

# Remote Access Concentrator SNMP MIB Reference

Marketing Release 5.1

Part No. 118361-A Rev. A  
September 1997



**Copyright © 1997 Bay Networks, Inc.**

All rights reserved. Printed in the USA. September 1997.

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Bay Networks, Inc.

The software described in this document is furnished under a license agreement and may only be used in accordance with the terms of that license. A summary of the Software License is included in this document.

**Trademarks**

Remote Annex, Annex Manager, and Bay Networks are registered trademarks and BayStack, Quick2Config, System 5000, Bay Networks Press, and the Bay Networks logo are trademarks of Bay Networks, Inc.

Microsoft, MS, MS-DOS, Win32, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

All other trademarks and registered trademarks are the property of their respective owners.

**Restricted Rights Legend**

Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Notwithstanding any other license agreement that may pertain to, or accompany the delivery of, this computer software, the rights of the United States Government regarding its use, reproduction, and disclosure are as set forth in the Commercial Computer Software-Restricted Rights clause at FAR 52.227-19.

**Statement of Conditions**

In the interest of improving internal design, operational function, and/or reliability, Bay Networks, Inc. reserves the right to make changes to the products described in this document without notice.

Bay Networks, Inc. does not assume any liability that may occur due to the use or application of the product(s) or circuit layout(s) described herein.

Portions of the code in this software product are Copyright © 1988, Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms of such portions are permitted, provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that such portions of the software were developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from such portions of the software without specific prior written permission.

SUCH PORTIONS OF THE SOFTWARE ARE PROVIDED “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

In addition, the program and information contained herein are licensed only pursuant to a license agreement that contains restrictions on use and disclosure (that may incorporate by reference certain limitations and notices imposed by third parties).

## Bay Networks, Inc. Software License Agreement

**NOTICE:** Please carefully read this license agreement before copying or using the accompanying software or installing the hardware unit with pre-enabled software (each of which is referred to as “Software” in this Agreement). BY COPYING OR USING THE SOFTWARE, YOU ACCEPT ALL OF THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT. THE TERMS EXPRESSED IN THIS AGREEMENT ARE THE ONLY TERMS UNDER WHICH BAY NETWORKS WILL PERMIT YOU TO USE THE SOFTWARE. If you do not accept these terms and conditions, return the product, unused and in the original shipping container, within 30 days of purchase to obtain a credit for the full purchase price

**1. License Grant.** Bay Networks, Inc. (“Bay Networks”) grants the end user of the Software (“Licensee”) a personal, nonexclusive, nontransferable license: a) to use the Software either on a single computer or, if applicable, on a single authorized device identified by host ID, for which it was originally acquired; b) to copy the Software solely for backup purposes in support of authorized use of the Software; and c) to use and copy the associated user manual solely in support of authorized use of the Software by Licensee. This license applies to the Software only and does not extend to Bay Networks Agent software or other Bay Networks software products. Bay Networks Agent software or other Bay Networks software products are licensed for use under the terms of the applicable Bay Networks, Inc. Software License Agreement that accompanies such software and upon payment by the end user of the applicable license fees for such software.

**2. Restrictions on use; reservation of rights.** The Software and user manuals are protected under copyright laws. Bay Networks and/or its licensors retain all title and ownership in both the Software and user manuals, including any revisions made by Bay Networks or its licensors. The copyright notice must be reproduced and included with any copy of any portion of the Software or user manuals. Licensee may not modify, translate, decompile, disassemble, use for any competitive analysis, reverse engineer, distribute, or create derivative works from the Software or user manuals or any copy, in whole or in part. Except as expressly provided in this Agreement, Licensee may not copy or transfer the Software or user manuals, in whole or in part. The Software and user manuals embody Bay Networks’ and its licensors’ confidential and proprietary intellectual property. Licensee shall not sublicense, assign, or otherwise disclose to any third party the Software, or any information about the operation, design, performance, or implementation of the Software and user manuals that is confidential to Bay Networks and its licensors; however, Licensee may grant permission to its consultants, subcontractors, and agents to use the Software at Licensee’s facility, provided they have agreed to use the Software only in accordance with the terms of this license.

**3. Limited warranty.** Bay Networks warrants each item of Software, as delivered by Bay Networks and properly installed and operated on Bay Networks hardware or other equipment it is originally licensed for, to function substantially as described in its accompanying user manual during its warranty period, which begins on the date Software is first shipped to Licensee. If any item of Software fails to so function during its warranty period, as the sole remedy Bay Networks will at its discretion provide a suitable fix, patch, or workaround for the problem that may be included in a future Software release. Bay Networks further warrants to Licensee that the media on which the Software is provided will be free from defects in materials and workmanship under normal use for a period of 90 days from the date Software is first shipped to Licensee. Bay Networks will replace defective media at no charge if it is returned to Bay Networks during the warranty

period along with proof of the date of shipment. This warranty does not apply if the media has been damaged as a result of accident, misuse, or abuse. The Licensee assumes all responsibility for selection of the Software to achieve Licensee's intended results and for the installation, use, and results obtained from the Software. Bay Networks does not warrant a) that the functions contained in the software will meet the Licensee's requirements, b) that the Software will operate in the hardware or software combinations that the Licensee may select, c) that the operation of the Software will be uninterrupted or error free, or d) that all defects in the operation of the Software will be corrected. Bay Networks is not obligated to remedy any Software defect that cannot be reproduced with the latest Software release. These warranties do not apply to the Software if it has been (i) altered, except by Bay Networks or in accordance with its instructions; (ii) used in conjunction with another vendor's product, resulting in the defect; or (iii) damaged by improper environment, abuse, misuse, accident, or negligence. THE FOREGOING WARRANTIES AND LIMITATIONS ARE EXCLUSIVE REMEDIES AND ARE IN LIEU OF ALL OTHER WARRANTIES EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Licensee is responsible for the security of its own data and information and for maintaining adequate procedures apart from the Software to reconstruct lost or altered files, data, or programs.

**4. Limitation of liability.** IN NO EVENT WILL BAY NETWORKS OR ITS LICENSORS BE LIABLE FOR ANY COST OF SUBSTITUTE PROCUREMENT; SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES; OR ANY DAMAGES RESULTING FROM INACCURATE OR LOST DATA OR LOSS OF USE OR PROFITS ARISING OUT OF OR IN CONNECTION WITH THE PERFORMANCE OF THE SOFTWARE, EVEN IF BAY NETWORKS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL THE LIABILITY OF BAY NETWORKS RELATING TO THE SOFTWARE OR THIS AGREEMENT EXCEED THE PRICE PAID TO BAY NETWORKS FOR THE SOFTWARE LICENSE.

**5. Government Licensees.** This provision applies to all Software and documentation acquired directly or indirectly by or on behalf of the United States Government. The Software and documentation are commercial products, licensed on the open market at market prices, and were developed entirely at private expense and without the use of any U.S. Government funds. The license to the U.S. Government is granted only with restricted rights, and use, duplication, or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c)(1) of the Commercial Computer Software—Restricted Rights clause of FAR 52.227-19 and the limitations set out in this license for civilian agencies, and subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause of DFARS 252.227-7013, for agencies of the Department of Defense or their successors, whichever is applicable.

**6. Use of Software in the European Community.** This provision applies to all Software acquired for use within the European Community. If Licensee uses the Software within a country in the European Community, the Software Directive enacted by the Council of European Communities Directive dated 14 May, 1991, will apply to the examination of the Software to facilitate interoperability. Licensee agrees to notify Bay Networks of any such intended examination of the Software and may procure support and assistance from Bay Networks.

**7. Term and termination.** This license is effective until terminated; however, all of the restrictions with respect to Bay Networks' copyright in the Software and user manuals will cease being effective at the date of expiration of the Bay Networks copyright; those restrictions relating to use and disclosure of Bay Networks' confidential information shall continue in effect. Licensee may terminate this license at any time. The license will automatically terminate if Licensee fails to comply with any of the terms and conditions of the license. Upon termination for any reason, Licensee will immediately destroy or return to Bay Networks the Software, user manuals, and all copies. Bay Networks is not liable to Licensee for damages in any form solely by reason of the termination of this license.

**8. Export and Re-export.** Licensee agrees not to export, directly or indirectly, the Software or related technical data or information without first obtaining any required export licenses or other governmental approvals. Without limiting the foregoing, Licensee, on behalf of itself and its subsidiaries and affiliates, agrees that it will not, without first obtaining all export licenses and approvals required by the U.S. Government: (i) export, re-export, transfer, or divert any such Software or technical data, or any direct product thereof, to any country to which such exports or re-exports are restricted or embargoed under United States export control laws and regulations, or to any national or resident of such restricted or embargoed countries; or (ii) provide the Software or related technical data or information to any military end user or for any military end use, including the design, development, or production of any chemical, nuclear, or biological weapons.

**9. General.** If any provision of this Agreement is held to be invalid or unenforceable by a court of competent jurisdiction, the remainder of the provisions of this Agreement shall remain in full force and effect. This Agreement will be governed by the laws of the state of California.

Should you have any questions concerning this Agreement, contact Bay Networks, Inc., 4401 Great America Parkway, P.O. Box 58185, Santa Clara, California 95054-8185.

LICENSEE ACKNOWLEDGES THAT LICENSEE HAS READ THIS AGREEMENT, UNDERSTANDS IT, AND AGREES TO BE BOUND BY ITS TERMS AND CONDITIONS. LICENSEE FURTHER AGREES THAT THIS AGREEMENT IS THE ENTIRE AND EXCLUSIVE AGREEMENT BETWEEN BAY NETWORKS AND LICENSEE, WHICH SUPERSEDES ALL PRIOR ORAL AND WRITTEN AGREEMENTS AND COMMUNICATIONS BETWEEN THE PARTIES PERTAINING TO THE SUBJECT MATTER OF THIS AGREEMENT. NO DIFFERENT OR ADDITIONAL TERMS WILL BE ENFORCEABLE AGAINST BAY NETWORKS UNLESS BAY NETWORKS GIVES ITS EXPRESS WRITTEN CONSENT, INCLUDING AN EXPRESS WAIVER OF THE TERMS OF THIS AGREEMENT.



## *Revision Level History*

Revision	Description
A	<b>Initial release.</b>





## About This Guide

Conventions . . . . .	xii
Acronyms . . . . .	xiii
Ordering Bay Networks Publications . . . . .	xiv
Bay Networks Customer Service . . . . .	xiv
How to Get Help . . . . .	xv

## Chapter 1

### Simple Network Management Protocol (SNMP)

SNMP Protocol Overview . . . . .	1-1
SNMP Commands . . . . .	1-2
Using SNMP set to Send Commands to the RAC . . . . .	1-4

## Chapter 2

### Configuring the RAC for SNMP

Configuring the SNMP Agent . . . . .	2-1
Defining the Community String . . . . .	2-3
Defining Trap Hosts and Traps . . . . .	2-4
Defining the Contact String . . . . .	2-6
Defining the Location String . . . . .	2-6
Defining the disabled_modules Parameter . . . . .	2-7
Defining the allow_snmp_sets Parameter . . . . .	2-7

## Chapter 3

### Private MIBs and Standard MIB Support

MIB Object Hierarchy . . . . .	3-2
Understanding MIB Objects . . . . .	3-2
Restrictions on Standard MIBs . . . . .	3-3
RFC 1213 MIB-II Restrictions . . . . .	3-3
RFC 1243 AppleTalk MIB Restrictions . . . . .	3-4
RFC 1389 RIPv2 MIB Restrictions . . . . .	3-5
RFC 1623 Ethernet MIB Restrictions . . . . .	3-5
RFC 1406 DS1 MIB Restrictions . . . . .	3-6
Private Enterprise MIBs . . . . .	3-7
Location of Private MIB Files . . . . .	3-7
Private MIB Filenames . . . . .	3-7

## Chapter 4

### Call Management

Active Call Statistics . . . . .	4-1
Active Modem Calls . . . . .	4-2
Call History Statistics . . . . .	4-3
Modem Call History Statistics . . . . .	4-3
Modem MIBs . . . . .	4-4
Modem Identification . . . . .	4-4
Modem Control . . . . .	4-4
Modem Statistics . . . . .	4-4



**Chapter 5**

**Error Handling**

Error Handling ..... 5-1

    Error Counters ..... 5-2

    Thresholds ..... 5-6

    Proprietary Traps ..... 5-8

**Chapter 6**

**Troubleshooting**

**Index**

Table 1-1. Supported SNMP Commands .....	1-2
Table 2-1. Supported Standard SNMP Traps .....	2-6
Table 3-1. Standard MIBs Supported by the RAC .....	3-1
Table 3-2. RFC 1213 MIB-II Objects .....	3-3
Table 3-3. RFC 1243 AppleTalk .....	3-4
Table 3-4. RFC 1389 RIPv2 MIB Objects .....	3-5
Table 3-5. RFC 1623 Ethernet MIB Objects .....	3-5
Table 3-6. RFC 1406 DS1 MIB Objects .....	3-6
Table 3-7. Private MIB Filenames .....	3-7
Table 3-8. Prefixes for MIB Object Names Related to the RAC .....	3-9
Table 5-1. Current Error Counter MIBs .....	5-2
Table 5-2. Interval Error Counter MIBs .....	5-4
Table 5-3. Total Error Counter MIBs .....	5-5
Table 5-4. Threshold MIBs .....	5-6
Table 5-4. Thresholds (continued) .....	5-7
Table 5-5. Proprietary Traps .....	5-8



## About This Guide







This manual is intended for the person responsible for installation, configuration, and day-to-day administration of the Remote Access Concentrator. For brevity, this manual often uses the acronym *RAC* to refer to the Remote Access Concentrator.

This manual assumes its readers have a basic familiarity with network administration and with the basic concepts of Integrated Services Digital Network (ISDN) and Channelized T1/E1.

If you want	Go to
An overview of the SNMP protocol, and a description of the SNMP commands	<a href="#">Chapter 1</a>
To configure the RAC SNMP agent, define traps and trap hosts, define community strings and other parameters	<a href="#">Chapter 2</a>
A description of exceptions and restrictions placed on standard MIBs by the RAC SNMP agent	<a href="#">Chapter 3</a>
A description of active call statistics, call history statistics, and the private enterprise MIBs for the RAC	<a href="#">Chapter 4</a>
A description of how the RAC handles errors and error reporting	<a href="#">Chapter 5</a>
To troubleshoot problems encountered when using SNMP to communicate with your RAC	<a href="#">Chapter 6</a>

## Conventions

This manual uses the following printing conventions:

Convention:	Represents:
<code>special type</code>	In examples, <code>special type</code> indicates system output.
<b>special type</b>	Bold <b>special type</b> indicates user input.
	In command examples, this notation indicates that pressing  enters the default value.
<b>bold</b>	Bold indicates commands, pathnames, or filenames that must be entered as displayed.
<i>italics</i>	In the context of commands and command syntax, lowercase italics indicate variables for which the user supplies a value.
[ ]	In command dialog, square brackets indicate default values. Pressing  selects this value. Square brackets appearing in command syntax indicate optional arguments.
{ }	In command syntax, braces indicate that one, and only one, of the enclosed value must be entered.
	In command syntax, this character separates the different options available for a parameter.
	Notes provide important information.
	Warnings inform you about conditions that can have adverse effects on processing.
	Cautions notify you about dangerous conditions.

## Acronyms

ASN.1	abstract syntax notation one
BRI	Basic Rate Interface
CCITT	International Telegraph and Telephone Consultative Committee (now ITU-T)
DLCMI	Data Link Control Management Interface
GUI	graphical user interface
HDLC	high-level data link control
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
LAN	local area network
LAT	local area transport
MAC	media access control
MIB	Management Information Base
OID	object identifier
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First (Protocol)
PPP	Point-to-Point Protocol
RAC	Remote Access Concentrator
RFC	Request For Comment
SLIP	Serial Line Interface Protocol
SMDS	Switched Multimegabit Data Service
SMI	Structure of Management
SNMP	Simple Network Management Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
Telnet	Telecommunication Network
TFTP	Trivial File Transfer Protocol
UDP	Unreliable Datagram Protocol
WAN	wide area network

## Ordering Bay Networks Publications

To purchase additional copies of this document or other Bay Networks publications, order by part number from Bay Networks Press™ at the following numbers:

- Phone--U.S./Canada: 888-422-9773
- Phone--International: 510-490-4752
- FAX--U.S./Canada and International: 510-498-2609

The Bay Networks Press catalog is available on the World Wide Web at [support.baynetworks.com/Library/GenMisc](http://support.baynetworks.com/Library/GenMisc). Bay Networks publications are available on the World Wide Web at [support.baynetworks.com/Library/tpubs](http://support.baynetworks.com/Library/tpubs).

## Bay Networks Customer Service

You can purchase a support contract from your Bay Networks distributor or authorized reseller, or directly from Bay Networks Services. For information about, or to purchase a Bay Networks service contract, either call your local Bay Networks field sales office or one of the following numbers:

Region	Telephone number	Fax number
United States and Canada	800-2LANWAN; then enter Express Routing Code (ERC) 290, when prompted, to purchase or renew a service contract  508-916-8880 (direct)	508-916-3514
Europe	33-4-92-96-69-66	33-4-92-96-69-96
Asia/Pacific	61-2-9927-8888	61-2-9927-8899
Latin America	561-988-7661	561-988-7550

Information about customer service is also available on the World Wide Web at [support.baynetworks.com](http://support.baynetworks.com).




## How to Get Help

If you purchased a service contract for your Bay Networks product from a distributor or authorized reseller, contact the technical support staff for that distributor or reseller for assistance.

If you purchased a Bay Networks service program, call one of the following Bay Networks Technical Solutions Centers:

Technical Solutions Center	Telephone number	Fax number
Billerica, MA	800-2LANWAN	508-916-3514
Santa Clara, CA	800-2LANWAN	408-495-1188
Valbonne, France	33-4-92-96-69-68	33-4-92-96-69-98
Sydney, Australia	61-2-9927-8800	61-2-9927-8811
Tokyo, Japan	81-3-5402-0180	81-3-5402-0173





# Chapter 1

## Simple Network Management Protocol (SNMP)

**T**his chapter describes the Simple Network Management Protocol (SNMP) and the SNMP agent provided by the RAC. This chapter includes the following sections:

- SNMP Protocol Overview
- SNMP Commands

### SNMP Protocol Overview

SNMP provides an easy and efficient means of managing the RAC. It operates over the UDP (Unreliable Datagram Protocol), which is part of the TCP/IP protocol suite.

- The Structure of Management Information (SMI), defined in RFC 1155, is a framework that describes what kinds of information can be manipulated using SNMP. Using SMI, objects are placed as nodes in an object tree. The object tree groups logically related objects into subtrees. Each of these subtrees is called a MIB (Management Information Base).
- MIBs located on the SNMP Network Management Station describe the information that is relayed from the agents.
- The SNMP network management station or application can send queries to the SNMP agent located in each RAC.
- Each SNMP agent collects information about its RAC and provides that information to the Network Management Station. The agent process acts as a server in a typical client-server model.

## SNMP Commands

The SNMP agent software in the RAC supports the SNMP commands **get**, **get-next**, **set**, and **trap** as defined in RFC 1157. [Table 1-1](#) describes these commands.

Table 1-1. Supported SNMP Commands

Action	Description
get	Retrieves the value of a specific object from one of the supported MIBs.
get-next	Traverses the MIB tree to retrieve the next object's management information.
set	Modifies the values of MIB objects. The RAC private enterprise MIB and several objects in the standard MIBs allow you to configure the RAC from an SNMP management station on the network rather than using the <b>na</b> utility or CLI <b>admin</b> command.
trap	Asynchronously reports significant events.

When the **allow\_snmp\_sets** parameter is enabled, the RAC accepts SNMP **set** commands from IP host addresses and communities that have read-write access permissions. When disabled, the RAC rejects all SNMP **set** commands; the RAC SNMP agent returns the error *no such name* for the first object in the **set** command (for more details, see *allow\_snmp\_sets* in the *Remote Access Concentrator Software Reference*).

SNMP version 1 is not a secure protocol. SNMP bypasses the RAC's security measures. If security is a concern, the administrator should consider taking the following security measures:

- Define the community strings for specific IP addresses with read-write access.
- Disable all other SNMP access by defining none or read-only access.
- Institute filters on any firewall router to block SNMP traffic from outside the local network. This is good practice in all cases, not just concerning the RAC.

The specifics of using the SNMP commands depend on the management station (see your SNMP management station documentation). The MIB definitions in the files provided in the directory `/annex_root/src/snmp` must be compiled and included in your management station database before you can manage the RAC.

## Using SNMP set to Send Commands to the RAC

The private enterprise MIB objects allow you to change the configuration of the RAC. These configuration changes do not take effect until the RAC is rebooted.

Using the SNMP **set** command, you can broadcast a message, reset a port or subsystem, and reboot the RAC.

- To broadcast a message, use SNMP **set** to write the message to the MIB object **anxBcastMsg** and then **set** the broadcast type to the MIB object **anxBcast**.
- To reset a RAC subsystem, use SNMP **set** to write the desired type (**all**, **macros**, **motd**, **nameserver**, **security**) to the MIB object **anxReset**.
- To reset the RAC, use SNMP **set** to write the desired value to the MIB object **anxReset**.
- To reset a single internal modem, use SNMP **set** to write a value to **mdmCtlReset**.
- To reboot the RAC, **set** the desired image name using the MIB object **anxBootImage** and **set** any boot warning message using the MIB object **anxBootMsg**. For a delayed boot, **set** the boot time using the MIB object **anxBootTime**. Then **set** the boot type using the MIB object **anxBoot**.



To change the RAC's configuration using **set**, SNMP must be enabled at boot time. Make sure the argument **snmp** is not disabled in the **disabled\_modules** parameter. For more details, see *disabled\_modules* in the *Remote Access Concentrator Software Reference*.

You cannot configure filters through SNMP.



## Chapter 2

# Configuring the RAC for SNMP

**B**efore an SNMP network management application can monitor or manage the RAC, the RAC must be configured for SNMP. This entails defining certain configuration data, including the SNMP agent, the SNMP community string, and related RAC parameters. This chapter describes how to configure the RAC for SNMP in the following sections:

- Configuring the SNMP Agent
- Defining the Community String
- Defining Trap Hosts and Traps
- Defining the Contact String
- Defining the Location String
- Defining the `disabled_modules` Parameter
- Defining the `allow_snmp_sets` Parameter

## Configuring the SNMP Agent

Entries in the **gateway** section of the configuration file, which is downloaded during RAC initialization, both enable the SNMP agent and define the operating characteristics of the SNMP daemon that controls the SNMP agent.

The **gateway** section of the configuration file contains four optional keywords for configuring the RAC SNMP agent:

- **community**
- **traphost**
- **contact**
- **location**

Details of these keywords, as well as the parameters you use with SNMP, are described in the following sections.

A sample entry in the **gateway** section of the configuration file looks like this:

```
% gateway
annex 132.245.6.34
    host 132.245.1.01 gateway 132.245.7 metric 1 hardwired
    net 132.245.9.0 gateway 132.245.2.3 metric 1 hardwired
    snmp contact john smith ext 370
    snmp location computer room
end
%include snmp_inc_file
end
```

Note that the example above includes a file named **snmp\_inc\_file**, and executes the commands within that file. It is not necessary to use an include file in this manner. You can simply list all your SNMP commands within the **gateway** section of the configuration file.

It is often convenient to use an SNMP include file to define community names, trap hosts and other SNMP characteristics of the RAC. A sample of what can be included in an SNMP include file is shown below:

```
snmp traphost 123.245.1.2
snmp traphost 132.245.6.50:1049
snmp traphost 132.245.33.233:1048 mycommstring
end
```

In the first trap host entry, neither a port nor a community string is defined. In this case, the port defaults to the well-known trap port 162, and the community string defaults to *public*.

In the second trap host entry, the trap port is specified as 1049. The community string is not specified, and again defaults to *public*.

In the last entry, the trap port is specified as 1048, and the community string is specified as *mycommstring*.



## Defining the Community String

When an SNMP request is received by the agent in the RAC, the agent performs three tests to authenticate the request. The tests are as follows:

- Each SNMP message contains a community string in its header. The receiving SNMP agent tries to match the message's string with an existing community string list. If there is no match, the SNMP agent discards the message without responding to the sender and the MIB-II object **snmpInBadCommunityNames** is incremented.
- When the community string match is found, the sender's IP address is checked against the IP address for the matching community string(s). If there is no match, the SNMP agent discards the message without responding to the sender and the MIB-II object **snmpInBadCommunityNames** is incremented.
- If the community string and the IP address in the SNMP request matches one of the configured community strings, the access mode is checked for that community. If the access is read-write, the SNMP request is processed. If the access is read-only and the SNMP is a **get** or **get next**, the request is processed. In all other cases (access is none or access is read-only and the request is a **set**), an error **noSuchName** is returned to the sender and the MIB-II object **snmpBadCommunity** is incremented.

The keyword **community** defines an SNMP community name from which the RAC responds to requests. At system start-up, the SNMP agent requires at least one community string to be defined in the configuration file. If the file does not contain a community string, the RAC defaults to the community name *public* (unless SNMP is disabled in the parameter **disabled\_modules**).

The SNMP agent authenticates an SNMP request through the use of access permissions. The configuration file format for SNMP defines the IP address and access modes. Security is set by defining community strings that have none, read-only, or read-write access to the MIB variables. The format is shown below:

```
snmp community <string> <IP address> <access>
```

You can use a wild card (\*) to define the IP address. Using a wild card allows anybody with that community string to have access.

You can specify up to ten SNMP community names in the **gateway** section of the configuration file, but each community requires a separate line. The RAC adds these communities to the SNMP agent's community table.

## Defining Trap Hosts and Traps

The RAC employs two methods for defining the host addresses it uses when generating SNMP trap messages.

- The first method defines up to ten static trap hosts using the **anxTrapHostTable** objects in the proprietary MIB. However, the changes you make directly through the MIB are lost when the RAC reboots. If you want your changes to be permanent, you must use the configuration as described below.
- The second method loads the trap hosts (if any) from the configuration file into the Trap Host Table. You can modify this table by adding or deleting trap hosts in the configuration file.

Traps are unsolicited administrative messages generated by SNMP agents on the network. The keyword **traphost** defines the host to which SNMP traps are sent. For the RAC to generate traps, one or more trap host addresses must be defined in the **gateway** section of the configuration file along with the SNMP community string. You can also temporarily add or modify the trap host definitions using the **anxTrapHostTable** objects in the MIB. All generated trap messages use the first community string defined in the configuration file (if the file does not contain a community string, the RAC defaults to *public*).

You can specify up to ten static trap hosts in the configuration file, but each host requires a separate line. Specify a trap host using its IP address (RFC 1157 provides more details on communities and traps). The syntax is:

```
snmp traphost <ipaddr>[:udp port number][community string]
```

You can configure the port number. The following example shows how to use this feature when specifying a trap host.

```
snmp traphost 123.245.1.2
snmp traphost 132.245.6.50:1049 mycommstring
```

In the first trap host entry, neither a port nor a community string is defined. In this case, the port defaults to the well-known trap port 162, and the community string defaults to *public*.

In the second trap host entry, the trap port is specified as 1049, and the community string is specified as *mycommstring*.

[Table 2-1](#) describes the standard SNMP traps supported by the RAC. The proprietary traps and descriptions are provided in [Chapter 5](#).

Table 2-1. Supported Standard SNMP Traps

Trap	Description
coldstart	Upon initialization of the SNMP agent at boot time.
linkUp	Upon initialization of each network interface.
linkDown	Upon de-configuration of any network interface.

## Defining the Contact String

The keyword **contact** defines the object that identifies the person responsible for managing the RAC, as supported by MIB-II. The syntax is:

**snmp contact** *string*

The *string* can include information about how to contact the person, e.g., *M. Law, x 370* (MIB-II object `contact`).

## Defining the Location String

The keyword **location** defines the object that describes the RAC's location; e.g., *computer room*. Specifying this string defines the value of the MIB-II object **sysLocation**.

The syntax is:

**snmp location** *string*

## Defining the `disabled_modules` Parameter

The parameter **`disabled_modules`** allows you to turn off certain features during software initialization (e.g., enter **`lat`**, **`ppp`**, **`slip`** to turn these features off). If you disable SNMP, the RAC discards all SNMP messages it receives. By default, the SNMP agent on the RAC is enabled (for more details, see *disabled\_modules* in the *Remote Access Concentrator Software Reference*).

## Defining the `allow_snmp_sets` Parameter

The RAC's default setting for the **`allow_snmp_sets`** parameter does not permit parameter value changes because the SNMP **`set`** command's header transmits the community string in clear text, which may be a security risk. To modify parameters through SNMP, you must first set **`allow_snmp_sets`** to **`yes`** using the **`na`** utility or the **`admin`** command. You cannot set this parameter using SNMP. If security is a concern, you can take the following measures:

1. **Edit the `%gateway` section of the configuration file for the RAC to define at least one community string with read/write privileges. Change the line:**

```
snmp community public
```

**to:**

```
snmp community am_gui * read-write
```

The RAC accepts SNMP **`sets`** only from sources using the community string **`am_gui`**. You can further restrict the access by including the IP address of the Annex Manager.

```
snmp community am_gui 192.9.200.55 read-write
```

2. **Invoke na, configure the RAC to accept and process SNMP command sets, and exit na:**

```
% na
command: annex 129.9.200.62
command: set annex allow_snmp_sets y
command: quit
```

3. **Enter the selected community string when invoking Annex Manager or Quick2Config Annex.**

# Chapter 3

## Private MIBs and Standard MIB Support

This chapter describes the private enterprise MIBs for the RAC, and lists the exceptions and restrictions placed on standard MIBs by the RAC SNMP agent. This chapter includes:

- MIB Object Hierarchy
- Understanding MIB Objects
- Restrictions on Standard MIBs
- Private Enterprise MIBs

The RAC supports the standard MIBs listed in [Table 3-1](#) with some restrictions.

Table 3-1. Standard MIBs Supported by the RAC

MIB	Defined in...	For information on restrictions...
MIB-II	RFC 1213	See Table 3-2 on page 3-3.
AppleTalk MIB	RFC 1243	See Table 3-3 on page 3-4.
Dot3 Ethernet-like Statistics MIB	RFC 1623	See Table 3-5 on page 3-5.
Rip2 MIB	RFC 1389	See Table 3-4 on page 3-5.
DS1 MIB	RFC 1406	See Table 3-6 on page 3-6.

The Ethernet MIBs defined in RFC 1623 are supported as read objects only. For information about MIB restrictions, see Table 3-5 on page 3-5.



The Capabilities Statement (filename: **xylo.cap**) in the *lannex\_root/src/snmp* directory contains additional information about support for specific MIB objects.

Most parameters do not map to standard MIB objects. Instead, they map to MIB objects in a proprietary (or private enterprise) MIB specific to the RAC and other Remote Annexes. The private MIB also contains objects that provide status and statistics information to the network manager (see [Chapter 4](#)).

## MIB Object Hierarchy

MIBs define the hierarchy of managed objects. MIB objects represent data that the RAC can retrieve or configuration information that it can modify.

## Understanding MIB Objects

RFC 1155 (*Structure and Identification of Management Information for TCP/IP-based Internets*) describes the layout and encoding of exchanged data objects. The SMI uses the ISO standard ASN.1 (Abstract Syntax Notation One) to define a method for describing a hierarchical name space for managed information.

Each object has:

- A name (also referred to as an Object Identifier [OID]).
- A syntax and an encoding. In addition to the basic integer and octet string data types, several special types are defined (e.g., *IP Address*, *Network Address*, *Counter*, *Gauge*, *TimeTicks*). RFC 1212 (*Concise MIB Definitions*) is an easier-to-read form used in most standard MIBs today. It defines the private enterprise MIB.



## Restrictions on Standard MIBs

The SNMP agent does not use all objects in the supported standard MIBs. Also, there may be restrictions on the standard MIB objects that are supported. This section lists the supported standard MIBs and outlines the differences between the RAC parameters and specific standard MIB objects. [Table 3-1](#) lists the supported standard MIBs.

### RFC 1213 MIB-II Restrictions

The RAC supports RFC 1213's *system*, *interfaces*, *at*, *ip*, *icmp*, *tcp*, *udp*, and *snmp* groups. It does not support the *egp* group. In addition, some individual objects have the restrictions outlined in [Table 3-2](#).

Table 3-2. RFC 1213 MIB-II Objects

Object Name	get/set Restrictions	Read Object Limitations
ifAdminStatus	Read only	Returns only <i>up</i> (1) and <i>down</i> (2)
ifOperStatus	None	Returns only <i>up</i> (1) and <i>down</i> (2)
atEntry	Cannot create new rows	None
ipRouteEntry	Cannot create new rows	None
ipRouteProto	None	Returns only <i>local</i> (2), <i>icmp</i> (4), and <i>rip</i> (8)
ipRouteType	None	Returns only <i>invalid</i> (2), <i>direct</i> (3), <i>indirect</i> (4)
ipNetToMediaEntry	Cannot create new rows	None
ipNetToMediaType	Writes only <i>invalid</i> (2), <i>dynamic</i> (3), and <i>static</i> (4)	Returns only <i>dynamic</i> (3) and <i>static</i> (4)

## RFC 1243 AppleTalk MIB Restrictions

The RAC does not support the *llap*, *rtmp*, *kip*, *zip*, and *nbp* groups. It supports the *aarp*, *atport*, *ddp*, and *atecho* groups with the restrictions listed in [Table 3-3](#).

Table 3-3. RFC 1243 AppleTalk

Object Name	Restrictions	Read Object Limitations
atportType	Read only	None
atportNetStart	Not supported	Not applicable
atportNetEnd	Not supported	Not applicable
atportNetAddress	Not supported	Not applicable
atportStatus	Read only	None
atportZone	Read only	None
atportIfIndex	Read only	None
ddpOutRequests	Not supported	Not applicable
ddpInLocalDatagrams	Not supported	Not applicable
ddpNoProtocolHandlers	Not supported	Not applicable
ddpBroadcastErrors	Not supported	Not applicable
ddpShortDDPErrors	Not supported	Not applicable
ddpHopCountErrors	Not supported	Not applicable

## RFC 1389 RIPv2 MIB Restrictions

The RAC supports *rip2GlobalGroup*, *rip2IfStatTable*, and *rip2IfConfTable*. It does not support *rip2PeerTable*. [Table 3-4](#) describes additional restrictions.

Table 3-4. RFC 1389 RIPv2 MIB Objects

Object Name	Restrictions	Read Object Limitations
rip2IfStatStatus	Read only	None
rip2IfConfDomain	Not supported	Not applicable
ripIfConfAuthKey	Not supported	Not applicable
ripIfConfStatus	Read only	None

## RFC 1623 Ethernet MIB Restrictions

The RAC supports RFC 1623's *dot3StatsTable* and *dot3CollTable* with the restrictions outlined in [Table 3-5](#).

Table 3-5. RFC 1623 Ethernet MIB Objects

Object Name	Restrictions	Read Object Limitations
dot3StatsSQETestErrors	Not supported	Not applicable
dot3StatsInternalMac ReceiveErrors	Not supported	Not applicable
dot3StatsEtherChipSet	Read only	None
dot3CollIndex	Not supported	Not applicable

## RFC 1406 DS1 MIB Restrictions

All DS1 MIB objects necessary to configure the RAC PRI interface are supported, but for some objects you are limited to setting default values. The RAC supports this MIB with the restrictions described in [Table 3-6](#).

Table 3-6. RFC 1406 DS1 MIB Objects

Object Name	Restrictions
MIB Tables	
dsx1CurrentTable	Not supported
dsx1IntervalTable	Not supported
dsx1TotalTable	Not supported
DSX1ConfigTable	
dsx1LineType	Not supported; use anxt1dsx1LineType
dsx1TimeElapsed	Not supported
dsx1ValidIntervals	Not supported
dsx1SendCode	Not supported
dsx1CircuitIdentifier	Display string length limited to 128 bytes
dsx1TransmitClockSource	Not supported
dsx1Fd1	Not supported

## Private Enterprise MIBs

The private enterprise MIB file provides the object descriptions for the hardware, software, ports, parameters, and commands groups for all Remote Access Concentrator and Remote Annex products.



RACs support a subset of the private MIB objects. For example, the RAC does not support objects related to parallel ports and async ports.

### Location of Private MIB Files

The private MIBs reside in the `/annex_root/src/snmp` directory.

### Private MIB Filenames

The software distribution kit provides the MIB files listed in [Table 3-7](#). Ask your local system administrator for the location of these MIB files on your system.

Table 3-7. Private MIB Filenames

MIB Filename	Description
xylo.smi	Describes the structure of Bay Networks Remote Access Concentrator MIBs.
xylo-anx.mib	Contains MIB objects related to configuring the Model 8000 RAC and Model 5399 RAC (for example, RAC-wide configuration settings).
xylo-protocol.mib	Contains the protocol-related private MIB groups.
xylo-wan.mib	Contains the MIB objects related to WANs (either PRI or T1).

*(continued on next page)*

Table 3-7. Private MIB Filenames (continued)

MIB Filename	Description
xylo-modem.mib	Contains all the private MIB objects for modem status and configuration.
xylo-callmgmt.mib	Contains the private MIB active call and call history objects.
xylo.trp	Contains the trap definitions for all the private traps.
xylo-trpobj.mib	Contains the trap host table object and all trap threshold objects.

Most of the configuration parameters are provided as objects with read-write access permission in the private enterprise MIB. A number of these parameters are in the standard MIBs that the SNMP agent supports.

Most MIB object names for the parameters in the private enterprise MIB are preceded by the string:

*“.iso.org.dod.internet.private.enterprises.xylogics.annex.”*

## MIB Prefixes

All MIB object names have prefixes that are used to organize them into groups. [Table 3-8](#) lists these prefixes and the corresponding MIB files that contain the MIB objects with these prefixes.



There are other settable MIB objects included in the standard MIBs supported by the SNMP agent. The read-only objects defined in the various MIBs allow the SNMP management station to monitor many MIB variables.

Table 3-8. Prefixes for MIB Object Names Related to the RAC

Prefix	Corresponding MIB File
actcall	xylo-callmgmt.mib
anx	xylo-anx.mib
callhist	xylo-callmgmt.mib
gp	xylo-wan.mib
mdm	xylo-modem.mib
call	xylo-wan.mib
anxt1	xylo-wan.mib
wan	xylo-wan.mib
gsy	xylo-anx.mib
radius	xylo-anx.mib
mdmCall	xylo-callmgmt.mib





# Chapter 4

## Call Management

The RAC maintains call statistics for active calls and call history statistics for previous calls. This chapter describes active call statistics, call history statistics, and the modem MIBs for the RAC.

This chapter includes the following sections:

- Active Call Statistics
- Call History Statistics
- Modem MIBs

### Active Call Statistics

The RAC collects statistics for active calls and makes the information available to management applications through SNMP MIB objects. Active call statistics are also available through the command line interface (CLI). The statistics are used to generate call information for monitoring the current state of the RAC, and for general troubleshooting.

The active call statistics are updated each time you query them, either through the CLI or SNMP. The MIB object names and descriptions for active call statistics are listed in the `activecall` table in the `xylo-callmgmt.mib` file.



This MIB file is included in the software distribution kit. Ask your local system administrator for the location of this MIB file on your system.

The activecall table is indexed by the MIB objects anxt1ChanIndex (1001 for WAN 1, 1002 for WAN 2), and anxt1ChanNumber (1 through the number referenced by the MIB object anxt1TotChan). For example, to get the username for an active call on WAN 1, channel 19, execute the following command (with a MIB browser or other SNMP tool):

```
get actcallusername.1001.19
```



Use anxt1WanIfIndex.1 to get the interface number for WAN 1, and anxt1WanIfIndex.2 to get the interface number for WAN 2.

## Active Modem Calls

A further level of granularity is provided for active modem calls. The mdmCallStatTable in the **xylo-callmgmt.mib** file provides additional statistics specifically for active modem calls (as opposed to active synchronous or TA calls). This table is indexed by the modem number (1 through the number referenced by the MIB object totalmodems).

For example, to get the receive baud rate for an active modem call on modem number 16, execute the following command (with a MIB browser or other SNMP tool):

```
get mdmCallStatRxBaudRate.16
```



When querying for modem statistics, it is important to determine the state of the modem. If the modem state is active, the statistics reflect the value for that current call on a given modem. If the modem state is not active (idle, busied out, failed, etc.), the statistics are not reliable.

## Call History Statistics

The RAC maintains generic call history statistics for terminated calls (both completed calls and calls that failed to connect) and makes the information available to management applications through SNMP MIB objects. The statistics are used to generate call information for accounting purposes and capacity planning. The call history MIB object names and descriptions are listed in the callHistTable in the **xylo-callmgmt.mib** file.



This MIB file is included in the software distribution kit. Ask your local System Administrator for the location of this MIB file on your system.

The objects in the callHistTable are indexed by the MIB object callHistIndex which is a unique index assigned to each call in the order in which it was terminated.

You can configure the number of calls that are logged in the callHistTable with the callHistMaxCalls MIB object. The default value is zero.



The larger the number specified in callHistMaxCalls, the greater the memory resources used by the network management module within the RAC. Each call logged in the callHistTable uses approximately 500 bytes of memory.

## Modem Call History Statistics

A further level of granularity is provided for modem call history statistics. The mdmCallHistAsyTable in the **xylo-callmgmt.mib** file provides additional call history statistics specifically for terminated modem calls (as opposed to terminated synchronous or TA calls). The objects in the mdmCallHistAsyTable are indexed by the MIB object callHistIndex which is a unique index assigned to each call in the order in which it was terminated.

## Modem MIBs

This section describes the three categories of modem MIBs that are located in the **xylo-modem.mib** file. All the tables in this file are indexed by the modem number (1 through number referenced by the MIB object **totalmodems**). The individual MIB objects apply to all modems in the RAC and use zero for the instance.



This MIB file is included in the software distribution kit. Ask your local system administrator for the location of this MIB file on your system.

### Modem Identification

The objects in the `mdmIdTable` identify the hardware and software revisions of the modems.

### Modem Control

There are two MIB objects in the `mdmCtlObjects` group (`mdmCtlResetAll` and `mdmCtlReadConfig`) that apply to all modems. They allow you to reset all modems and read the modem configuration file, respectively. See the MIB object descriptions in the **xylo-modem.mib** file for more details.

The objects in the `mdmCtlTable` reset and set the state of individual modems. See the MIB object descriptions in the **xylo-modem.mib** file for more details.

### Modem Statistics

The objects in the `mdmStatTable` provide cumulative modem statistics for each modem. See the MIB object descriptions in the **xylo-modem.mib** file for more details.



# Chapter 5

## Error Handling

**T**his chapter describes error handling and error reporting.

### Error Handling

The RAC handles errors and error reporting through the use of a number of error counters, thresholds, and traps.

## Error Counters

The RAC makes use of a number of error counter MIB objects (current, interval, and total) for the DS1 WAN interfaces. The counters store the error conditions as described in [Table 5-1](#), [Table 5-2](#), and [Table 5-3](#).

Table 5-1. Current Error Counter MIBs

MIB Object Name	Description
anxt1CurrentIndex	The index value of the DS1 interface for the current interval.
anxt1CurrentOofs	The number of OOF (Out Of Frame) events for the current interval. (An event begins when any two out of four consecutive frame synchronizing bits are received from the network interface are incorrect.) An OOF state ends when reframe occurs.
anxt1CurrentBpvs	The number of bipolar violation errors for the current interval. (Bipolar violation is the occurrence of two consecutive pulses with the same polarity.)
anxt1CurrentCrcs	The number of CRC errors for the current interval. (DS1 signal from incoming call does not agree with DS1 signal from the network.)
anxt1CurrentCs	The number of DS1 frames which are replicated or deleted in the current interval.
anxt1CurrentRnacs	The number of network alarms occurring in the current interval. (This is expressed in seconds with at least one alarm occurring per second.)

*(continued on next page)*

Table 5-1. Current Error Counter MIBs (continued)

MIB Object Name	Description
anxt1CurrentEsfError	The extended superframe errors count in the current interval.
anxt1CurrentLofc	The loss of framing errors count in the current interval.
dsx1CurrentUASs	The number of unavailable seconds encountered by a DS1 interface in the current 15 minute interval.
dsx1CurrentPCVs	The number of path encoding violations encountered by a DS1 interface in the current 15 minute interval.
dsx1CurrentBESs	The number of bursty errored seconds encountered by a DS1 interface in the current 15 minute interval.
dsx1CurrentCSSs	The number of controlled slip seconds encountered by a DS1 interface in the current 15 minute interval.
dsx1CurrentESs	The number of errored seconds encountered by a DS1 interface in the current 15 minute interval.
dsx1CurrentSEFs	The number of severely errored framing seconds encountered by a DS1 interface in the current 15 minute interval.

Table 5-2. Interval Error Counter MIBs

MIB Object Name	Description
anxt1IntervalIndex	The index value of the DS1 interface for the selected interval.
anxt1IntervalNumber	A number between 1 and 96, where 1 is the most recently completed 15 minute interval and 96 is the least recently completed 15 minute interval (this assumes that all 96 intervals are valid).
anxt1IntervalOofs	The number of OOF events for the selected interval. (An event begins when any two out of four consecutive frame synchronizing bits received from the network interface are incorrect.) An OOF state ends when reframe occurs.
anxt1IntervalBpvs	The number of bipolar violation errors for the selected interval. (Bipolar violation is the occurrence of two consecutive pulses with the same polarity.)
anxt1IntervalCrcs	The number of CRC errors for the selected interval. (DS1 signal from incoming call does not agree with DS1 signal from the network.)
anxt1IntervalCs	The number of DS1 frames which are replicated or deleted in the selected interval.
anxt1IntervalRnacs	The number of network alarms occurring in the selected interval. (This is expressed in seconds with at least one alarm occurring per second.)
anxt1IntervalEsfError	The extended superframe errors count in the selected interval.
anxt1IntervalLofc	The loss of framing errors count in the selected interval.



Table 5-3. Total Error Counter MIBs

MIB Object Name	Description
anxt1TotalIndex	The index value of the DS1 interface for the selected interval.
anxt1TotalOofs	The number of OOF events for the selected interval. (An event begins when any two out of four consecutive frame synchronizing bits are received from the network interface are incorrect.) An OOF state ends when reframe occurs.
anxt1TotalBpvs	The number of bipolar violation errors for the selected interval. (Bipolar violation is the occurrence of two consecutive pulses with the same polarity.)
anxt1TotalCrcs	The number of CRC errors for the selected interval. (DS1 signal from incoming call does not agree with DS1 signal from the network.)
anxt1TotalCs	The number of DS1 frames which are replicated or deleted in the selected interval.
anxt1TotalRnacs	The number of network alarms occurring in the selected interval. (This is expressed in seconds with at least one alarm occurring per second.)
anxt1TotalEsfError	The extended superframe errors count in the selected interval.
anxt1TotalLofc	The loss of framing errors count in the selected interval.

## Thresholds

The threshold values are user selectable and range from 0 to 65535. When the MIB counter meets or exceeds the threshold value, the corresponding trap is sent to the trap hosts. Setting the threshold value to 0 disables the corresponding error trap while setting the threshold to a higher number reduces the network traffic. [Table 5-4](#) lists the threshold MIB objects, descriptions, and corresponding error counters. (Error counter MIBs are in parentheses.)

Table 5-4. Threshold MIBs

MIB Object Name	Description
wanBpvThreshold	The threshold which, when met or exceeded, triggers the wanBpvThreshTrap to be sent. (anxt1CurrentBpvs)
wanOofThreshold	The threshold which, when met or exceeded, triggers the wanOofThreshTrap to be sent. (anxt1CurrentOofs)
wanEsThreshold	The threshold which, when met or exceeded, triggers the wanEsThreshTrap to be sent. (dsx1CurrentESs)
wanCvThreshold	The threshold which, when met or exceeded, triggers the wanCvThreshTrap to be sent. (dsx1CurrentPCVs)
wanEsfThreshold	The threshold which, when met or exceeded, triggers the wanEsfThreshTrap to be sent. (anxt1CurrentEsfs)
wanSesThreshold	The threshold which, when met or exceeded, triggers the wanSesThreshTrap to be sent. (dsx1CurrentSEFs)

*(continued on next page)*

Table 5-4. Thresholds (continued)

MIB Object Name	Description
wanUasThreshold	The threshold which, when met or exceeded, triggers the wanUasThreshTrap to be sent. (dsx1CurrentUASs)
wanBesThreshold	The threshold which, when met or exceeded, triggers the wanBesThreshTrap to be sent. (dsx1CurrentBESs)
wanLofcThreshold	The threshold which, when met or exceeded, triggers the wanLofcThreshTrap to be sent. (anxt1CurrentLofcs)
wanCssThreshold	The threshold which, when met or exceeded, triggers the wanCssThreshTrap to be sent. (dsx1CurrentCSSs)
ds0ErrorThreshold	This object defines the threshold for number of consecutive calls that the ds0 fails to accept after which the ds0ErrorTrap is sent to the trap host(s). Setting this object to zero disables the trap.

## Proprietary Traps

The RAC generates various SNMP traps. Some alarm traps are generated when the error counters have reached the error threshold. Clear traps are generated after the associated trap has been sent. By default, all traps are disabled. [Table 5-5](#) lists the proprietary traps and descriptions.

Table 5-5. Proprietary Traps

MIB Object Name	Description
callBeginTrap	This trap is generated when an incoming call is detected or an outbound call is generated by the RAC. This trap is sent to the trap host defined in the configuration file.
callEndTrap	This trap is generated when a call is terminated. The trap is sent to the trap host defined in the configuration file. The trap includes MIB objects as variable bindings. The trap is controlled by the callENDTrapThresh MIB object.
unexpectedDisconnectTrap	This trap is generated when a call is disconnected unexpectedly. A call is considered to disconnect unexpectedly when the one of the following occurs: <i>protocolError</i> <i>localHangup</i> <i>timeoutHDLC</i> <i>maxLogonTimeout</i> Or, when a call is handled by a modem, the unexpected disconnect can be caused by <i>poorSignalQ</i> or <i>failRetrain</i> .

(continued on next page)

Table 5-5. Proprietary Traps (continued)

MIB Object Name	Description
forcedCallDisconnectTrap	This trap is generated when a call is disconnected due to inactivity. The RAC has the following activity timers that trigger this trap: <i>cliInactivityTimeout</i> - The amount of time (in minutes) that the RAC waits before hanging up the call. <i>inactivityTimeout</i> - The amount of time (in minutes) the RAC waits before terminating the call. Uses the value of <i>gpTimerInactivityTimer</i> . <i>netActivityTimeout</i> - Similar to the activity timeout.
modemErrorTrap	This trap is sent to the trap host(s) when a modem failure is detected automatically, the modem is busied out due to a percentage failure threshold over a time period or a consecutive failure threshold.
wanAlarmTrap	This trap is generated when a Red, Yellow, or Blue alarm is detected.
wanAlarmClearTrap	This trap is generated when at least 15 seconds have elapsed without a Red, Yellow, or Blue alarm.
wanBpvThreshTrap	This trap is generated when the number of bipolar violation errors in a 15 minute interval equals or exceeds the threshold defined by <i>wanBpvThreshold</i> .

(continued on next page)

Table 5-5. Proprietary Traps (continued)

MIB Object Name	Description
wanBpvClearTrap	This trap is generated when a 15 minute interval is terminated without the number of bipolar errors meeting or exceeding wanBpvThreshold.
wanOofThreshTrap	This trap is generated when the number of frame errors equals or exceeds the threshold defined by the wanOofThreshold.
wanOofClearTrap	This trap is generated when a 15 minute interval is terminated without the number of frame errors meeting or exceeding wanOofThreshold.
wanEsThreshTrap	This trap is generated when a number of errored seconds errors equals or exceeds the threshold defined by wanEsThreshold.
wanEsClearTrap	This trap is generated when a 15 minute interval is terminated without the number of errored seconds meeting or exceeding wanEsThreshold.
wanCvThreshTrap	This trap is generated when the number of CRC6 errors equals or exceeds the threshold defined by wanCvThreshold.
wanCvClearTrap	This trap is generated when a 15 minute interval is terminated without the number of CRC6 errors meeting or exceeding wanCvThreshold.
wanEsfThreshTrap	This trap is generated when the number of ESF errors equals or exceeds the threshold defined by anxEsfThreshold.

*(continued on next page)*

Table 5-5. Proprietary Traps (continued)

MIB Object Name	Description
wanEsfClearTrap	This trap is generated when a 15 minute interval is terminated without the number of ESF errors meeting or exceeding wanEsfThreshold.
wanSesThreshTrap	This trap is generated when the number of severely errored seconds errors equals or exceeds the threshold defined by wanSesThreshold.
wanSesClearTrap	This trap is generated when a 15 minute interval is terminated without the number of severely errored seconds meeting or exceeding wanSesThreshold.
wanUasThreshTrap	This trap is generated when the number of unavailable seconds errors equals or exceeds the threshold defined by the wanUasThreshold.
wanUasClearTrap	This trap is generated when a 15 minute interval is terminated without the number of unavailable seconds meeting or exceeding wanUasThreshold.
wanBesThreshTrap	This trap is generated when the number of bursty errored seconds errors equals or exceeds the threshold defined by wanBesThreshold.
wanBesClearTrap	This trap is generated when a 15 minute interval is terminated without the number of bursty errored seconds meeting or exceeding wanBesThreshold.

*(continued on next page)*

Table 5-5. Proprietary Traps (continued)

MIB Object Name	Description
wanLofcThreshTrap	This trap is generated when the number of loss of frame count errors equals or exceeds the threshold defined by wanLofcThreshold.
wanLofcClearTrap	This trap is generated when a 15 minute interval is terminated without the number of loss of frame count errors meeting or exceeding wanLofcThreshold.
wanCssThreshTrap	This trap is generated when the number of controlled slip seconds errors equals or exceeds the threshold defined by wanCssThreshold.
wanCssClearTrap	This trap is generated when a 15 minute interval is terminated without the number of controlled slip seconds meeting or exceeding wanCssThreshold.
wanFailWanTrap	This trap is generated when a WAN module fails initialization procedures.
modemUnavailableTrap	This trap is generated when no modems are available to handle an incoming call.
ds0ErrorThresholdTrap	This trap is generated when the number of consecutive failures meets or exceeds ds0ErrorThreshold.



# Chapter 6

## Troubleshooting

This chapter provides answers to the questions frequently asked when you are unable to use SNMP to communicate with the RAC.

- Why am I unable to use SNMP to communicate with the RAC?

In order to enable SNMP communication with a RAC, the RAC SNMP agent must be up and the specified community string must match one of the RAC community strings. Check the configuration file to make sure that your host is not disabled through the community string.

To perform operations which change parameters (e.g., the **set** command), the **allow\_snmp\_sets** parameter on the RAC must be set to **Y** (Yes). You can set this parameter only by using the non-SNMP **na** and **admin** tools, or by using the ROM monitor.

- How can I tell if the SNMP agent is running on the RAC?

SNMP cannot be used to communicate with a RAC unless the RAC is running the SNMP daemon process. You should use the CLI **su** command to go to the **superuser** or **administrative** mode from the direct connection to the RAC, and then use the CLI **proc** command to find a line for the **snmpd** process. For example, this line may read:

```
407  0 S0 7e8d0 103c 7f7f4 12 12  0   18 0:00.017 ? snmpd.
```

If there is no **snmpd** process:

- Check the RAC configuration file to make sure there is at least one SNMP community defined. Reboot the RAC if you change the configuration file.
- Check the **disabled\_modules** parameter using **admin** or **na**. If this parameter indicates that SNMP is disabled, the RAC discards SNMP messages. Remove **snmp** from the list of disabled modules and reboot the RAC.

- Why can't I use an SNMP **set** to change parameters on the RAC?

You may be prevented from changing parameter values based on the value of the **allow\_snmp\_sets** parameter. If **allow\_snmp\_sets** is set to **disabled**, you can use the **telnet** command to reach the RAC and run the superuser CLI **admin** command to set **allow\_snmp\_sets** to **yes**.

The RAC default value does not allow changes to configuration parameters through SNMP. The RAC SNMP agent uses SNMP version 1. The only security check performed by the SNMP agent is to match the community string in an incoming SNMP packet with the defined community strings for the RAC. Since the community string in the SNMP packet is transmitted in the clear (not encrypted), there is a potential security risk in allowing changes to the configuration through SNMP messages. If you wish to configure the RAC using SNMP, please be aware of this situation.

- Why do I see a "Parameter does not exist" message?

Not all parameters apply to all RAC hardware configurations or all software releases. Therefore, you may see this message when you use the **show** or **set** commands.

**A**

active call statistics 4-1  
     active modem calls 4-2  
 allow\_snmp\_sets parameter 1-2, 2-7

**B**

Bay Networks Press xiv

**C**

call history statistics 4-3  
     modem call history statistics 4-3  
 call management 4-1  
 community string  
     defining 2-3  
 configuration parameters  
     global port  
         allow\_snmp\_sets 1-2  
     RAC  
         allow\_snmp\_sets 1-2  
 contact string  
     defining 2-6  
 customer support  
     programs xiv  
     Technical Solutions Centers xv

**D**

disabled\_modules parameter 1-4, 2-7

**G**

gateway  
     entries  
         for SNMP community 2-4  
         for SNMP trap hosts 2-5  
     uses for configuring SNMP agent 2-1

**M**

MIBs 1-1  
     MIB object hierarchy 3-2  
     prefixes for MIB Object Names 3-8

    RAC restrictions on 3-6  
     MIBs supported by RAC  
         AppleTalk MIB, and restrictions 3-4  
         DS1 MIB objects 3-6  
         DS1 MIB, and object restrictions 3-6  
         Ethernet MIB, and object restrictions 3-5  
         MIB-II, and object restrictions 3-3  
         RIPv2 MIB, and object restrictions 3-5  
     modem MIBs 4-4

**P**

printing conventions xii  
 publications, ordering xiv

**S**

SNMP  
     commands 1-2  
         get 1-2  
         get-next 1-2  
         set 1-2  
         using to send commands to RAC 1-4  
     configuring the RAC for 2-1  
     definition 1-1  
     gateway entry for community string 2-4  
     gateways file entry for trap hosts 2-5  
     MIB object hierarchy 3-2  
     overview 1-1  
     SNMP agent configuration  
         defining allow\_snmp\_sets  
             parameter 2-7  
         defining community string 2-3  
         defining contact string 2-6  
         defining disabled\_modules  
             parameter 2-7  
         defining location string 2-6  
         defining trap hosts and traps 2-4  
     supported standard traps 2-6  
     understanding MIB objects 3-2  
     standard MIBs supported by RAC  
         list of 3-1  
         restrictions on 3-3



supported traps 3-1

## T

Technical Solutions Centers xv

trap hosts

    defining 2-4

traps

    supported by RAC 3-1

troubleshooting 6-1

## U

Unreliable Datagram Protocol 1-1