstT1Average           OBJECT-TYPE
    SYNTAX INTEGER (0..999)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The average response time-out value in tenths of seconds."
    ::= { wcbstEntry 8 }

wcbstT1Deviation         OBJECT-TYPE
    SYNTAX INTEGER (0..999)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The average deviation from the T1Average values in tenths
        of seconds."
    ::= { wcbstEntry 9 }

wcbstT1Timeouts         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Number of T1 timeouts. A T1 timeout occurs when an
        expected response is lost."
    ::= { wcbstEntry 10 }

wcbstT2Timeouts         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Number of T2 timeouts. A T2 timeout occurs if the higher
        layer does not have data ready to send before a WTP
        supervisory response is required."
    ::= { wcbstEntry 11 }
```

```

wcbstTiTimeouts          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Number of Ti timeouts. A Ti timeout occurs when a
        connection is idle for an INACTIVE TIMEOUT period."
    ::= { wcbEntry 12 }

wcbstFramesRecv          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP frames received."
    ::= { wcbEntry 13 }

wcbstIframesRecv        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames received."
    ::= { wcbEntry 14 }

wcbstIframesDiscarded    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames which were discarded as
        duplicates."
    ::= { wcbEntry 15 }

wcbstDataBytesRecv       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of data bytes received."
    ::= { wcbEntry 16 }

wcbstFramesXmit          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP frames transmitted."
    ::= { wcbEntry 17 }

wcbstIframesXmit         OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames transmitted."
    ::= { wcbEntry 18 }

wcbstDataBytesXmit       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of WTP data frames transmitted."
    ::= { wcbEntry 19 }

```

```

wcbstIframesRetransmitted OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Total number of data frames which were lost and had to be
        retransmitted."
    ::= { wcbEntry 20 }

wcbstState OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The internal software state of the LLC control block."
    ::= { wcbEntry 21 }

nApplication OBJECT IDENTIFIER ::= { norandNET 104 }
gwSession OBJECT IDENTIFIER ::= { nApplication 1 }

    gs OBJECT IDENTIFIER ::= { gwSession 1 }

gsReset OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the gs group."
    ::= { gs 1 }

gsLastReset OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time elapsed since the last reset of gs group."
    ::= { gs 2 }

-- The Gateway Statistics group

gsClockTicks OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of .01 second timer ticks since the timer was
        reset."
    ::= { gs 3 }

gsNetworkErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times the network has been lost."
    ::= { gs 4 }

gsHostDataCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks received from the host
        computer."
    ::= { gs 5 }

```

```

gsHostHaltCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of HALT commands received from the host task"
    ::= { gs 6 }

gsHostInvalidCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of invalid messages received from the host
        task."
    ::= { gs 7 }

gsHostDiscardCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of blocks from the host computer which were
        discarded."
    ::= { gs 8 }

gsConnectCount      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of transport connections."
    ::= { gs 9 }

gsCloseCount        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times a transport connection was closed."
    ::= { gs 10 }

gsBlocksXmit        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of blocks passed to the network."
    ::= { gs 11 }

gsDataXmit          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks passed to the network."
    ::= { gs 12 }

gsBlocksRecv        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of blocks received from the network."
    ::= { gs 13 }

```

```

gsDataRecv          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks received from the network."
    ::= { gs 14 }

gsTermResetCount    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times a session was reset by a terminal."
    ::= { gs 15 }

gsTermContinueCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times a session was continued after a
        transport connection was lost."
    ::= { gs 16 }

gsTermInvalidCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of invalid blocks received from terminals."
    ::= { gs 17 }

gsConnectionErrors  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of transport connection errors."
    ::= { gs 18 }

gsConnectionResets  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times an active transport connection was
        reset."
    ::= { gs 19 }

gsRetransmissions   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks which were retransmitted."
    ::= { gs 20 }

gsRecvSeqErrors     OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks which were discarded due to
        receive sequence errors."
    ::= { gs 21 }

```



```
gsXmitErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of transport transmission errors."
    ::= { gs 22 }

gsTermDiscardCount    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of discarded blocks received from terminals."
    ::= { gs 23 }

gsHostDelayMax        OBJECT-TYPE
    SYNTAX INTEGER (0..9999)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The maximum host delay time. Host delay is the time
        a message is received from a terminal until the host
        responds. Times less than the maximum host delay
        time are counted as transactions times. Host delay
        time is measured in hundredths of seconds."
    ::= { gs 24 }

gsHostDelayTotal      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total host delay time in hundredths of seconds,
        not including delay times which exceeded the maximum
        host delay."
    ::= { gs 25 }

gsHostTransactions    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of times the host responded to a
        terminal message in a time less than the maximum host
        delay time."
    ::= { gs 26 }

gsRecvErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of transport layer receive errors."
    ::= { gs 27 }
```

```

gsTermDelayMax          OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The maximum terminal transaction time in hundredths
        of seconds. Round-trip times which exceed the
        maximum delay are not included as terminal
        transactions."
    ::= { gs 28 }

gsTermDelayThreshold OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Terminal transactions which take longer than the
        value set in gsTermDelayThreshold are counted in
        gsTermThresholdCount. This threshold is specified in
        1/100th seconds."
    ::= { gs 29 }

gsTermDelayTotal        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total time for all terminal transactions."
    ::= { gs 30 }

gsTermTransactions      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of terminal transactions. Terminal
        transaction information is sent to the controller
        from terminals."
    ::= { gs 31 }

gsTermThresholdCount    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of terminal transactions which exceed the
        threshold value set in gsTermDelayThreshold."
    ::= { gs 32 }

gsTermMaxCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of terminal transactions which exceed the
        maximum transaction time set in gsTermDelayMax."
    ::= { gs 33 }

```

```

gsHostMaxCount          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of transactions which exceed the maximum
        host delay time set in gsHostDelayMax."
    ::= { gs 34 }

gsTermDelayTraceOn     OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Terminal transaction delay diagnostics are enabled
        by setting gsTermDelayTraceOn to a non-zero value.
        The results of this trace are found in the tdTable."
    ::= { gs 35 }

gsHostActive           OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Indicates whether the host is active and
        communicating with the controller. A value of 1 =
        active, a value of 0 = inactive. The default is 0.
        Once the host has communicated with the controller,
        this value will always show 'active' (1) UNLESS the
        cpctPortInactTimeout value in the cpcTable is set to
        a non-zero value. If the controller has not received
        a response from the host in the time (in seconds)
        specified by cpctPortInactTimeout, gsHostActive gets
        set to 0."
    ::= { gs 36 }

rcbs                    OBJECT IDENTIFIER ::= { gwSession 2 }

-- The RCBS Table

rcbsTableReset         OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the rcbsTable table."
    ::= { rcbs 2 }

rcbsTableLastReset     OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time elapsed since the last reset of rcbsTable
        table."
    ::= { rcbs 3 }

-- Table Definition

rcbsTable              OBJECT-TYPE
    SYNTAX SEQUENCE OF RcbsEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "SESSION control block status table. Each entry
        corresponds to a terminal session."
    ::= { rcbs 4 }

```

```

-- Row Definition

rcbsEntry          OBJECT-TYPE
    SYNTAX RcbEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { rcbsIndex }
    ::= { rcbsTable 1 }

-- Columnar Object Definition

RcbEntry ::=
    SEQUENCE {
        rcbsIndex          INTEGER,
        rcbsTerminal       INTEGER,
        rcbsType           INTEGER,
        rcbsState          INTEGER,
        rcbsLLCIndex       INTEGER,
        rcbsHostDataCount  Counter,
        rcbsBlocksXmit     Counter,
        rcbsDataXmit       Counter,
        rcbsBlocksRecv     Counter,
        rcbsDataRecv       Counter,
        rcbsTermResetCount Counter,
        rcbsTermContinueCount Counter,
        rcbsCloseCount     Counter,
        rcbsHostTransactions Counter,
        rcbsHostDelayTotal Counter,
        rcbsHostDelayLast  INTEGER,
        rcbsTermResetTime  INTEGER,
        rcbsTermInactTime  INTEGER,
        rcbsHostInactTime  INTEGER,
        rcbsConnectionErrors Counter,
        rcbsNetAddress     PhysAddress,
        rcbsSeqErrors      Counter,
        rcbsTermTransactions Counter,
        rcbsTermDelayTotal Counter,
        rcbsTermDelayLast  INTEGER,
        rcbsTermDelayLongest INTEGER,
        rcbsTermThresholdCount Counter,
        rcbsTermMaxCount   Counter,
        rcbsTermDelayTraceOn INTEGER,
        rcbsHostAlias      DisplayString
    }

-- Leaf Definition

rcbsIndex          OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Non-zero table index."
    ::= { rcbsEntry 1 }

rcbsTerminal       OBJECT-TYPE
    SYNTAX INTEGER (0..127)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Terminal session identifier"
    ::= { rcbsEntry 2 }

```

```

rcbsType          OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The terminal power up type as passed by the terminal
        at the beginning of an RTC session. (Unused until
        V4.16 of terminal software). Values: 11, 21, 46, 47,
        61 indicate 3270 emulation; 62 = 5250 emulation, 63 =
        VT220; Any other value indicates Native mode."
    ::= { rcbsEntry 3 }

rcbsState          OBJECT-TYPE
    SYNTAX INTEGER (0..99)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Session state. 0=reset state, 1=active state."
    ::= { rcbsEntry 4 }

rcbsLLCIndex      OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Transport connection control block identifier.
        Corresponds to an lcbstIndex value."
    ::= { rcbsEntry 5 }

rcbsHostDataCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks received from the host."
    ::= { rcbsEntry 6 }

rcbsBlocksXmit    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Total number of blocks passed to the network."
    ::= { rcbsEntry 7 }

rcbsDataXmit      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks passed to the network."
    ::= { rcbsEntry 8 }

rcbsBlocksRecv    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Total number of blocks received from the terminal."
    ::= { rcbsEntry 9 }

```

```

rcbsDataRecv          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks received from the terminal."
    ::= { rcbsEntry 10 }

rcbsTermResetCount    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times the terminal has reset the session."
    ::= { rcbsEntry 11 }

rcbsTermContinueCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times a terminal has continued a session
        after the transport connection was lost and
        re-established."
    ::= { rcbsEntry 12 }

rcbsCloseCount        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "Number of times an active transport connection used
        by the terminal session was closed."
    ::= { rcbsEntry 13 }

rcbsHostTransactions  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of times the host responded to a
        message from the terminal in a time less than the
        maximum host delay time."
    ::= { rcbsEntry 14 }

rcbsHostDelayTotal    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total host delay for the terminal in hundredths
        of seconds, not including times which exceeded the
        maximum host delay time."
    ::= { rcbsEntry 15 }

rcbsHostDelayLast     OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The host response time, in hundredths of seconds,
        for the last message sent from the terminal."
    ::= { rcbsEntry 16 }

```

```
rcbsTermResetTime      OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time, in seconds, since the terminal reset its
        session."
    ::= { rcbsEntry 17 }

rcbsTermInactTime      OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time, in seconds, since a message was received
        from the terminal."
    ::= { rcbsEntry 18 }

rcbsHostInactTime      OBJECT-TYPE
    SYNTAX INTEGER (0..2147483647)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time, in seconds, since a message was received
        from the host for the terminal."
    ::= { rcbsEntry 19 }

rcbsConnectionErrors  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of times the terminal session lost a
        transport connection."
    ::= { rcbsEntry 20 }

rcbsNetAddress         OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The network address of the remote terminal defined
        by rcbsTerminal."
    ::= { rcbsEntry 21 }

rcbsSeqErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The number of messages discarded due to receive
        sequence errors."
    ::= { rcbsEntry 22 }

rcbsTermTransactions  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of terminal transactions for the
        terminal defined by rcbsTerminal."
    ::= { rcbsEntry 23 }
```

```

rcbsTermDelayTotal      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total time for all transactions for the terminal
        defined by rcbsTerminal."
    ::= { rcbsEntry 24 }

rcbsTermDelayLast       OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time for the last transaction for the terminal
        defined by rcbsTerminal, in 1/100th seconds."
    ::= { rcbsEntry 25 }

rcbsTermDelayLongest    OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The time for the longest transaction for the
        terminal defined by rcbsTerminal, in 1/100th seconds"
    ::= { rcbsEntry 26 }

rcbsTermThresholdCount  OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only --read-reset
    STATUS mandatory
    DESCRIPTION
        "The total number of transactions which exceeded the
        value specified by gsTermDelayThreshold."
    ::= { rcbsEntry 27 }

rcbsTermMaxCount        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of transactions which exceeded the
        value specified by gsTermDelayMax."
    ::= { rcbsEntry 28 }

rcbsTermDelayTraceOn    OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Terminal delay tracing is enabled for the terminal
        defined by rcbsTerminal if rcbsTermDelayTraceOn is
        set to a non-zero value. The results of the trace is
        found in the tdTable. For this to work, global
        terminal delay trace must be turned off - to do this
        set gsTermDelayTraceOn to 0 (this value is found in
        the gsTable)."
    ::= { rcbsEntry 29 }

```



```

rcbsHostAlias          OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..16))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The alias of the host computer associated with the
        RCB."
    ::= { rcbsEntry 30 }

hd                      OBJECT IDENTIFIER ::= { gwSession 3 }

-- The HD Table

hdTableReset          OBJECT-TYPE
    SYNTAX INTEGER { true(1), false(2) }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Reset variable for the host delay table."
    ::= { hd 2 }

hdTableLastReset      OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time elapsed since the last reset of the host
        delay table."
    ::= { hd 3 }

-- Table Definition

hdTable               OBJECT-TYPE
    SYNTAX SEQUENCE OF HdEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Host delay table. Host delay is defined as the time
        from when a terminal message is received until the
        host replies. Each entry represents a bucket in the
        host delay table."
    ::= { hd 4 }

-- Row Definition

hdEntry               OBJECT-TYPE
    SYNTAX HdEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { hdBucket }
    ::= { hdTable 1 }

-- Columnar Object Definition

HdEntry ::=
    SEQUENCE {
        hdHostDelay INTEGER,
        hdCount      Counter,
        hdBucket     INTEGER
    }

-- Leaf Definition

```

```

hdHostDelay      OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"The host delay time. The value in each row defines
a host delay bucket bounded by the value in the
preceeding row (or 0) and the value in the row.
Delay times are defined in 1/100th seconds."
 ::= { hdEntry 1 }

hdCount          OBJECT-TYPE
SYNTAX Counter
ACCESS read-only --read-reset
STATUS mandatory
DESCRIPTION
"The total number of times the host responded to a
message from the terminal in a time bounded by the
associated host delay time."
 ::= { hdEntry 2 }

hdBucket         OBJECT-TYPE
SYNTAX INTEGER (1..20)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The bucket number used to identify a bucket in the
host delay table (hdTable)."
 ::= { hdEntry 3 }

td              OBJECT IDENTIFIER ::= { gwSession 4 }

-- The TD Table

tdTableReset    OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Reset variable for the tdTable."
 ::= { td 2 }

tdTableLastReset OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The time elapsed since last reset of tdTable."
 ::= { td 3 }

-- Table Definition

tdTable         OBJECT-TYPE
SYNTAX SEQUENCE OF TdEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Terminal delay table. Terminal delay is the time
used to send a message from a terminal to a host plus
the time used to send a response. Host processing
and queueing time is not included. Each entry in the
table contains a response time bucket and an
count. Set gsTermDelayTraceOn=1 to enable for all
terminals. Set rcbsTermDelayTraceOn=1 and
gsTermDelayTraceOn=0 to enable for single terminal."
 ::= { td 4 }

```

```

-- Row Definition

tdEntry          OBJECT-TYPE
SYNTAX TdEntry
ACCESS not-accessible
STATUS mandatory
INDEX { tdBucket }
 ::= { tdTable 1 }

-- Columnar Object Definition

TdEntry ::=
SEQUENCE {
    tdTrxnTime  INTEGER,
    tdCount     Counter,
    tdBucket    INTEGER
}

-- Leaf Definition

tdTrxnTime      OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"The round trip transaction time. The value in each
row defines a terminal delay bucket bounded by the
value in the preceeding row (or 0) and the value in
the row. Delay times are defined in 1/100th seconds"
 ::= { tdEntry 1 }

tdCount         OBJECT-TYPE
SYNTAX Counter
ACCESS read-only --read-reset
STATUS mandatory
DESCRIPTION
"The total number of times the terminal delay fell
within the time interval bounded by the associated
tdTrxnTime time."
 ::= { tdEntry 2 }

tdBucket        OBJECT-TYPE
SYNTAX INTEGER (1..20)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The bucket number used to identify a bucket in the
terminal delay table (tdTable)."
 ::= { tdEntry 3 }

ahost          OBJECT IDENTIFIER ::= { nApplication 2 }

ahstReset      OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Reset variable for the ahost group."
 ::= { ahost 1 }

ahstLastReset  OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The time elapsed since the last reset of ahost group."
 ::= { ahost 2 }

```

```
ahstFramesOut      OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of frames from the gateway sent to the host."
    ::= { ahost 3 }

ahstFramesIn       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of frames received from the host."
    ::= { ahost 4 }

ahstIdleDetected   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times that the host went inactive. The
        time out is set in the gateway from the user interface."
    ::= { ahost 5 }

ahstNoErrors       OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of host interface errors detected by the
        gateway."
    ::= { ahost 6 }

ahstSyntaxErrors   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of invalid commands received. Incremented
        whenever a ?1 error is sent to the host."
    ::= { ahost 7 }

ahstLengthErrors   OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of host frames that are too long. Increments
        whenever a ?4 is issued by the gateway."
    ::= { ahost 8 }

ahstDisabledErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of commands sent to disabled terminals.
        Increments whenever the gateway issues a ?5 response."
    ::= { ahost 9 }
```

```

ahstGapErrors          OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times the Gap time out is exceeded. The time
        out value is set with the CMT5 command and is measured
        by the start of a command and the receipt of a carriage
        return <CR>. Increments whenever the gateway issues a
        ?6 response."
    ::= { ahost 10 }

ahstCheckErrors        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times the host command fails the LRC or
        CRC-16 check. LRC/CRC error checking is enabled by the
        CMT8 command. Increments whenever the controller sends a
        ?8 response."
    ::= { ahost 11 }

ahstNoBuffersErrors    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times the gateway cannot receive a command
        from the host because of the lack of local buffers.
        Increments when the gateway sends a ?10 response."
    ::= { ahost 12 }

ahstSequenceErrors    OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of times the host sends an R command before the
        host initialized the gateway with CMT commands.
        Increments when the gateway sends a ?11 response."
    ::= { ahost 13 }

-- ahstOtherErrors { ahost 14 } is obsolete and has been
-- deleted.

ahstParityErrors        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of async parity errors."
    ::= { ahost 15 }

ahstFrameErrors        OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of async character framing errors."
    ::= { ahost 16 }

```

```

    ahstNoiseErrors      OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "Number of UART signal samples with ambiguous results."
        ::= { ahost 17 }

    ahstBreakErrors      OBJECT-TYPE
        SYNTAX Counter
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "Number of async break characters received."
        ::= { ahost 18 }

nControl                OBJECT IDENTIFIER ::= { norandNET 105 }

    powerUp              OBJECT IDENTIFIER ::= { nControl 1 }

        pwrPowerUpCount  OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                "Power-up count"
            ::= { powerUp 1 }

        pwrNextPowerUpTime OBJECT-TYPE
            SYNTAX TimeTicks
            ACCESS read-write
            STATUS mandatory
            DESCRIPTION
                "Next power-up time (Reboots the device)."
            ::= { powerUp 2 }

    softwareDownload     OBJECT IDENTIFIER ::= { nControl 2 }

        sdServerIpAddress OBJECT-TYPE
            SYNTAX IpAddress
            ACCESS read-write
            STATUS mandatory
            DESCRIPTION
                "TFTP server IP address"
            ::= { softwareDownload 2 }

        sdScriptFilename  OBJECT-TYPE
            SYNTAX DisplayString (SIZE (0..80))
            ACCESS read-write
            STATUS mandatory
            DESCRIPTION
                "Download script filename (May include path)."
            ::= { softwareDownload 3 }

        sdStartTime       OBJECT-TYPE
            SYNTAX TimeTicks
            ACCESS read-write
            STATUS mandatory
            DESCRIPTION
                "The amount of time to delay before beginning the software
                download."
            ::= { softwareDownload 1 }

```

```

sdStatus          OBJECT-TYPE
  SYNTAX INTEGER {
      sdPending(1),
      sdSTStopped(2),
      sdInProgress(3),
      sdTerminated(4),
      sdSuccess(5),
      sdError(6),
      pwrNPUT(7),
      tftpError(8)
  }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Status of the current software download."
  ::= { softwareDownLoad 4 }

sdErrorString     OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..40))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Description of sdStatus field"
  ::= { softwareDownLoad 5 }

sdCheckPoint     OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
  "An application variable intended to contain a number
  relating the progress of the current software download."
  ::= { softwareDownLoad 6 }

sdSetActivePointers OBJECT-TYPE
  SYNTAX INTEGER {
      none(1),
      boot(2),
      data(3),
      both(4)
  }
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
  "If the device reboots due to the expiration of the
  pwrNextPwrUpTime timer, this value specifies which active
  pointers will be toggled prior to rebooting."
  ::= { softwareDownLoad 7 }

sdTerminate      OBJECT-TYPE
  SYNTAX INTEGER {
      true(1),
      false(2)
  }
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
  "Terminate the current software download."
  ::= { softwareDownLoad 8 }

```

END

ANSI (American National Standards Institute)

A private organization that coordinates some United States (US) standards setting. It also approves some US standards that are often called ANSI standards. ANSI also represents the US to the International Organization for Standards (ISO).

ASN.1 (Abstract Syntax Notation One)

A machine independent data definition language used by ISO protocols. Most notable use is for SNMP and the MIBs. ASN.1 is defined in ISO documents 8824.2 and 8825.2. See also: BER.

Assigned Numbers

The codes (numbers in hexadecimal, decimal, or binary format) officially recognized and assigned as part of the Internet standards.

Backbone

The main connectivity device of a distributed system. All systems that have connectivity to the backbone connect to each other. This does not stop systems from setting up private arrangements with each other to bypass the backbone for cost, performance, or security.

BER (Basic Encoding Rules)

Basic Encoding Rules. Standard rules for encoding data units described in ASN.1. Sometimes incorrectly lumped under the ASN.1 term, which refers only to the abstract syntax description language, not the encoding technique. See also: ASN.1.

BGP (Border Gateway Protocol)

A suggested routing protocol for autonomous systems, based on experience with EGP.

Collision

When more than one LAN station attempts communications on the same transmission medium simultaneously.

Community

Defines a level of authentication and access control at the SNMP agent. The SNMP agent establishes one community for each desired combination of authentication and access control characteristics.

Connection-Oriented

Data communication method in which communication works through link establishment, data transfer, and link release, such as TCP and SPX protocols. See also: connectionless, TCP.

Connectionless

Data communication method in which communication occurs between hosts with no previous setup. Packets between two hosts may take different routes, each independent from the other. UDP, IPX, and IP are connectionless protocols. See also: connection-oriented, UDP.

DIX Ethernet (DEC Intel Xerox Ethernet)

Also called Ethernet 2. See Ethernet.

Dot Address

Dot address refers to the common notations for IP addresses of the form 0.0.0.0; where each zero represents one decimal byte of a four-byte IP address. Also called dotted decimal notation. See also: IP address.

EGP (Exterior Gateway Protocol)

A hop-oriented, reachability, routing protocol used by routers in an internet. The routing information is distributed to the routers which connect autonomous systems. EGP is used in the Internet core system. See also: Core Gateway, BGP, IGP.

Ethernet

A general term indicating both 802.3 and DIX Ethernet (also called Ethernet 2.0). See also: 802.3

Frame

A series of bytes of data encapsulated with a header (and trailer). Frame is often used interchangeably with packet, although technically a packet refers to data from the network layer of the protocol stack.

Gateway (GW)

In current usage, “gateway” and “application gateway” refer to translating systems that convert data traveling from one environment to another.

Get

A basic SNMP PDU operation that lets a management station tell an agent to retrieve a value that agent stores. A way to retrieve management information.

HHC (Hand-Held Computer)

A generic acronym for a hand-held computer, including the 4000 Series (43XX, 44XX, 4500) and the PEN*KEY or 6000 Series (61XX, 62XX, 63XX, 6400, 65XX, 66XX).

Also the trademark of another company.

Hop

Used in vector distance routing. A hop equals one data link. A path to the final destination on a net is a series of hops away from the origin.

Host

A customer’s host computer.

A computer that provides services directly to users, such as the user’s computer. In TCP/IP, an IP addressed device.

A large computer that serves many users, such as a minicomputer or mainframe.

Host Address

The part of an internet address that designates which node on the (sub)network is being addressed. Also called host number.

IANA (Internet Assigned Numbers Authority)

The central registry for various Internet protocol parameters, such as port, protocol, and enterprise numbers, and options, codes, and types. See also: assigned numbers.

ICMP (Internet Control Message Protocol)

The protocol that handles errors and control messages at the IP layer. ICMP is actually a part of the IP protocol layer. It can generate error messages, test packets, and informational messages related to IP.

IEEE

Institute of Electrical and Electronic Engineers.

IGP (Interior Gateway Protocol)

A protocol, such as RIP or OSPF, used within an autonomous system. Contrast with EGP, which is used between autonomous systems. See also: EGP, RIP.

Integrated Gateway/Access Point (IGAP)

Operates as a wired bridge or a wireless access point. Connects to or coexists with existing wired LANs that conform to ANSI/IEEE 802.3 and DIX Ethernet specifications. In current usage, “gateway” and “application gateway” refer to translating systems that convert data traveling from one environment to another.

internet

While a network, “internet” usually refers to a collection of networks interconnected with routers. See also: network.

Internet

(Note the capital “I”.) The world’s largest internet, including large national backbone nets (such as MILNET, NSFNET, and CREN) and many regional and local networks world-wide. The Internet uses the TCP/IP suite. Networks with only Email connectivity are not considered on the Internet.

IP (Internet Protocol)

The network layer for the TCP/IP Protocol Suite. It is a connectionless, best-effort packet switching protocol that offers a common layer over dissimilar networks.

IP Address

An IP defined 32-bit address usually in dotted decimal notation. See also: dot address, IP, network address, subnet address, host address.

IPX (Internetwork Packet eXchange)

A protocol derived from XNS and used by Novell’s NetWare. A router with IPX routing can interconnect LANs for Novell NetWare clients and servers to communicate. IPX provides a “best effort” delivery service and is equivalent to TCP/IP’s IP. See also: LAN.

ISO (International Organization for Standardization)

A voluntary, nontreaty organization founded in 1946 responsible for creating international standards. See also: ANSI, OSI.

LAN (Local Area Network)

A group of network devices in which each device can communicate through a wired or wireless link. The wired link may have several segments joined by repeaters and bridges. The LAN is characterized by the relatively short distance it is designed to cover, a high speed of operation, and relatively low error rates. The geographic scope of LANs is limited to thousands of feet or closely-spaced building complexes.

Layer

A modular portion of a stacked protocol that consists of one or more semi-independent protocols. Each layer builds on the layer beneath it and feeds information to the protocols in the layers above it. TCP/IP has five layers of protocols, and OSI has seven.

LLC (Logical Link Control)

The upper portion of the data link layer, defined in the IEEE 802.2 standard. The LLC layer represents a uniform interface to the user of the data link service, usually a network layer.

MAC (Media Access Control) Address

The “built-in” hardware address of a device connected to a shared media. See also: MAC Sublayer, Ethernet, token ring.

Management Agent

Program that resides on the managed device and responds to information requests from the management station (GETs or SETs). May provide unsolicited information about extraordinary events (TRAPs) to the station.

Management Station

The management station talks to network management agents, which resides in the managed nodes. It provides an interface or is responsible for monitoring and controlling the network or a portion of the network.

MIB (Management Information Base)

A database which contains information about management objects used as access points at the management agent for the management station. The information identifies the management element and specifies its type and access mode (Read-Only, Read-Write). See also: SNMP.

Network

A computer data communications system which interconnects computer systems at various sites. A network may include any combination of LANs, MANs, or WANs. See also: LAN, WAN, internet.

Network Address

In TCP/IP, the network portion of an IP address. For a class A network, the network address is the first byte of the IP address. For a class B network, the network address is the first two bytes of the IP address. For a class C network, the network address is the first three bytes of the IP address. In each case, the remainder is the host address. In the Internet, assigned network addresses are globally unique. See also: Internet, IP address, subnet address, host address.

Network Management Protocol

Allows management stations and management agents to exchange information.

Node

A connection point. An addressable device attached to a computer network. See also: host, router.

OID (Object Identifier)

A unique SNMP identifier assigned to each object. OIDs consist of a sequence of integer values represented in dot notation or address. See also: dot address.

OSI (Open Systems Interconnection)

The seven layer suite of protocols, designed by ISO committees, to be the international standard computer network architecture. See also: ISO, OSI Reference Model.

OSI Reference Model

A seven-layer structure designed to describe computer network architectures and their data communication methods.

Packet

The unit of data sent across a packet switching network. In TCP/IP, it has come to refer to the data package sent across a physical network as in Ethernet packet. See also: Frame.

PDU (Protocol Data Unit)

An OSI term for “packet.” A data object exchanged by protocol machines (entities) within a given layer. In TCP/IP, it defines the type of SNMP message being sent.

Port

Physically, an access point to a computer, multiplexer device, or network. Logically, a pointer to a TCP/IP application.

Protocol

A formal description of message formats and the rules computers must follow to exchange those messages.

Protocol Stack

A modular set of functions, one at each layer of the stack, that form a network service. Each layer of the stack uses the services of the module beneath it.

Repeater

An electronic device used to regenerate, at full strength, signals which have become too weak. Used with digital signals, it ignores extraneous voltages such as noise.

RFC (Request For Comments)

1. The document series, begun in 1969, which describes the Internet suite of protocols, related experiments, and other Internet-related information and ideas.
2. Initial form for regulations from many consortia and government agencies.

RIP (Routing Information Protocol)

An IGP supplied with UNIX and most routers. As “routed” (pronounced route-dee), it was developed for the Berkeley UNIX (BSD) operating system. It lets a group of hosts, located on a LAN, share routing information. RIP is used on many LANs and is also called local routing process.

Router

A device that forwards traffic between networks or subnetworks. It operates at the OSI Network Layer (layer 3) or the IP layer in TCP/IP. See also: bridge, gateway, EGP, IGP.

Set

A basic SNMP PDU operation that lets a management station tell an agent to change a value that agent stores. A way to manipulate management information.

SNMP (Simple Network Management Protocol)

A technology on which companies build management systems. The standard protocol for managing TCP/IP networks. The Internet standard protocol for managing nodes on an IP network. See also: MIB.

Subnet Address

The network-significant part of the locally administered part of a network IP address. Identified using a subnet (or address) mask. See also: address mask, IP address, network address, host address.

Subnetwork

A group of OSI-end and intermediate systems under control of a single administrative domain and using a common network access protocol.

TCP (Transmission Control Protocol)

An Internet standard transport protocol in the Internet protocols suite for reliable, connection-oriented, full-duplex streams. Contrasts with UDP.

TCP/IP (Transmission Control Protocol, Internet Protocol)

Most networks combine IP with a higher-level protocol called Transport Control Protocol (TCP), which establishes a virtual connection between a destination and a source.

IP by itself is something like a postal system. It allows you to address a package and drop it in the system, but there is no direct link between you and the recipient. TCP/IP establishes a connection between two hosts so they can send messages back and forth for a period of time.

Telnet

The virtual terminal protocol in the Internet suite of protocols. Users on one host can access another host and work as users of that host.

Token Ring

A network access method and topology where a token is sequentially passed from station to station. Stations wishing to send data must wait for the token before transmitting data. In a token ring, the next logical station is the next physical station on the ring.

Transmission

The electrical transfer of a signal, message, or other form of data from one location to another.

Trap

A basic SNMP operation that sends unsolicited SNMP agent messages, about extraordinary events that occurred, for the SNMP management station.

UDP (User Datagram Protocol)

An Internet standard transport protocol that exchanges datagrams without acknowledgements or guaranteed delivery.

WAN (Wide Area Network)

A network that covers a large geographic area. It is often used to connect two or more LANs together. This usually involves a variety of methods to maintain communications between all nodes in the network. For example, microwave communications, fiber-optic connections or leased telephone lines.

General Index

NOTE:

This index covers all topics. Those in italics are figures, those in bold are tables.

A

Abstract Syntax Notation.1. *See* ASN.1

Access Point. *See* AP MIB groups

AP MIB groups definitions

bridging parameters

addr, 4-41

brg, 4-40

brgState, 4-43

bridgeStats, 4-46

rt, 4-36

control group

powerUp, 4-48

softwareDownload, 4-49

interface information

nifx, 4-19

pmsg, 4-32

portState, 4-22

portStats, 4-26

ptxq, 4-29

product OID, 4-13

SNMP group

community, 4-34

trapTarget, 4-35

system information

criticalErrors, 4-18

dir, 4-16

fsinfo, 4-14

hw, 4-13

segment, 4-15

AP MIB groups summary

bridging parameters

addr, **4-10**

brg, **4-10**

brgState, **4-10**

bridgeStats, **4-11**

rt, **4-9**

control group

powerUp, **4-12**

softwareDownload, **4-12**

interface information

nifx, **4-4**

pmsg, **4-7**

portState, **4-5**

portStats, **4-5**

ptxq, **4-6**

product OIDs, products, **4-2**

SNMP group

community, **4-8**

trapTarget, **4-8**

system information

criticalErrors, **4-4**

dir, **4-3**

fsinfo, **4-3**

hw, **4-2**

segment, **4-3**

ASN.1, 2-2

definition, 2-4

G

Gateway. *See* GW MIB groups

Gets, 2-2

GW MIB groups definitions

application layer group

ahost, 5-41

gs, 5-28

hd, 5-39

rcbs, 5-33

td, 5-40

control group, powerUp, 5-44

product OIDs, 5-12

SNMP group

community, 5-16

trapTarget, 5-17

system information

criticalErrors, 5-14

dir, 5-13

hw, 5-12

transport groups

hlit, 5-23

wcbst, 5-24

wst, 5-18

GW MIB groups summary

application layer group

ahost, **5-11**

gs, **5-8**

hd, **5-10**

rcbs, **5-9**

td, **5-10**

control group, powerUp, **5-11**

product OIDs, products, **5-2**

SNMP group

community, **5-4**

trapTarget, **5-4**

system information

criticalErrors, **5-3**

dir, **5-3**

hw, **5-2**

transport groups

hlit, **5-6**

wcbst, **5-6**

wst, **5-5**

I

IGAP MIB groups definitions

application layer group

ahost, 6-79

gs, 6-66

hd, 6-77

rcbs, 6-71

td, 6-78

bridging parameters

addr, 6-49

brg, 6-48

brgState, 6-51

bridgeStats, 6-54

rt, 6-45

control group

powerUp, 6-82

softwareDownload, 6-82

interface information

nifx, 6-27

pmsg, 6-40

portState, 6-30

portStats, 6-34

ptxq, 6-37

product OIDs, 6-21

SNMP group

community, 6-42

trapTarget, 6-44

system information

criticalErrors, 6-26

dir, 6-24

fsinfo, 6-22

hw, 6-21

segment, 6-23

transport groups

hlit, 6-61

wcbst, 6-62

wst, 6-56

IGAP MIB groups summary

application layer group

ahost, **6-19**

gs, **6-16**

hd, **6-18**

rcbs, **6-17**

td, **6-18**

bridging parameters

addr, **6-11**

brg, **6-11**

brgState, **6-11**

bridgeStats, **6-12**

rt, **6-10**

control group

powerUp, **6-19**

softwareDownload, **6-20**

interface information

nifx, **6-5**

pmsg, **6-8**

portState, **6-6**

portStats, **6-6**

ptxq, **6-7**

product OIDs, products, **6-2**

SNMP group

community, **6-9**

trapTarget, **6-9**

system information

criticalErrors, **6-4**

dir, **6-4**

fsinfo, **6-3**

hw, **6-3**

segment, **6-4**

transport groups

hlit, **6-14**

wcbst, **6-14**

wst, **6-13**

Installation, 1-2

Integrated Gateway/Access Point. *See*
IGAP MIB groups

Introduction, 1-2

M

MIB groups definitions for AP

- bridging parameters
 - addr, 4-41
 - brg, 4-40
 - brgState, 4-43
 - bridgeStats, 4-46
 - rt, 4-36
- control group
 - powerUp, 4-48
 - softwareDownload, 4-49
- interface information
 - nifx, 4-19
 - pmsg, 4-32
 - portState, 4-22
 - portStats, 4-26
 - ptxq, 4-29
- product OID, 4-13
- SNMP group
 - community, 4-34
 - trapTarget, 4-35
- system information
 - criticalErrors, 4-18
 - dir, 4-16
 - fsinfo, 4-14
 - hw, 4-13
 - segment, 4-15

MIB groups definitions for GW

- application layer group
 - ahost, 5-41
 - gs, 5-28
 - hd, 5-39
 - rcbs, 5-33
 - td, 5-40
- control group, powerUp, 5-44
- product OIDs, 5-12
- SNMP group
 - community, 5-16
 - trapTarget, 5-17
- system information
 - criticalErrors, 5-14
 - dir, 5-13
 - hw, 5-12
- transport groups
 - hlit, 5-23
 - wcbst, 5-24
 - wst, 5-18

MIB groups definitions for IGAP

- application layer group
 - ahost, 6-79
 - gs, 6-66
 - hd, 6-77
 - rcbs, 6-71
 - td, 6-78
- bridging parameters
 - addr, 6-49
 - brg, 6-48
 - brgState, 6-51
 - bridgeStats, 6-54
 - rt, 6-45
- control group
 - powerUp, 6-82
 - softwareDownload, 6-82
- interface information
 - nifx, 6-27
 - pmsg, 6-40
 - portState, 6-30
 - portStats, 6-34
 - ptxq, 6-37
- product OIDs, 6-21

- SNMP group
 - community, 6-42
 - trapTarget, 6-44
- system information
 - criticalErrors, 6-26
 - dir, 6-24
 - fsinfo, 6-22
 - hw, 6-21
 - segment, 6-23
- transport groups
 - hlit, 6-61
 - wcbst, 6-62
 - wst, 6-56

MIB groups summary for AP

- bridging parameters
 - addr, **4-10**
 - brg, **4-10**
 - brgState, **4-10**
 - bridgeStats, **4-11**
 - rt, **4-9**
- control group
 - powerUp, **4-12**
 - softwareDownload, **4-12**
- interface information
 - nifx, **4-4**
 - pmsg, **4-7**
 - portState, **4-5**
 - portStats, **4-5**
 - ptxq, **4-6**
- product OIDs, products, **4-2**
- SNMP group
 - community, **4-8**
 - trapTarget, **4-8**
- system information
 - criticalErrors, **4-4**
 - dir, **4-3**
 - fsinfo, **4-3**
 - hw, **4-2**
 - segment, **4-3**

MIB groups summary for GW

- application layer group
 - ahost, **5-11**
 - gs, **5-8**
 - hd, **5-10**
 - rcbs, **5-9**
 - td, **5-10**
- control group, powerUp, **5-11**
- product OIDs, products, **5-2**
- SNMP group
 - community, **5-4**
 - trapTarget, **5-4**
- system information
 - criticalErrors, **5-3**
 - dir, **5-3**
 - hw, **5-2**
- transport groups
 - hlit, **5-6**
 - wcbst, **5-6**
 - wst, **5-5**

MIB groups summary for IGAP

- application layer group
 - ahost, **6-19**
 - gs, **6-16**
 - hd, **6-18**
 - rcbs, **6-17**
 - td, **6-18**
- bridging parameters
 - addr, **6-11**
 - brg, **6-11**
 - brgState, **6-11**
 - bridgeStats, **6-12**

- rt, **6-10**
- control group
 - powerUp, **6-19**
 - softwareDownload, **6-20**
- interface information
 - nifx, **6-5**
 - pmsg, **6-8**
 - portState, **6-6**
 - portStats, **6-6**
 - ptxq, **6-7**
- product OIDs, products, **6-2**
- SNMP group
 - community, **6-9**
 - trapTarget, **6-9**
- system information
 - criticalErrors, **6-4**
 - dir, **6-4**
 - fsinfo, **6-3**
 - hw, **6-3**
 - segment, **6-4**
- transport groups
 - hlit, **6-14**
 - wcbst, **6-14**
 - wst, **6-13**

MIB-2, basic information, 2-4

MIBs, 2-2

- access mode, 2-5
- ASN.1, 2-2
- example tree, 2-3
- installation, 1-2
- introduction, 1-2
- Norand basic information, 3-2
- object identifier, 2-2
- specific order of loading, 1-2

N

Network management

- architecture, 2-2
- components, 2-1
- functions, 2-1
- goal, 2-1
- simple protocol, 2-2

O

Objects, management, 1-3

Overview, SNMP, 1-3

OWLMIBS contents

- DEVICES, 1-3
- MIBS, 1-2, 1-3
- SYMBOLS, 1-3

P

Product contents, 1-1

S

Sets, 2-2

SNMP

- access control, 2-5
- architecture, 2-2
- basic operations, 2-2
- design, 2-2
- introduction, 1-3
- overview, 1-3
- traps, 2-5