SNMP Management Guide
09.23.2003

9033678

ENTERASYS NETWORKS™

NOTICE

Enterasys Networks reserves the right to make changes in specifications and other information contained in this document and its web site without prior notice. The reader should in all cases consult Enterasys Networks to determine whether any such changes have been made.

The hardware, firmware, or software described in this document is subject to change without notice.

IN NO EVENT SHALL ENTERASYS NETWORKS BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS DOCUMENT, WEB SITE, OR THE INFORMATION CONTAINED IN THEM, EVEN IF ENTERASYS NETWORKS HAS BEEN ADVISED OF, KNEW OF, OR SHOULD HAVE KNOWN OF, THE POSSIBILITY OF SUCH DAMAGES.

Enterasys Networks, Inc.
50 Minuteman Road
Andover, MA 01810

Part Number: 9033678 September 2003

ENTERASYS NETWORKS, INC.
PROGRAM LICENSE AGREEMENT

BEFORE OPENING OR UTILIZING THE ENCLOSED PRODUCT, CAREFULLY READ THIS LICENSE AGREEMENT.

This document is an agreement ("Agreement") between the end user ("You") and Enterasys Networks, Inc. on behalf of itself and its Affiliates (as hereinafter defined) ("Enterasys") that sets forth Your rights and obligations with respect to the Enterasys software program (including any accompanying documentation, hardware or media) ("Program") in the package and prevails over any additional, conflicting or inconsistent terms and conditions appearing on any purchase order or other document submitted by You. "Affiliate" means any person, partnership, corporation, limited liability company, or other form of enterprise that directly or indirectly through one or more intermediaries, controls, or is controlled by, or is under common control with the party specified. This Agreement constitutes the entire understanding between the parties, and supersedes all prior discussions, representations, understandings or agreements, whether oral or in writing, between the parties with respect to the subject matter of this Agreement. The Program may be contained in firmware, chips or other media.

BY INSTALLING OR OTHERWISE USING THE PROGRAM, YOU REPRESENT THAT YOU ARE AUTHORIZED TO ACCEPT THESE TERMS ON BEHALF OF THE END USER (IF THE END USER IS AN ENTITY ON WHOSE BEHALF YOU ARE AUTHORIZED TO ACT, "YOU" AND "YOUR" SHALL BE DEEMED TO REFER TO SUCH ENTITY) AND THAT YOU AGREE THAT YOU ARE BOUND BY THE TERMS OF THIS AGREEMENT, WHICH INCLUDES, AMONG OTHER PROVISIONS, THE LICENSE, THE DISCLAIMER OF WARRANTY AND THE LIMITATION OF LIABILITY. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT OR ARE NOT AUTHORIZED TO ENTER INTO THIS AGREEMENT, ENTERASYS IS UNWILLING TO LICENSE THE PROGRAM TO YOU AND YOU AGREE TO RETURN THE UNOPENED PRODUCT TO ENTERASYS OR YOUR DEALER, IF ANY, WITHIN TEN (10) DAYS FOLLOWING THE DATE OF RECEIPT FOR A FULL REFUND.

IF YOU HAVE ANY QUESTIONS ABOUT THIS AGREEMENT, CONTACT ENTERASYS NETWORKS, LEGAL DEPARTMENT AT (978) 684-1000.

You and Enterasys agree as follows:

1. LICENSE. You have the non-exclusive and non-transferable right to use only the one (1) copy of the Program provided in this package subject to the terms and conditions of this Agreement.

2. RESTRICTIONS. Except as otherwise authorized in writing by Enterasys, You may not, nor may You permit any third party to:

(i) Reverse engineer, decompile, disassemble or modify the Program, in whole or in part, including for reasons of error correction or interoperability, except to the extent expressly permitted by applicable law and to the extent the parties shall not be permitted by that applicable law, such rights are expressly excluded. Information necessary to achieve interoperability or correct errors is available from Enterasys upon request and upon payment of Enterasys' applicable fee.

(ii) Incorporate the Program, in whole or in part, in any other product or create derivative works based on the Program, in whole or in part.

(iii) Publish, disclose, copy, reproduce or transmit the Program, in whole or in part.

(iv) Assign, sell, license, sublicense, rent, lease, encumber by way of security interest, pledge or otherwise transfer the Program, in whole or in part.

(v) Remove any copyright, trademark, proprietary rights, disclaimer or warning notice included on or embedded in any part of the Program.

3. APPLICABLE LAW. This Agreement shall be interpreted and governed under the laws and in the state and federal courts of the Commonwealth of Massachusetts without regard to its conflicts of laws provisions. You accept the personal jurisdiction and venue of the Commonwealth of Massachusetts courts. None of the 1980 United Nations Convention on Contracts for the International Sale of Goods, the United Nations Convention on the Limitation Period in the International Sale of Goods, and the Uniform Computer Information Transactions Act shall apply to this Agreement.

4. EXPORT RESTRICTIONS. You understand that Enterasys and its Affiliates are subject to regulation by agencies of the U.S. Government, including the U.S. Department of Commerce, which prohibit export or diversion of certain technical products to certain countries, unless a license to export the Program is obtained from the U.S. Government or an exception from obtaining such license may be relied upon by the exporting party.

If the Program is exported from the United States pursuant to the License Exception CIV under the U.S. Export Administration Regulations, You agree that You are a civil end user of the Program and agree that You will use the Program for civil end uses only and not for military purposes.

If the Program is exported from the United States pursuant to the License Exception TSR under the U.S. Export Administration Regulations, in addition to the restriction on transfer set forth in Sections 1 or 2 of this Agreement, You agree not to (i) reexport or release the Program, the source code for the Program or technology to a national of a country in Country Groups D:1 or E:2 (Albania, Armenia, Azerbaijan, Belarus, Bulgaria, Cambodia, Cuba, Estonia, Georgia, Iraq, Kazakhstan, Kyrgyzstan, Laos, Latvia, Libya, Lithuania, Moldova, North Korea, the People's Republic of China, Romania, Russia, Rwanda, Tajikistan, Turkmenistan, Ukraine, Uzbekistan, Vietnam, or such other countries as may be designated by the United States Government), (ii) export to Country Groups D:1 or E:2 (as defined herein) the direct product of the Program or the technology, if such foreign produced direct product is subject to national security controls as identified on the U.S. Commerce Control List, or (iii) if the direct product of the technology is a complete plant or any major component of a plant, export to Country Groups D:1 or E:2 the direct product of the plant or a major component thereof, if such foreign produced direct product is subject to national security controls as identified on the U.S. Commerce Control List or is subject to State Department controls under the U.S. Munitions List.

5. UNITED STATES GOVERNMENT RESTRICTED RIGHTS. The enclosed Program (i) was developed solely at private expense; (ii) contains "restricted computer software" submitted with restricted rights in accordance with section 52.227-19 (a) through (d) of the Commercial Computer Software-Restricted Rights Clause and its successors, and (iii) in all respects is proprietary data belonging to Enterasys and/or its suppliers. For Department of Defense units, the Program is considered commercial computer software in accordance with DFARS section 227.7202-3 and its successors, and use, duplication, or disclosure by the Government is subject to restrictions set forth herein.

6. DISCLAIMER OF WARRANTY. EXCEPT FOR THOSE WARRANTIES EXPRESSLY PROVIDED TO YOU IN WRITING BY ENTERASYS, ENTERASYS DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON- INFRINGEMENT WITH RESPECT TO THE PROGRAM. IF IMPLIED WARRANTIES MAY NOT BE DISCLAIMED BY APPLICABLE LAW, THEN ANY IMPLIED WARRANTIES ARE LIMITED IN DURATION TO THIRTY (30) DAYS AFTER DELIVERY OF THE PROGRAM TO YOU.

7. LIMITATION OF LIABILITY. IN NO EVENT SHALL ENTERASYS OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS, PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR RELIANCE DAMAGES, OR OTHER LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM, EVEN IF ENTERASYS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS FOREGOING LIMITATION SHALL APPLY REGARDLESS OF THE CAUSE OF ACTION UNDER WHICH DAMAGES ARE SOUGHT.

THE CUMULATIVE LIABILITY OF ENTERASYS TO YOU FOR ALL CLAIMS RELATING TO THE PROGRAM, IN CONTRACT, TORT OR OTHERWISE, SHALL NOT EXCEED THE TOTAL AMOUNT OF FEES PAID TO ENTERASYS BY YOU FOR THE RIGHTS GRANTED HEREIN.

8. AUDIT RIGHTS. You hereby acknowledge that the intellectual property rights associated with the Program are of critical value to Enterasys and, accordingly, You hereby agree to maintain complete books, records and accounts showing (i) license fees due and paid, and (ii) the use, copying and deployment of the Program. You also grant to Enterasys and its authorized representatives, upon reasonable notice, the right to audit and examine during Your normal business hours, Your books, records, accounts and hardware devices upon which the Program may be deployed to verify compliance with this Agreement, including the verification of the license fees due and paid Enterasys and the use, copying and deployment of the Program. Enterasys' right of examination shall be exercised reasonably, in good faith and in a manner calculated to not unreasonably interfere with Your business. In the event such audit discovers non-compliance with this Agreement, including copies of the Program made, used or deployed in breach of this Agreement, You shall promptly pay to Enterasys the appropriate license fees. Enterasys reserves the right, to be exercised in its sole discretion and without prior notice, to terminate this license, effective immediately, for failure to comply with this Agreement. Upon any such termination, You shall immediately cease all use of the Program and shall return to Enterasys the Program and all copies of the Program.

9. OWNERSHIP. This is a license agreement and not an agreement for sale. You acknowledge and agree that the Program constitutes trade secrets and/or copyrighted material of Enterasys and/or its suppliers. You agree to implement reasonable security measures to protect such trade secrets and copyrighted material. All right, title and interest in and to the Program shall remain with Enterasys and/or its suppliers. All rights not specifically granted to You shall be reserved to Enterasys.

10. ENFORCEMENT. You acknowledge and agree that any breach of Sections 2, 4, or 9 of this Agreement by You may cause Enterasys irreparable damage for which recovery of money damages would be inadequate, and that Enterasys may be entitled to seek timely injunctive relief to protect Enterasys' rights under this Agreement in addition to any and all remedies available at law.

11. ASSIGNMENT. You may not assign, transfer or sublicense this Agreement or any of Your rights or obligations under this Agreement, except that You may assign this Agreement to any person or entity which acquires substantially all of Your stock or assets. Enterasys may assign this Agreement in its sole discretion. This Agreement shall be binding upon and inure to the benefit of the parties, their legal representatives, permitted transferees, successors and assigns as permitted by this Agreement. Any attempted assignment, transfer or sublicense in violation of the terms of this Agreement shall be void and a breach of this Agreement.

12. WAIVER. A waiver by Enterasys of a breach of any of the terms and conditions of this Agreement must be in writing and will not be construed as a waiver of any subsequent breach of such term or condition. Enterasys' failure to enforce a term upon Your breach of such term shall not be construed as a waiver of Your breach or prevent enforcement on any other occasion.

13. SEVERABILITY. In the event any provision of this Agreement is found to be invalid, illegal or unenforceable, the validity, legality and enforceability of any of the remaining provisions shall not in any way be affected or impaired thereby, and that provision shall be reformed, construed and enforced to the maximum extent permissible. Any such invalidity, illegality or unenforceability in any jurisdiction shall not invalidate or render illegal or unenforceable such provision in any other jurisdiction.

14. TERMINATION. Enterasys may terminate this Agreement immediately upon Your breach of any of the terms and conditions of this Agreement. Upon any such termination, You shall immediately cease all use of the Program and shall return to Enterasys the Program and all copies of the Program.

Contents

Figures

Tables

Preface

SNMPV1, SNMPV2C, AND SNMPV3 AGENT OVERVIEW

The SNMP Agent on the XP includes the current IETF defined elements of SNMPv1, SNMPv2c, and SNMPv3 protocols. This document describes how to manage X-Pedition firmware using the Simple Network Management Protocol (SNMP) version 1 defined in RFC 1157.

TIP: SNMP version 1 traps are unsolicited events sent by the firmware to a management system on port udp/162. Traps are unconfirmed events that serve to optimize status collection. The design of reliable management systems should never rely on SNMP traps alone to recognize fault conditions in a network. Polling is required to build a reliable fault management system. Please see Request for Comments (RFC) 1215 and RFC 1224 for details.

BASIC AGENT CONFIGURATION

This section describes how to configure the SNMP agent on an X-Pedition using the Command Line Interface (CLI) of the X-Pedition. More information on the X-Pedition CLI is available in PDF format via http://www.enterasys.com/support/manuals.

Configuring the agent for SNMPv1 (the default mode) means defining at least one SNMP community string. A community string is like a password that identifies the authorization level of the management station. Messages sent to an X-Pedition that contain invalid community strings are discarded and cause the snmpInBadCommunities counter to increment. By default, the SNMP agent is not enabled until you set an SNMP community string via the CLI—there are no default SNMP community strings built into the X-Pedition (they must be installed via the configuration file).

Configuring the agent for SNMPv2c implies the same community string requirements mentioned for SNMPv1. Community based SNMPv2c allows for support of SNMPv2 PDU and data types. You should select this mode of operation if you do not want to use the SNMPv3 secure mode of operation, but need to have access to Counter64 data types or wish to send acknowledged types (notifies). Unlike SNMPv3 which provides authentication and encryption features, SNMPv1 and SNMPv2c are not secure protocols. Messages containing community strings are sent in plain text from manager application to

agent. This means that anyone with a protocol decoder and access to the wire can capture, modify, and replay messages. To prevent such attacks, the X-Pedition requires that you authenticate and encrypt messages sent on the wire (for secure access). When using SNMPv1 or SNMPv2c, it is prudent to protect your network element by applying Access Control Lists (ACL) to the SNMP agent—this will prevent unauthorized access to your network elements and route your SNMP traffic through trusted networks only.

Examples

The following trap configuration commands set the SNMP agent to process messages from the source IP address 10.50.1.1 only—any other source IP address will be dropped. The X-Pedition will process and reply only to SNMP messages that contain the community string "public."

```
snmp set community public privilege read
acl mgmt_only permit udp 207.135.89.1 any any any
acl mgmt_only apply service snmp
```

The next example configures the agent's identity and sets the MIB object sysName to "XP8-3," sysContact to "IT dept," sysLocation to "building 1 closet," and the enterprise sysHwChassisId to "s/n 12345." This last object is a non-standard object defined in the ctron-ssr-hardware MIB and provides a way to specify the serial number for X-Pedition systems that do not have a serial number programmed into static memory. The system group RFC (1907) maps to the commands below.

```
system set name xp8-3
system set contact "IT dept"
system set location "building 1 closet"
snmp set chassis-id "s/n 12345"
```

Security

Simple Network Management Protocol Version 3 (SNMPv3) is an interoperable standards-based protocol for network management. SNMPv3 provides secure access to devices by a combination of authenticating and encrypting packets over the network. The security features provided in SNMPv3 are:

- **Message integrity**. Ensuring that a packet has not been tampered with in-transit.
- **Authentication**. Determining the message is from a valid source.

- **Encryption**. Scrambling the contents of a packet to prevent it from being seen by an unauthorized source.

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy that is set up for a user and the group in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level will determine which security mechanism is employed when handling an SNMP packet. Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. The following table identifies what the combinations of security models and levels mean:

SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	MD5 or SHA-1	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA1 algorithms.
v3	authPriv	MD5 or SHA-1	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA1 algorithms. Provides 56-bit encryption based on the CBC-DES (DES-56) standard in addition to authentication.

Access Control

SNMPv3 also provides fine-granularity access control to management information. The View-Based Access Control Model (VACM) allows subsets of management information to be organized into “views.” There are three types of views: read, write, and notify. Management information that is in a user's view gives the user the corresponding access-level to that management information -- either read, write, or notify. Individual users are organized into groups, and using VACM it is possible to pre-define which views will be available to a given group based on the security model and security level used to request access.

In other words, VACM gives the ability to allow or deny access to any individual item of management information depending on the user's group membership and the level of security provided by the communications channel.

Reliability

In addition to better security and better access control, SNMPv3 also provides a higher degree of reliability for notifying management stations when critical events occur.

Traditionally SNMP agents communicated events to SNMP managers via “traps.” However, if a temporary network problem prevents the reception of the trap by the manager, then the trap is lost. SNMPv3 provides “informs” which are a more reliable form of traps. The SNMP agent initiates the inform process by sending an inform request to the manager. The manger responds to the inform request to acknowledge receipt of the message. If the inform is not received by the manager, the inform request will time-out and a new inform request will be sent. Subsequent inform requests will be sent as previous requests time-out until either an acknowledgement is received from the manager, or until a pre-specified retry-count is reached.

Configuration Overview

Configuring the XP for SNMPv3 support consists of the following steps:

1. **Creating Users**. Users are created and configured to optionally use a secure authentication protocol and/or encryption.
2. **Creating Groups**. Groups are created and assigned a read view, a write view, and a notify view.
3. **Assigning Users**. Users are assigned to their respective groups.
4. **Defining Views**. Views are defined as collections of OID subtrees. Appropriate views are created for the groups that were previously created.

5. **Defining Targets**. Management targets are defined for receiving traps or informs.
6. **Configuring Target Parameters**. Parameters for communicating with designated targets are defined.
7. **Creating Notification Filters**. Optionally, notification filters can be created to constrain the types of traps or informs that a given management target will receive.
8. **Configuring Informs**. Optionally, any targets can be configured to receive SNMPv3 Informs.
9. **Assigning Aliases to Interfaces**. Used to assign additional identification information to any interface handled by the ifXTable (i.e., physical ports, IP interfaces, IPX interfaces, VLANs, and SmartTRUNKs).

SNMP Agent Design Parameters

This section provides parameter limitations for an X-Pedition running the SNMP agent.

- Number of SNMP V1 communities supported: 5 read or read-write
- Number of SNMP V1 Trap targets: 32
- Maximum number of Concurrent PDU processed: 1
- Trap Queue Depth (number of traps waiting to be sent): 64 traps
- Minimum Trap transmit interval: 2 seconds per trap
- Maximum Trap transmit interval: 1 hour (RFC 2115)

Creating Users

Users can be configured to use a combination of authentication and/or privacy. To take advantage of the security features of SNMPv3, it is recommended that users be configured with at least authentication which helps to eliminate most of the potential security risks associated with SNMPv1.

To create a new user, use the **snmp set user** command. For a detailed explanation of the **snmp set user** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*. Most commonly, users will be configured to authenticate with the local SNMP engine only, however, individual users can be configured to authenticate with remote authoritative SNMP engines if necessary (see *Configuring Informs* on page xxvi). The example below illustrates the creation of a new user named “jane” and configured to use the HMAC-SHA-96 authentication protocol with an initial authentication password of “foo.”

```
xp(config)# snmp set user jane engine-id local auth sha1 auth-password foo
```

The next example shows how to configure a user to use both encryption and authentication. In this case the user will be configured to use the HMAC-MD5-96 authentication protocol along with CBC-DES encryption for privacy. In the example the user is configured to use “foo” for the authentication password and “bar” for the privacy password.

```
xp(config)# snmp set user john engine-id local auth md5 auth-password foo
priv des priv-password bar
```

After saving the configuration to make it active, authentication and privacy keys will be generated and localized to the Engine-ID specified (in this case the local Engine-ID). Note that neither the resulting keys nor the original plaintext passwords will be visible in the XP's configuration file. If the user's passwords are lost they cannot be recovered and the user will need to be reconfigured using the **snmp set user** command.

Once your users have been created, the individual users can be grouped to assign access rights based on the level of security the user uses when remotely accessing the XP (see *Creating Groups* on page xx).

Creating Groups

Groups facilitate the assignment of access rights to specific users. Users who require the same level of access should be grouped together into the same group. Different groups can then be created with the necessary access rights. To create groups, use the **snmp set group** command. For a detailed explanation of the **snmp set group** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*. The example below illustrates the creation of the group named “opers.” Based on the example, users belonging to the opers group who are authenticated will have read access to the “restricted” view, no write access, and notify access to the “all” view. Users belonging to the opers group who are not authenticated will have no access.

```
xp(config)# snmp set group opers v3 auth read restricted notify all
```

The next example shows the creation of the “admins” group. In this case, group members will only have partial access if they are authenticated without privacy, but full access if they are authenticated with privacy.

```
xp(config)# snmp set group admins v3 auth read internet notify all
xp(config)# snmp set group admins v3 priv read all write all notify all
```

Both examples make use of several built-in views that are provided with the XP. However, to fully utilize the access-control capabilities of SNMPv3, you should consider defining your own views (see *Defining Views* on page xxii).

Assigning Users

Once users and groups have been created, the users must be assigned to the groups so that the individual users will inherit the access rights of the group. To assign a user to a group, use the **snmp set user-to-group** command. For a detailed explanation of the **snmp set user-to-group** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*. The example below illustrates how to assign the user named "root" to the group named "admins." The user root will then inherit the access rights configured for the "admins" group.

```
xp(config)# snmp set user-to-group root to admins
```

You may also assign an SNMPv1 or SNMPv2c community to a group through the **group** option of the **snmp set community-to-group** command. The following example shows how to assign the "Custom" community to the group "V1Users" and configure the SNMPv1 security model:

```
xp(config)# snmp set community-to-group Custom to V1Users v1
```

The next example illustrates the process of creating a user, creating a group, and assigning the user to the group. First we will create the user "jane" for use with the local SNMP engine, using the HMAC-SHA-96 authentication protocol, and the authentication password "foo". Next the "operators" group will be created with read, write, and notify access to the built-in "all" view for users using the SNMPv3 protocol with authentication enabled. Then the user "jane" will be assigned to the "operators" group.

```
xp(config)# snmp set user jane engine-id local auth sha1 auth-password foo
xp(config)# snmp set group operators v3 auth read all write all notify all
xp(config)# snmp set user-to-group jane to operators
```

This process will give the user "jane" read, write, and notify access to the built-in "all" view when she is successfully authenticated by the local SNMP engine. Note that this example makes use of one of the built-in views. For information on how to define custom views, see *Defining Views* on page xxii.

Defining Views

Views are a collection of subtrees of management information that are used to define a subset of the OID space that a group of users may have access to. There are three types of views: read, write, and notify. Read views give read access to management information, while write views give write access to management information. Notify views allow notifications—either traps or informs—to be sent to a management target. A view needs to be defined only once, and can then be used as any combination of read, write, or notify views when creating groups.

To define a new view, use the **snmp set view** command. For a detailed explanation of the **snmp set view** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*. The example below illustrates how to define a view named "vrrp" that will give access to the 1.3.6.1.2.1.68 subtree (vrrpMIB). In this example, the vrrp view gives access to only the vrrpMIB. All other management information is excluded from the view. Groups can be assigned read, write, and/or notify access to the vrrp view when the groups are created.

```
xp(config)# snmp set view vrrp subtree 1.3.6.1.2.1.68 type include
```

The next example illustrates the use of multiple subtrees and the **exclude** keyword to restrict access to a particular subtree. In this example, "myview" grants access to all of the Internet OID space *excluding* the 1.3.6.1.6.3.18 subtree (snmpCommunityMIB). If assigned to a group, this view can give read, write, and/or notify access to all of the Internet view except the snmpCommunityMIB.

```
xp(config)# snmp set view myview subtree 1.3.6.1 type include
xp(config)# snmp set view myview subtree 1.3.6.1.6.3.18 type exclude
```

Using Masks

The use of masks allows for complex selection of subtrees without needing to specify potentially several dozen **snmp set view** commands. Masks are particularly useful for selecting a particular row from a table. The example below uses the mask 0xff:bf to prevent access to instance 35 of the ifTable. The bits of the mask are used to indicate which bytes of the subtree OID are significant. A one in the mask indicates a significant byte in the OID while a zero indicates an insignificant, or "wild card" byte. Written out in bit notation the mask is: 1111 1111 1011 1111. Notice that the zero (10th bit) matches up with the column header of the ifTable OID (10th byte). Combined with the subtree 1.3.6.1.2.1.2.2.1.1.35, the zero has the effect of selecting ALL columns in the ifTable,

while the trailing ones select ONLY the 35th row of the table. Without masks, this could only be accomplished by entering **snmp set view** commands for each of the 22 columns of the ifTable.

```
xp(config)# snmp set view myview subtree 1.3.6.1.2.1.2.2 type include
xp(config)# snmp set view myview subtree 1.3.6.1.2.1.2.2.1.1.35 mask 0xff:bf
type exclude
```

NOTE: When creating a mask, as in the previous example, there will not always be enough bits to completely fill the last byte. In such a case the remaining bits should be padded with ones.

Defining Targets

Use the **snmp set target** command to define which management targets should receive notifications when events occur. For a detailed explanation of the **snmp set target** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*.

Notifications can be sent in the form of either traps or informs. Traps are unreliable because the receiver does not send acknowledgments when it receives traps. The sender cannot determine if the traps were received. However, an SNMP entity that receives an inform request acknowledges the message with an SNMP response PDU. If the sender never receives the response, the inform request can be sent again. Thus, informs are more likely to reach their intended destination. However, informs consume more resources in the agent and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network.

The example below illustrates how to define an SNMPv1 target named “manager” whose IP address is 10.10.10.10 and can receive traps. Any event that generates a notification will cause a trap to be sent to the target. The XP will use the security-name string “public” to authenticate with the SNMPv1 target.

```
xp(config)# snmp set target manager ip-address 10.10.10.10 security-name
public
```

The next example shows how to define a target named “foo” using SNMPv3 user-based authentication rather than security-name strings. In this context, the **security-name** option

refers to the name of the user which will be used when communicating with the target. This example assumes that the user “jane” already exists.

```
xp(config)# snmp set target foo ip-address 10.10.10.10 v3 auth security-name
jane
```

The **type** option can be used to specify that informs should be sent to the target rather than traps. The sending of informs does require that an SNMPv3 user be created using the proper authoritative SNMP Engine-ID (see *Configuring Informs* on page xxvi for more information). Again this example assumes that the user “jane” has already been created with the appropriate SNMP Engine-ID.

```
xp(config)# snmp set target foo ip-address 10.10.10.10 v3 auth security-name
jane type inform
```

Additionally, a target-params entry can be utilized (see *Configuring Target Parameters* on page xxiv) to apply the same configuration parameters to multiple targets. This can be accomplished via the **param** option. The following example illustrates the use of the **param** option. When used, the security model, security level, and security-name are all obtained from the specified target-params entry. In this case, both “foo” and “bar” will use the same security model, security level, and security-name. This example assumes that the “v3targets” target-params entry already exists.

```
xp(config)# snmp set target foo ip-address 10.10.10.10 param v3targets
xp(config)# snmp set target bar ip-address 10.10.10.11 param v3targets
```

Configuring Target Parameters

As mentioned in the previous section, target-parameters allow several targets to share the same security model, security level, and security-name. This can be useful if all of your targets are going to utilize the same security model. In this way, the security parameters need to only be specified once, and later if a parameter must be changed, it needs to only be changed once rather than needing to be changed for every target. To configure a set of target-parameters to be shared by multiple targets, use the **snmp set target-params** command. For a more detailed explanation of the **snmp set target-params** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*.

The following example shows how to create a set of target-parameters named “global” and make use of those parameters in defining multiple targets. Both targets “foo” and “bar” will utilize the SNMPv1 security model and the “public” security-name string.

```
xp(config)# snmp set target foo ip-address 10.10.10.10 param global
xp(config)# snmp set target bar ip-address 10.10.10.11 param global
xp(config)# snmp set target-params global v1 security-name public
```

Additionally, the **filter** option allows notification filtering to be applied to a group of targets. For more information on notification filters, see *Creating Notification Filters* on page xxv.

Creating Notification Filters

The **snmp set filter** command facilitates the creation of notification filters. Notification filters can prevent specific types of notifications from being sent to a given group of management targets. Once the filter is created using the **snmp set filter** command, it is associated with targets via the **filter** option of the **snmp set target-params** command. For a detailed explanation of the various options available with the **snmp set filter** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*.

The **category** and **subtree** options are used to specify which notifications to filter. The **category** option acts as a shortcut for specifying common notification subtrees. When used with the **category** option, the **subtree** option is not available. As with view subtrees, filter subtrees can also be modified using the **mask** option (see the explanation on masking in *Defining Views* on page xxii for more information).

One important point that should be mentioned is that subtrees that are **included** in a filter will be *filtered out*. In other words, a subtree that is included in the filter will not have its notifications sent to the associated targets. Subtrees that are **excluded** from a filter will be the *only* subtrees that will be left *unfiltered* and have notifications sent.

The example below illustrates the use of a notification filter named “noVRRP” to prevent the target “foo” from receiving any VRRP notifications. Here we use the pre-defined “vrrp” category to specify the OID subtree to include in the filter.

```
xp(config)# snmp set filter noVRRP category vrrp type included
xp(config)# snmp set target foo ip-address 10.10.10.10 param bar
xp(config)# snmp set target-params bar filter noVRRP
```

This next example illustrates the use of multiple subtrees to filter out all notifications except BGP and OSPF notifications. Here, the “routing” filter will prevent the target “foo” from receiving any notifications from sources other than **bgp** and **ospf**.

```
xp(config)# snmp set filter routing category bgp type excluded
xp(config)# snmp set filter routing category ospf type excluded
xp(config)# snmp set target foo ip-address 10.10.10.10 param bar
xp(config)# snmp set target-params bar filter routing
```

Configuring Informs

As mentioned in *Defining Targets* on page xxiii, informs provide a more reliable means by which to notify SNMP managers of events that occur at an SNMP agent. However, informs require that the inform request be sent using the manager's SNMP Engine-ID as the authoritative ID. This means that a user needs to be configured using the manager's Engine-ID as well as the local Engine-ID.

To configure the XP router to send SNMPv3 informs follow the steps outlined below.

1. Create a local user with appropriate notify access.
2. Configure the user with the necessary credentials for authenticating with the manager.
3. Create an SNMPv3 target configured to use the user for authentication.

Following is an example configuration using the user “Informer” for sending inform requests to the SNMP manager with the SNMP Engine-ID `0x00:11:22:33:44:55:66:77:88:99:aa:bb`. The user “Informer” will then be used to authenticate with the manager using the password “foo” when an inform is sent.

```
xp(config)# snmp set user Informer engine-id local
xp(config)# snmp set user Informer engine-id
0x00:11:22:33:44:55:66:77:88:99:aa:bb auth sha1 auth-password foo
xp(config)# snmp set group InformSenders v3 noauth notify all
xp(config)# snmp set user-to-group Informer to InformSenders
xp(config)# snmp set target InformTarget ip-address 10.10.10.10 v3 auth
security-name Informer type inform
```

Note that the user has been configured to work on both the remote manager and on the local system. When communicating with the manager, the user will use the HMAC-SHA-96 authentication protocol, but on the local system the user does not require authentication. However, the user has only been assigned notify access on the local system, so the user account cannot be used to read or write to or from the local system. Additionally, note that the user must also exist in the manager's SNMP configuration database in order for the user to successfully authenticate with the manager.

Configuration Notes

Built-In Users

Out-of-the-box the XP is shipped with a “very-secure” configuration as described in RFC 2574 APPENDIX A. In other words, there are no built-in users configured. This means that at least one user will need to be configured locally using the XP's CLI commands before more users can be created remotely using SNMP and the USM “clone-from” mechanism.

Built-In Views

As noted above, the XP is shipped with no built-in users configured. However, there are built-in views based on the recommended initial configuration for “initial-semi-security-configuration” defined in RFC 2575 APPENDIX A. Since no users exist initially, the XP will still be “very-secure” yet easy to configure since a built-in group and built-in views are already provided. The built-in entries are permanent -- in other words they cannot be deleted. They can, however, be modified to either increase or decrease the security they provide. The entries can be modified by using the CLI commands available on the XP, or remotely by SNMP. If these entries are modified by SNMP, a corresponding CLI command will be saved in the startup configuration.

TRAP CONFIGURATION PROCEDURE

Configuring SNMPv1 traps on the X-Pedition is a two step process. First, you must specify one or more management stations’ IPv4 address (these addresses are referred to as “targets”). For every trap generated in the agent, each target will receive a copy of the trap sent. The second step is to define which traps the target should receive. By default, all traps except for SNMPv1 authentication traps are enabled when you enable a trap target.

In the following example, two trap targets are defined, but only one is active. The X-Pedition can send all supported traps except for SNMPv1 authentication, OSPF, and VRRP. Link Down/Up traps for Ethernet port three on module one are also disabled.

```
snmp set target 10.50.24.55 community public status enable
snmp set target 10.60.21.23 community bgs status disable owner mrm
snmp disable port-trap et.1.3
snmp disable trap authentication
snmp disable trap ospf
snmp disable trap vrrp
```

The X-Pedition sends authentication traps when it receives SNMP packets with invalid community strings. A common security attack on an SNMP agent is to send an invalid message, then capture the authentication trap to learn the community string. When using SNMPv1, turn off authentication traps or use community strings in traps that are different from your read community strings. You may also configure linkUp and linkDown traps per port via the ifXTable's ifLinkUpDownEnabled MIB object, introduced in RFC 2233. Using the **snmp disable trap link-up-down** command will disable linkUp and linkDown traps on all ports. The **snmp disable port-trap et.1.1** command affects only those ports listed.

NOTE: The **snmp disable port-trap** command handles linkUp and linkDown traps only. Other traps are system-wide and not configurable on a per-port basis.

Trap categories include:

```
xp(config)# snmp disable trap ?
  link-up-down      - Link up/down generic trap
  authentication    - Authentication generic trap
  frame-relay       - DLCI up/down trap
  ospf              - sixteen different OSPF traps
  spanning-tree     - newRoot and topologyChange traps
  bgp               - bgpEstablished and bgpBackwardTransition traps
  vrrp              - NewMaster and authFailure traps
  environmentals    - temperature, fan, power supply traps
```

Below is a sample enabled mode report showing the status of an SNMP trap subsystem:

```
xp# snmp show trap
Trap Target Table:
Index    Trap Target Addr   Community String   Status    Port     Owner
1.       10.50.6.4          public             enabled   162      monitor
2.       10.60.21.23        bgs                disabled  162      mrm
Traps by Type:
Authentication trap   :disabled
Frame Relay           :enabled
OSPF                  :disabled
Spanning Tree         :enabled
BGP                   :enabled
VRRP                  :disabled
Environmental         :enabled
Link Up/Down          :enabled
Link Up/Down traps disabled by phisical port:
et.1.3
Trap source address   :10.1.1.1
Trap transmit rate    :1 per 2 seconds
```

Defining Targets

Use the **snmp set target** command to define which management targets should receive notifications when events occur. For a detailed explanation of the **snmp set target** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*.

Notifications can be sent in the form of either traps or informs. Traps are unreliable because the receiver does not send acknowledgments when it receives traps. The sender cannot determine if the traps were received. However, an SNMP entity that receives an inform request acknowledges the message with an SNMP response PDU. If the sender never receives the response, the inform request can be sent again. Thus, informs are more likely to reach their intended destination. However, informs consume more resources in the agent and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network.

The example below illustrates how to define an SNMPv1 target named “manager” whose IP address is 10.10.10.10 and can receive traps. Any event that generates a notification will cause a trap to be sent to the target. The XP will use the security-name string “public” to authenticate with the SNMPv1 target.

```
xp(config)# snmp set target manager ip-address 10.10.10.10 security-name
public
```

The next example shows how to define a target named “foo” using SNMPv3 user-based authentication rather than security-name strings. In this context, the **security-name** option refers to the name of the user which will be used when communicating with the target. This example assumes that the user “jane” already exists.

```
xp(config)# snmp set target foo ip-address 10.10.10.10 v3 auth security-name jane
```

The **type** option can be used to specify that informs should be sent to the target rather than traps. The sending of informs does require that an SNMPv3 user be created using the proper authoritative SNMP Engine-ID (see *Configuring Informs* on page xxvi for more information). Again this example assumes that the user “jane” has already been created with the appropriate SNMP Engine-ID.

```
xp(config)# snmp set target foo ip-address 10.10.10.10 v3 auth security-name
jane type inform
```

Additionally, a target-params entry can be utilized (see *Configuring Target Parameters* on page xxiv) to apply the same configuration parameters to multiple targets. This can be accomplished via the **param** option. The following example illustrates the use of the **param** option. When used, the security model, security level, and security-name are all obtained from the specified target-params entry. In this case, both “foo” and “bar” will use the same security model, security level, and security-name. This example assumes that the “v3targets” target-params entry already exists.

```
router(config)# snmp set target foo ip-address 10.10.10.10 param v3targets
router(config)# snmp set target bar ip-address 10.10.10.11 param v3targets
```

Configuring Target Parameters

As mentioned in the previous section, target-parameters allow several targets to share the same security model, security level, and security-name. This can be useful if all of your targets are going to utilize the same security model. In this way, the security parameters need to only be specified once, and later if a parameter must be changed, it needs to only be changed once rather than needing to be changed for every target. To configure a set of target-parameters to be shared by multiple targets, use the **snmp set target-params** command. For a more detailed explanation of the **snmp set target-params** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*.

The following example shows how to create a set of target-parameters named "global" and make use of those parameters in defining multiple targets. Both targets "foo" and "bar" will utilize the SNMPv1 security model and the "public" security-name string.

```
router(config)# snmp set target foo ip-address 10.10.10.10 param global
router(config)# snmp set target bar ip-address 10.10.10.11 param global
router(config)# snmp set target-params global v1 security-name public
```

Additionally, the **filter** option allows notification filtering to be applied to a group of targets. For more information on notification filters, see *Creating Notification Filters* on page xxv.

Creating Notification Filters

The **snmp set filter** command facilitates the creation of notification filters. Notification filters can prevent specific types of notifications from being sent to a given group of management targets. Once the filter is created using the **snmp set filter** command, it is associated with targets via the **filter** option of the **snmp set target-params** command. For a detailed explanation of the various options available with the **snmp set filter** command, see the *Enterasys X-Pedition Command Line Interface Reference Manual*.

The **category** and **subtree** options are used to specify which notifications to filter. The **category** option acts as a shortcut for specifying common notification subtrees. When used with the **category** option, the **subtree** option is not available. As with view subtrees, filter subtrees can also be modified using the **mask** option (see the explanation on masking in *Defining Views* on page xxii for more information).

One important point that should be mentioned is that subtrees that are **included** in a filter will be *filtered out*. In other words, a subtree that is included in the filter will not have its notifications sent to the associated targets. Subtrees that are **excluded** from a filter will be the *only* subtrees that will be left *unfiltered* and have notifications sent.

The example below illustrates the use of a notification filter named “noVRRP” to prevent the target “foo” from receiving any VRRP notifications. Here we use the pre-defined “vrrp” category to specify the OID subtree to include in the filter.

```
router(config)# snmp set filter noVRRP category vrrp type included
router(config)# snmp set target foo ip-address 10.10.10.10 param bar
router(config)# snmp set target-params bar filter noVRRP
```

This next example illustrates the use of multiple subtrees to filter out all notifications except BGP and OSPF notifications. Here, the “routing” filter will prevent the target “foo” from receiving any notifications from sources other than **bgp** and **ospf**.

```
router(config)# snmp set filter routing category bgp type excluded
router(config)# snmp set filter routing category ospf type excluded
router(config)# snmp set target foo ip-address 10.10.10.10 param bar
router(config)# snmp set target-params bar filter routing
```

Configuring Informs

As mentioned in *Defining Targets* on page xxiii, informs provide a more reliable means by which to notify SNMP managers of events that occur at an SNMP agent. However, informs require that the inform request be sent using the manager's SNMP Engine-ID as the authoritative ID. This means that a user needs to be configured using the manager's Engine-ID as well as the local Engine-ID.

To configure the XP router to send SNMPv3 informs follow the steps outlined below.

1. Create a local user with appropriate notify access.
2. Configure the user with the necessary credentials for authenticating with the manager.
3. Create an SNMPv3 target configured to use the user for authentication.

Following is an example configuration using the user “Informer” for sending inform requests to the SNMP manager with the SNMP Engine-ID `0x00:11:22:33:44:55:66:77:88:99:aa:bb`. The user “Informer” will then be used to authenticate with the manager using the password “foo” when an inform is sent.

```
xp(config)# snmp set user Informer engine-id local
xp(config)# snmp set user Informer engine-id
0x00:11:22:33:44:55:66:77:88:99:aa:bb auth sha1 auth-password foo
xp(config)# snmp set group InformSenders v3 noauth notify all
xp(config)# snmp set user-to-group Informer to InformSenders
xp(config)# snmp set target InformTarget ip-address 10.10.10.10 v3 auth
security-name Informer type inform
```

Note that the user has been configured to work on both the remote manager and on the local system. When communicating with the manager, the user will use the HMAC-SHA-96 authentication protocol, but on the local system the user does not require authentication. However, the user has only been assigned notify access on the local system, so the user account cannot be used to read or write to or from the local system. Additionally, note that the user must also exist in the manager's SNMP configuration database in order for the user to successfully authenticate with the manager.

USER-DEFINED TRAP FEATURE

RMON I (RFC 1757) defines the Alarm and Event group which allows for reliable alarms and more efficient polling. Alarms are said to be more reliable in that they can be logged on the device in an event table that a management station can poll. When you use this MIB, polling is more efficient because the device polls internally, based on a user-defined interval (saving the network from wasting bandwidth in exchange for consuming more cycles and memory). Using RMON I alarms allows you to scale a management station to poll more network elements and reduce the time taken to poll any individual device.

To configure an X-Pedition network element, use RMON I to send a trap and log critical trap conditions for fault management. To do this, use the following steps:

1. Identify Managed Object to monitor by its OID.
2. Configure an Event control row to describe the alarm.
3. Configure an Alarm control row to define alarm condition and polling interval.

Suppose you would like to an X-Pedition to create an event when a module is hot swapped into the chassis or any new IP interface is defined via the configuration file. To begin, locate a suitable Managed Object (e.g., ifTableLastChanged from RFC 2233) and define how often the system should poll that value (e.g., every 5 minutes). The configuration items to add might be as follows:

```
xp# config
xp(config)# rmon event index 15 type both community public \
description “Interface added or Module hot swapped in” owner “help desk
x4155”
xp(config)# rmon alarm index 20 variable 1.3.6.1.2.1.31.1.5.0 interval 300 \
startup both type absolute-value rising-threshold 1 falling-threshold 1 \
rising-index 15 falling-index 15 owner “help desk x4155”
xp(config)# save active
```

HOW THE X-PEDITION AGENT LIMITS THE RATE AT WHICH TRAPS ARE SENT

Subsystems or *tasks* running in the X-Pedition create SNMP traps or informs by first inserting them into a queue managed by the SNMP task. Once a trap exists in the queue, a timer is enabled. This timer goes off every two seconds, at which point the agent will extract (in FIFO order) one trap off the queue and send it to the management station specified; hence, the rate at which traps are sent is one every two seconds.

You may queue up to 64 traps, ranging in size from 50-150 bytes (currently, this value is fixed). An experimental RFC 1224 defined parameters for configuration, but never became standard practice. The **snmp show statistics** command will describe the current state of the queue.

```
xp# snmp show statistics
       -
       2 traps sent
                 0 traps in queue
                 0 traps dropped due to queue overflow
                 0 traps dropped due to send failures
```

If a trap cannot be sent for reasons such as “no route to host,” the X-Pedition exponentially increases the wait time (in seconds) from 2 to 4, and continues to attempt to send the trap up to a total of 8 times. Before dropping a trap, the X-Pedition will display the following error message on the console, once for each trap dropped, specifying the reason for the fail and the destination trap address. Turning off SNMP will clear the trap queue.

```
2002-03-26 21:45:03 %SNMP-E-TRAP, send trap pdu to host “20.21.2.12” failed :
No route to host
```

The Frame Relay MIB (RFC 2115) defines an object for rate limiting trap emissions from a network device. FrTrapMaxRate allows you to set the delay between sending traps—the default value is 2 seconds plus the value defined by the user via SNMP or the CLI (i.e., 2 + **snmp set trap-rate <*seconds*>**).

SETTING THE SNMP AGENT SOURCE IP ADDRESS FOR TRAPS

In firmware releases 3.0 or newer, users can define the trap source address. The SNMP task will bind a UDP port to either an IP interface or to an IP Unicast address. If you specify an interface name from one of the user-defined IP interfaces, the X-Pedition will choose the first non-loopback interface. For instance, the interface *to_admin_net* has two IP addresses defined:

```
interface create ip to_admin_net address-netmask 207.135.88.141/26
interface add ip to_admin_net address-netmask 135.78.23.15/29
```

In a routed network where an X-Pedition is participating in the routing protocols, the following model is typically deployed. To configure a router for SNMP Access so that the management station doesn't have to keep track of all the different interfaces, use an IP Address on the loopback interface as follows.

```
! use address from RFC 1918 (Private addresses)
ip-router global set router-id 10.1.1.1
interface add ip lo0 address-netmask 10.1.1.1/32
! this router has three interfaces and participates in OSPF backbone
interface create to_admin_net address-netmask 207.135.88.121/28
interface create to_mgt_net address-netmask 207.136.88.121/28
interface create to_engr_net address-netmask 207.137.88.121/28
ospf create area backbone
ospf add interface to_admin_net to-area backbone
ospf add interface to_mgt_net to-area backbone
ospf add interface to_engr_net to-area backbone
ospf add stub-host 10.1.1.1 to-area backbone cost 1
ospf start
snmp set community public privilege read
! make traps use the source IP address of the first non-loopback ipaddr
snmp set trap-source lo0
```

A management station configured with routing protocols or talking to a router via a default route that runs routing protocols can now reach the X-Pedition over any to_*_net interface by using the 10.1.1.1 address. Traps sent from the router will use the 10.1.1.1 source IP Address instead of the default behavior (the IP address of the interface chosen to send the trap to the management station—in this case, one of the to_*_net addresses).

USING THE X-PEDITION AS A LAYER-2 SWITCH

When running the X-Pedition as a Layer-2 switch, you may route by creating an IP Interface on a single port or a group of ports, or use an *out-of-band interface* for better security. To create an interface on a set of ports use the following:

```
vlan create UserSubnet port-based id 66
! add all eight ports of slot 1 to this vlan
vlan add ports et.1.1-8 to UserSubnet
interface create ip xpMgtIf address-netmask 10.1.1.1/32 vlan UserSubnet
```

To create a single port VLAN, use the port option to create an interface:

```
interface create ip xpMgtIf address-netmask 10.1.1.1/32 port et.1.4
```

To install an IP Address on the out-of-band port “en0” of the active Control Module:

```
interface add ip en0 address-netmask 10.1.1.1/32
```

DEFINING WHICH MIB MODULES WILL BE ACTIVE

The X-Pedition allows users to enable or disable MIBs in the agent. Depending on how the system is deployed, it may make sense to provide only a smaller set of MIBs for access to management stations. This makes discovery via MIB walks run faster. RFC 1907 SNMPv2-MIB expands on the original system group defined in RFC1213 by providing the sysORTable. The sysORTable lists the MIBs and revision information for each MIB currently exported by the agent to any management station. The CLI also provides a means for viewing this list and, in addition, lists all MIBs that are currently disabled or “offline.”

Detail on how the MIB was implemented is available in the AGENT-CAPABILITIES MIB which defines specific implementation details for each MIB listed in the following table. The file **ssr-agent-capabilities.txt** contains the details specific to a given software revision. The sysORTable’s sysORID object links the agent capability MIB module to a release using a specific object identifier. Applications can use this to adjust their polling/configuration strategy knowing the device’s capabilities explicitly instead of having to do trial sets or gets.

NOTE: Enterasys MIB documents are available on-line at:
http://www.enterasys.com/support/mibs

The **snmp show mibs** command allows you to view this list and, in addition, all MIBs currently disabled or "offline." The following example depicts this capability:

```
xp# snmp show mibs
Supported MIBs
==============
Name                                    Version      Status
----                                    -------      ------
SNMPv2-MIB                              1907         online
EtherLike-MIB                           2358         online
IF-MIB                                  2233         online
IP-MIB                                  2011         online
IP-FORWARD-MIB                          2096         online
UDP-MIB                                 2013         online
TCP-MIB                                 2012         online
BGP4-MIB                                1657         offline
OSPF-MIB                                1850         offline
RIPv2-MIB                               1724         offline
BRIDGE-MIB                              1493+2674    online
FRAME-RELAY-DTE-MIB                     2115         online
PPP-LCP-MIB                             1471         online
PPP-IP-NCP-MIB                          1473         online
PPP-BRIDGE-NCP-MIB                      1474         online
DS1-MIB                                 2495         online
DS3-MIB                                 2496         online
SONET-MIB                               1595         online
ATM-MIB                                 1695         online
RADIUS-AUTH-CLIENT-MIB                  2618         online
RMON-MIB                                1757         online
RMON2-MIB                               2021         online
VRRP-MIB                                2787         online
DVMRP-MIB                               Draft #4     offline
IGMP-MIB                                Draft #5     offline
MAU-MIB                                 2668         online
FDDI-MIB                                1512         online
DEC-ELAN-MIB                            elanv32      online
NOVELL-RIPSAP-MIB                       2/94         online
NOVELL-IPX-MIB                          4/21/94      online
```

Continued on next page....

```
CTRON-CDP-MIB                       8/27/99      online
CTRON-SSR-POLICY-MIB                7/21/99      online
CTRON-SSR-CONFIG-MIB                8/17/99      online
CTRON-SSR-HARDWARE-MIB              12/18/99     online
CTRON-SSR-SERVICE-STATUS-MIB        8/4/98       online
CTRON-SSR-CAPACITY-MIB              11/05/98     online

CTRON-MIB2-EXTENSION                8/28/98      online
CT-CONTAINER-MIB                    3/26/98      online
LAG-MIB                             1/24/00      online
CT-DOWNLOAD-MIB                     2/2/99       online

APPLETALK-MIB-II                    1742         online
SNMP-FRAMEWORK-MIB                  2271         online
SNMP-MPD-MIB                        2272         online
SNMP-NOTIFICATION-MIB               2273         online
SNMP-TARGET-MIB                     2273         online

SNMP-USER-BASED-SM-MIB              2574         online
SNMP-VIEW-BASED-ACM-MIB             2575         online
SNMP-COMMUNITY-MIB                  2576         online
HOST-RESOURCES-MIB                  2790         online
ENTITY-MIB                          2737         online
```

1

Tier I—Critically Managed Object Polling List

1.1 DEVICE-SPECIFIC MANAGED OBJECTS

The managed objects described in this section should be polled every 60 seconds.

1.1.1 Time of operation

MIB Values: sysUpTime (1.3.6.1.2.1.1.3)

Description: The time (in hundredths of a second) since the network management portion of the system last initialized.

Affected areas: All devices attached to this network element—including the network element.

Severity: Critical

Related Trap: coldStart

Solution: Check for router software crash—a “core” file may be present on the PCMCIA card.

1.1.2 Power supply status

MIB values: sysHwPowerSupply (1.3.6.1.4.1.52.2501.1.1.4)

Description: The number and status of power supplies powering the Shelf/Chassis.

Affected areas: Device itself

Severity: Marginal

Related Traps: envPowerSupplyFailed, envPowerSupplyRecovered

Solution: Replace or re-seat power supply. Verify line conditioning (i.e., that the A/C power line is protected with surge and spike compensation).

1.1.3 Fan tray status

MIB values: sysHwFan (1.3.6.1.4.1.52.2501.1.1.5)

Description: The current state of the fans located inside the Shelf/Chassis.

Affected areas: Device itself

Severity: Marginal

Related Traps: envFanFailed, envFanRecovered.

Solution: Replace or clean Fan tray. Verify that the environment meets requirements for dust, humidity, and so forth. Since the XP-2000 does not have hardware support for monitoring tray status, its sysHwFan value is set to unknown(3).

1.1.4 Chassis temperature

MIB values: sysHwTemperature (1.3.6.1.4.1.52.2501.1.1.6)

Description: Operational status of chassis ambient temperature.

Affected areas: Device itself

Severity: Critical

Related Trap: envTempExceeded, envTempNormal.

Solution: If the router's temperature exceeds the normal range, this object will change from normal(1) to outOfRange(2). On XP-2000s, the temperature sensor is not present and the value returned for sysHwTemperature will be unknown(3). If sysHwFan is in state of notWorking(2) and sysHwTemperature changes to outOfRange, the X-Pedition firmware will cause a system reboot. The router may include multiple sensors. If one of the sensors is tripped, the sysHwTemperature object will change from normal (1) to outOfRange (2).

1.1.5 Switching fabric status (XP-8600 only)

MIB values: sysHwSwitchingFabric (1.3.6.1.4.1.52. 2501.1.1.19)

Description: The current state of the switching fabric located inside the Shelf/Chassis.

Affected areas: Device itself

Severity: Marginal

Related Trap: envLineModuleFailure

Solution: Replace switching fabric module.

1.2 TOPOLOGY-RELATED MANAGED OBJECTS

1.2.1 Spanning tree topology change count

MIB values: dot1dStpTopChanges(1.3.6.1.2.1.17.2.4)

Description: The total number of topology changes detected by this bridge since the management entity was last reset or initialized.

Affected areas:

All devices attached to this network element including the network element. Adjacent devices will report linkDown trap and ifOperStatus on links changing to down(1) or lowerLayerDown(7)

Severity: Critical

Related Trap: topologyChange(2), newRoot(1).

Solution: Check Network Element specified as the root bridge for system reload. If the device did not reload, check network links to see if the root bridge was isolated due to a link failure.

Automatically generated actions:

Identify current root bridge and report to this to management. Poll for dot1dStpTimeSinceLTopologyChange to determine exactly when the last topology change occurred and log how often this condition happens. During topology recalculation, broadcast storms may occur due to Layer-2 loops.

1.2.2 Link operational states

MIB values: ifOperStatus (1.3.6.1.2.1.2.2.1.8) for VLAN ifTypes, IP Interfaces or VLANS

Description: Physical or virtual Link Status depends on ifType in ifTable. Per RFC 2233, the current operational state of the interface may be the following.

The testing(3) state indicates that no operational packets can be passed. If ifAdminStatus is down(2) the ifOperStatus should also be down(2). If ifAdminStatus changes to up(1), the ifOperStatus should change to up(1) if the interface is ready to transmit and receive network traffic; it should change to dormant(5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down(2) state if and only if there is a fault that prevents it from going to the

up(1) state; and it should remain in the notPresent(6) state if the interface has missing (typically hardware) components.

Affected areas: All devices attached to ports, upper layer or lower layer systems

Alarm value: The following values represent a non-working link: down (2), lowerLayerDown(7), dormant(5), notPresent(6).

Severity: Critical

Related Traps: linkDown, linkUp.

Solution: Investigate through SNMP Management

Automatically generated actions:

The ifType value of the interface determines what kind of interface the ifOperStatus represents. SNMP management application actions examine the remote end of the link per topology, then query and report ifOperStatus status. The router can verify critical or marginal fault condition based on validation of reachability between key end station devices. On an X-Pedition router, IP Interfaces have ifType ipForward(142). Loss of a logical IP interface tends to be more critical than loss of a physical port (ifType ethernet(6), gigEthernet(117)). The X-Pedition maps IP Interfaces to IP VLANS (137) to zero, one or more physical interfaces. See Appendix B.

1.3 TIER I - TRAP PROTOCOL DATA UNITS (RFC 1215)

1.3.1 coldStart

Enterprise	Trap #
1.3.6.1.4.1.52.3.9.20.1.3	0

Description

The SNMP agent on an X-Pedition has initialized and is operational.

System	Enterprise Value	Defined in
XP-8000 XP-8600 XP-2000	1.3.6.1.4.1.52.3.9.20.1.3 1.3.6.1.4.1.52.3.9.20.1.4 1.3.6.1.4.1.52.3.9.20.1.5	ctron-oids.txt
ER16 SSRM XP-2400	1.3.6.1.4.1.5624.2.1.23 1.3.6.1.4.1.5624.2.1.24 1.3.6.1.4.1.5624.2.1.42	enterasys-oids-mib.txt

Information

This trap is issued when the firmware on the X-Pedition Router has completed initialization (initialization occurs upon power up).

Background Action

Attempt to identify reason for system restart. Typical reasons include:

- Power loss and restoration
- Power supply failed

NOTE: If the Primary CM fails in a dual-CM environment, the Backup CM boots and sends the coldStart trap.

- Operator issued the reboot command
- Firmware crashed (can be verified by checking for crash trace file on PCMCIA flash module)

Related Managed Objects

Use sysUpTime to track system status. Most time stamps in mibs are an instance of this key managed object.

1.3.2 warmStart

Enterprise	Trap #
1.3.6.1.4.1.52.3.9.20.1.3	1

Description
The SNMP agent on an X-Pedition has re-initialized and is operational.

System	Enterprise Value	Defined in
XP-8000 XP-8600 XP-2000	1.3.6.1.4.1.52.3.9.20.1.3 1.3.6.1.4.1.52.3.9.20.1.4 1.3.6.1.4.1.52.3.9.20.1.5	ctron-oids.txt
ER16 SSRM XP-2400	1.3.6.1.4.1.5624.2.1.23 1.3.6.1.4.1.5624.2.1.24 1.3.6.1.4.1.5624.2.1.42	enterasys-oids-mib.txt

Information
This trap is issued when an operator removes the **snmp stop** command from the command line interface. SNMP will re-initialize and send a warmStart trap to indicate this event.

Background Action
When SNMP is enabled during the initial boot sequence, the router sends a coldStart trap.

1.3.3 linkDown

Enterprise Trap
1.3.6.1.4.1.52.3.9.20.1.3 2

Description
SNMP agent detected that a logical or physical link state has transitioned to a downed state. The trap contains the ifIndex of the physical or logical port that is down. IfOperStatus and ifAdminStatus identify how the link went down—by operator request, due to a connection failure caused by disconnecting a cable, or because the device at the other end of the link was disconnected.

Information
An ifIndex is an integer from 1 to 65535 used to represent how a network path is plumbed. See Appendix B for information on how to use the ifTable, ifXTable, and ifStackTable to determine the impact of the downed link. For example, if you combine three physical links to form a SmartTRUNK, the loss of one link does not bring down the logical port.

Background Action

Attempt to identify the ifOperStatus (operational state of the link), the ifType (type of interface—e.g., ethernet, WAN, VLAN, SmartTRUNK), the ifDescr (text description values for a given link), and the ifLastChanged (time stamp of the link change) to record the exact information for the link that went down. Use the ifStackTable to find out to what the port is plumbed—a port may be plumbed to a SmartTRUNK, VLAN, or both.

Related Managed Objects

When you use ifOperStatus in ifTable, you must poll ifLastChange to track the interface link state. See Appendix B for ways to optimize the polling of large numbers of ports on an X-Pedition to provide for better performance and scalability.

1.3.4 linkUp

Enterprise	Trap #
1.3.6.1.4.1.52.3.9.20.1.3	3

Description

The SNMP agent detected that a logical or physical link has transitioned state *up*. The trap contains the ifIndex of the physical port or logical port. IfOperStatus and ifAdminStatus identify whether the link went online via operator command or due to link restoration.

Information

An ifIndex is an integer from 1 to 65535 used to represent how a network path is plumbed. See Appendix B for information on how to use the ifTable, ifXTable, and ifStackTable to determine the impact of the downed link. For example, if you combine three physical links to form a SmartTRUNK, the loss of one link does not bring down the logical port.

Background Action

Attempt to identify the ifOperStatus (operational state of the link), the ifType (type of interface—e.g., ethernet, WAN, VLAN, SmartTRUNK), the ifDescr (text description values for a given link), and the ifLastChanged (time stamp of the link change) to record the exact information for the link that went down. Use the ifStackTable to find out to what the port is plumbed—a port may be plumbed to a SmartTRUNK, VLAN, or both.

Related Managed Objects

When you use ifOperStatus in ifTable, you must poll ifLastChange to track the interface link state. See Appendix B for ways to optimize the polling of large numbers of ports on an X-Pedition to provide for better performance and scalability.

1.3.5 authenticationFailure

Enterprise	Trap #
1.3.6.1.4.1.52.3.9.20.1.3	4

Description

An authenticationFailure trap signifies that the SNMP agent received a protocol message that is not properly authenticated. While implementations of the SNMP must be capable of generating this trap, they must also be capable of suppressing the emission of such traps through an implementation-specific mechanism.

Information

The router sends this trap when an invalid community string is found by the SNMP agent in an SNMP request. This trap indicates that one of the following events occurred:

- A management station was configured incorrectly
- The SNMP agent was configured incorrectly
- An intruder is attempting to access the SNMP agent

Background Action

Verify the address of the intruder using the **snmp show access** command. This reports the IP Addresses and timestamps of the last five SNMP requests.

Related Managed Objects

The snmpInBadCommunityNames MIB counter increments when the X-Pedition agent receives SNMP requests that have invalid community names. You can use snmpInBadCommunityUses to diagnose configuration errors where a community name has read privileges but not read-write privileges.

2

RFC 1493 BRIDGE-MIB

The bridge MIB represents one instance of a Spanning Tree—by default, the first instance. Per VLAN Spanning Tree (PVST) support is available in release 3.0 firmware.

2.0.1 newRoot

Enterprise	Trap #
1.3.6.1.2.1.17	1

Description

The newRoot trap indicates that the topology change timer expired, and that the sending agent is the new root of the Spanning Tree. The X-Pedition sends this trap immediately after the sending agent is elected as the new root.

Information

Layer-2 switches, hubs, and repeaters do not specify a time to live field in the packet. When you connect these types of devices such that a physical loop can occur, *broadcast storms* (duplicate links between bridges that allow redundant paths through the network) occur. Spanning Tree Protocol (STP) attempts to solve broadcast storms by removing loops from the physical topology. By electing one bridge that serves as the root and blocking all upstream links to the root bridge, no loops occur. The bridge with the lowest priority value in a network is the root bridge. When you send this trap, network connectivity will return to normal—prior to sending this trap, however, the network will remain unstable as each 802.1d device attempts to assume the role of bridge and puts all ports in forwarding mode (creating temporary loops in the Layer-2 topology).

Background Action

Verify that the root bridge is a stable, topologically central device in your network. If a root device is on the edge of the topology, spanning tree will take longer to converge. Minimal convergence for standard STP is 30-90 seconds.

Related Managed Objects

dot1dStpTopChanges reports the number of times the Layer-2 topology changed since the system booted. dot1dStpDesignatedRoot indicates the MAC address of the root bridge. If dot1dBaseBridgeAddress does not equal dot1dStpDesignatedRoot, dot1dStpRootPort will indicate the ifIndex of the port that leads to the root bridge or to a device closer to the root bridge.

2.0.2 topologyChange

Enterprise	Trap #
1.3.6.1.2.1.17	2

Description

A topologyChange trap is sent by a device running Spanning Tree when any of its configured ports transitions from the Learning state to the Forwarding state, or from the Forwarding state to the Blocking state. If the topology change causes this device to become the new root bridge, the router sends the newRoot trap only—not the topology change trap.

Information

If Layer-2 loops exist in the network, connectivity for a portion of the network may be impacted during the transition to a new root bridge. If connectivity is interrupted, a link has been lost or the remote end bridge has stopped due to a cycle in power, reconfiguration, or over-utilization.

Background Action

Either a subsidiary bridge or a link has been lost, causing a previously blocked route to the bridge to enter into forwarding state. Check sysUpTime on related bridges and ifOperState on links for utilization and configuration problems.

3

RFC 1850 OSPF-TRAP-MIB

3.0.1 ospfVirtIfStateChange

Enterprise 1.3.6.1.2.1.14.16.2

Trap # 1

Description

An ospfVirtIfStateChange trap signifies a change in the state of an OSPF virtual interface. The X-Pedition generates this trap when an event occurs in the hardware or firmware that changes the interface state. For example, the router generates this trap when the interface state regresses (e.g., changes from Point-to-Point to Down) or progresses to a terminal state (e.g., Point-to-Point).

Information

When a virtual link is lost, an area connected to the backbone via that virtual link may be disconnected from the backbone.

Background Action

Examine the transit area for lost connectivity between the two routers participating in the virtual link.

Related Managed Objects

ospfVirtIfState

3.0.2 ospfNbrStateChange

Enterprise 1.3.6.1.2.1.14.16.2

Trap # 2

Description

An ospfNbrStateChange trap signifies a change in the state of a non-virtual OSPF neighbor. The X-Pedition generates this trap when an event occurs in the hardware or

firmware that changes the interface state. For example, the router generates this trap when the neighbor state regresses (e.g., changes status from Attempt or Full to 1-Way or Down) or progresses to a terminal state (e.g., 2-Way or Full). When a neighboring OSPF device is connected via a physical link, and the physical link transitions to or from the connected state, the designated router generates the trap. A designated router transitioning to a downed state will indicate this change by sending the ospfIfStateChange trap.

Information

In order for two routers on the same subnet to form an adjacency, they must synchronize their link-state databases. When the synchronization is complete, the designated router will send the ospfVirtNbrStateChange trap to the management station that adjacency has transitioned to Full State. Adjacency refers to the relationship established between 2 routers that are not separated by another router—this relationship progresses through states on each router based on messages they exchange. The router also sends this trap when an adjacency changes from Full State to an inferior state. Any state other than Full indicates an out-of-sync link-state database.

Background Action

Examine the connectivity between the two routers on that particular subnet. If you made any configuration changes to one of the routers, make sure that certain parameters like hello interval, router dead interval, and subnet mask are identical on that interface for both routers. If you are still unable to determine why the trap was generated, check the CPU utilization.

Related Managed Objects

ospfNbrEvents

3.0.3 ospfVirtNbrStateChange

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	3

Description

An ospfVirtNbrStateChange trap signifies that there has been a change in the state of an OSPF virtual neighbor. For example, the router generates this trap when the neighbor state regresses (e.g., changes status from Attempt or Full to 1-Way or Down) or progresses to a terminal state (e.g., 2-Way or Full)

Information
In order for two routers on the same subnet to form an adjacency, they must synchronize their link-state databases. When the synchronization is complete, both routers will send the ospfVirtNbrStateChange trap to the management station that adjacency has transitioned to Full State. Adjacency refers to the relationship established between 2 routers that are not separated by another router—this relationship progresses through states on each router based on messages they exchange. The router also sends this trap when an adjacency changes from Full State to an inferior state. Any state other than Full indicates an out-of-sync link-state database.

Background Action
When a virtual link is lost, an area connected to the backbone via that virtual link may disconnect from the backbone.

Related Managed Objects
ospfVirtNbrEvents

3.0.4 ospfIfConfigError

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	4

Description
An ospfIfConfigError trap signifies that a packet was received on a non-virtual interface from a router whose configuration parameters conflict with this router's configuration parameters. Note that the ospfConfigErrorType event of optionMismatch results in a trap only if the mismatch will prevent the adjacency from forming.

Information
Routers on the same subnet should have identical hello, router dead interval, and network mask and authentication values. If these values aren’t the same and the X-Pedition receives an OSPF packet from another router, the X-Pedition will send this trap to the management station.

Background Action
Examine the configuration files on both routers for the parameters specified above.

Related Managed Objects
ospfIfHelloInterval, ospfIfRtrDeadInterval, ospfIfPollInterval, ospfIfAuthType

3.0.5 ospfVirtIfConfigError

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	5

Description
An ospfVirtIfConfigError trap signifies that a packet was received from a virtual interface on a router whose configuration parameters conflict with this router's configuration parameters. Note that the ospfConfigErrorType event of optionMismatch results in a trap only if the mismatch will prevent the adjacency from forming.

Information
Routers talking over the same virtual link should have identical hello, router dead interval, and authentication values. If these values are not the same and a router receives an OSPF packet from another router, the router will send this trap to the management station.

Background Action
Examine the configuration files on both routers for the parameters specified above.

Related Managed Objects
ospfVirtHelloInterval, ospfVirtIfRtrDeadInterval, ospfVirtIfAuthType.

3.0.6 ospfIfAuthFailure

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	6

Description
An ospfIfAuthFailure trap signifies that a packet was received on a non-virtual interface from a router whose authentication key or type conflicts with this router's authentication key or type.

Information
A router received and OSPF packet with an invalid authentication key or type.

Background Action
Examine the router configuration file for an identical key. The key is an OSPF configuration option which identifies a password or *key* used to authenticate messages. If the keys configured on both routers are not the same, the routers will not be able to authenticate protocol messages.

Related Managed Objects
opfIfAuthKey, ospfIfAuthType.

3.0.7 ospfVirtIfAuthFailure

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	7

Description
An ospfVirtIfAuthFailure trap signifies that a packet was received on a virtual interface from a router whose authentication key or type conflicts with this router's authentication key or type.

Information
A router received an OSPF packet with an invalid authentication key or type over a virtual link.

Background Action
Examine the router configuration file for identical authentication key and type.

Related Managed Objects
opfVirtIfAuthType, ospfVirtIfAuthKey

3.0.8 ospfIfRxBadPacket

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	8

Description
The ospfIfRxBadPacket trap signifies that an OSPF packet was received on a non-virtual interface and cannot be parsed.

Information
This trap usually indicates a hardware or software error. The router sending the trap cannot parse the OSPF packet. Check the firmware versions on the routers to ensure that they are using compatible versions of OSPF.

Background Action
Examine version of OSPF running on the router that sent the trap and the router that originated the OSPF packet.

Related Managed Objects
N/A

3.0.9 ospfVirtIfRxBadPacket

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	9

Description
The ospfVirtIfRxBadPacket trap signifies that an OSPF packet was received on a virtual interface and cannot be parsed.

Information
This trap usually indicates a hardware or software error. The OSPF packet could not be parsed by the router sending the trap. Check the firmware versions on the routers to ensure that they are using compatible versions of OSPF.

Background Action
Examine version of OSPF running on the router that sent the trap and the router that originated the OSPF packet.

Related Managed Objects
N/A

3.0.10 ospfTxRetransmit

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	10

Description
The ospfTxRetransmit trap signifies than an OSPF packet was retransmitted on a non-virtual interface. All packets that may be retransmitted are associated with an LSDB entry that includes the LS type, LS ID, and Router ID.

Information
This indicates the receiving router is busy and did not acknowledge receipt of the packet within the required time frame.

Background Action
Look at the link errors and CPU utilization on the *far end router* (i.e., the destination router for OSPF packet transmission) for unusual conditions. Unusual conditions may include numerous link errors or extremely high CPU utilization.

Related Managed Objects
ospfIfRetransInterval, ospfNbrLsRetransQLen.

3.0.11 ospfVirtTxRetransmit

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	11

Description
The ospfTxRetransmit trap signifies than an OSPF packet was retransmitted on a virtual interface. All packets that may be retransmitted are associated with an LSDB entry that includes the LS type, LS ID, and Router ID.

Information
This indicates the receiving router is busy and did not acknowledge receipt of the packet within the required time frame.

Background Action
Look at the link errors and CPU utilization on the *far end router* (i.e., the destination router for OSPF packet transmission) for unusual conditions. Unusual conditions may include numerous link errors or extremely high CPU utilization.

Related Managed Objects
ospfVirtIfRetransInterval

3.0.12 ospfOriginateLsa

Enterprise **Trap #**
1.3.6.1.2.1.14.16.2 12

Description
An ospfOriginateLsa trap signifies that the router originated a new LSA. Typically, you should not invoke this trap for simple refreshes of LSAs (which happen every 30 minutes), only when an LSA (re)originates due to a topology change. This trap does not include LSAs that are flushed because they reached MaxAge.

Information
This indicates network changes—including links on the originating router.

Background Action
This trap indicates both normal and potential error conditions. Check the router and interface specified in the trap for any unusual activity.

Related Managed Objects
ospfOriginateNewLsas

3.0.13 ospfMaxAgeLsa

Enterprise **Trap #**
1.3.6.1.2.1.14.16.2 13

Description
The ospfMaxAgeLsa trap signifies that one of the LSAs in the router's link-state database has reached MaxAge.

Information
A router has an entry in the LSDB that has not been “refreshed” within the max age time-out (1 hour). This can occur when a remote router does not respond or is disconnected.

Background Action
This indicates that some remote router exited abnormally and the LSA was not flushed out.

Related Managed Objects
ospfLsdbAge

3.0.14 ospfLsdbOverflow

Enterprise
1.3.6.1.2.1.14.16.214

Trap

Description
An ospfLsdbOverflow trap signifies that the number of LSAs in the router's link-state database exceeds ospfExtLsdbLimit.

Information
The router has insufficient memory to hold a new LSA.

Background Action
Check memory utilization. If possible change the area to a stub area or *NSSA*.

Related Managed Objects
ospfExternLsaCount, ospfRxNewLsas, ospfExitOverflowInterval.

3.0.15 ospfLsdbApproachingOverflow

Enterprise
1.3.6.1.2.1.14.16.2

Trap
15

Description
The ospfLsdbApproachingOverflow trap signifies that the number of LSAs in the router's link-state database exceeds ninety percent of ospfExtLsdbLimit.

Information
The LSA has reached a user defined (ospfExtLsdbLimit) threshold for number of LSAs in a given router.

Background Action
Increase the ospfExtLsdbLimit or change area to Stub area or NSSA.

Related Managed Objects
ospfExternLsaCount, ospfRxNewLsas

3.0.16 ospfIfStateChange

Enterprise	Trap #
1.3.6.1.2.1.14.16.2	16

Description
The ospfIfStateChange trap signifies a change in the state of a non-virtual OSPF interface. Generate this trap when the interface state regresses (e.g., goes from DR to Down) or progresses to a terminal state (i.e., Point-to-Point, DR Other, DR, or Backup).

Information
If this router was a Designated Router for a given IP interface that changed states to down, the Backup Designated Router (BDR) will assume the role of a Designated Router (DR) for that Interface. Any router with a non-zero priority on that subnet will compete with the remaining routers for the role of BDR.

WARNING: If the DR is lost for a given subnet and there is no BDR elected, the Link State Database (LSDB) Table may not be able to keep up to date with additions and deletions and a loss of connectivity may occur.

Background Action
If there IS a BDR, no functionality (connectivity) is lost. If no BDR exists, assign a non-zero priority to one of the remaining routers on the subnet of the failed interface.

Related Managed Objects
ospfLsdbTable.

4

RFC 1657 BGP-MIB

4.0.1 bgpEstablished

Enterprise
1.3.6.1.2.1.15.7

Trap #
1

Description
The BGP Established event occurs when the BGP FSM enters the ESTABLISHED state.

Information
This indicates that a normal peering relationship was established between two BGP routers.

Background Action
None.

Related Managed Objects
bgpPeerTable's bgpPeerAdminStatus, bgpPeerFsmEstablishedTransitions, and bgpPeerFsmEstablishedTime

4.0.2 bgpBackwardTransition

Enterprise
1.3.6.1.2.1.15.7

Trap #
2

Description
The bgpBackwardTransition event occurs when the BGP FSM moves from a higher numbered state to a lower numbered state.

Information
This trap indicates a lost TCP connection between two peering BGP routers.

Background Action

Check the connectivity between the routers first, then check for configuration changes on the router.

Related Managed Objects

bgpPeerTable's bgpPeerAdminStatus, bgpPeerFsmEstablishedTransitions, and bgpPeerFsmEstablishedTime

5

VRRP-MIB (RFC 2787)

5.0.1 vrrpTrapNewMaster

Enterprise	Trap #
1.3.6.1.2.1.68.0.1	1

Description
The newMaster trap indicates that the sending agent has transitioned from “Backup” state to “Master” state.

Information
The virtual router interface was lost and a new router has assumed Master state. This trap does not indicate lost connectivity, rather that the network is now in a marginal state.

Background Action
Check the connectivity and state of the router that was configured to be the master.

Related Managed Objects
Poll vrrpStatsBecomeMaster to identify when a router became master, then examine vrrpOperMasterIpAddr and vrrpOperPriority to identify the interfaces for which this router became master.

5.0.2 vrrpTrapAuthFailure

Enterprise	Trap #
1.3.6.1.2.1.68.0.2	2

Description
The vrrpAuthFailure trap signifies that a packet was received from a router whose authentication key or type conflicts with this router's authentication key or type. Implementation of this trap is optional.

Information
This trap indicates a configuration error or potential attack.

Background Action
Verify that all routers configured to participate in a virtual router interface are configured with identical authentication keys and types.

Related Managed Objects
Poll vrrpStatsPasswdSecurityViolations to identify this fault condition. Once a fault occurs, retrieve vrrpOperAuthType, vrrpOperAuthKey, and vrrpOperHMACMD5Key to collect configuration information per fault interface and to report configuration errors associated with these values. The configuration on this device must match the configuration on the other VRRP router.

6

CTRON-SSR-TRAP-MIB-V1

6.0.1 envPowerSupplyFailed

Enterprise 1.3.6.1.4.1.52.2501.10

Trap # 1

Description
A power supply on the sending device has failed. The sysHwPowerSupply object identifies the failed supply.

Information
When you configure an X-Pedition is configured with redundant power supplies, the system will load balance them. When you remove a power supply, the supply fails, or there is an interrupt in power to the supply, the router sends this trap.

Background Action
Verify power supply exists in the chassis. On the XP-2000, the secondary power supply is built in.

Related Managed Objects
sysHwPowerSupply

6.0.2 envPowerSupplyRecovered

Enterprise 1.3.6.1.4.1.52.2501.10

Trap # 2

Description
A power supply on the sending device has recovered after a failure. The sysHwPowerSupply object identifies the recovered supply.

Information
A power supply was inserted, powered on, or had power restored to it.

Background Action
None. Indicates normal operation.

Related Managed Objects
sysHwPowerSupply

6.0.3 envFanFailed

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10	3

Description
A Fan tray on the sending device has failed. The sysHwFan object identifies the failed fan tray.

Information
You can remove and replace the Fan trays on the XP-8000/8600s, the XP-2000 requires field service to repair.

Background Action
Visually inspect network element and repair.

Related Managed Objects
sysHwFan

6.0.4 envPowerFanRecovered

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10	4

Description
A Fan tray on the sending device recovered after failure. The sysHwFan object identifies the recovered Fan tray.

Information
The device detected a working fan. This indicates a normal state.

Background Action
None.

Related Managed Objects
sysHwFan

6.0.5 envTempExceeded

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10	5

Description
The temperature inside the chassis of the sending device has exceeded normal operating temperature. The sysHwTemperature object identifies the current temperature status (normal or out of range).

Information
When the X-Pedition exceeds its normal operating temperature, it will send the environmental temperature exceeded trap, write a message to the console/syslog, then cycle the power.

Background Action
Verify environmental conditions of the area enclosing the X-Pedition to verify that the router has sufficient ventilation.

Related Managed Objects
sysHwTemperature

6.0.6 envTempNormal

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10	6

Description
The temperature inside the chassis on the sending device has returned to normal operating temperature. The sysHwTemperature object identifies the current temperature status (normal or out of range).

Information
Normal operating temperature has been restored.

Background Action
None

Related Managed Objects
sysHwTemperature

6.0.7 envHotSwapIn

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10	7

Description
A line module, switch fabric (XP-8000 and ER16), or Backup Control Module was inserted into the chassis. sysHwModuleSlotNumber identifies the slot that contains the module.

Information
A module was inserted into a chassis on the network. The system detected it and powered it up.

Background Action
Review the sysHwModuleTable variable in the ctron-ssr-hardware.txt MIB to determine the card type and port information.

Related Managed Objects
sysHwModuleTable, sysHwPortTable, ifTable, ifXTable, ifStackTable, ifTableLastChange, ifStackTableLastChange.

6.0.8 envHotSwapOut

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10	8

Description
A module was powered off (and possibly removed) from the chassis. sysHwModuleSlotNumber identifies the slot that contained or powered down the module.

Information
A card was inserted into a chassis—the system detected it and powered it up.

Background action
Review the sysHwModuleTable in ctron-ssr-hardware.txt to determine the module's current state. With the module hot swapped out, the ifTable ifOperStatus will change to noPresent for physical ports on the module and any VLANs, IP interfaces, or SmartTRUNKs configured on this module will transition to ifOperState of lowerLayerDown.

Related managed objects
sysHwModuleTable, sysHwPortTable, ifTable, ifXTable, ifStackTable, ifTableLastChange, and ifStackTableLastChange.

6.0.9 envBackupControlModuleOnline

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10	9

Description
The Backup Control Module has assumed the role of the Primary Control Module (this trap may indicate a degradation in redundancy). A firmware crash or hardware failure occurred on the Primary Control Module—polling the sysHwControlModuleBackupState will indicate which of these events occurred. This trap occurs after the router sends a coldStart trap.

Information
When operating the X-Pedition in a Dual CM environment, the router will launch the Backup CM and keep it on standby. If heartbeat messages from the Primary CM stop coming for at least 20 seconds, the Primary CM will fail over and the Backup CM will take control of the router.

Except in the case of a CM fail-over, slot CM/0 contains the Primary Control Module and slot CM/1 contains the Backup Control Module. If slot CM/1 does not contain a Control Module or if some other line card resides in the slot, the router will mark the state of this object as not installed.

States for object sysHwControlModuleBackupState are as follows:

unknown(1)	Status unavailable to SNMP agent
inactive(2)	Backup Control Module is offline
standby(3)	CM in backup slot CM/1 is standby
notInstalled(4)	No Backup CM installed in Slot: CM/1
active(5)	CM in backup slot CM/1 is active

Background action

Reread the object sysHwControlModuleBackupState in ctron-ssr-hardware.txt to determine current state of the former Primary Control Module. If failure was a software crash, the state should become standby(1). If the former Primary Control Module was unable to boot its firmware image, the state becomes inactive. Verify that the firmware image and the configuration file exist on the device. A state of notInstalled may signal either the module has been removed or has failed at a hardware level.

Related managed objects

sysHwModuleTable, sysHwControlModuleBackupState, and sysUpTime.

6.0.10 envBackupControlModuleFailure

Enterprise

1.3.6.1.4.1.52.2501.10

Trap

10

Description

A Backup Control Module that was on standby has changed to inactive or notInstalled. Poll sysHwControlModuleBackupState for the current state of the Backup Control Module.

Information
States for object sysHwControlModuleBackupState are as follows:

unknown(1)	Status unavailable to SNMP agent
inactive(2)	Backup Control Module is offline
standby(3)	CM in backup slot CM/1 is standby
notInstalled(4)	No Backup CM installed in Slot: CM/1
active(5)	CM in backup slot CM/1 is active

Background action
Verify that the Backup CM has not been removed. If the Backup CM is installed, it has failed and must be replaced.

Related managed objects
sysHwModuleTable, sysHwControlModuleBackupState, and sysUpTime.

6.0.11 envLineModuleFailure

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10	11

Description
A line module error condition was detected and the module has changed to an offline status.

Information
This is an indication of a hardware failure or, if the line module contains a processor, a software failure. The router does not generate this trap when a user hot swaps a line module—instead, it generates the envHotSwapOut trap.

Background action
Verify that the module has not been removed. If the module is installed, it is not functioning properly and should be replaced with a working module.

Related managed objects

sysHwModuleTable and sysUpTime

6.0.12 envCPUThresholdExceeded

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10	12

Description

The CPU utilization has exceeded the value of capCPUMaxThreshold. After this trap occurs, it will not re-occur until the utilization drops below capCPUMinThreshold. Poll capCPUMinThreshold and capCPUMaxThreshold to determine the configured threshold settings.

Information

The current level of activity on the router caused the CPU utilization to exceed the maximum threshold configured.

Background action

Monitor capCPUCurrentUtilization to see if the utilization is still above the threshold. If the utilization reaches 100%, it may not be possible to receive timely responses to SNMP queries. Check for abnormal network traffic activity. This could be caused by network misconfiguration or Denial of Service (DOS) attacks. Verify that the capCPUMaxThreshold and capCPUMinThreshold values are proper for the expected CPU utilization which results due to the traffic on your network.

Related managed objects

capCPUCurrentUtilization and sysUpTime

6.0.13 polAclDenied

Enterprise	Trap #
1.3.6.1.4.1.52.2501.10.3	1

Description

The polAclDenied trap indicates that a message was dropped due to a “deny” ACL. The polAclName and ifIndex values identify the ACL and the interface from which it was invoked.

7

Tier II—Functional MIB Objects

You may poll these MIB variables at 15 minute intervals—you can perform basic security, systems capacity, and configuration checks here.

7.0.1 System Status

MIB values: ifAdminStatus (1.3.6.1.2.1.2.2.1.7)

Description: Desired Port Status. The desired state of the interface. The testing (3) state does not allow you to pass operational packets.

Sample period: N/A

Affected areas: All devices attached to ports

Alarm value: N/A

Severity: Warning

Solution: Use TACACS or RADIUS to identify the ID of the user who last changed the configuration to administratively bring down the interface. The **system show active** and **system show telnet-access** commands will report the method and time of the last configuration change.

7.0.2 SNMP Authentication Failure

MIB values: snmpInBadCommunityNames (1.3.6.1.2.1.11.4)

Description: The total number of SNMP Messages delivered to the SNMP protocol entity that use an SNMP community name not known to the entity. You can use the snmpEnableAuthenTraps object or the SNMP disable trap authentication command to enable or disable traps of this type.

NOTE: Sending these traps is one approach used to hack into a system if the attacker knows that sending a bad set/get request will cause the router to send an authentication trap from which he can extract an SNMP community string. Secure systems do not allow you to send authentication traps.

Sample period: 86,400 seconds (24 hours)

Affected areas: Security

Severity: Informational

solution: check agent and management station for identical community strings.

7.0.3 Router is Unable to Route Some IP Data

MIB values: ipOutNoRoutes (1.3.6.1.2.1.4.12)

Description: The number of IP datagrams discarded because no route was found to transmit them to their destination(s). This counter includes any packets counted in ipForwDatagrams which meet the “no-route” criterion and any datagrams a host cannot route because its default gateways are down.

Sample period: 1 hour

Affected areas: None

Severity: Marginal

Solution: If delta (the change in the value between sampling periods) reports a high number, check the routing tables for potential errors and for clients requesting access to unavailable or un-reachable networks (IcmpOutDestUnreachs).

7.0.4 Router is Dropping Valid IP Datagrams

MIB values: ipRoutingDiscards (1.3.6.1.2.1.4.23)

Description: The number of valid routing entries chosen to discard. Discarding such entries can free up buffer space for other routing entries.

Sample period: 1 hour

Affected areas: Routing IP Data

Severity: Critical

Solution: If delta (the change in the value between sampling periods) reports a high number, check to see why flows are routing to CPU (Layer-3 table misses). IP Datagrams destined for the CPU are either new flows (to learn) or control protocols such as ICMP, UDP/OSPF, UDP/RIP, or UDP/SNMP. IP datagram loss may impact router connectivity.

7.0.5 Router is Failing to Reassemble IP Datagrams

MIB values: ipReasmFails (1.3.6.1.2.1.4.16)

Description: The number of failures detected by the IP reassembly algorithm (e.g., timed out, errors). Note, this does not represent a count of all discarded IP fragments. Some algorithms, notably the algorithm in RFC 815, can lose track of the number of fragments.

Sample period: 1 hour

Affected areas: Routing UDP/IP Data

Severity: Critical

solution: If delta (the change in the value between sampling periods) reports a high number, check to see where the datagrams are being sent from. ipAdEntReasmMaxSize defines the max size of an IP Packet per interface.

7.0.6 Router is Dropping UDP Datagrams

MIB values: udpInErrors (1.3.6.1.2.1.7.3)

Description: The number of received UDP datagrams that were not delivered for reasons other than the lack of an application connected as a listener on the UDP port or because the incoming data rate exceeds the router's capacity to process or buffer messages.

Sample period: 1 hour

Affected areas: Routing UDP/IP Data

Severity: Critical

solution: If delta (the change in the value between sampling periods) report a high number, check to see why the CPU is processing UDP datagrams. Some routing protocols such as OSPF and RIP use UDP datagrams to carry control traffic. UDP Datagram loss may lead to network connectivity problems.

7.0.7 Current CPU Utilization

MIB values: capCPUCurrentUtilization(1.3.6.1.4.1.52. 2501.2.1.1.1)

Description: The current CPU Utilization of the Control Module specified.

Sample period: 1 hour

Affected areas: The current device

Severity: Critical

Related Trap: None

Solution: Verify the cause of the high utilization and configure the X-Pedition appropriately.

7.0.8 Current Layer-2 Learning Rate

MIB values: capCPUL2LearningRate(1.3.6.1.4.1.52. 2501.2.1.1.5)

Description: The current Layer-2 Learning Rate.

Sample period: 1 hour

Affected areas: The current device

Severity: Marginal

Related Trap: None

Solution: Verify the cause of the high learning rate. You may consider adjusting the aging interval.

7.0.9 Current Layer-2 Aging Rate

MIB values: capCPUL2AgingRate(1.3.6.1.4.1.52. 2501.2.1.1.6)

Description: The current Layer-2 aging rate.

Sample period: 1 hour

Affected areas: The current device

Severity: Marginal

Related Trap: None

Solution: Verify the cause of the high aging rate. You may consider adjusting the aging interval.

7.0.10 Current Layer-3 Learning Rate

MIB values: capCPUL3LearningRate(1.3.6.1.4.1.52. 2501.2.1.1.2)

Description: The current Layer-3 learning rate.

Sample period: 1 hour

Affected areas: The current device

Severity: Marginal

Related Trap: none

Solution: Verify the cause of the high learning rate. You may consider adjusting the aging interval.

7.0.11 Current Layer-3 Aging Rate

MIB values: capCPUL3AgingRate(1.3.6.1.4.1.52. 2501.2.1.1.3)

Description: The current Layer-3 aging rate.

Sample period: 1 hour

Affected areas: The current device

Severity: Marginal

Related Trap: none

Solution: Verify the cause of the high aging rate. You may consider adjusting the aging interval.

7.0.12 Current Layer-3 Collision Rate

MIB values: capCPUL3HashCollisions(1.3.6.1.4.1.52. 2501.2.1.1.4)

Description: The current Layer-3 Hash Collision Rate.

Sample period: 1 hour

Affected areas: Device itself

Severity: Marginal

Related Trap: none

Solution: Adjust the hashing algorithm to minimize collisions.

7.0.13 Current NIA Receive Rate

MIB values: capCPUNIAReceiveRate(1.3.6.1.4.1.52. 2501.2.1.1.7)

Description: The current rate at which the CPU is receiving frames.

Sample period: 1 hour

Affected areas: Device itself

Severity: Marginal

Related Trap: none

Solution: Verify what is causing the high arrival rate at the CPU and possibly adjust the aging intervals or hash algorithm.

7.0.14 Current NIA Transmit Rate

MIB values: capCPUNIATransmitRate(1.3.6.1.4.1.52. 2501.2.1.1.8)

Description: The current rate at which the CPU is transmitting frames. This includes frames for management and routing protocols.

Sample period: 1 hour

Affected areas: The current device

Severity: Marginal

Related Trap: None

Solution: Verify the cause of the high transmit rate from the CPU (e.g., a high number of routing and switching functions are enabled and in use by a large number of users).

8

RMON I/II Support in the X-Pedition

8.1 IMPLEMENTATION DETAILS

Use RMON I to show Layer-2/bridged traffic patterns. Use RMON II to show Layer-3/routed traffic patterns of IP and IPX protocols. The X-Pedition protocol directory has over 500 protocols in the IP and IPX protocol domains. RMON II provides limited extensibility of protocol reporting. It allows for renaming applications known by destination UDP and TCP ports. The key to using RMONII on the X-Pedition is knowing that custom designed Layer-3 ASICs used in routing and Layer-4 bridging decode and account for traffic on a per-flow basis. Bridged traffic is accounted for by a Layer-2 ASIC. Hence, bridged traffic is not accounted for by RMON II and routed traffic is not accounted for by RMON I.

Data Availability

The Layer-2 and Layer-3 ASIC are swept every 14 seconds. This is the period of granularity for host and matrix counters. For etherStats counters, the updates are less than a second.

The flow bridging command is required for RMON I Layer-2 matrix information to be collected. By default, the X-Pedition operates in Address-bridging mode (see X-Pedition Native CLI Reference Manual).

The RMON data is collected on the traffic entering the router via one of the line cards. This data represents traffic that is uni-directional. This is illustrated in the example display screen below which shows traffic that enters port et.5.5. To see the traffic in the reverse direction, RMON data would need to be collected on the port attached to the destination. The traffic on this port would represent the uni-directional traffic in the other direction.

The combination represents the total bi-directional traffic between hosts connected to the router.

```
xp# rmon show al-matrix et.5.5
RMON II Application Layer Host Table
Index: 500, Port: et.5.5, Inserts: 4, Deletes: 0, Owner: monitor
SrcAddr  DstAddr          PacketsOctetsProtocol
-------------------      --------------------------
10.50.89.8815.15.15.3    1771272562ip-v4
10.50.89.8815.15.15.3    1125211192tcp
10.50.89.8815.15.15.3    1122210967telnet
10.50.89.8815.15.15.3    3   225    www-http
xp#
```

The following examples help illustrate how collected data should be interpreted based on the X-Pedition implementation information described above.

For example, if the host (host-1) is connected to an Ethernet port (e.g., et.15.1) and another host (host-2) is connected to a different Ethernet port (e.g., et.15.2), and they are bridging traffic (NOT routing) to each other, data will be collected as it comes into each port (if each port is configured to collect data). If the command **rmon show hosts et.15.1** is executed, the data will show the MAC address of host-1 and nothing more because that is all that is connected to port et.15.1. This data will show as follows for address-bridging mode (the default setting):

```
xp# rmon show hosts et.15.1
RMON I Host Table
Index: 500, Port: et.15.1, Owner: monitor
Address  InPkts           InOctets  OutPkts   OutOctetsOut BcstOut Mcst
-------  ------           --------  -------   ------------------------
003030:3030300            0         837303    1175873920       0
```

OutPkts and OutOctets are counted because the traffic is seen coming into port et.15.1 and is labeled with relationship to the host: Packets are being transmitted from the host. InPkts and InOctets show no data because the hardware can collect only statistics coming into the port, not going out of it.

The data for flow-bridging mode will appear as follows:

```
Index: 500, Port: et.15.1, Owner: monitor
Address  InPkts           InOctets  OutPkts   OutOctetsOut BcstOut Mcst
-------  ------           --------  -------   ------------------------
002020:20202020752       1328128   0         0        0       0
003030:3030300           0         20752     1328128  0       0
```

Both hosts show because the data is displaying in the flow. From host-1 (MAC address 003030:303030) to host-2 (Mac address 002020”202020). Again, data will be displayed with reference to the port but will be labeled with reference to the host that is connected to the port. Still, the InPkts and InOctets are not counted for host-1 (MAC address 002020:202020) because we only see traffic coming into port et.15.1. And, the OutPkts and OutOctets are not counted for host-2 (MAC address 002020:202020) because host-2 is not connected to port et.15.1.

If the network configuration is changed to reflect multiple hosts on a port, then it is possible to see both InPkts/InOctets and OutPkts/OutOctets at 0 (for flow-bridging mode) or counted (for flow bridging mode) for each host detected on a port.

For example, suppose host-1 and host-3 connect to ethernet port et.15.1 but host-2 connects to a different ethernet port (et.15.2). If host-1 bridges (not *routes*) traffic to host-2, host-3 bridges traffic to host-1, and host-2 bridges traffic to host-1, data will be collected as it arrives by each port configured to do so. If a user executes the **rmon show hosts et.15.1** command, the router will display the MAC address of host-1 and host-3 because nothing else connects to the port. This data will appear as follows for address-bridging mode (the default):

```
Index: 500, Port: et.15.1, Owner: monitor
Address  InPkts           InOctets  OutPkts   OutOctetsOut BcstOut Mcst
-------  ------           --------  -------   ------------------------
003030:3030300           0         2768      177152   0       0
004040:4040400           0         0         0        0       0
```

Host-3 (MAC address 004040:404040) shows no data collected because the data was directed to host-1 and was not bridged through the X-Pedition. Yet, the X-Pedition detected the traffic from both hosts and created an L2 table entry for both. Host-1 (MAC address 003030:303030) shows traffic that is being transmitted to host-2. Host-2 is not

displayed because it is not connected to port et.15.1. This data will show as follows for flow-bridging mode.

```
Index: 500, Port: et.15.1, Owner: monitor
Address  InPkts           InOctets  OutPkts   OutOctetsOut BcstOut Mcst
-------  ------           --------  -------   ------------------------
002020:202020333333       21333312  0         0        0       0
003030:303030333330       21333120  333333    21333312 0       0
004040:4040400            0         333330    21333120 0       0
```

For port et.15.1, a flow is established through the X-Pedition from host-1 (MAC address 003030:303030) to host-2 (MAC address 002020:202020) and a go-nowhere flow is established from host-3 (MAC address 004040:404040) to host-1 (MAC address 003030:303030). Even though host-3 (MAC address 004040:404040) has data directed to host-1 (MAC address 003030:303030) without passing through the X-Pedition, the router still detects the traffic and records it in a flow. This is why host-1 (MAC address 003030:303030) shows data for both InPkts/InOctets and OutPkts/OutOctets—data from host-3 is going to host-1 and data from host-1 is going to host-2. It is important to note that the data reflects what is seen on the input of port et.15.1. As such, the InPkts/InOctets for host-1 (MAC address 003030:303030) do not reflect the absolute total of data received by host-1. Data from host-2 to host-1 is recorded in a flow seen on the input of port et.15.2. This data will show as follows:

```
Index: 501, Port: et.15.2, Owner: monitor
Address  InPkts           InOctets  OutPkts   OutOctetsOut BcstOut Mcst
-------  ------           --------  -------   ------------------------
002020:2020200            0         314071    20100544 0       0
003030:303030314071       20100544  0         0        0       0
```

In this case, the total data being received by host-1 is the sum of InPkts/InOctets for host-1 (MAC address 003030:303030) from what is seen on ports et.15.1 and et.15.2, and any other port passing traffic to host-1.

When routing packets, the data is collected by RMON II tables and all output will look much like the flow bridging output. For example, two hosts (host-1 and host-3) are connected to a switch or repeater that is connected to an Ethernet port (et.15.1). Another host (host-2) is connected to Ethernet port et.15.2. If host-2 routes traffic to host-2, host-3 routes traffic to host-2, and host-2 routes traffic to both host-1 and host-3, the router will collect data as it comes into each port if the ports are configured to collect data. If the user enters the **rmon show al-host et.15.1** command, the data will show the IP addresses and

the protocols they are using for host-1, host-2, and host-3. This data will appear as follows:

```
xp# rmon show al-host et.15.1
RMON II Application Layer Host Table
Index: 500, Port: et.15.1, Owner: monitor
Address  InPktsInOctets   OutPkts   OutOctetsProtocol
-----------------------------------------------------------
10.10.0.20   0           1997708   161809707ip-v4
10.10.0.20   0           1997708   161809707tcp
10.10.0.20   0           1997708   161809707ftp
10.10.0.239954153236193330          0  ip-v4
10.10.0.239954153236193330          0  tcp
10.10.0.239954153236193330          0  ftp
10.10.0.20   0           1997708   161809626ip-v4
10.10.0.20   0           1997708   161809626tcp
10.10.0.20   0           1997708   161809626ftp
```

Because data is observed on port et.15.1, there is no data shown for OutPkts/OutOctets from host-2 (IP address 10.20.0.2), and no data shown for InPkts/InOctets on host-1 (IP address 10.10.0.2) or host-3 (IP address 10.30.0.2). Furthermore, if a user enters the **rmon show al-host et.15.2** command, the data will show the IP addresses and the protocols they are using for host-1, host-2, and host-3 in reference to port et.15.2. This data will appear as follows:

```
Index: 501, Port: et.15.2, Inserts: 9, Deletes: 0, Owner: monitor
Address  InPktsInOctets   OutPkts   OutOctetsProtocol
-----------------------------------------------------------
10.10.0.2100180081145800 0          0  ip-v4
10.10.0.2100180081145800 0          0  tcp
10.10.0.2100180081145800 0          0  ftp
10.20.0.20   0           2003653   162295893ip-v4
10.20.0.20   0           2003653   162295893tcp
10.20.0.20   0           2003653   162295893ftp
10.30.0.2100185381150093 0          0  ip-v4
10.30.0.2100185381150093 0          0  tcp
10.30.0.2100185381150093 0          0  ftp
```

Again, this data is in reference to what is seen on the input port of et.15.2 and labeled in reference to the host with OutPkts/OutOctets routed from host-2 (IP address 10.20.0.2) and InPkts/InOctets routed to host-1 (IP address 10.10.0.2) and host-3 (IP address 10.30.0.2).

By default, the RMON I/II functionality is disabled and requires a configuration command **rmon enable** to turn on the functionality and make it available for use by an SNMP management station. The functionality is divided into three categories (lite, standard, and professional) and can be enabled in any combination using the **rmon set lite**, **rmon set standard**, and **rmon set professional** configuration commands. The “lite” category includes only the etherStats and History groups from RMON I. The “standard” category includes all RMON I groups. The “professional” category includes all RMON II groups.

NOTE: The probeConfig group in RMON II allows the X-Pedition to reboot using SNMP. It is important to protect the X-Pedition from unauthorized users by using ACLs to protect the SNMP agent.

8.2 MEMORY REQUIREMENTS

Until RMON is enabled, no memory is used by this feature. Once enabled, the amount of memory RMON uses is determined automatically by the number of ports and the level of functionality selected (lite, standard, or professional).

1MB	Base initialization
60K	Lite only (per port)
120K	Standard only (per port)
140K	Professional only (per port)
140K	Lite and standard (per port)
160K	Lite and professional (per port)
220K	Standard and professional (per port)
240K	Lite, standard, and professional (per port)

Example calculation using the information above.

X-Pedition 2000 with 32 ports, only lite and standard enabled:
1MB + (32 * 140K) = 5.48MB of memory

X-Pedition 8600 with 120 ports, lite, standard, and professional enabled
1MB + (120 * 240K) = 29.8MB of memory

Memory allocated to RMON can be returned to the system at any time by removing the **rmon enable** command.

The size of the data tables are a function of the number of hosts participating in the traffic recorded by RMON. If the automatic memory calculation is insufficient for your network, the data tables may exceed the RMON memory size. Should this happen, the oldest data is pruned from the table to make room for new entries. The following error message will be displayed the first time the memory limit is reached:

```
%SNMP-W-RMON_MEM, RMON memory max'ed out. You may want to add more memory to
RMON.
```

If this occurs, more system memory can be made available to RMON by using the **rmon set memory** <*memory in Megabytes*> command from configuration mode. This command overrides the automatic calculation and sets the memory limit to a user-specified value. The following values are allowed:

X-Pedition 2000/2400	12 MB
X-Pedition 8000	32MB
X-Pedition 8600/ER16	96 MB

Configuration Example

The following is a sample RMON configuration. For a more detailed description of RMON capabilities using the CLI, please refer to the Enterasys Networks Native CLI Reference Manual.

```
xp(config)# show
Running system configuration:
!
! Last modified from Telnet (10.50.89.88) on 2002-04-05 16:52:28
!
1 : port flow-bridging et.5.(3-8)Green40 address-netmask 192.16.40.51/24 port
gi.1.1
!
2 : interface add ip en0 address-netmask 10.50.6.9/16
!
3 : system set contact “usama”
4 : system set location Enterasys Systems
5 : system set name “nms”
6 : system set location junkinconfig
!
7 : rmon set ports all-ports
8 : rmon set lite default-tables yes
9 : rmon set standard default-tables yes
!
! Set RMON Pro Group with Default Tables ON, cap memory at 4 meg
! Pro: protocolDir, protocolDist, addressMap, al/n1-Matrix,
! al/nl-Host,
! al/nl-matrixTopN. userHistory, probeConfig.
! Default Tables: one control row per dataSource for
! protocolDist, addressMap,
! al/nl-Host, al/nl-Matrix
!
10 : rmon set professional default-tables yes
11 : rmon set memory 4
12 : rmon enable
```

8.3 TROUBLESHOOTING RMON PROBLEMS

If you are not seeing the information you expected with an **rmon show** command, or if the network management station is not collecting the desired statistics, use the **rmon show status** command to check the RMON configuration on the X-Pedition. See command output below:

```
xp# rmon show status
RMON Status
----------------
* RMON is ENABLED
* RMON initialization successful.
+-----------------------------------+
|  RMON Group Status   |
+----------+-----------+-----------+
|  Group| Status|     Default|
+----------+-----------+-----------+
|  Lite | On|         Yes|
+----------+-----------+-----------+
|  Std  | On|         Yes|
+----------+-----------+-----------+
|  Pro  | On|         Yes|
+----------+-----------+--------*--+

RMON is enabled on: et.5.1, et.5.2, et.5.3, et.5.4, et.5.5, et.5.6, et.5.7,
et.5.8

RMON Memory Utilization
-----------------------------------
Total Bytes Available:   48530436

Total Bytes Allocated to RMON:4000000
          Total Bytes Used:2637872
          Total Bytes Free:1362128
```

1. Verify that RMON has been enabled.
2. Make sure that at least one of the RMON support levels (Lite, Standard, Professional) is on.
3. Ensure that RMON is enabled on the ports for which you want statistics.
4. Check to see that RMON has not run out of memory (Total Bytes Free).
5. If **default-tables yes** is not included in the **rmon set lite| standard| professional** command, no statistics will be collected. The CLI may be used to create control table entries to collect data. See the X-Pedition User Reference Manual for detailed information about using RMON with the CLI commands.

6. If your RMON/SNMP applications are unable to create control table entries, use the **snmp show status** command to check for errors and to verify that a community or user is configured with write access to the RMON control tables. Verify that you can ping the X-Pedition and that no ACLs prevent SNMP to access the router.
7. If traffic includes Netbios or vines traffic, it will not be recognized by RMON. The X-Pedition decodes over 500 protocols—to determine if a protocol is decoded, use the **rmon show protocol-directory** <*protocol name*> or use **all-protocols** to get the list of all decoded protocols.

NOTE: The RMON statistics are collected by hardware at wire speed. The hardware design imposes some restrictions on the kind of information that is available. The design of the X-Pedition hardware ASICs uses different data paths for bridged and routed traffic. This means that Routed traffic can be collected in RMON II tables and bridged traffic can be collected in RMON I tables. RMON II data can be collected for bridged traffic in RMON II tables if a port is configured for Layer-4 bridging mode.

8. The **rmon add ports all-ports** command adds only LAN ports. RMON I etherStats data is available from the CLI when a service profile is applied to a wan port or VC/DLCI. Standard RMON Mibs do not support subinterfaces, hence there is currently no MIB to retrieve per VC/DLCI data.
9. For PPP interfaces, the serial port provides a 1 to 1 mapping to the PPP ifIndex. RMON I/II can be collected for the first port of a WAN card only.
10. The RMON CLI can be used to filter RMON reports that are quite large. This allows users to locate specific information without searching through the entire report. Alternately, users may install filters to help limit what appears in a CLI report. Filters are expressions that describe what you really want to see. See example filters below:

```
xp# rmon show cli-filters
RMON CLI Filters
Id Filter
-- ------
1 (inpkts >= 0)
2 (inpkts >= 0 and outoctets>= 0)
3 srcmac 222222222222 and (outoctets >= 0)
4 inpkts > 500
5 inpkts > 0
6 srcmac 2222222222222
You have not selected any filter
```

A CLI filter is defined with the **rmon set cli-filter**<*filter number*><*expression*> command from configure mode and filters may be applied in enable mode using the **rmon**

apply cli-filter<*filter-number*> command. Subsequent RMON show commands will apply this filter to the output.

A

X-Pedition MIB Descriptions

A.1 IETF MIB SUPPORT

This section describes IETF MIBs supported by the X-Pedition Firmware.

Table A-1 Supported IETF MIBs

Layer-1	Rev E8.0	Rev E8.1	Rev E8.2/E8.3	Rev E9.0
EtherLike-MIB	2358	2358	2358	2358
MAU-MIB	2668	2668	2668	2668
SONET-MIB	1595 (1)	1595 (1)	1595 (1)	1595 (1)
DS1-MIB	2495	2495	2495	2495
DS3-MIB	2496	2496	2496	2496
FDDI-SMT73-MIB	1512	1512	1512	1512
IEEE 802.3ad LAG-MIB	N/A	N/A	Draft#3.1	Draft#3.1

Layer-2	Rev E8.0	Rev E8.1	Rev E8.2/E8.3	Rev E9.0
FRAME-RELAY-MIB	2115	2115	2115	2115
BRIDGE-MIB	1493	1493	1493	1493
Q-BRIDGE-MIB	2674	2674	2674	2674
P-BRIDGE-MIB	2674	2674	2674	2674
PPP-LCP-MIB	1471	1471	1471	1471
PPP-SEC-MIB	1472	1472	1472	1472

PPP-IP-NCP-MIB	1473	1473	1473	1473
PPP-BRIDGE-NCP-MIB	1474	1474	1474	1474
RMON-MIB	1757	1757	1757	1757
ATM-MIB	1695	1695	1695	1695

Layer-3	Rev E8.0	Rev E8.1	Rev E8.2/E8.3	Rev E9.0
BGP4-MIB	1657 (2)	1657 (2)	1657 (2)	1657 (2)
RIPv2-MIB	1724 (2)	1724 (2)	1724 (2)	1724 (2)
OSPF-MIB	1850 (2)	1850 (2)	1850 (2)	1850 (2)
RMON2-MIB	2021	2021	2021	2021
IP-FORWARD-MIB	2096	2096	2096	2096
IP-MIB	2011	2011	2011	2011
DVMRP	Draft#4	Draft#4	Draft#4	Draft#4
IGMP	Draft#5	Draft#5	Drat#5	Draft#5
VRRP-MIB	Draft#9	Draft #9	2787	2787
APPLETALK-MIB	N/A	1742	1742	1742

System-Related	Rev E8.0	Rev E8.1	Rev E8.2/E8.3	Rev E9.0
Entity MIB	N/A	N/A	N/A	2737
Host Resources MIB	N/A	N/A	N/A	2790
IF-MIB	2233	2863	2863	2863
RADIUS-AUTH-CLIENT-MIB	2618	2618	2618	2618
SNMPv2-MIB	1907	1907	1907	1907

UPD-MIB	2013	2013	2013	2013
TCP-MIB	2012	2012	2012	2012

SNMP	Rev E8.0	Rev E8.1	Rev E8.2/E8.3	Rev E9.0
SNMPv1	N/A	N/A	N/A	1157
SNMPv2c	N/A	N/A	N/A	1901
SNMP-FRAMEWORK-MIB	N/A	N/A	3411 (3)	3411
SNMP-MPD-MIB	N/A	N/A	3412 (3)	3412
SNMP-NOTIFICATION-MIB	N/A	N/A	3413 (3)	3413
SNMP-TARGET-MIB	N/A	N/A	3413 (3)	3413
SNMP-USER-BASED-SM-MIB	N/A	N/A	3414 (3)	3414
VACM for the SNMP	N/A	N/A	N/A	3415
SNMP-COMMUNITY-MIB	N/A	N/A	3584 (3)	3584

Notes:

1. Implemented only on 2 port ATM card. POS OC3/12 card will implement POS MIB in future release.
2. MIB offline by default. Traps are configured separately with the **snmp set trap** command.
3. These SNMPv3 MIBs are implemented as read-only. The SNMPv3 administrative model is implemented, but the SNMPv3 protocol is not supported in this release.

Table A-2 Supported Enterprise MIBs

Enterprise MIBs	Rev E8.0	Rev E8.1	Rev E8.2/E8.3	Rev E9.0
NOVELL-IPX-RIP-SAP-MIB	2/94	2/94	2/94	2/94
NOVELL-IPX-MIB	4/21/94	4/21/94	4/21/94	4/21/94
CTRON-SSR-POLICY-MIB	7/21/99	7/21/99	7/21/99	7/21/99
CTRON-SSR-CONFIG	8/17/99	8/17/99	8/17/99	8/17/99
CTRON-SSR-HARDWARE-MIB	12/18/99	12/18/99	12/18/99	12/18/99
CTRON-SSR-SERVICE-MIB	8/4/99	8/4/99	8/4/99	8/4/99
CTRON-SSR-CAPACITY-MIB	11/5/98	11/5/98	11/5/98	11/5/98
DEC-ELAN-MIB	First Release	First Release	First Release	First Release
CTIF-EXT-MIB	8/24/98	8/24/98	8/24/98	8/24/98
CTRON-CHASSIS-MIB	5/23/97	5/23/97	5/23/97	5/23/97
CT-CONTAINER-MIB	5/26/98	5/26/98	5/26/98	5/26/98
CTRON-DOWNLOAD-MIB	N/A	2/2/99	2/2/99	2/2/99
CTRON-CDP-MIB	N/A	N/A	8/28/99	8/28/99

A.2 ENTERPRISE MIB DESCRIPTIONS

NOVELL-IPX-RIPSAP (Novell Netware)

NOVELL-IPX (Novell Netware)
Provide information on Novell IPX protocol.

CTRON-SSR-POLICY
Provides a means to monitor and modify Layer-3 access control lists (ACLs) as well as Layer-2 filters.

CTRON-SSR-CONFIG
Provides for retrieval and download of the X-Pedition configuration file via TFTP. It also allows users to retrieve the startup bootlog file for analysis of configuration errors.

CTRON-SSR-HARDWARE
Provides detailed inventory information and environmental status of the X-Pedition.

CTRON-SSR-SERVICE
Provides status for services like routing protocols, Spanning tree, and RMON running on the X-Pedition.

CTRON-SSR-CAPACITY
Provides system, CPU, and memory statistics.

DEC-ELAN-MIB
Provides additional FDDI status and statistics not provided by the FDDI-MIB.

CTIF-EXT-MIB
Provides proprietary extensions to MIB-II objects.

CTRON-CHASSIS-MIB
Only available in the Matrix E5 and E7 router modules. Provides environmental information about the chassis.

CT-CONTAINER-MIB
Provides status and mapping of logical and physical components.

CTRON-DOWNLOAD-MIB
Provides for downloading firmware images via TFTP.

CTRON-CDP-MIB
Provides for control and monitoring of the Cabletron Discovery Protocol.

B

Implementing IF-MIB RFC 2233 on the X-Pedition

This appendix describes X-Pedition SNMP agent support for IETF RFC 2233. This MIB supersedes RFC 1573 which supersedes the interfaces group in RFC 1213 (MIB II). In particular, RFC 2233 provides the following features:

- Full 64-bit counters per port via the ifXTable.
- Module hot swap support via the ifCounterDiscontinuityTime field in ifXTable.
- ifStackTable presents complete connectivity picture for management applications from port, VLAN, SmartTRUNK (port group), and Layer-3 interfaces.

B.1 X-PEDITION IFTABLE MODEL

In RFC 1213, the ifTable contained rows for each physical interface at the IP level. The bridge MIB (RFC 1493) kept statistics at a Layer-2 level but used a different indexing scheme (dot1dBasePort vs. ifIndex). Both tables provided port status (up/down) and counters for packets, and the bridge MIB even provided a foreign key to index into the ifTable in cases where a device could bridge and route traffic on the same physical port.

To understand the ifTable in the X-Pedition, you must first understand the X-Pedition in terms of how to build current data networks. Data networks are built using two classes of devices: Layer-2 switches and Layer-3 routers. Switches are fast (wire-speed) and have huge physical port densities (now in the hundreds of ports). Routers are much slower than switches when moving packets from port to port and have far fewer ports (tens of ports), yet they provide the software (routing protocols) crucial to building scalable networks—as well as specialized links for wide area networks that link geographically to remote locations.

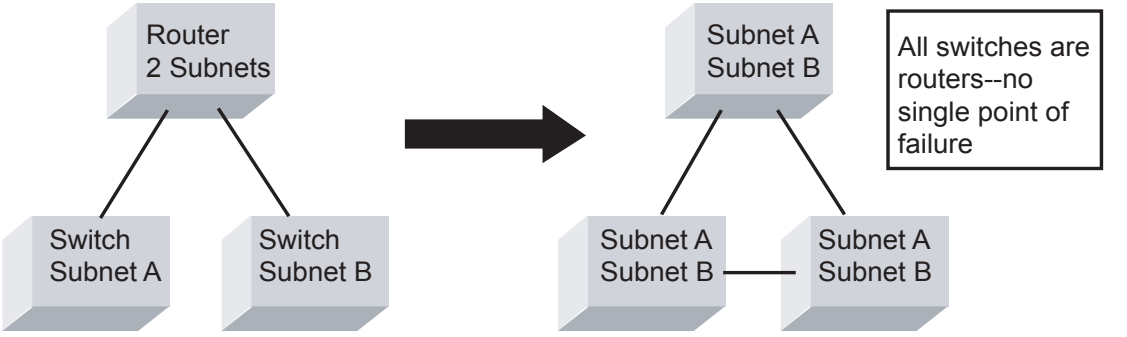
The X-Pedition is a type of device called a switch router. In addition to providing full bridge and routing functions, it provides integrated Layer-3 per protocol VLAN and port-based/Layer-2 VLAN per IEEE 802.1Q/RFC 2674 in high density port

configurations. For example, the XP-8000/8600 may have a complement of 240 fast Ethernet ports using 16-port fast ethernet modules. The X-Pedition offers incredible flexibility in building traditional data networks that use one type of network device where any port can be made into a Layer-2 or Layer-3 subnet through simple CLI configuration. Instead of physical separation of Layer-2 and Layer-3 functions into switches and routers, it is clear that switches will replace traditional routers.

In OSPF Network Design Solutions, 1998 Cisco Press, page 64, Thomas M Thomas II describes basic internetworking components. The author notes the following about LAN Switches:

> One of the more interesting advances in switching technology has been the recent addition of OSPF to switches. Many of today's newer switches actually have the power of the OSPF protocol included within them. This makes for a very powerful combination and makes us wonder: What will happen to routers in the years to come?

Figure B-1 Combining Layer-2 and Layer-3 Functionality



This consolidation of networking capabilities in the X-Pedition leads to a moderate increase in the complexity of the X-Pedition MIB design.

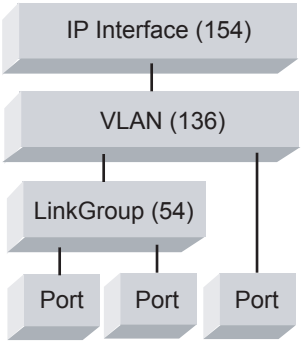
In the current world of Layer-2 switches and Layer-3 routers, ifTable would contain simple physical ports from which the user would assign one or more IP Addresses. VLANS may exist in a separate SNMP table, RFC 2674, which at first glance seems best for traditional Layer-2 switching but doesn't fit well in a multi-layer switch environment. A management station must go through a complex and cumbersome data retrieval process to understand the Layer-2 to Layer-3 topology and to map the complete, combined representation above. It requires performing table gets across multiple Layer-2 devices to

learn what ports are in what VLAN. It then must figure out what routers connect to what ports to learn the Layer-3 IP subnet assignments.

In contrast to representing Layer-3 routers with the ifTable (with interfaces tied to ports and Layer-2 switches), RFC 2674 enables the X-Pedition agent to build a complete picture of a data network in a consistent and logical fashion using just two standard tables. The ifTable must contain entries for all key logical and physical layers. IP Addresses are not tied to physical ports anymore, they are properties of subnets (vlans) which are simply broadcast domains. This new model represents the following type of interfaces:

1. **Physical Port**. A physical component connected to the network. It can be identified by the ifConnectorPresent object in the ifXTable. The X-Pedition firmware reports a string "Physical Port:" in ifDescr. Counters can be found in the ifTable and IfXTable(High Capacity) or in a specific MIB for that port type. The EtherLike-MIB, SONET-MIB, and DS1-MIB all contain specific data and error counters. If a user adds the **port description** <*port*><*user description*> command to the configuration file, this string will be appended to ifDescr. The ifName object reports the CLI name of the port.
2. **Logical Port**. A virtual entity that makes a set of physical ports appear as one port to the upper services such as Spanning Tree or IP interfaces for the purpose of improving reliability and increasing bandwidth. See IEEE 820.3ad Link Aggregation (ifType PropMultiplexor) or Multi-Link PPP for examples of logical ports. Logical ports also include sub interfaces such as Frame Relay DLCI and ATM PVC.
3. **VLAN**. A set of ports that make up a broadcast domain. The X-Pedition supports broadcast domains that are IP only, Layer-2, and others. An IP VLAN explicitly disallows other Layer-2 and Layer-3 protocols. However, IETF standards define only port-based—not protocol-based VLANs (the X-Pedition supports both). IF-MIB statistics such as ifInOctets reported for VLAN ifTypes in the ifTable and ifXTable report the SUM of the individual physical ports in the VLAN. If logical ports are added to the VLAN, these are used to determine the actual physical ports of sub-interface ports to create an aggregate counter for the logical port. Be sure to use High Capacity Counters if the aggregate speed is over 200Mbps per RFC2233.
4. **IP Interface**. A logical entity that maintains one or more IP addresses and maintains a one-to-one relationship with its broadcast domain/VLAN.

Figure B-2 IF-MIB Layered Model



The figure above shows three Ethernet ports—two of which have been configured as a link aggregation group (IEEE 802.3ad). A Virtual LAN (VLAN) is then formed from the link group and another port to form a broadcast domain (IEEE 802.1Q). Defining an interface and attaching it to a VLAN allows VLAN traffic to be routed. Traffic between VLANs routes internally to the switch—as if the ports were connected directly to an external Router, but with major savings in cabling, rack space, power, and money.

The model in Figure B-2 provides key benefits to management stations. Using the ifStackTable, a management station can quickly determine the Layer-2 and Layer-3 relationship that exists on the network without having to know ahead of time how to link or join networking layers that exist across multiple SNMP tables. In the standards arena, there appears to be a rift in current ifStackTable designs between LAN and WAN technologies. WAN MIBs such as SONET and ATM and DS1-3 MIBs make liberal use of the ifTable and ifStackTable, while the IEEE prefers to think of ways to increase scaling problems and complexity issues. As seen in the LAG-MIB and the Q-BRIDGE-MIB, not using precise ifType enumerations for standard protocols and inventing new and unique tables and indexes introduces scaling problems and complexity issues. Simple, pragmatic use of the ifSTackTable helps solve connectivity issues and provides the simplicity through conformity needed to isolate problems using only two tables: the *ifTable* and *ifStackTable*. Other benefits also arise from using these tables. For instance, summary polling is easier using ifTable/ifStackTable when dealing with devices such as switch-routers that have high port density.

B.2 IFXTABLE SUPPORT

RFC 2233 defines the ifXTable, which provides additional capabilities. These capabilities include:

ifName Defines the string that matches the CLI object. On the X-Pedition, ifDescr="Physical port: et.1.3" ifName = "et.1.3"

ifConnectorPresent (T/F) Object can identify the presence of physical as opposed to logical ports in the ifTable.

ifLinkUpDownTrapEnable Object provides the ability to stop traps on a per port basis.

ifCounterDiscontinuityTime Object tracks hot swap events.

ifAlias Tracks an interface through reboots and hot swaps. The value is read-only to begin with and is guaranteed to be unique.

ifPromiscuousMode Object allows you to detect ports that have port mirroring enabled. Interfaces of this type accept all packets—regardless of the MAC address.

ifHC*(octets) Provide 64-bit counter access to byte packet counters. This is very useful for gigabit or link aggregations like VLANs and SmartTRUNKs where 32-bit counters will wrap (reach their maximum values and reset to zero) in as few as 32 seconds.

B.3 ENHANCED POLLING FEATURES FOR HIGH-DENSITY PORT DEPLOYMENTS

VLAN row entries (both port- and protocol-based) are particularly useful when monitoring large groups of ports on an X-Pedition. The VLAN counters kept per VLAN reflect the sum of the counters for the ports contained in the VLAN. ifLastChange represents the last time an entry in the table changed its ifOperStatus state. For a VLAN entry, ifLastChange reflects the amount of time since the system last booted. As such, it makes polling for port status capable of being more efficient since users need to poll only the VLAN's ifLastChange. Once a VLAN ifLastChange value changes, only those ports in that VLAN need their ifOperStatus and ifLastChanged values examined. RFC2674 defines additional MIB information on IEEE 802.1Q/P VLANS such as the VlanId in the dot1qVlanCurrentTable (another method for determining port membership and a way to provision VLANS using dynamic protocols). Enterasys product development disagrees with some of the recommendations in this RFC. In real management situations, particularly the integration of the ifTable, the features described previously for high-density systems (500+ ports), integrating logical layers both above and below the VLAN, and identifying the type of VLAN (protocol- or port-based) are necessary to adequately describe the X-Pedition configuration in a manner recognized by industry standards.

B.4 HOT SWAP SCENARIO

This section describes what a management application will see when a module is hot swapped. Potential scenarios include:

- Hot swap the card *out*
- Hot swap the card *out* and *replace* with a card of the *same type*
- Hot swap the card *out* and *replace* with a card of a *different type*

When you press the hot swap button or issue the **hotswap** command, the X-Pedition Agent will set ifCounterDiscontinuity equal to sysUpTime and the ifOperStatus will change to notPresent.

An RFC 2233-aware management application that graphs performance counters such as ifInOctets should throw away any delta (i.e., the change in value between data samples) it finds when the ifCounterDiscontinuityTime object's value differs between two polls. A management application that is not RFC 2233-aware, such as HP OV's xnmgraph, would graph a flat line because the counters return the values the X-Pedition saved when you removed the card (the X-Pedition saves the ifTable counters only). Other tables indexed by ifIndex do not have the notion of state and, as such, will return noSuchName even though the ifIndex exists. A management application that is RFC 2233-aware should not graph or save the delta in cases where the ifCounterDiscontinuityTime value is not the same between data samplings. sysUpTime represents a similar discontinuity indicator but on a global level.

Typically, older SNMP agents will return noSuchName when faced with a hot swap. Furthermore, they often return the same ifIndex value (tied to a slot on the chassis) and typically fail to supply new ifIndex values. On the X-Pedition, a hot swap refers not only to removing physical modules and cards, but to all types of entries in the ifTable (VLANs, SmartTRUNKs, IP interfaces). If you create and activate a VLAN, then negate and recreate it with the same name, the X-Pedition will reuse the same ifIndex.

The X-Pedition diagnostic command, **snmp-debug ifmib show all**, displays the current ifIndex to port mapping as well as the ifStackTable mappings from the CLI.

B.5 NEW OPERATIONAL STATES FOR IFOPERSTATUS

RFC 2233 added three new operational states to ifOperStatus: dormant, notPresent, and lowerLayerDown. The X-Pedition SNMP Agent uses these new states as follows. When you delete an item in the ifTable, the ifOperStatus changes to notPresent. The dormant

state is used when the X-Pedition is “half-configured.” For example, if the configuration file had an IP Interface defined and connected to a VLAN but no physical or logical ports were added (e.g., VCs, DLCIs, or SmartTRUNKS), the state of the VLAN will be dormant.

The lowerLayerDown state, used by IP Interfaces and VLANS, clearly indicates the “layered” feature of a multi-layer switch. By traversing the ifStackTable from a given ifIndex that reports lowerLayerDown(7), you can easily pinpoint the exact layer at which the problem occurred—its ifOperState will be down(2).

C

X-Pedition Layer-2 and Bridging MIB Implementation Notes

This section describes the implementation of the BRIDGE-MIB, Q-BRIDGE-MIB, and P-BRIDGE-MIB on the X-Pedition. Currently, you can configure and monitor PVST through the command line interface.

C.1 UNDERSTANDING THE BRIDGE-MIB

The BRIDGE-MIB (RFC 1493) is composed of a set of groups used to configure and monitor spanning tree protocol (STP) bridges running transparent mode, and provision destination-based MAC address filters (but not monitor them). The BRIDGE-MIB monitors and configures only the default instance of spanning tree as defined in IEEE 802.1d-1990 documents at http://www.ieee.org. A bridge consists of a set of "ports" (as found in dot1dBaseNumPorts.0) and relates back to the ifTable/ifStackTable via the dot1dBasePortTable. In the BRIDGE MIB, individual ports are identified by dot1dBasePort index—this index is not the same as the IF-MIB's ifIndex. You must use the dot1dBasePortIfIndex object to relate a dot1dBasePort to an ifIndex when you need to know what type of port is part of this logical bridge. In the X-Pedition, the dot1dBaseNumPorts.0 is the total number of physical LAN and Link Aggregation ports.

In the past, a port was almost always a corporeal entity, most likely a FDDI, Ethernet, or Token-ring port. A classic transparent Ethernet bridge was simply a set of physical ports. Today, ports may be physical or logical entities and bridging is just another software feature in a multi-functional network device.

Physical and logical ports are defined by ifType in the ifTable. A list of enumerated types is available at ftp://ftp.isi.edu/mib/ianaiftype.mib. To distinguish between physical or logical ports, use the ifXTable's ifConnectorPresent (T/F) instead of the ifType value. Physical ports on the X-Pedition that support bridging include:

- ethernetCsmacd(6)

- fddi(15)
- v35(45)
- hssi(46)
- sonet(39)

Logical ports on the X-Pedition you could use with transparent bridging include:

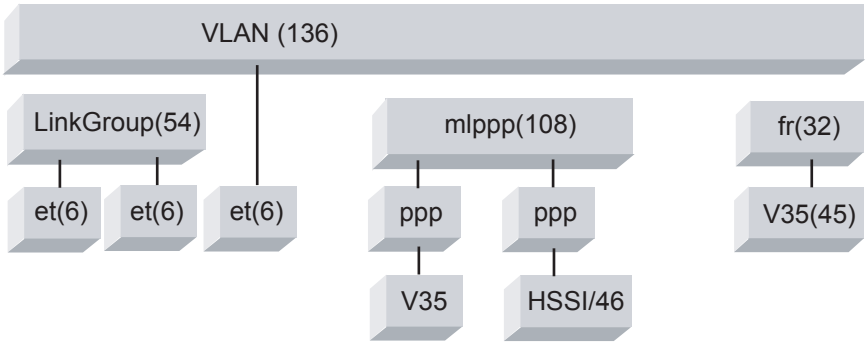
- ppp(23)—Point to Point Protocol
- pppMultilinkBundle(108)—Multi-link PPP (similar to link aggregation)
- propMultiplexor(54)—Link Aggregation (SmartTRUNKs)
- frameRelay(32)—DLCI added to non-default VLAN may bridge

Mapping Frame Relay DLCI's to the BRIDGE-MIB is a special case. Per RFC 2115, DLCI's are not in the ifTable/ifStackTable; as such, mapping DLCI's to dot1dBasePortTable is incoherent without the use of an enterprise MIB to map DLCI's to spanning tree instances. This case is becoming less of a concern since bridging over WAN links using frame relay is generally a bad idea.

NOTE: The X-Pedition WAN interfaces do not bridge when WAN ports are part of the default (VID=1) VLAN. This is an implementation-specific default. To enable bridging, create a separate VLAN and place LAN and WAN ports into the new VLAN. PPP allows you to enable and disable bridging on a per-port basis—not on a DLCI-by-DLCI basis for FRAME-RELAY.

Figure C-1 depicts an example of potential types of Layer-2 configurations available to the X-Pedition bridging function. In this example, the VLAN is composed of four ports. The Bridge MIB dot1dBasePortTable has dot1dBasePortIfIndex instances that refer to the entries marked in bold.

Figure C-1 X-Pedition Layer-2 Configurations



The dot1dBasePortTable also provides a list of all bridge-capable ports—regardless of their bridging state. To determine which dot1dBasePort matches to the CLU, use the **system show hardware** command. In the following example, dot1dBasePort 97 and 98 map to the Ethernet module in slot 6, ports 1 and 2. You may also use the **snmp-debug ifmib all** diagnostic command to find the ifIndex for these ports.

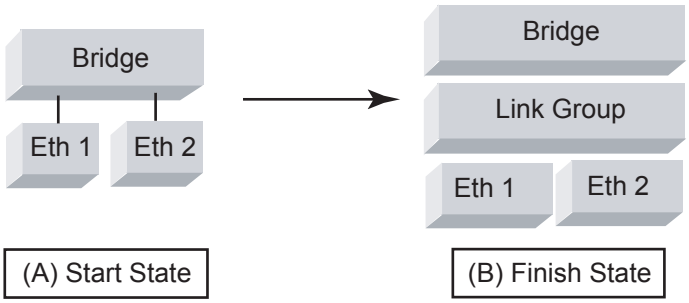
```
Slot     6,  Module: 10/100-TX  Rev. 0.0
    Port:   et.6.1,  Media Type: 10/100-Mbit Ethernet,  Physical Port: 97
    Port:   et.6.2,  Media Type: 10/100-Mbit Ethernet,  Physical Port: 98
```

What ports are listed in the BRIDGE-MIB? The BRIDGE-MIB does not assume that ports may have spanning tree enabled or disabled. Hence, all ports that are capable of bridging are listed in the dot1dBasePortTable.

C.2 REMOVING TWO PHYSICAL PORTS FROM A BRIDGE AND REPLACING THEM WITH ONE LINK AGGREGATION PORT

This section shows how the BRIDGE-MIB responds to a configuration sequence where two ports are placed into a link aggregation (SmartTRUNK).

Figure C-2 Link Aggregation Configuration



In the starting state (A), the dot1dBasePortTable will report two entries for ports 1 and 2. When the link group is created via the **smarttrunk create st.1 protocol no-protocol** command, the dot1dBaseNumPorts counter will increment by 1. Next, entering the **smarttrunk add ports et.1.1-2 to st.1** command adds ports et.1.1 and et.1.2 to the SmartTRUNK and results in dot1dBaseNumPorts.0 returning 1—effectively hiding the two physical ports from the BRIDGE-MIB (B).

The Q-BRIDGE-MIB dot1qVlanCurrentTable will also report the new dot1dBasePort index for the link group. A management application will need to consult the dot1dBasePortTable for the ifIndex to determine what kind of port the bridge detects. Polling ifTableLastChange.0 will provide the ability to detect port changes in a device.

Table C-1 ifType Values

ethernetCsmacd (6)	Physical ethernet port
ieee8023adLag (161)	IEEE 802.3ad Link Aggregation
propMultiplexor (54)	Proprietary Multiplexing (SmartTRUNKs)

D

Backward Compatibility

Backward compatibility has been improved for management applications such as MRTG that currently don't follow RFC 2233 ifMib changes but need to determine the ifSpeed of the interface. Currently, the X-Pedition SNMP agent reports an ifSpeed of 0 for all virtual layers (VLANs, IP interfaces, SmartTRUNKs). With the **snmp set retro-mib-ifspeed set** command, the ifSpeed of IP Interfaces, ifType virtualIpAddress(112) or ipForward(142), will report the ifSpeed of the first physical port associated with the interface. The atTable will also become visible for applications that continue to use this deprecated MIB. The console message reports this behavior as follows:

```
xp(config)# snmp set retro-mib-ifspeed
xp(config)# save active
%SNMP-I-RETRO_IFSPEED, ifSpeed for IP Interfaces will use speed of first
operational port
```

This feature allows you to create IP Interfaces with the following:

```
xp(config)# interface create ip mls0 address-netmask 1.2.3.4/16 port et.5.5
```

D.1 GROUP MIB/RFC REPLACEMENT (PRIOR VERSION)

MIB2 has been parcelled out to unique MIBs from each of the various groups. The breakdown is as follows:

Interfaces	IF-MIB / RFC 2233 (1573)
System	SNMPv2-MIB / RFC 1907
IP	IP-MIB / RFC 2011
IpRouteTable	IP-FORWARDING-MIB / RFC 2096
UDP	UDP-MIB / RFC 2012
TCP	TCP-MIB / RFC 2013

Objects in MIB2 that were marked as “deprecated” or obsolete are not implemented in the X-Pedition SNMP Agent. Below is a simple mapping of common areas of concern:

Object	New Object
atTable	ipNetToMedia
ipRouteTable	ipCidrRouteTable
ifInNUoctets	ifInMulticastPkts + ifInBroadcastPkts
ifOutNOctets	ifOutMulticastPkts + ifOutBroadcastPkts

NOTE: RFC 2233 no longer supports these managed objects.

E

Security and Auditing Protocols

Security provides controlled access to the network infrastructure through a set of Network Elements. **Changes to the configuration of network elements account for nearly half of all network outages**. The protocols discussed in this section help reduce outages caused by improper network management—by providing centralized access control and accounting, you can easily trace outages caused by improper configurations back to their origins. The following is a brief description of security protocols and how to configure them on the X-Pedition.

E.1 RADIUS (REMOTE ACCESS DIAL-UP SECURITY)

RFC 2865 defines RADIUS accounting and RFC 2866 defines the RADIUS protocol. RADIUS is the IETF Standard for AAA Authentication, Authorization, and Accounting. Designed by Livingston (now part of Lucent), RADIUS is the security protocol of choice for many network vendors. The X-Pedition includes a RADIUS client implementation. This client also has a MIB which instruments the RADIUS protocol for a client. With RADIUS accounting enabled, the X-Pedition also generates accounting messages for SNMP sets—just like any CLI changes.

```
system set password enable some-enable-mode-password
radius accounting shell all
radius authentication login
radius set host 15.1.1.3
radius set host 120.1.1.2
radius set timeout 15
radius set key “SOME SHARED KEY”
radius set last-resort password
radius enable
```

E.2 TACACS+ (TERMINAL ACCESS CONTROLLER ACCESS CONTROL SYSTEM)

Cisco Systems has maintained development of TACACS—now called TACACS-Plus (TACACS+). However, Cisco Systems routers also implement RADIUS (TACACS+ configures the same as RADIUS). TACACS+ is defined by draft-grant-tacacs-02-txt. Since many customers with Cisco routers may have CiscoSecure, the X-Pedition interoperates with this security protocol.

```
tacacs-plus accounting shell all
tacacs-plus authentication login
tacacs-plus set host 15.1.1.3
tacacs-plus set host 120.1.1.2
tacacs-plus set key “SOME SHARED KEY”
tacacs-plus set last-resort password
```

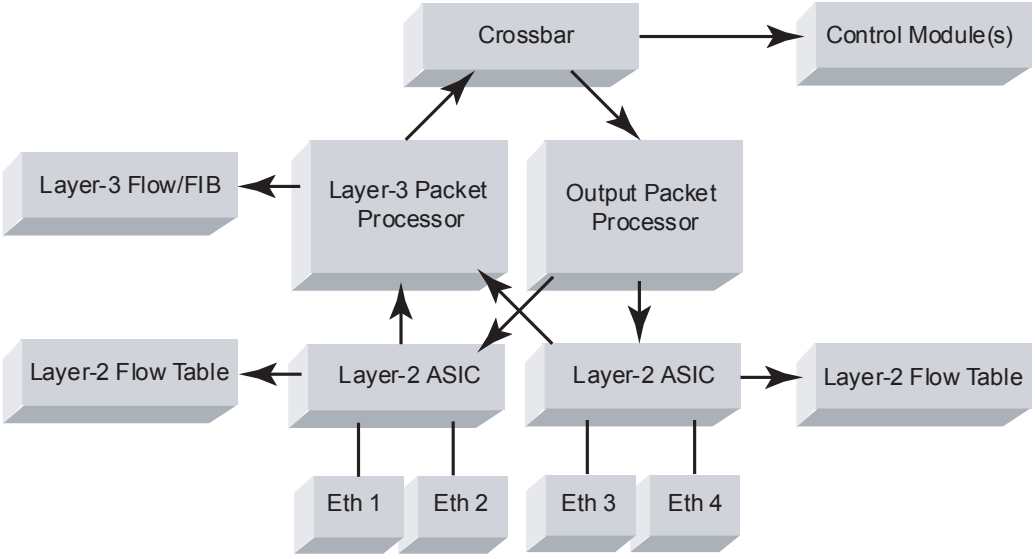
F

Monitoring Device Capacity

The Capacity MIB and **system show capacity** command enumerate and describe the status of all X-Pedition resources. Resources include memory, CPU utilization, queues, file-system capacity, and the number of empty slots. Any resource that may become depleted over time will appear in this table in a normalized form.

To comprehend resource utilization on the X-Pedition, users must understand how packets flow in one port and out another. The following figure provides a view of one type of line card available for the X-Pedition.

Figure F-1 Packet Flow Example



The relationships depicted in the illustration above help identify where to monitor an X-Pedition for capacity. The Layer-2 flow tables maintain micro-flow status for existing Layer-2 traffic, distinguished from Layer-3 traffic by the ethertype and destination MAC address. Layer-3 IP traffic will always use a destination MAC address of the next hop

router unless a user runs “Cisco-style” unnumbered interfaces typical in point-to-point WAN links. All X-Pedition Ethernet ports report the same MAC address—unlike traditional datacom equipment. This makes identifying a chassis with 480 ports easy at Layer-2. The system MAC address is reported in ifPhysAddress or via the **system show hardware** command.

When a packet enters a port, the Layer-2 ASIC decodes its header and, if the packet is to be bridged (based on the destination MAC address), a lookup is made to the Layer-2 flow table. If the lookup fails to find an entry in the flow table, the packet is sent to the CPU for learning. Otherwise, the packet is sent to the output port via the crossbar, then to the output packet processor which can queue packets into each of the 4 hardware queues it maintains for output. If the incoming packet is a routed IP packet, the packet is sent to the Layer-3 packet processor for decoding. The Layer-3 packet processor performs a lookup of the flow based on the current Layer-3 flow switching mode and, if no match is found, the packet is sent to the CPU to be learned. If a match is found, the packet will forward to the exit port via the crossbar.

Given the above, the key memories to watch can begin with the main CPU for basic utilization, as well as what the CPU is spending time doing (e.g., learning Layer-2 and -3 flows, running control protocols, or answering management queries). Aging rates can also be examined to fine-tune the system. Next, the Layer-2 and Layer-3 flow tables should be monitored for current capacity as well as flow misses. The capMemTable in the CAPACITY-MIB provides details for each line card on total/current usage, as well as failures. Failures are key to watch as they can occur even when the capacity used is less

than the capacity available. This MIB also has a CLI equivalent. The following diagram shows how to display the capCpuMemory table from the console.

```
xp# system show capacity memory
Capacity MIB Storage Information:
Type     Description      Size      Free      Used     Block    Remov    Fail
----     -----------      ----      ----      ----     -----    -----    ----
CPU      Internal CPU     8388608   8109714   278894   16       True     0
FLASH    Internal Flash   756       737       19       64       True     0
PCMCIA   PCMCIA Module    15876     8705      7171     64       True     0
L2HW     port et.1.1      5888      5887      1        64       True     0
L2HW     port et.1.2      5888      5887      1        64       True     0
L2HW     port et.1.3      5888      5887      1        64       True     0
L2HW     port et.1.4      5888      5887      1        64       True     0
L2HW     port et.1.5      5888      5887      1        64       True     0
L2HW     port et.1.6      5888      5884      4        64       True     0
L2HW     port et.1.7      5888      5887      1        64       True     0
L2HW     port et.1.8      5888      5887      1        64       True     0
L2HW     port et.2.1      5888      5887      1        64       True     0
L2HW     port et.2.2      5888      5886      2        64       True     0
L2HW     port et.2.3      5888      5887      1        64       True     0
L2HW     port et.2.4      5888      5887      1        64       True     0
L2HW     port et.2.5      5888      5887      1        64       True     0
L2HW     port et.2.6      5888      5887      1        64       True     0
L2HW     port et.2.7      5888      5887      1        64       True     0
L2HW     port et.2.8      5888      5883      5        64       True     0
L2HW     port gi.3.1      65536     65533     3        64       True     0
L2HW     port gi.3.1      65536     65535     1        64       True     0
L3HW     module 1         118784    118321    463      64       True     78
L3HW     module 2         118784    118321    463      64       True     296
L3HW     module 3         118784    118321    463      64       True     550
```

Index

R

S

T

U

V

W