

**Lucent Technologies**  
Bell Labs Innovations



# ***StarKeeper*<sup>®</sup> II NMS SNMP Proxy Agent Guide**

255-114-761  
Issue 1  
Release 10.0

**Copyright © 1998 Lucent Technologies**  
**All Rights Reserved**  
**Printed in USA**

*Datakit* and *StarKeeper* are registered trademarks of Lucent Technologies.

HP, HP-UX, HP-VUE are registered trademarks of Hewlett-Packard, Inc.

INFORMIX-SQL is a registered trademark of Relational Database Systems, Inc.

Motif and OSF/Motif are registered trademarks of Open Software Foundation, Inc.

The information in this document is subject to change without notice.  
Lucent Technologies assumes no responsibility for any errors that may appear in this document.

This document was produced by Customer Training and Information Products (CTIP).

---

# Contents

---

	<b>Preface</b>	xix
	■ Purpose of the Document	xix
	■ Terminology	xx
	■ Document Conventions	xx
	■ Training	xxi
<b>1</b>	<b>Introduction</b>	1-1
	■ SNMP Concepts	1-1
	■ Features of the <i>StarKeeper II</i> NMS SNMP Proxy Agent	1-2
	What the SNMP Proxy Agent Provides	1-3
	Functional Capabilities of the SNMP Proxy Agent	1-4
	■ Connectivity	1-5
	■ Database Provisioning	1-6
	■ Software Processes	1-6
	How an SNMP Request is Processed	1-6
	Daemon Processes	1-8
	■ Where Do You Go From Here?	1-9
<b>2</b>	<b>Administration and Maintenance Procedures</b>	2-1
	■ System Administration Tasks and Responsibilities	2-1
	Prerequisites	2-1
	Installation	2-2
	Logging In	2-2
	Starting the SNMP Proxy Agent	2-2
	Stopping the SNMP Proxy Agent	2-3
	SNMP Database Configuration	2-3
	Synchronizing the SNMP Database	2-3
	Testing	2-3

---

# Contents

BNS-2000 Initializing, Backing Up, and Restoring the SNMP Database	2-4
Troubleshooting	2-4
Guidelines for Processing Set Requests	2-5
Cleaning Up Log Files	2-6

---

<b>3</b>	<b>Database Management</b>	<b>3-1</b>
	■ Overview	3-1
	■ SNMP Proxy Agent Planning	3-1
	■ Sample Scenario	3-3
	■ MASTERVIEW	3-5
	■ Synchronizing the SNMP Database	3-5
	sksnmpndsync	3-5
	sksnmpdbsync	3-6
	■ Flow Control and Usage Counts	3-6
	Usage Counts	3-6
	Flow Control Parameters	3-7
	■ SNMP Proxy Agent Motif User Interface	3-7
	Forms Usage	3-9
	Special HP VUE and Motif Capabilities	3-10
	Keyboard Shortcuts	3-10
	Using the Help Facility	3-11
	Error Messages	3-11
	System Information	3-11
	Network Interfaces	3-13
	Fields on the Network Interface Administration Form	3-14
	Network Interface Provisioning	3-16
	Customer Views	3-18
	Fields on the Customer View Administration Form	3-19
	Subscriber	3-22
	Subscriber Information Administration Form Fields	3-23
	Adding a Subscriber	3-25
	Flow Control	3-27
	Allowed Requests by All Subscribers	3-27

---

# Contents

	Single Subscriber Requests Allowed	3-29
	Report Generation	3-30
	Node Usage Report	3-30
	Subscriber Reports	3-30
	Sample Subscriber Reports	3-31
■	SNMP Proxy Agent ASCII User Interface	3-35
	Operations Ring Menu	3-37
	Entering Data on Forms	3-37
	Getting On-line Help	3-38
	Error Messages	3-38
	Option 1: System Information	3-38
	Option 2: Network Interfaces	3-40
	Fields on the Network Interface Administration Form	3-41
	Option 3: Customer Views	3-46
	Fields on the Customer Views Form	3-46
	Option 4: Customer View Members	3-49
	Fields on the Customer View Members Form	3-49
	Option 5: Subscriber Information	3-53
	Subscriber Information Administration Form Fields	3-53
	Option 6: Flow Control and Usage	3-58
	Changing Throttling Counts or Usage Counts	3-58
	Verifying Flow Control and Usage Information	3-60
	Sample Usage Reports	3-61
<b>4</b>	<b>Troubleshooting</b>	4-1
■	SNMP Proxy Agent Dependencies	4-1
■	Monitoring the SNMP Proxy Agent Daemon Processes	4-1
	Displaying the Log File	4-2
	Using the Trace Facility	4-2
■	Troubleshooting the TCP/IP Subsystem	4-2
	ping	4-2
	ifconfig	4-3
	netstat	4-3

---

# Contents

■ Troubleshooting the StarKeeper II NMS Core System	4-3
■ Troubleshooting by Symptom	4-4
SNMP Proxy Agent Doesn't Come Up	4-4
SNMP Proxy Agent Comes Down Abruptly	4-4
SNMP Proxy Agent Doesn't Respond to Queries from any SNMP Managers	4-5
SNMP Proxy Agent Responds with NoSuchName Error	4-5
SNMP Proxy Agent Responds with TooBig Error	4-6
SNMP Proxy Agent Responds with AccessDenied Error	4-6
SNMP Proxy Agent Responds with General Error or BadValue Error	4-6
SNMP Proxy Agent Responds with ReadOnly Error	4-6
SNMP Proxy Agent Sends Authentication-Failure Trap	4-6

---

<b>5</b>	<b>SNMP Manager Configuration</b>	5-1
■	Prerequisites for SNMP Managers	5-1
■	Distribution of MIB Files	5-1
■	Guidelines for SNMP Manager Applications Development	5-2

---

<b>A</b>	<b>User Reference</b>	A-1
■	SNMP Proxy Agent Commands	A-1
■	Crontab for cnmsadm	A-24
■	Planning Forms	A-25

---

<b>B</b>	<b>Error and Log Messages</b>	B-1
■	SNMP Proxy Agent Error and Log Messages	B-1
■	Error Messages from sksnmpndsync	B-5

---

# Contents

---

<b>C</b>	<b>Supported SNMP MIB Objects and Traps</b>	<b>C-1</b>
	■ Overview	C-1
	■ Standard MIB-II Objects Supported by the SNMP Proxy Agent	C-2
	MIB-II Objects That Return Their Value	C-2
	MIB-II Objects That Return Null Value	C-3
	MIB-II Objects That Return noSuchName Error	C-3
	■ Standard MIB-II Traps Supported by the SNMP Proxy Agent	C-3
	■ Proxy Agent Behavior	C-4
	CNM System MIB	C-4
	CNM DS1 MIB	C-4
	CNM DS3 MIB	C-5
	CNM SIP MIB	C-5
	CNM SMDS MIB	C-6
	CNM FRM MIB	C-6
	CNM Enhanced FRM MIB	C-6
	General MIB Issues	C-7
	■ Summary of Private CNM MIB Objects	C-9
	CNM System MIB	C-9
	CNM DS1 MIB	C-10
	CNM DS3 MIB	C-13
	CNM SIP MIB	C-14
	CNM SMDS MIB	C-15
	CNM Frame Relay MIB	C-20
	CNM Enhanced Frame Relay MIB	C-22
	■ MIB Specification of Private CNM MIBs	C-28
	Format of Definitions	C-28
	CNM System MIB	C-28
	System Group Of Objects (lucent-cnm-system)	C-29
	Interfaces Group of Objects (lucent-cnm-interfaces)	C-31
	Interfaces Configuration Table	C-32
	Interfaces Status Table	C-34
	CNM Frame Relay MIB	C-36
	Frame Relay Group of Objects (lucent-cnm-fr)	C-36

---

# Contents

Frame Relay Configuration Table	C-36
Frame Relay Measurements Table	C-38
Frame Relay PVC-Level Measurements Table	C-42
Frame Relay PVC-Level Status Table	C-44
CNM SMDS MIB	C-46
SMDS Group of Objects (lucent-cmn-smds)	C-47
SMDS Configuration Table	C-47
SMDS SNI Addresses Table	C-51
SMDS Individual Address Screening Table	C-53
SMDS Group Address Screening Table	C-55
SMDS Member-Group Address Table	C-57
SMDS Group-Member Address Table	C-60
SMDS Disagreements Table	C-63
SMDS Disagreements Log Table	C-67
SMDS ICI Table	C-69
SMDS CIC Table	C-71
SMDS Set Individual Address Screening Table	C-72
SMDS Set Group Address Screening Table	C-76
■ CNM SIP MIB	C-81
SIP Group of Objects (lucent-cnm-sip)	C-81
SIP Configuration Table	C-81
SIP Measurements Table	C-83
SIP Level 3 Error Log Table	C-86
■ CNM DS1 MIB	C-90
DS1 Group of Objects (lucent-cnm-ds1)	C-90
DS1 Configuration Table	C-90
DS1 Status Table	C-93
DS1 Error Counts Table	C-94
DS1 Total Error Counts Table	C-102
DS1 PLCP Error Counts Table	C-108
DS1 PLCP Status Table	C-110
■ CNM DS3 MIB	C-112
DS3 Group of Objects (lucent-cnm-ds3)	C-112
DS3 Configuration Table	C-112

---

# Contents

DS3 Status Table	C-115
DS3 Error Counts Table	C-116
DS3 PLCP Error Counts Table	C-121
DS3 PLCP Status Table	C-123
■ CNM Enhanced Frame Relay MIB	C-125
Frame Relay Group of Objects (lucent-cnm-efr)	C-125
Enhanced-Frame-Relay Interface-Level Configuration Table	C-126
Enhanced Frame Relay Measurements Table	C-132
Enhanced Frame Relay PVC Level Configuration Table	C-141
Frame Relay PVC Level Measurements Table	C-146
Frame Relay PVC Level Status Table	C-154

---

<b>D</b>	<b>Database Schema</b>	D-1
■	dis_logs	D-1
■	grpmem	D-1
■	grpscr	D-2
■	ifdci	D-2
■	ifstatus	D-3
■	iftype	D-3
■	ifindex	D-4
■	indaddr	D-4
■	indscr	D-5
■	ip	D-5
■	ip_throt	D-6
■	memgrp	D-6
■	mibview	D-6
■	mibviewmem	D-7
■	node_throt	D-7
■	pe_logs	D-7
■	scr_log	D-8
■	scr_reqts	D-9

---

## Contents

■ snicfg	D-10
■ snmpcounts	D-11
■ snmpsys	D-12

---

<b>GL</b>	<b>Glossary</b>	GL-1
-----------	-----------------	------

---

<b>IN</b>	<b>Index</b>	IN-1
-----------	--------------	------

---

# Figures

---

## **1 Introduction**

Figure 1-1.	Communication between SNMP Manager and Agent	1-2
Figure 1-2.	SNMP Managers and Proxy Agent	1-3
Figure 1-3.	Sample Connectivity Diagram for SNMP Manager and Central Office	1-5
Figure 1-4.	SNMP Proxy Agent Processes	1-7
Figure 1-5.	SNMP Proxy Agent Daemon Processes	1-8

---

# Figures

---

# Tables

---

## **3 Database Management**

Table 3-1.	Keys Used on Forms	3-37
------------	--------------------	------

---

# Tables

---

# Procedures

---

## **2 Administration and Maintenance Procedures**

Procedure 2-1.	Starting the SNMP Proxy Agent Processes	2-2
Procedure 2-2.	Reinitializing the Database:	2-4
Procedure 2-3.	Provisioning Network Interfaces for CNM Service	2-5

---

## **3 Database Management**

Procedure 3-1.	Planning for the SNMP Proxy Agent	3-2
Procedure 3-2.	Accessing the SNMP Proxy Agent Motif User Interface	3-7
Procedure 3-3.	Using Forms (General Method)	3-9
Procedure 3-4.	Entering or Verifying System Information	3-12
Procedure 3-5.	Adding a Network Interface	3-16
Procedure 3-6.	Updating a Network Interface	3-17
Procedure 3-7.	Deleting a Network Interface	3-17
Procedure 3-8.	Adding a Customer View	3-20
Procedure 3-9.	Updating a Customer View	3-20
Procedure 3-10.	Deleting a Customer View	3-22
Procedure 3-11.	Entering a Subscriber	3-25
Procedure 3-12.	Updating a Subscriber	3-26
Procedure 3-13.	Deleting a Subscriber	3-27
Procedure 3-14.	Restricting the Number of Subscriber Queries Allowed	3-28
Procedure 3-15.	Adjusting the Number of Single Subscriber Requests Allowed	3-29
Procedure 3-16.	Generating the Node Usage Report	3-30
Procedure 3-17.	Generating Subscriber Reports	3-31
Procedure 3-18.	Re-initializing a Subscriber's Usage and Error Counts to Zero	3-35
Procedure 3-19.	Displaying the Main Menu	3-36
Procedure 3-20.	Choosing an Option from the Operations Ring Menu	3-37
Procedure 3-21.	Changing System Information	3-39
Procedure 3-22.	Verifying System Information	3-40
Procedure 3-23.	Adding a Network Interface	3-43
Procedure 3-24.	Deleting a Network Interface	3-43
Procedure 3-25.	Changing a Network Interface	3-44
Procedure 3-26.	Verifying a Network Interface	3-44
Procedure 3-27.	Adding a Customer View	3-47

---

## Procedures

Procedure 3-28.	Deleting a Customer View	3-47
Procedure 3-29.	Changing a Customer View	3-47
Procedure 3-30.	Verifying a Customer View	3-48
Procedure 3-31.	Adding a Customer View Member	3-50
Procedure 3-32.	Deleting a Customer View Member	3-51
Procedure 3-33.	Verifying Customer View Members	3-51
Procedure 3-34.	Adding Subscriber Information	3-55
Procedure 3-35.	Deleting Subscriber Information	3-56
Procedure 3-36.	Changing Subscriber Information	3-56
Procedure 3-37.	Verifying Subscriber Information	3-57
Procedure 3-38.	Changing Throttling Counts or Reinitializing Usage Counts	3-59
Procedure 3-39.	Generating Usage Reports	3-61

---

# Screens

---

<b>3</b>	<b>Database Management</b>	
Screen 3-1.	<i>StarKeeper</i> II NMS SNMP Proxy Agent Main Window	3-7
Screen 3-2.	Provisioning Menu	3-8
Screen 3-3.	Sample Form	3-9
Screen 3-4.	System Information Administration Form	3-12
Screen 3-5.	Network Interface Administration Form	3-13
Screen 3-6.	Network Interface: Node Address Dialog	3-16
Screen 3-7.	Customer View Administration Form	3-19
Screen 3-8.	Customer View: Interface Index Addition Dialog	3-21
Screen 3-9.	Subscriber Information Administration Form	3-23
Screen 3-10.	Subscriber: Read Customer View Dialog	3-26
Screen 3-11.	Flow Control: Node Requests by All Subscribers	3-28
Screen 3-12.	Flow Control: Node Requests by a Single Subscriber	3-29
Screen 3-13.	Node Usage Report	3-31
Screen 3-14.	Subscriber Usage of Nodes Report	3-32
Screen 3-15.	SNMP Packets by Subscriber Report	3-32
Screen 3-16.	SNMP PDU Errors by Subscriber Report	3-34
Screen 3-17.	Usage Counts: Re-initialize Usage Counts	3-35
Screen 3-18.	SNMP Database Administration Menu	3-36
Screen 3-19.	Sample Operations Ring Menu	3-36
Screen 3-20.	System Information Operations Ring Menu	3-38
Screen 3-21.	System Information Administration Form	3-39
Screen 3-22.	Verify System Information Form	3-40
Screen 3-23.	Network Interfaces Operations Ring Menu	3-40
Screen 3-24.	Network Interface Administration Form	3-41
Screen 3-25.	Verify Network Interfaces Form	3-45
Screen 3-26.	Customer View Operations Ring Menu	3-46
Screen 3-27.	Customer Views Form	3-46
Screen 3-28.	Verify Customer Views Form	3-48
Screen 3-29.	Customer View Members Operations Ring Menu	3-49
Screen 3-30.	Customer View Members Form	3-49
Screen 3-31.	Verify Customer View Members Form	3-52
Screen 3-32.	Subscriber Information Operations Ring Menu	3-53
Screen 3-33.	Subscriber Information Administration Form	3-53
Screen 3-34.	Verify Subscriber Information Form	3-57

---

## Screens

Screen 3-35.	Flow Control and Usage Information Operations Ring Menu	3-58
Screen 3-36.	Flow Control and Usage Administration Form	3-59
Screen 3-37.	Verify Usage Form	3-61
Screen 3-38.	Report of SNMP PDUs Processed	3-63

---

## A

### User Reference

Screen A-1.	Backup Media Screen	A-3
Screen A-2.	Restore Media Screen	A-12
Screen A-3.	Restore from Directory Screen	A-12

---

## Preface

---

The *StarKeeper II* Network Management System (NMS) SNMP Proxy Agent is an optional add-on software package for the *StarKeeper II* NMS Core System. It enables customers who subscribe to SMDS or Frame Relay services to manage their own private virtual networks by allowing them to monitor the interfaces they use on the public data network.

The *StarKeeper II* NMS SNMP Proxy Agent supports SMDS and Frame Relay services on BNS-2000 networks and certain Frame Relay services on BNS-2000 VCS networks.

The SNMP Proxy Agent software is installed on the host processor for the *StarKeeper II* NMS Core System. (See the *StarKeeper II* NMS Planning Guide for the models that support the SNMP Proxy Agent.)The *StarKeeper II* NMS Core System must also be connected to a TCP/IP Local Area Network (LAN).

## Purpose of the Document

This guide covers the installation, administration, and maintenance of the *StarKeeper II* NMS SNMP Proxy Agent and is intended for the *StarKeeper II* NMS administrator.

This guide includes procedural and reference information.

Before you install the SNMP Proxy Agent software, read **Chapter 1, Introduction** and **Chapter 2, Administration and Maintenance Procedures** to get an overview of the product and your administrative tasks.

Refer to **Chapter 3, Database Management** when you are ready to register an SNMP Manager in the SNMP database. **Chapter 5, SNMP Manager Configuration** describes the information that the administrator of an SNMP Manager needs in order to begin using the SNMP Proxy Agent.

**Chapter 4, Troubleshooting** and the **Appendices** include information that is useful for troubleshooting and maintaining the system.

---

## Terminology

---

The following terms are used frequently in this guide. For a complete list of terms and their definitions, refer to the **Glossary**.

- **SNMP Manager**—a customer's SNMP management station.  
An SNMP Manager controls and monitors the customer's private data network. With the SNMP Proxy Agent, an SNMP Manager can manage the part of the public network that it uses. The SNMP Manager can send requests for information and asynchronously receive SNMP traps.
- **Subscriber**—a customer who subscribes to SMDS or Frame Relay services. Each SNMP Manager is identified as a subscriber and assigned a Subscriber Identifier in the SNMP database.
- **Customer**—the person responsible for the set-up, administration and management of the private virtual network. The customer is located at the customer premises and manages the private virtual network through the use of SNMP.

## Document Conventions

---

Typographical conventions are used in this guide to distinguish certain kinds of information. For example, one convention represents text displayed on your screen, and another indicates information that you type at your keyboard.

The following list summarizes the conventions used in this guide:

- Commands and text that you type appear in bold, such as **sksnmpcf**.
- Keyboard keys appear in boxes, such as `ENTER`. A key sequence is noted by two keys. For example, `CONTROL G`.
- To enter a key sequence, you press and hold down the first key, then press the second key and release both.
- Field names appear in bold type, as in **Interface Index**.
- Variable entries are enclosed in greater than/less than brackets. For example, `<object_id>`.
- Selections appear in capital letters, as in `ADD`.
- Software processes, file and directory names, and environment variables appear in italics, such as *sksnmpd*.
- Recommended Prerequisites

The *StarKeeper* II NMS administrator responsible for the SNMP Proxy Agent should be knowledgeable with the following:

- The functionality and features of the *StarKeeper* II NMS Core System

- 
- TCP/IP concepts
  - SNMP concepts
  - The administration of Local Area Networks (LANs)
  - The administration of the UNIX® operating system
  - SMDS and Frame Relay Services

This person should also be familiar with the following products:

- HP-UX® operating system
- OSF/Motif™ user environment
- HP VUE®
- *StarKeeper* II NMS
- BNS-2000
- BNS-2000 VCS

## **Training**

---

To get information about training courses and schedules

- In the U.S.A., call the Customer Information Center at 1-888-LUCENT8, Option 2.
- In Europe, contact the Customer Assistance Contact in your country
- In other global locations, contact an International Enrollment Coordinator at +1-407-767-2798.

Customers may also obtain training information by accessing the World Wide Web at:

[www.lucent.product-training.com/catalog](http://www.lucent.product-training.com/catalog)



---

# Introduction

# 1

---

This chapter covers basic SNMP concepts and describes the features and architecture of the *StarKeeper II NMS SNMP Proxy Agent* software.

## SNMP Concepts

*Simple Network Management Protocol (SNMP)* is an open industry-standard network management protocol. SNMP enables an *SNMP Manager* and an *agent* to communicate.

An *SNMP Manager* controls and monitors the customer's private data network. It can also request information from the *agent*.

An *agent* is a device which carries out the requests of the SNMP Manager.

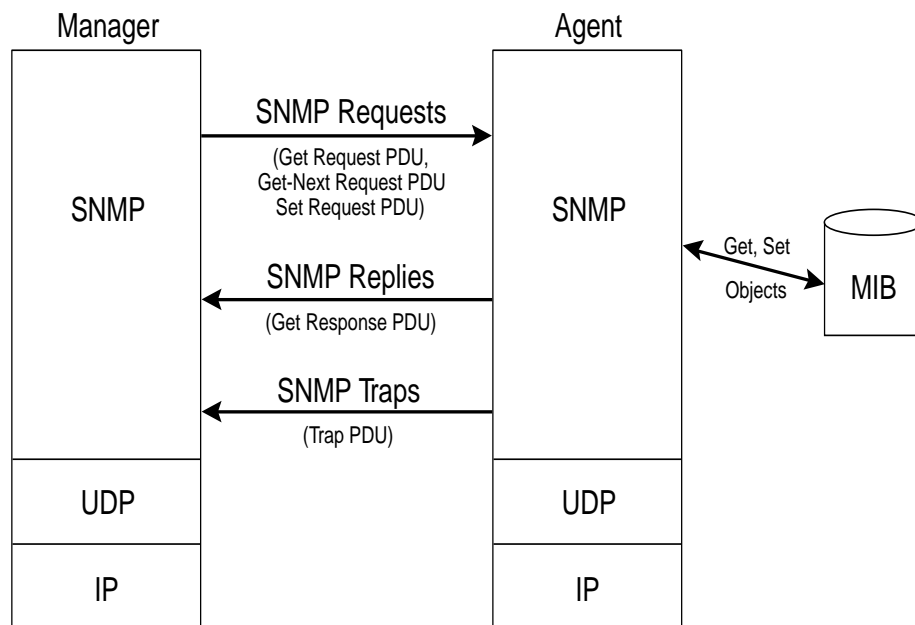
In addition, the agent automatically notifies the SNMP Manager when changes occur to network devices that may impact the network, without the manager's explicit request. This type of information transfer is called a *trap*.

An *SNMP Proxy Agent* is a technique that allows the proxy to take SNMP requests and translate them into transactions needed by managed machines that do not support SNMP.

The management information accessed through SNMP is conceptually organized into a database called the *Management Information Base* or *MIB*.

SNMP uses the User Datagram Protocol (UDP) at the transport layer and the Internet Protocol at the network layer for communication. Each SNMP Manager and SNMP agent are identified by a valid IP address.

Figure 1-1 shows how information flows between the SNMP Manager and agent and which network layers exist on the SNMP Manager and agent.

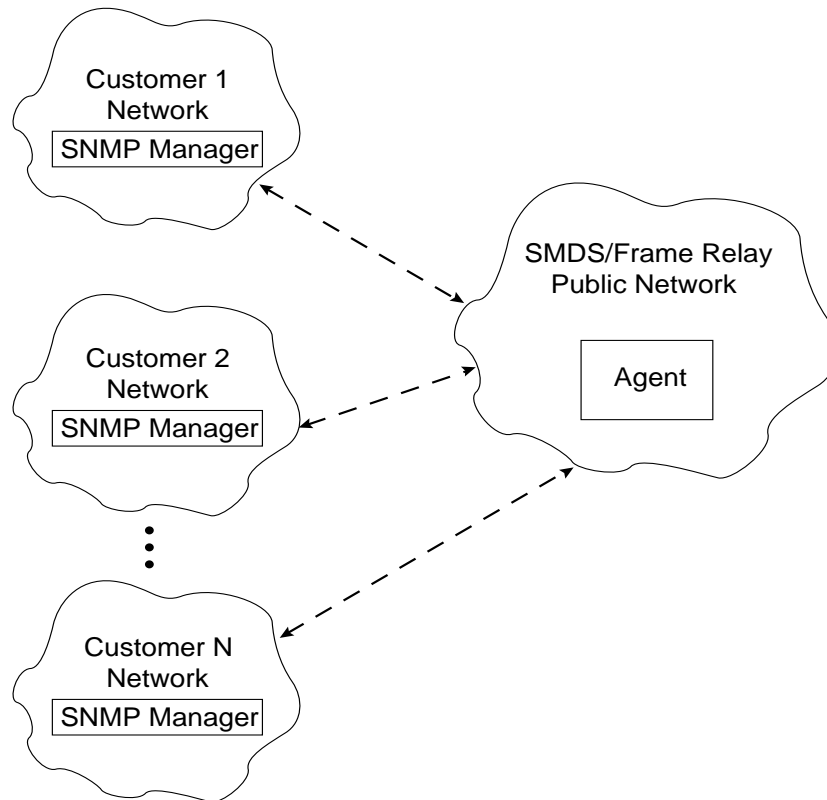


**Figure 1-1. Communication between SNMP Manager and Agent**

## Features of the *StarKeeper II* NMS SNMP Proxy Agent

The *StarKeeper II* NMS SNMP Proxy Agent enables customers who subscribe to SMDS or Frame Relay services to manage the public network interfaces they use.

The *StarKeeper II* NMS SNMP Proxy Agent communicates with many SNMP Managers, as illustrated in Figure 1-2. The customers' SNMP Managers manage their private data networks, as illustrated by the bubbles on the left side of Figure 1-2. With the SNMP Proxy Agent, customers can also monitor the network interfaces they use on the SMDS/Frame Relay public network, as shown by the right side of Figure 1-2.



---

**Figure 1-2. SNMP Managers and Proxy Agent**

### **What the SNMP Proxy Agent Provides**

---

The SNMP Proxy Agent allows an SNMP Manager to do the following:

- Retrieve performance measurements, error counts, and logs.
- Retrieve configuration information.
- Monitor the current status of an interface.
- Receive traps (alarms) on changes in the status of an interface.
- Receive authentication-failure traps.

## **Functional Capabilities of the SNMP Proxy Agent**

The *StarKeeper* II NMS SNMP Proxy Agent supports the following functional capabilities:

- Supports SMDS and certain Frame Relay services.  
SMDS interfaces are provided by the Access Interface (AI) modules on a BNS-2000 node. Frame Relay interfaces are provided by the M2-Frame Relay Module (M2-FRM) and the Frame Relay Module (FRM) on a BNS-2000 node and the FRM on a BNS-2000 VCS node.
- Supports the following SNMP primitives:
  - **Get Request PDU**—For each variable in the Get Request PDU, the SNMP Proxy Agent returns the value of the object.
  - **Get-Next Request PDU**—For each variable in the Get-Next Request PDU, the SNMP Proxy Agent returns the value of the next object in its MIB.
  - **Get Response PDU**—The SNMP Proxy Agent responds to the SNMP Manager's Get Request PDU or Get-Next Request with this PDU.
  - **Set Request PDU**—For each variable in the Set Request PDU, the SNMP Proxy Agent adds or deletes the value of the object.
  - **Trap PDU**—The SNMP Proxy Agent sends unsolicited traps to the SNMP Manager.
- Assures that only SNMP Managers that are registered in the SNMP database gain access to the SNMP Proxy Agent.
- Allows SNMP Managers to get information only for the interfaces that have been assigned to them.
- Restricts the number of requests that an SNMP Manager can make to a particular node. Refer to "Flow Control and Usage" in **Chapter 3, Database Management** for more details.
- Supports a subset of MIB II Objects  
The SNMP Proxy Agent supports the *system* group, a subset of the *interfaces* group, and the *snmp* group of objects.
- Supports Lucent Enterprise MIBs  
Lucent Enterprise MIBs identify additional network management information that an SNMP Manager can access. These MIBs are registered as enterprise vendor-specific MIBs and are described in **Appendix C, Supported SNMP MIB Objects and Traps**.



**NOTE:**

The Proxy Agent continues to support the superseded AT&T Enterprise MIBs, which were previously used by the product. Both

versions of the MIBs define the same objects; they simply name them differently. However, we recommend that the Lucent MIBs be used.

## Connectivity

Figure 1-3 is an example of connectivity between an SNMP Manager and the SNMP Proxy Agent. In this example, the SNMP Manager uses the public SMDS/Frame Relay network to gain access to the SNMP Proxy Agent that is connected to the same network. IP routing configuration must be made at the SNMP Manager and the SNMP Proxy Agent routers to ensure proper connectivity.

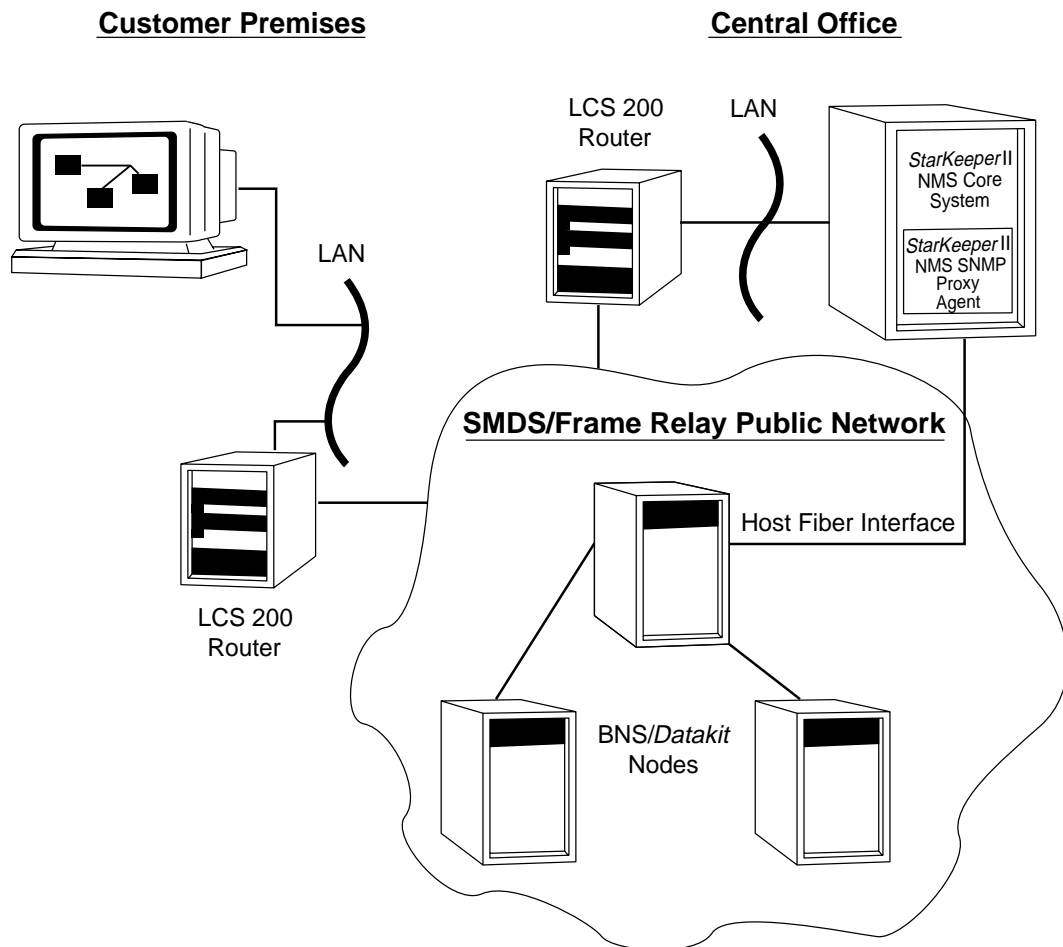


Figure 1-3. Sample Connectivity Diagram for SNMP Manager and Central Office

## Database Provisioning

---

The SNMP Proxy Agent accesses data in two databases: the *StarKeeper* II NMS Core System database and the SNMP database. The SNMP database is administered using the **sksnmpcf** or the **sksnmpxadm** command. Before SNMP Managers can access data about the SMDS or Frame Relay interfaces they use, the *StarKeeper* II NMS administrator must enter the following information in the SNMP database:

- System Information—identifies the system, where the SNMP Proxy Agent resides, and the organization or person to contact when there is a problem with the SNMP Proxy Agent.
- Network Interfaces—identifies the interfaces that the SNMP Proxy Agent supports. Each interface is assigned a unique Id called an *Interface Index*. An interface is identified by node name, module address, and port as well as other attributes of the interface.
- Customer Views and Customer View Members—a Customer View is a name for a set of Interface Indices that an SNMP Manager can manage. A member of a Customer View is an Interface Index.
- Subscriber Information—identifies the SNMP Manager by a unique name (*Subscriber Identifier*), IP address, and a set of passwords (*community strings*) and Customer Views, as well as several other attributes.

In addition there are two other types of information kept in the SNMP database:

- Flow Control Information—provides the capability to limit the number of requests an SNMP Manager can make to a particular node in a 15-minute interval.
- Usage Information—provides counts on the SNMP usage by a particular SNMP Manager.

See **Chapter 3, Database Management** for more information.

## Software Processes

---

This section describes how SNMP requests are processed and the internal processes of the SNMP Proxy Agent.

### How an SNMP Request is Processed

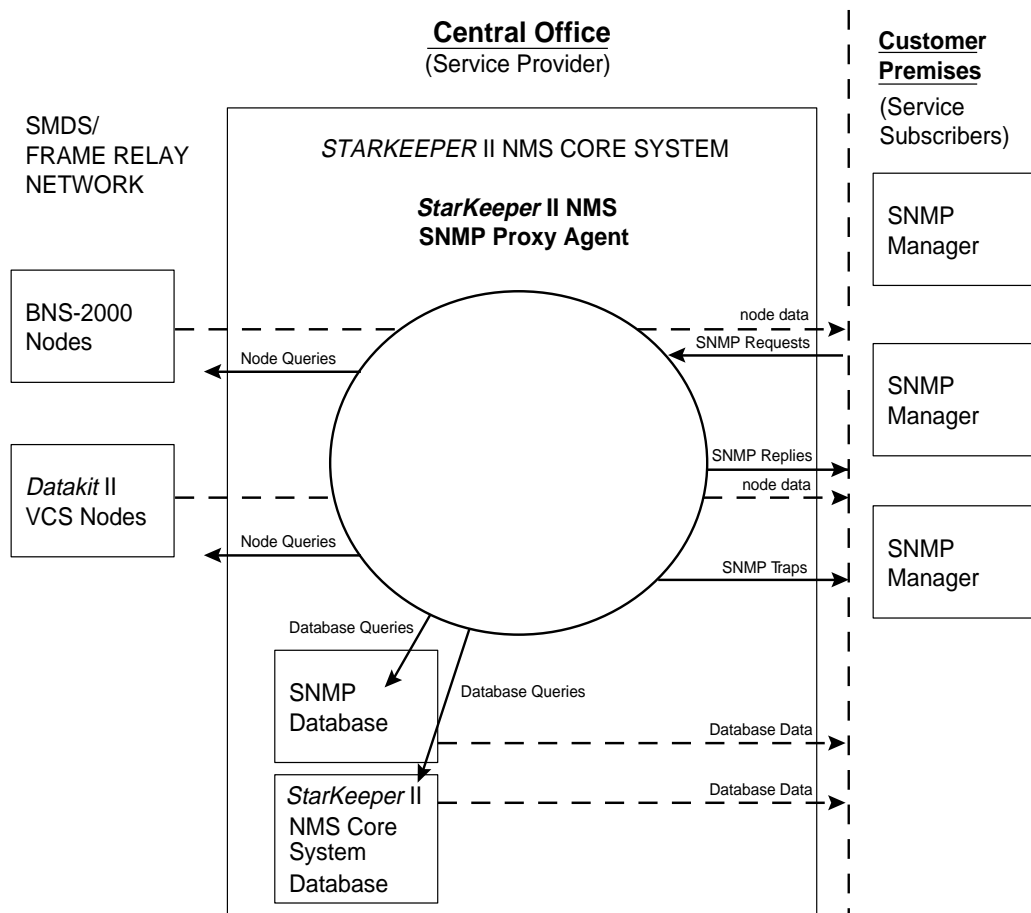
---

The SNMP Proxy Agent processes an SNMP request (Get Request PDU or Get-Next Request PDU) by completing these steps:

1. The SNMP Proxy Agent verifies that the request is from an SNMP Manager registered in the SNMP database.
2. The SNMP Proxy Agent verifies that the community string (password) in this SNMP request is valid.

3. The SNMP Proxy Agent verifies that the SNMP Manager has access capabilities for the requested information by checking the SNMP Manager's assigned Customer View in the SNMP database.
4. The SNMP Proxy Agent retrieves the appropriate network management information from the *StarKeeper II* NMS Core System database, SNMP database, or the node.
5. The SNMP Proxy Agent translates the information into SNMP format and sends it to the SNMP Manager.

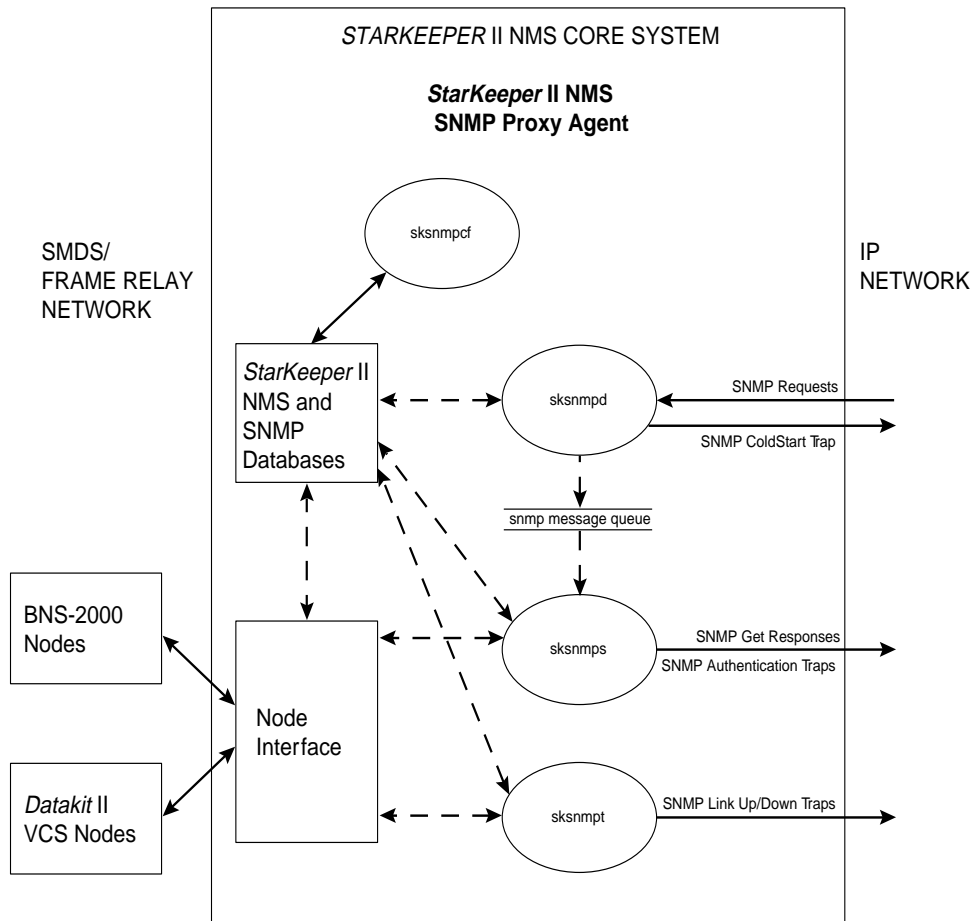
Figure 1-4 shows the SNMP Proxy Agent processes and the paths that information takes.



**Figure 1-4. SNMP Proxy Agent Processes**

## Daemon Processes

Figure 1-5 shows the SNMP Proxy Agent processes responsible for processing requests, responses, and traps.



**Figure 1-5. SNMP Proxy Agent Daemon Processes**

When you start the SNMP Proxy Agent, these processes are created:

- *sksnmpd*—A daemon process that receives SNMP packets from the network and queues them for subsequent processing by *sksnmps*.
- *sksnmps*—This is a server process that is responsible for getting information from the nodes, the *StarKeeper II NMS Core System* database, and the SNMP database, and sending the SNMP responses to the SNMP Managers.

- *sksnmpt*—This process is responsible for sending status traps to the SNMP Managers.

## **Where Do You Go From Here?**

---

<b>Situation</b>	<b>Reference</b>
To administer and maintain the system	<b>Chapter 2</b>
To administer the SNMP database	<b>Chapter 3</b>
To troubleshoot the system	<b>Chapter 4</b>
To copy the CNM MIBs	<b>Chapter 5</b>
To find out about the Proxy Agent commands	<b>Appendix A</b>
To see error and log messages	<b>Appendix B</b>
To find out about the supported MIBs	<b>Appendix C</b>
To learn about the database structure	<b>Appendix D</b>



---

# Administration and Maintenance Procedures

# 2

---

## System Administration Tasks and Responsibilities

---

This chapter gives you an overall view of the system administration and maintenance tasks that are your responsibility as administrator of the SNMP Proxy Agent.

Additional information on the commands mentioned in this section is provided in **Appendix A, User Reference**.

### Prerequisites

---

The *StarKeeper II* NMS SNMP Proxy Agent software can be installed on a Hewlett-Packard 715/75, 720, or 730. The *StarKeeper II NMS Planning Guide* lists all of the supported configurations.

Before installing the *StarKeeper II* NMS SNMP Proxy Agent software, make sure the following have been completed:

- The *StarKeeper II* NMS Core System software must be installed on the HP system before you add the SNMP Proxy Agent software.
- The appropriate software licenses must be installed on the *StarKeeper II* NMS Core System. Licensing is required for all *StarKeeper II* NMS software. Software licensing is required for all of the *StarKeeper II* NMS application packages before they can be loaded.

Before the SNMP Proxy Agent can support SMDS or Frame Relay interfaces, the appropriate licenses must be installed on the *StarKeeper II* NMS Core System. Separate *StarKeeper II* NMS SNMP Proxy Agent licenses are required for SMDS and for Frame Relay support.

For detailed information on software licensing, see the *StarKeeper II NMS Installation Guide*.

## Installation

---

For instructions on installing the *StarKeeper II* NMS SNMP Proxy Agent software, refer to *Getting Started with StarKeeper II NMS for Software Package System and Upgrade*. The SNMP Proxy Agent software is, by default, installed in the following base directory: `/usr2/SNMP`.

## Logging In

---

To use any of the SNMP Proxy Agent commands, log in to your *StarKeeper II* NMS Core System as **cnmsadm**. When you log in as **cnmsadm**, you inherit the SNMP Proxy Agent environment variables defined in the file `$(SNMP_ROOT)/etc/env/SNMP.vars`.

## Starting the SNMP Proxy Agent


---

Before starting the SNMP Proxy Agent for the first time, connect the *StarKeeper II* NMS Core System processor to a TCP/IP Local Area Network (LAN). Refer to the appropriate LAN manual for instructions on adding your *StarKeeper II* NMS Core System processor to a LAN.

### Procedure 2-1. Starting the SNMP Proxy Agent Processes

---

1. Log in as **cnmsadm**.
2. Verify that the *StarKeeper II* NMS Core System software is running. The *StarKeeper II* NMS Core System must be running when you start the SNMP Proxy Agent.
3. If you are starting the software for the first time, configure the SNMP database before allowing SNMP Managers to access the SNMP Proxy Agent. For details, see **Chapter 3, Database Management**.
4. Enter the **sksnmpstart** command.
5. When the system prompts you for a password, enter the **root** password.
6. After you enter the correct **root** password, the SNMP Proxy Agent daemon processes start. See **Appendix A, User Reference** for more detail on **sksnmpstart**.
7. To verify the operational state of the SNMP Proxy Agent, use **sksnmpstat**. See **Appendix A, User Reference** for details on **sksnmpstat**.

 **NOTE:**  
If you reboot the *StarKeeper II* NMS Core System, the SNMP Proxy Agent will automatically start up. You do not have to start it manually.

If you manually start up the *StarKeeper* II NMS Core System software, the *StarKeeper* II NMS Core System start-up script automatically invokes the **sksnmpstart** command to start up the SNMP Proxy Agent processes. You will need to enter the root password before the SNMP Proxy Agent processes can start up.

## **Stopping the SNMP Proxy Agent**

---

If you need to stop the Core System, you must stop the SNMP Proxy Agent first. If you do not stop the SNMP Proxy Agent first, the SNMP Proxy Agent terminates abnormally.

Use the **sksnmpstop** command to stop the SNMP Proxy Agent.

## **SNMP Database Configuration**

---

Before you allow SNMP Managers to access the SNMP Proxy Agent, register the SNMP Managers and identify the network interfaces in the SNMP database.

To configure the database, you use either the ASCII or Motif version of the SNMP Proxy Agent menus and screens. For more details, refer to **Chapter 3, Database Management**.

## **Synchronizing the SNMP Database**

---

In addition to the data you provide on network interfaces in the SNMP database, configuration and measurement information is needed from the nodes and from the *StarKeeper* II NMS Core System database. To collect this necessary data, you use two commands: **sksnmpndsync** and **sksnmpdbsync**. These commands are described in **Chapter 3, Database Management**.

## **Testing**

---

After you start the SNMP Proxy Agent and configure the SNMP database, test the SNMP Proxy Agent by using one of these commands: **sksnmpget**, **sksnmpnext**, **sksnmpset**, or **sksnmpwalk**. These commands are used to verify that the SNMP Proxy Agent can get and send MIB data. In terms of data retrieval, these tools act like SNMP Managers.

Your local system must be registered as a subscriber in the SNMP database in order to use these tools.

Try the following simple tests:

```
sksnmpget <address> <community> sysDescr.0
sksnmpnext <address> <community> system
sksnmpset <address> <community> [<object_id> <type> <value>]
sksnmpwalk <address> <community> system
```

## **BNS-2000 Initializing, Backing Up, and Restoring the SNMP Database**

---

The SNMP database *snmpdb* is created automatically when you install the SNMP Proxy Agent software. In rare instances, you may need to use the **sksnmpidb** command to recreate and reinitialize the database.



### **WARNING:**

*The **sksnmpidb** command removes all of the existing data in the database and also removes the database itself.*

### **Procedure 2-2. Reinitializing the Database:**

---

1. Back up the data in the SNMP database using the **sksnmpbdb** command.
2. Enter **sksnmpidb**.
3. Restore the database using the **sksnmprdb** command.

Backups should be done after the SNMP database is initially configured, and afterwards on a regular basis and after you make extensive changes. The SNMP database can be backed up to a directory that you specify or to tape.

## **Troubleshooting**

---

If problems occur, look in the log file, *\$SNMP\_LOG/snmp\_agent.log*. This log lists general events and errors that occur when the SNMP Proxy Agent is running, such as when the SNMP Proxy Agent was started or stopped, and if there have been any unauthorized or invalid attempts by SNMP Managers to access the SNMP Proxy Agent.

The log file is a flat ASCII file. To browse through the log file, you can use such UNIX tools as **pg**, **cat**, or **vi**.

Refer to **Chapter 4, Troubleshooting** and **Appendix B, Log and Error Messages** for more information.

### **Procedure 2-3. Provisioning Network Interfaces for CNM Service**

---

Before you can bring your network interfaces (module ports) into CNM service, complete the following:

1. Make sure that *StarKeeper II* NMS is running on your Core System.
2. Configure the nodes on your Core System so that *StarKeeper II* NMS can monitor the nodes.
3. Configure the ports (SMDS or Frame Relay) on your BNS-2000 or BNS-2000 VCS nodes.
4. Run **skload** and **cfg\_sync** to make sure that the *StarKeeper II* NMS Core System database is populated with the node's module port information.
5. Make sure that the Core System is collecting SMDS and Frame Relay measurements from the nodes.
6. If you have SMDS interfaces, use *Network Builder* to logically administer your SMDS network.
7. Enter the network interfaces, Customer Views, and subscribers into the SNMP database by using the **sksnmpcf** or the **sksnmpxadm** command.
8. Make sure that the SNMP Proxy Agent is running on your Core System.

### **Guidelines for Processing Set Requests**

---

CNM customers, using their SNMP Managers or SNMP management applications, can submit Set Requests. For example, an SNMP Manager can add or delete SMDS individual or group screening addresses. These requests are stored in the SNMP Proxy Agent database.

To process these requests and update the node and the *StarKeeper II* NMS Core System and SNMP Proxy Agent databases, you must execute the **sksnmpupd** command. Execute this command when you want to update the user's operating environment to reflect the stored Set Requests.

To examine the results of the **sksnmpupd** command, look at the file `$(SNMP_LOG)/snmp_set.log`. To browse through the log file, you can use such UNIX tools as **pg**, **cat**, or **vi**. If any requests fail, correct any error conditions and execute the **sksnmpupd** command again.

See **Appendix A, User Reference** for details on **sksnmpupd**. Refer to **Appendix C, Supported SNMP MIB Objects and Traps** to see which objects can be set.

## **Cleaning Up Log Files**

---

Periodically go to the log file directory, `$SNMP_LOG`, and check the size of your log files. Check log file `snmp_agent.log` and delete log entries that you no longer need. You can use the UNIX editor `vi` to clean up the log files.

---

## Overview

---

The SNMP Proxy Agent utilizes the INFORMIX® database management system. The SNMP database is created automatically when the SNMP Proxy Agent is installed.

Before SNMP Managers can access the SNMP Proxy Agent, you must administer the SNMP database so that it contains information about the SNMP Proxy Agent host system, the SNMP Managers, and the network interfaces that the SNMP Managers can manage.

This chapter provides the following:

- A description of how to register an SNMP Manager into the SNMP database
- A sample scenario that shows how a group of SNMP Managers could be defined in the SNMP database.
- Instructions for the menus and screens used to maintain the data in the SNMP database and to generate reports. This chapter describes the two versions of the user interface: the ASCII version and the Motif version. Refer only to the section that describes the version you are using.

## SNMP Proxy Agent Planning

---

Before entering information into the SNMP database, plan the system.

Use the planning forms in **Appendix A** to help you gather and organize the information you need. You may also find it helpful to read “Sample Scenario” to see how the information can be organized.

Once you have gathered and organized the appropriate information, use the following procedure.

### Procedure 3-1. Planning for the SNMP Proxy Agent

---

1. After the SNMP Proxy Agent is installed, enter System Information into the SNMP database.

System Information describes the host system for the SNMP Proxy Agent. For example, it includes the name of the host for the SNMP Proxy Agent and contact information the customer uses when there is a problem with the SNMP service.

2. Identify the network interfaces that the SNMP Proxy Agent supports.

Network interfaces are ports on a BNS-2000 node, BNS-2000 VCS node, or BNS-2000 MPC (concentrator). You must assign a unique numeric identifier (*Interface Index*) to each of the network interfaces that the SNMP Proxy Agent supports. An Interface Index can be any number from 1-999999.

The SNMP Managers use the Interface Indices in their SNMP queries to identify the Frame Relay or SMDS interfaces that they use.

Interface Indices do not have to be assigned sequentially. Rather than assign Interface Indices arbitrarily, develop a numbering plan that suits your needs.

3. Determine the set of network interfaces that an SNMP Manager can manage.

Each set of network interfaces that an SNMP Manager can manage is grouped together in a *Customer View*.

A Customer View is identified by a unique name and includes one or more Interface Indices as members.

When an SNMP Manager queries the SNMP Proxy Agent, it receives data for those Interface Indices that are included in its Customer View.

4. Identify each SNMP Manager in the SNMP database and assign a set of Customer Views and community strings to each SNMP Manager.

Each SNMP Manager must be identified by a unique Id called a *Subscriber Identifier*, its IP address, and a set of community strings (passwords) and Customer Views.

You can assign the same Customer View to more than one SNMP Manager.

Each SNMP Manager can have three community strings and three Customer Views:

- The *Read Community String* is used by the SNMP Manager when issuing SNMP Get and Get-Next Requests.
- The *Write Community String* is used by the SNMP Manager when issuing SNMP Set Requests.

- The *Trap Community String* is used by the SNMP Proxy Agent to send traps to the SNMP Manager. The administrator of the SNMP Manager should supply you with the Trap Community String.
- The *Read Customer View* defines the set of network interfaces that an SNMP Manager can manage.
- The *Write Customer View* defines the set of network interfaces that an SNMP Manager can modify (set).
- The *Trap Customer View* defines the set of network interfaces that an SNMP Manager receives traps for.

You can specify the same or different Customer Views for reading, writing, and for traps.

5. Make sure that the SNMP database is synchronized with the node and StarKeeper II NMS Core System database.

For more detail, refer to “Synchronizing the SNMP Database.”

## **Sample Scenario**

---

The following scenario illustrates how a group of management stations could be identified in the SNMP database.

Company XYZ has three divisions each with its own SNMP Manager: Sales, Marketing, and Headquarters. The company-owned LANs for Sales and Marketing are interconnected.

The Sales and Marketing SNMP Managers will manage the network interfaces for both Sales and Marketing. Headquarters monitors all network interfaces for Company XYZ.

The SNMP Manager in the Sales division connects to the network interface identified by Interface Index 10, Marketing uses Interface Index 11, and Headquarters uses Interface Index 100.

In addition, Marketing will receive traps for only the network interface it uses. Headquarters and the Sales division will receive traps for the same interfaces they monitor.

Look at the following sample Customer Profile Worksheet to see how the administrator of the SNMP Proxy Agent set up the SNMP database for Company XYZ so that these goals were met.

A blank Customer Profile Worksheet is included in **Appendix A, User Reference**.

SNMP PROXY AGENT —CUSTOMER PROFILE WORKSHEET						
CUSTOMER 'S NAME: <u>Company XYZ</u>						
NETWORK INTERFACES						
Interface Index	Node Name	Concen. module	trunk address	Module address	Port	Virtual Port
[ 10 ]	[ north/il/urbana ]	[ 0 ]	[ 0 ]	[10]	[1]	[0]
[ 11 ]	[ nj/suave/rico ]	[ 0 ]	[ 0 ]	[85]	[1]	[0]
[ 100 ]	[ east/ma/capecode ]	[ 0 ]	[ 0 ]	[95]	[3]	[0]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
CUSTOMER VIEWS						
Customer View	Description	Members				
[ XYZ Mrkt ]	[Marketing ]	[10				]
[ XYZ Sales ]	[Sales ]	[10,11				]
[ XYZ HQ ]	[Headquarters ]	[10,11,100				]
[ ]	[ ]	[ ]				]
[ ]	[ ]	[ ]				]
[ ]	[ ]	[ ]				]
[ ]	[ ]	[ ]				]
[ ]	[ ]	[ ]				]
SUBSCRIBER INFORMATION						
SUBSCRIBER IDENTIFIER: [XYZ Marketing ]						
IP Address: [135.22.45.89 ]						
		READ	WRITE	TRAP		
Community String: [public ] [public ] [public ]						
Customer View: [XYZ Sales ] [XYZ Sales ] [XYZ Mrkt ]						
SUBSCRIBER IDENTIFIER: [XYZ Sales Dept ]						
IP Address: [135.22.51.18 ]						
		READ	WRITE	TRAP		
Community String: [public ] [public ] [public ]						
Customer View: [XYZ Sales ] [XYZ Sales ] [XYZ Sales ]						
SUBSCRIBER IDENTIFIER: [XYZ HQ ]						
IP Address: [135.22.51.39 ]						
		READ	WRITE	TRAP		
Community String: [public ] [public ] [public ]						
Customer View: [XYZ HQ ] [XYZ HQ ] [ ]						

## **MASTERVIEW**

---

There is a Customer View called *MASTERVIEW* that is automatically created when the SNMP database is initialized. *MASTERVIEW*, as the name suggests, gives you a global view of your customer network. *MASTERVIEW* is a list of all the Interface Indices that you have defined. Each time you add or delete an Interface Index, *MASTERVIEW* is automatically updated.

If you assign *MASTERVIEW* to your local SNMP Manager, you will be able to get information about all the network interfaces (Interface Indices) in the SNMP database.

If you assign *MASTERVIEW* to any other SNMP Manager, that SNMP Manager can also get network management information about all the Interface Indices in the SNMP database.



**NOTE:**

The name *MASTERVIEW* is reserved for the system-generated Customer View. No other Customer View can be named *MASTERVIEW*.

## **Synchronizing the SNMP Database**

---

After you add a new network interface to the SNMP database or change an interface configuration, such as adding a new DLCI to a Frame Relay interface or changing the access class on a SMDS DS3 interface, additional customer network management information must be collected from the node and from the *StarKeeper II* NMS Core System database and stored in the SNMP database.

You use two commands to collect this data: **sksnmpndsync** and **sksnmpdbsync**. These commands run as cron processes, but can be executed manually to update the SNMP database immediately. These commands are described below and in **Appendix A, User Reference**.

### **sksnmpndsync**

---

The **sksnmpndsync** command retrieves SMDS Access Class information and Frame Relay DLCI numbers from the nodes for those Interface Indices that you have provisioned in the SNMP database. It retrieves the information and stores it in the SNMP database.

This information is automatically updated in the SNMP database by a cron process. As a cron process, this command retrieves information from the nodes for any **new** Interface Index that you have added to the SNMP database.

If you want the Access Class or DLCI information updated immediately, you must run this command manually to update the Interface Index in the SNMP database. You must run **sksnmpndsync** manually if you have done any of the following:

- Added a new Frame Relay or SMDS DS3 Interface Index
- Changed the SMDS Access Class for an SMDS port on the node that you have provisioned an Interface Index for.
- Added or deleted a Frame Relay DLCI number to a Frame Relay port on the node that you have provisioned an Interface Index for.

### **sksnmpdbsync**

---

The **sksnmpdbsync** command retrieves SMDS configuration information from the *StarKeeper II* NMS Core System database and assigns this information to the Interface Indices and Customer Views in the SNMP database. For example, this command retrieves such configuration information as SMDS individual/group addresses and SMDS individual/group screening addresses.

This command runs as a cron process. However, if you want the updates done immediately, you can run **sksnmpdbsync** manually. For more information on this cron process, refer to “Crontab for cnmsadm” in **Appendix A, User Reference**.

## **Flow Control and Usage Counts**

---

Some SNMP requests require the SNMP Proxy Agent to query the switch node directly for data, causing additional traffic and possible congestion on the node. The SNMP Proxy Agent enables you to restrict the number of queries made to a node and to get reports on SNMP traffic statistics.

### **Usage Counts**

---

The SNMP Proxy Agent maintains usage counts for each SNMP Manager and totals for all SNMP Managers. You can get reports on usage counts by using **sksnmpstat**, **sksnmpcf**, or **sksnmpxadm**.

**⇒ NOTE:**  
Only SNMP Managers that are CNM servers can request current cumulative SNMP usage statistics by using SNMP to query the *snmp* group of objects.

For each SNMP Manager, the SNMP Proxy Agent keeps track of several statistics, such as the number of times the SNMP Manager accesses the SNMP Proxy Agent, the number of SNMP errors an SNMP Manager has made, and the number of SNMP packets received and sent. These usage counts accumulate until you reinitialize them through **sksnmpcf** or **sksnmpxadm**.

The SNMP Proxy Agent also maintains totals for usage by all SNMP Managers from the time the SNMP Proxy Agent was last started. These usage counts are reinitialized when the SNMP Proxy Agent is restarted. These usage counts are known as *current\_totals*.

### **Flow Control Parameters**

---

The SNMP Proxy Agent enables you to restrict the number of queries made to a node by an individual SNMP Manager or by all SNMP Managers. For more information, see the description of the menu option in this chapter.

## **SNMP Proxy Agent Motif User Interface**

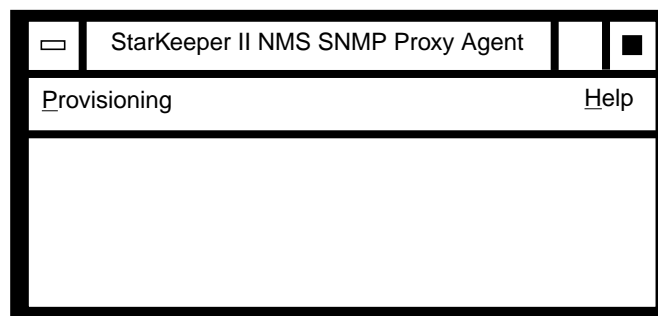
---

The SNMP Proxy Agent's Motif user interface utilizes the features of the OSF/Motif user environment and the Hewlett-Packard Visual User Environment (HP VUE). This section provides instructions for using this user interface.

### **Procedure 3-2. Accessing the SNMP Proxy Agent Motif User Interface**

---

1. Log in as **cnmsadm** in the Hewlett-Packard Visual User Environment (HP VUE).
2. Make sure that the environment variable DISPLAY is set correctly.
3. At the prompt, enter **sksnmpxadm**, or **sksnmpxadm &** to start the process in the background. The *StarKeeper II* NMS SNMP Proxy Agent main window displays.

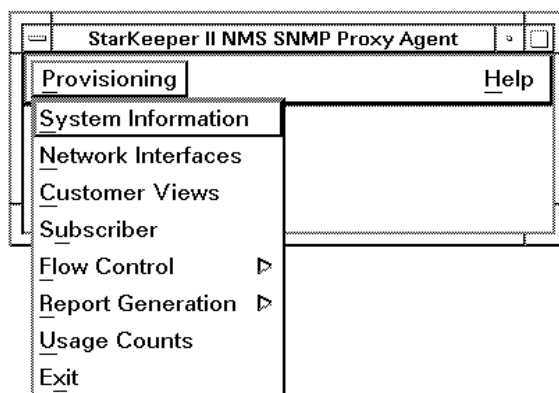


---

**Screen 3-1. StarKeeper II NMS SNMP Proxy Agent Main Window**

Choose an option from the *StarKeeper II NMS SNMP Proxy Agent* main window. There are two options available from the menu bar:

- **Provisioning**—provides access to the SNMP database.
  - **Help**—provides information on how to use this system. See **Using the Help Facility** for more information.
4. To access information in the SNMP database, choose **Provisioning**, then choose a menu option. Each of these menu options are described in detail later in this chapter.



### Screen 3-2. Provisioning Menu

- Select **System Information** to enter or verify general system information, such as the name of the host system that provides the SNMP Proxy Agent.
- Choose **Network Interfaces** to enter or verify all of the network interfaces that the SNMP Proxy Agent supports. Network interfaces are module ports on a BNS-2000 node, BNS-2000 VCS node, or BNS-2000 MPC (or concentrator). An ICI trunk is also a network interface. Each supported network interface must be assigned a unique Interface Index.
- Choose **Customer Views** to enter or verify the sets of Interface Indices that can be managed.
- Select **Subscriber** to enter or verify each SNMP Manager and assign Customer Views that the SNMP Manager can access.
- Use **Flow Control** to monitor and restrict the number of queries made to a node for a single subscriber or for all subscribers.
- Select **Report Generation** to get usage and error reports.
- Select **Usage Counts** to reset a subscriber's usage and error counts to zero.

- Choose **Exit** to exit the *StarKeeper II* NMS SNMP Proxy Agent Motif user interface.

## Forms Usage

Most menu options lead to forms where you can enter, maintain, and verify data in the SNMP database. The following procedure briefly describes the layout of most forms and describes how to add, change, and delete information in the SNMP database.

### Procedure 3-3. Using Forms (General Method)

1. To access a form, choose **Provisioning** from the menu bar in the *StarKeeper II* NMS SNMP Proxy Agent main window, then select one of the first four options.

When a form is displayed, the cursor is in the first field where information can be entered. Enter data using the keyboard.

Forms may contain one or more scroll boxes and an input area where data can be modified, as shown below.

The screenshot shows a window titled "SNMP Proxy Agent: Network Interfaces" with a menu bar containing "Edit", "View", and "Help". The main content area is titled "NETWORK INTERFACE ADMINISTRATION" and contains a table of network interfaces. Below the table are several input fields for configuration, including "Interface Index", "Node Address", "MTU", "Speed", "Contact", "Location", "Description", and "Subscriber".

IfIndex	Conc.	Mod.	Port	Vport	Mod Type	Nodename
1	0	30	4	0	frm_v35	east/ma/capecod
7	0	25	2	0	frm_v35	east/ma/capecod
9	0	2	3	0	frm_v35	north/11/urbana
10	0	10	1	0	frm_t1	north/11/urbana
11	0	85	1	0	frm_v35	nj/suave/rico
12	0	25	3	0	frm_v35	east/ma/capecod

Input fields below the table:

- Interface Index:
- Node Address:  list...
- MTU:  Speed:
- Contact:
- Location:
- Description:
- Subscriber:

Screen 3-3. Sample Form

A scroll box contains items that are presently configured in the SNMP database. For example, the scroll box in Screen 3-3 displays the network interfaces that are in the SNMP database. Use the scroll bars to move through the list.

The input area is used to add, delete, or update information in the SNMP database. When you select an item from a scroll box, the complete configuration information for the item displays in the input area.

2. There are four types of actions that you can complete on a form:
  - To add a record, the input area of the form must be blank. If the input area is not blank, choose **Edit** from the menu bar, then choose **Clear Form**. Supply data in all of the fields in the input area. To add the record to the SNMP database, choose **Edit** from the menu bar, then choose the **Add** option.
  - To change a record, select it from the scroll box, then modify the data in the input area. To update the SNMP database with your changes, choose **Edit** from the menu bar, then choose the **Update** option.
  - To delete a record, select it from the scroll box and review the information in the input area to verify that the right record has been selected. To delete the record, choose **Edit** from the menu bar, then choose the **Delete** option.
  - To cancel any changes made to the form before choosing the **Add**, **Update**, or **Delete** options, choose **Edit** from the menu bar, then choose the **Clear Form** option.

When you successfully modify the SNMP database, the input area of the form clears automatically.

3. To exit the window, choose **Edit**, then **Exit**.

### **Special HP VUE and Motif Capabilities**

---

This section highlights the HP VUE and Motif capabilities that are used in this version of the SNMP Proxy Agent user interface. For further information on HP VUE and Motif, refer to your HP documentation.

### **Keyboard Shortcuts**

You can use the mouse to move the cursor or you can use the keys on your keyboard.

- To choose a menu item from the menu bar with the keyboard, press and hold down A and type the underlined letter in the menu name. For example, to display the **Provisioning** menu, press **Ap**.

- To choose a menu item from a pull-down menu, type only the underlined letter. For example, to choose **Network Interfaces** from the **Provisioning** menu, type **n**.
- To move the cursor between fields on a form, use T.
- Pop-up command windows have a default action chosen from among the command buttons positioned at the base of the window. The default button is highlighted with a bold outline. You can press J to execute a window's default action.

### Using the Help Facility

The *StarKeeper* II NMS SNMP Proxy Agent provides on-line help. To get help from a window, select **Help** from the menu bar. The menu provides these features:

- information on the help facility (On Help)
- help for the current window (On Window)
- a table of contents for the current application's Help (Contents)
- displays the version number of the SNMP Proxy Agent software (On Version)

### Error Messages

Error messages are displayed in pop-up windows. An error must be corrected before the SNMP database can be successfully modified.

### System Information

---

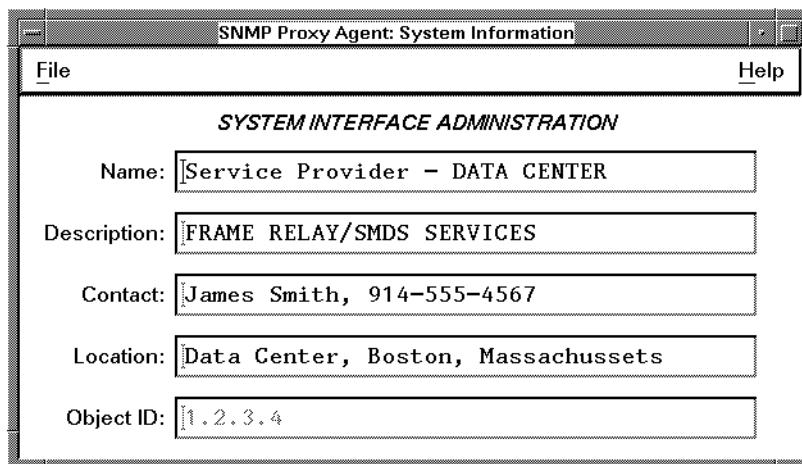
Choose System Information to enter or verify general system information. System Information should be entered right after the SNMP Proxy Agent is installed.

---

**Procedure 3-4. Entering or Verifying System Information**

---

1. In the *StarKeeper II* NMS SNMP Proxy Agent main window, choose **Provisioning**, then choose **System Information**. The System Information Administration form displays.



The screenshot shows a window titled "SNMP Proxy Agent: System Information". The window has a menu bar with "File" on the left and "Help" on the right. Below the menu bar, the text "SYSTEM INTERFACE ADMINISTRATION" is centered. The form contains five input fields, each with a label to its left: "Name:" with the value "Service Provider - DATA CENTER"; "Description:" with the value "FRAME RELAY/SMDS SERVICES"; "Contact:" with the value "James Smith, 914-555-4567"; "Location:" with the value "Data Center, Boston, Massachussets"; and "Object ID:" with the value "1.2.3.4".

---

**Screen 3-4. System Information Administration Form**

---

2. Use this form to verify the System Information and to make modifications. Use the following field descriptions to complete this form:
  - **Name**  
The name of the host system that provides the SNMP Proxy Agent.
  - **Description**  
The description of the services offered by the SNMP Proxy Agent.
  - **Contact**  
The name of the system's administrative contact and information on how to contact this person.
  - **Location**  
The location of the host system that provides the SNMP Proxy Agent.
  - **Object Id**  
The information in this field cannot be modified by the user. An entry is supplied automatically when the database is initialized.
3. After modifying the system information, choose **Edit**, then choose **Update System Information**.

- To exit the form, choose **Edit** from the menu bar, then choose **Exit**.

## Network Interfaces

Network interfaces are module ports on a BNS-2000 node, BNS-2000 VCS node, or BNS-2000 MPC (concentrator). An ICI trunk is also a network interface. Use this option to identify all the interfaces that the SNMP Proxy Agent supports. Each interface is identified by a unique Interface Index.

To access the Network Interface Administration form, in the *StarKeeper II* NMS SNMP Proxy Agent window, choose **Provisioning**, then choose **Network Interfaces**. Use this form to add, delete, or change network interfaces.

The screenshot shows a window titled "SNMP Proxy Agent: Network Interfaces" with a menu bar containing "Edit", "View", and "Help". The main content area is titled "NETWORK INTERFACE ADMINISTRATION" and contains a table with the following data:

Interface Index	Conc.	Mod.	Port	Vport	Mod Type	Nodename
1	0	30	4	0	frm_v35	east/ma/capecod
7	0	25	2	0	frm_v35	east/ma/capecod
9	0	2	3	0	frm_v35	north/il/urbana
10	0	10	1	0	frm_t1	north/il/urbana
11	0	85	1	0	frm_v35	nj/suave/rico
12	0	25	3	0	frm_v35	east/ma/capecod

Below the table are several input fields:

- Interface Index:
- Node Address:
- MTU:  Speed:
- Contact:
- Location:
- Description:
- Subscriber:

**Screen 3-5. Network Interface Administration Form**

The Network Interface Administration form has two sections:

- Use the scroll box in the upper section to select a network interface to change or delete.

- Use the lower section, the input area, to add a network interface or to change or delete a network interface that you have chosen from the scroll box.

## Fields on the Network Interface Administration Form

These fields appear on the Network Interface Administration form:

- **Interface Index**

The Interface Index identifies either a port on an SMDS or a Frame Relay module on a BNS-2000 node, BNS-2000 VCS node, BNS-2000 MPC (concentrator), or an ICI trunk on a BNS-2000 node. Each Interface Index must be unique.

Valid Interface Indices are 1 - 999999.

- **Node Address**

The Node Address contains the following information:

- **Conc. (Concentrator trunk module address)**

For a Frame Relay interface located on a concentrator, this is the module address (slot number) of the concentrator trunk *on the node* that connects the node to the concentrator.

For an SMDS interface or a Frame Relay interface located on a node, the default entry, **0** appears to indicate that the Concentrator trunk module address is not applicable.

- **Mod. (Node/Concentrator module address)**

The module address (slot number) of the SMDS or Frame Relay interface. If the network interface is on the node, select the interface's module address on the node. If the interface is on a concentrator, select the interface's module address on the concentrator.

- **Port**

The port number for the SMDS or Frame Relay interface. The port number is 0 if the interface is an ICI trunk.

- **Vport**

The virtual port number for the M2 Frame Relay interface. If the interface is not an M2 Frame Relay interface, **0** appears in this part of the field.

- **Mod Type (Interface Type)**

The valid interface types are listed below.

For **SMDS Service**:

- ait1 (DS1 Access Interface module)

- ait3p (DS1 Access Interface module)
- ait3 (DS3 Access Interface module)
- aie1 (European Access Interface module)
- aie3 (European Access Interface module)
- trkt3i (ICI trunk module)

**For Frame Relay Service:**

- frm (Frame Relay R3.0 module)
- frm\_t1 (Frame Relay t1 module)
- frm\_e1 (Frame Relay e1 module)
- frm\_v35 (Frame Relay v35 module)
- m2\_frm\_t1 (Frame Relay M2 t1 module)
- m2\_frm\_e1 (Frame Relay M2 e1 module)

— **Nodename**

The name of the node where the interface is located.

If the interface is on a concentrator and the concentrator is connected to the node by a concentrator trunk module, select the node name where the concentrator trunk module is located.

■ **MTU**

The Maximum Transfer Unit (MTU) for this network interface.

For SMDS, the interface MTU is 9188 bytes. For Frame Relay, the MTU can be from 262 to 4096 bytes. This field is not modifiable.

■ **Speed**

The speed of the network interface, represented by the number of bits per second. For example, at DS1 rate, the number of bits per second is 1,544,000. This field is not modifiable.

■ **Contact**

The name of the network interface's administrative contact and information on how to contact this person.

■ **Location**

A description of the physical location of the network interface.

■ **Description**

This is a comment that describes the interface, such as SMDS-T1 or Frame Relay T1.

■ **Subscriber**

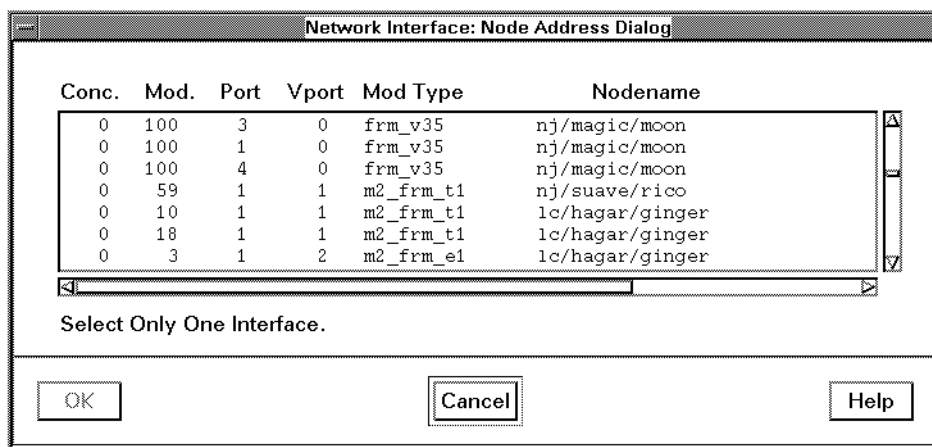
The subscriber that uses this interface. You may enter a Subscriber Identifier or some other description for the subscriber. If there is no subscriber assigned to this interface, you can enter **none**.

## Network Interface Provisioning

Before you provision an Interface Index to a module port, make sure that the node name, concentrators, slot and port numbers are configured in the *StarKeeper II* NMS Core System database. The slot and port information is entered into the Core System database by using the Core System command, **skload**.

### Procedure 3-5. Adding a Network Interface


1. In the *StarKeeper II* NMS SNMP Proxy Agent main window, choose **Provisioning**, then choose **Network Interfaces**. The Network Administration form appears.
2. Enter a unique Interface Index for this network interface.
3. To enter a Node Address, click on the **list...** button next to the Node Address field. The list of available node addresses appears in the Network Interface: Node Address Dialog, as shown below.



### Screen 3-6. Network Interface: Node Address Dialog

4. Select the network interface to be associated with the Interface Index, then select **OK**. The Node Address Dialog closes and you return to the Network Interface Administration form. The Node Address, MTU, and Speed fields are automatically filled in on the Network Interface Administration form.
5. Fill in the remainder of the Network Interface data. Refer to the definitions in "Fields on the Network Interface Administration Form."

6. After completing the form, choose **Edit** from the menu bar, then choose **Add Interface Index**.
7. When the database is successfully updated, the input area of the form is cleared and the new Interface Index is displayed in the scroll list.
8. To exit the form, choose **Edit**, then choose **Exit**.


 **NOTE:**  
When you add a network interface, it is automatically assigned to Customer View, *MASTERVIEW*.

---


### Procedure 3-6. Updating a Network Interface

---

1. In the *StarKeeper II* NMS SNMP Proxy Agent window, choose **Provisioning**, then choose **Network Interfaces**. The Network Interfaces Administration form appears.
2. Select an Interface Index to change from the scroll box. The information for the Interface Index appears in the input area in the lower section of the form.

 **NOTE:**  
By default, the list of network interfaces appears in Interface Index order. To select another method for sorting this list, choose **View** from the menu bar, then choose one of the sorting methods: Interface Index, node name, or module type.

3. Type in the new or corrected data. Refer to the definitions in “Fields on the Network Interface Administration Form.”

 **NOTE:**  
Information in the Interface Index, Node Address, MTU, and Speed fields cannot be modified. To change the Interface Index or the Node Address associated with the Interface Index, delete the existing Interface Index and add a new one.

4. After modifying a network interface, choose **Edit**, then choose **Update Network Interface**.  
  
When the database is successfully updated, the input area of the form is cleared.
5. To exit the form, choose **Edit**, then choose **Exit**.

---

### Procedure 3-7. Deleting a Network Interface

---

1. In the *StarKeeper II* NMS SNMP Proxy Agent main window, choose **Provisioning**, then choose **Network Interfaces**. The Network Interfaces Administration form appears.

2. Select an Interface Index to delete from the scroll box. The information for the Interface Index appears in the input area in the lower section of the form.



**NOTE:**

By default, the list of network interfaces appears in Interface Index order. To select another method for sorting this list, choose **View** from the menu bar, then choose one of the sorting methods: Interface Index, node name, or module type.

3. To delete the Interface Index, choose **Edit** from the menu bar, then choose **Delete Interface Index**. You are asked to confirm the deletion.
4. To confirm the deletion, choose **Yes**.  
When the database is successfully updated, the input area of the form is cleared.
5. To exit the form, choose **Edit**, then choose **Exit**.



**NOTE:**

When an Interface Index is deleted, the Interface Index is deleted from *MASTERVIEW* and from all other Customer Views that have this Interface Index as a member.

### Customer Views

---

A Customer View identifies a set of Interface Indices that an SNMP Manager can manage.

To access the Customer View Administration form, in the *StarKeeper II* NMS SNMP Proxy Agent window, choose **Provisioning**, then choose **Customer**

**Views.** Use this form to add, delete, or change information about a Customer View.

Customer View	Description
MASTerview	MASTerview has all defined ifindices
XYZMrkt	Marketing for Company XYZ
XYZSales	Sales for Company XYZ

Customer View:  Description:

IfIndex	Conc.	Mod.	Port	Vport	Mod Type	Nodename

### Screen 3-7. Customer View Administration Form

The Customer View Administration form has two sections:

- Use the scroll box in the upper section to select a Customer View to change or delete. When you select a Customer View, the current information for the Customer View appears in the input area.
- Use the input area, to create a Customer View or to delete one. The input area can also be used to change the description for a Customer View and to verify the list of members of a Customer View.

### Fields on the Customer View Administration Form

These fields appear on the Customer View Administration form:

- **Customer View**

Each Customer View must be a unique ID in the SNMP database. Customer View IDs are alphanumeric character strings. Customer View IDs can also contain hyphens (-) and underline characters (\_). Spaces are not allowed. For example, valid Customer View IDs are a\_b\_c, a-b, a-b\_1\_2.

- **Description**

A description of the Customer View.

- **List of Members**

The Interface Indices that are members of this Customer View.



**NOTE:**

Members can only be added to a Customer View **after** the Customer View has been added to the SNMP database.

### **Procedure 3-8. Adding a Customer View**

---

1. In the *StarKeeper II* NMS SNMP Proxy Agent main window, choose **Provisioning**, then choose **Customer Views**. The Customer View Administration form appears.
2. Enter a unique name for the Customer View and a description.
3. To add the Customer View, choose **Edit** from the menu bar, then choose **Add Customer View**.  
  
When the database is successfully updated, the input area of the form is cleared.
4. To exit the form, choose **Edit**, then choose **Exit**.

### **Procedure 3-9. Updating a Customer View**

---

This procedure covers three types of modifications:

- adding members to a Customer View
- deleting members from a Customer View
- modifying the description of a Customer View

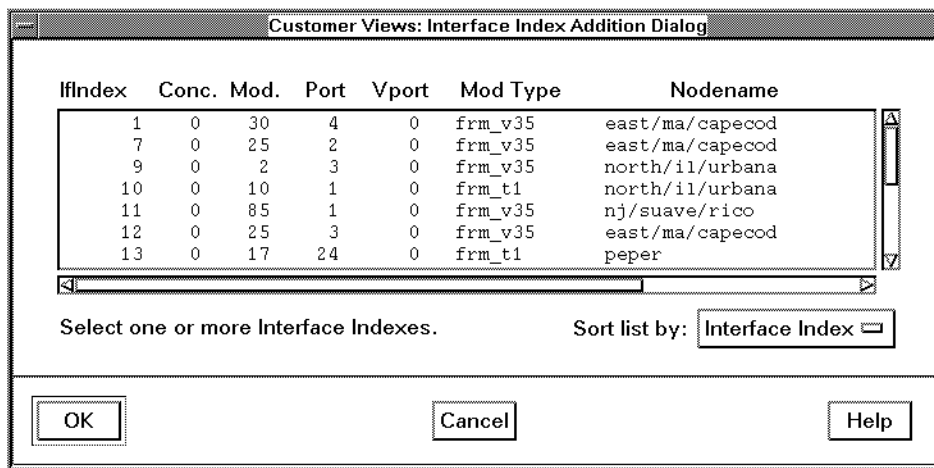
Before modifying a Customer View, make sure the following have been completed:

- The Customer View is in the SNMP database. To create a Customer View, see the procedure above, “Adding a Customer View.”
- Each network interface that will be a member of the Customer View must have an Interface Index. To add Interface Indices, see “Adding a Network Interface” earlier in this chapter.

To update a Customer View, follow these steps:

1. In the *StarKeeper II* NMS SNMP Proxy Agent main window, choose **Provisioning**, then choose **Customer Views**. The Customer View Administration form appears.

2. Select a Customer View from the scroll box. The current information for the Customer View appears in the input area of the form.  
If there are no interface indices in this Customer View, you receive a message that states that there are no interface indices configured for this Customer View. Choose **OK**.
3. Change the Customer View description, if needed.
4. To add members to the Customer View, follow these steps:
  - a. Choose **Edit** from the menu bar, choose **Administer Customer View**, then choose **Add Interface**. The Customer Views: Interface Index Addition Dialog displays the list of Interface Indices that can be added to the Customer View.



**Screen 3-8. Customer View: Interface Index Addition Dialog**

- b. Click on each Interface Index that you want to include in the Customer View, then select **OK**.



**NOTE:**

By default, the list of network interfaces appears in Interface Index order. To select another method for sorting this list, click and hold down the left mouse button on the field next to **Sort List by** and select a method for sorting.

- c. To close the dialog box, select **Cancel**.
5. To delete members from the Customer View, follow these steps:
  - a. From the Customer Views Administration form, choose **Edit** from the menu bar, choose **Administer Customer View**, then choose **Delete Interface**.

- b. Select one or more Interface Indices from the scroll box, then select **OK**.
  - c. To close the dialog box, select **Cancel**.
6. When you finish modifying the Customer View, choose **Edit** from the menu bar, then choose **Update Customer View**.  
When the database is successfully updated, the input area of the form is cleared.
7. To exit the form, choose **Edit**, then choose **Exit**.

### **Procedure 3-10. Deleting a Customer View**

---

1. In the *StarKeeper II NMS SNMP Proxy Agent* main window, choose **Provisioning**, then choose **Customer Views**.
2. Select the Customer View to delete from the scroll box. The information for the Customer View appears in the input area of the form.
3. To delete the Customer View, choose **Edit** from the menu bar, then choose **Delete Customer View**.  
When the database is successfully updated, the input area of the form is cleared.
4. To exit the form and return to the *StarKeeper II NMS SNMP Proxy Agent* main window, choose **Edit**, then choose **Exit**.



**NOTE:**

When a Customer View is deleted, all Interface Indices associated with this Customer View will be deleted from this Customer View.

## **Subscriber**

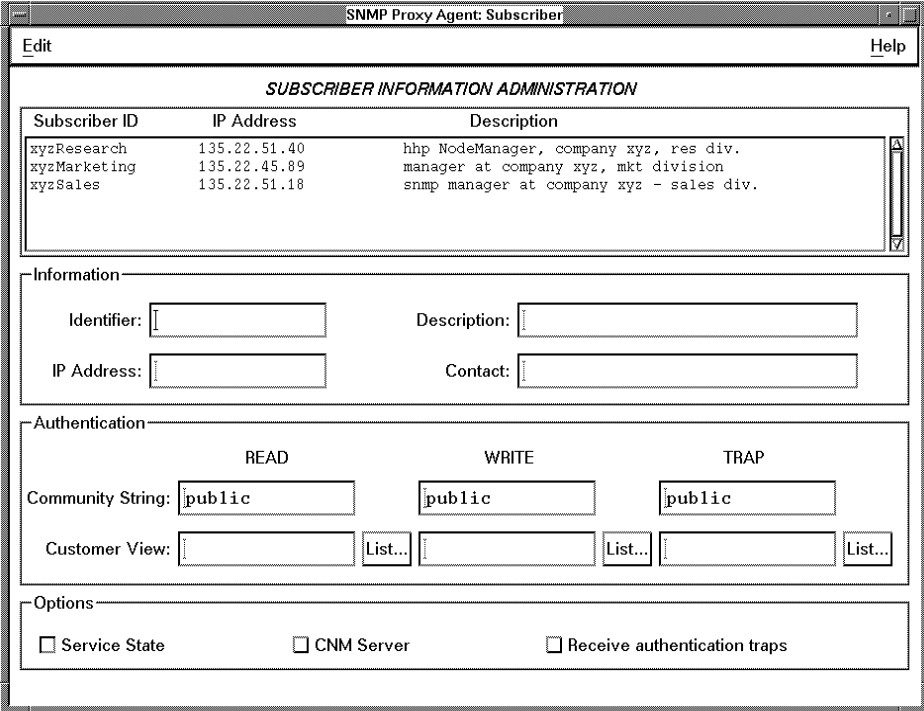
---

Each SNMP Manager must be identified by a unique Subscriber Identifier, its IP address, and a set of community strings and Customer Views.

To access the Subscriber Administration form, in the *StarKeeper II NMS SNMP Proxy Agent* window, choose **Provisioning**, then choose **Subscriber**. Use this form to add, delete, or change subscriber information.

The Subscriber Information Administration form has two sections:

- Use the scroll box in the upper section to select a subscriber to change or delete. When you select a subscriber, the current information for the subscriber appears in the lower section of the form.
- Use the lower section, the input area, to add, change or delete subscriber information.



**SUBSCRIBER INFORMATION ADMINISTRATION**

Subscriber ID	IP Address	Description
xyzResearch	135.22.51.40	hnp NodeManager, company xyz, res div.
xyzMarketing	135.22.45.89	manager at company xyz, mkt division
xyzSales	135.22.51.18	snmp manager at company xyz - sales div.

**Information**

Identifier:  Description:

IP Address:  Contact:

**Authentication**

READ                      WRITE                      TRAP

Community String:        

Customer View:  List...     List...     List...

**Options**

Service State     CNM Server     Receive authentication traps

**Screen 3-9. Subscriber Information Administration Form**

### Subscriber Information Administration Form Fields

These fields appear on the Subscriber Information Administration form:

- **Identifier**

A Subscriber Identifier is a name that identifies the SNMP Manager that will interface with the SNMP Proxy Agent.

Valid Subscriber Identifiers are alphanumeric character strings. The identifiers can also contain hyphens (-) and underline characters (\_). For example, **abc\_123** is a valid Subscriber Identifier. Spaces are not allowed.

- **Description**

A description of the SNMP Manager.

- **IP Address**

The IP address is the address of the host system where the subscriber's SNMP Manager is residing.

The IP address must be in the correct IP format. For example, 135.55.22.82 is a correctly formatted IP address.

The SNMP Proxy Agent uses this IP address to verify that an SNMP Manager is allowed access to the SNMP Proxy Agent and to send responses to requests.

- **Contact**

The name of the administrative contact responsible for the SNMP Manager and information on how to contact this person.

- **Community String (READ)**

The community string that this SNMP Manager will use in its SNMP queries to read (GET) data from the SNMP Proxy Agent.

The default Read Community String is **public**.

- **Community String (WRITE)**

The community string that this subscriber's SNMP Manager will use in its SNMP queries to change (SET) data on the *StarKeeper* II NMS Core System.

The default Write Community String is **public**.

- **Community String (TRAP)**

The community string that the SNMP Proxy Agent will need to send traps (alarms) to this SNMP Manager.

The default Trap Community String is **public**.

- **Customer View (READ)**

The Customer View that contains the interfaces that the SNMP Manager can access.



**NOTE:**

The Customer View (READ), Customer View (WRITE), and the Customer View (TRAP) must be in the SNMP Proxy Agent database. For more information, see "Adding a Customer View."

- **Customer View (WRITE)**

The Customer View that contains the interfaces that the SNMP Manager can change.

- **Customer View (TRAP)**

The Customer View that contains the interfaces that the SNMP Manager will receive traps for.

A trap will be sent to this SNMP Manager if an alarm occurs on any of the network interfaces that have been assigned to this Trap Customer View.



**NOTE:**

You can use the same Customer View for reading, writing, and for traps or you can specify different ones.

■ **Service State**

Indicate if this SNMP Manager is in service or out of service. If you want the SNMP Proxy Agent to respond to this SNMP Manager's requests, then turn on this feature. If this SNMP Manager is currently not operating or if you do not want the SNMP Proxy Agent to accept SNMP requests, then turn off this feature to indicate that this SNMP Manager is out of service.

By default, the Service State is set to **in**.

■ **CNM Server**

If the SNMP Manager is a Customer Network Management server, turn on this feature.

The SNMP Proxy Agent sends authentication traps to CNM servers whenever an SNMP request is received from an invalid SNMP Manager. Also, only CNM servers can request the SNMP Proxy Agent for current SNMP usage statistics.

By default, the CNM Server is set to **off**.

■ **Receive authentication traps**

An authentication-failure trap is generated whenever this SNMP Manager initiates a request to the SNMP Proxy Agent with an invalid community string. If the SNMP Manager should receive authentication traps, turn this feature on.

By default the Receive authentication traps is set to **off**.

## Adding a Subscriber

Follow these guidelines when you use this option:

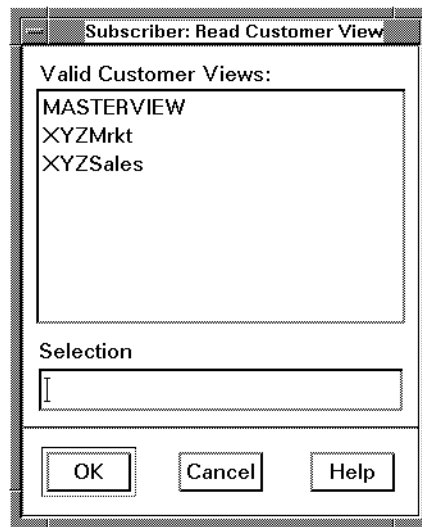
- Each Subscriber Identifier must be unique.
- Each IP Address must be unique in the database. You cannot enter an IP Address if it already exists in the SNMP database.
- Customer Views must have been defined previously in the SNMP database.
- The same Customer View can be assigned to Read, Write, and Trap Customer Views.

### Procedure 3-11. Entering a Subscriber

---

1. In the *StarKeeper II* NMS SNMP Proxy Agent window, choose **Provisioning**, then choose **Subscriber**. The Subscriber Information Administration form appears.
2. To complete the form, refer to the "Fields on the Subscriber Information Administration Form."

3. In the Information box, enter a unique identifier for the subscriber, a description, the subscriber's IP address and the name of the contact person at the subscriber's site.
4. In the Authentication box, enter the community strings for Read, Write and Trap. To enter a Customer View, click on the **List...** button next to the Customer View field. The Customer View Dialog appears, as shown below:



---

**Screen 3-10. Subscriber: Read Customer View Dialog**

5. Select a Customer View from the list or type one in the selection box, then select **OK**. The Customer View dialog closes and you return to the Subscriber Information Administration form.
6. To modify the entries in the Read, Write or Trap Customer View fields, use the **List...** button next to each of those fields.
7. In the Options box, select the options that apply to this subscriber.
8. After completing the form, choose **Edit** from the menu bar, then choose **Add Subscriber**.  
  
When the database is successfully updated, the input area of the form is cleared.
9. To exit the form, choose **Edit**, then choose **Exit**.

---

**Procedure 3-12. Updating a Subscriber**

1. In the *StarKeeper II* NMS SNMP Proxy Agent window, choose **Provisioning**, then choose **Subscriber**.

2. Select the subscriber to change from the scroll box. The information for the subscriber appears in the input area of the form.
3. Type in the new or corrected data. Refer to the definitions in “Fields on the Subscriber Information Administration Form.”



**NOTE:**

The IP address cannot be changed. If a subscriber has a new IP address, add the subscriber with the new address, then delete the subscriber information for the old IP address.

4. After modifying the subscriber information, choose **Edit**, then choose **Update Subscriber**.

When the database is successfully updated, the input area of the form is cleared.

5. To exit the form, choose **Edit**, then choose **Exit**.

**Procedure 3-13. Deleting a Subscriber**

---

1. In the *StarKeeper* II NMS SNMP Proxy Agent window, choose **Provisioning**, then choose **Subscriber**.
2. Select the subscriber to delete from the scroll box. The details for the subscriber appear in the input area of the form.
3. To delete the subscriber, choose **Edit** from the menu bar, then choose **Delete Subscriber**.

When the database is successfully updated, the input area of the form is cleared.

4. To exit the form, choose **Edit**, then choose **Exit**.

**Flow Control**

---

Use this option to restrict the number of queries made to a node. You can restrict the number of queries to a node for a single subscriber or for all subscribers.

**Allowed Requests by All Subscribers**

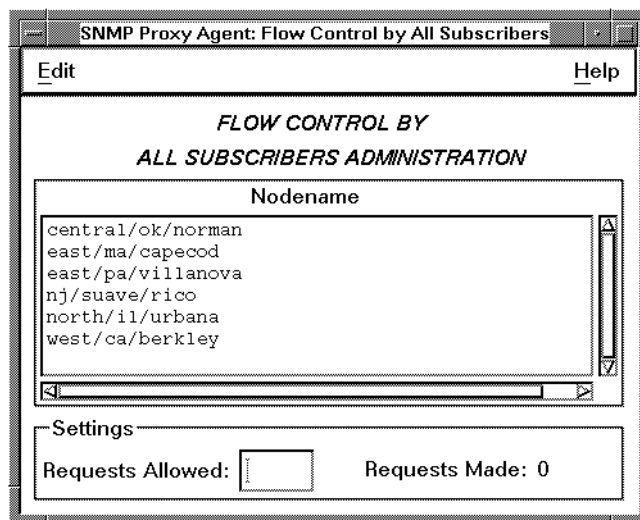
The following procedure explains the steps to use in order to increase or decrease the number of queries allowed to be made to a node by all subscribers.

---

**Procedure 3-14. Restricting the Number of Subscriber Queries Allowed**

---

1. In the *StarKeeper II* NMS SNMP Proxy Agent main window, choose **Provisioning**, choose **Flow Control**, then choose **Node Requests by All Subscribers**. The following window displays:



---

**Screen 3-11. Flow Control: Node Requests by All Subscribers**

---

2. From the scroll box, select a Nodename. The present settings are displayed:
  - **Requests Allowed**

The number of requests allowed to this node. The default value is 150.

When the number of actual requests to the node by all subscribers equals the number of requests allowed, the SNMP Proxy Agent will not process any more requests to the node for one 15-minute interval. During this penalty period, none of the requests made to this node by any subscriber are processed. At the end of the 15-minute interval, the value for Requests Made is re-initialized to zero and subscribers' requests will be processed.
  - **Requests Made**

The number of requests that have been made to the node since the last time this counter was re-initialized to zero. The data in this field cannot be modified by the user.
3. Enter a new value in the Requests Allowed field. Enter **0** if you want to prevent all subscribers from making requests to the node.
4. Choose **Edit** from the menu bar, then choose **Update Record**.

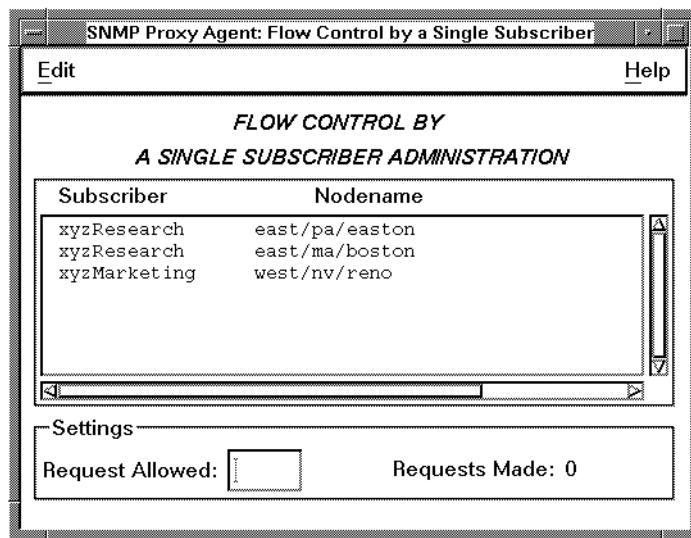
5. To exit the form, choose **Edit**, then choose **Exit**.

## Single Subscriber Requests Allowed

Use the following procedure to adjust the number of requests made to a node by a single subscriber by identifying the subscriber, the node, and entering a new value.

### Procedure 3-15. Adjusting the Number of Single Subscriber Requests Allowed

1. In the *StarKeeper II* NMS SNMP Proxy Agent main window, choose **Provisioning**, choose **Flow Control**, then choose **Node Requests by a Single Subscriber**. The following window displays:



### Screen 3-12. Flow Control: Node Requests by a Single Subscriber

2. Select a subscriber and Nodename from the scroll box. The present settings are displayed:
  - **Requests Allowed**

The number of requests allowed to this node by this subscriber. When this value is reached, the SNMP Proxy Agent will not process any more of the subscriber's requests to the node for one 15-minute interval. At the end of the 15-minute penalty period, the value for Requests Made is re-initialized to zero and subscribers' requests will be processed.
  - **Requests Made**

The number of requests made to the node by this subscriber since the last time the count was re-initialized to zero. The data in this field cannot be modified by the user.

3. Enter a new value in the Requests Allowed field. Enter **0** if you want to prevent the subscriber from making any requests to the node.
4. Choose **Edit** from the menu bar, then choose **Update Record**.
5. To exit the form, choose **Edit**, then choose **Exit**.

## Report Generation

---

Reports are available from the SNMP database that provide information by node or by subscriber. This section describes how to display these reports and provides samples of each report.

### Node Usage Report

The Node Usage Report displays the total number of node requests allowed and the total number of requests made for each node. To generate this report, complete these steps:

#### Procedure 3-16. Generating the Node Usage Report

---

1. In the *StarKeeper II* NMS SNMP Proxy Agent window, choose **Provisioning**, choose **Report Generation**, then choose **by Node**.
2. Select a node from the scroll box.
3. To generate the report, choose **Edit** from the menu bar, then choose **Display Report**.
4. After displaying the report, click on the **Cancel** button.
5. To exit the form, choose **Edit**, then choose **Exit**.

The following is an example of this report:

### Subscriber Reports

There are three reports on subscriber usage and errors:

- Subscriber Usage of Nodes Report — displays the total number of node requests allowed and the total number of requests made for each subscriber.
- SNMP Packets by Subscriber Report — displays usage statistics by subscriber.
- SNMP PDU Errors by Subscriber Report — displays error statistics

StarKeeper (R) II NMS SNMP DATABASE REPORTS				
Node Usage				
02/20/1996 10:03:51				
NODE	banana	Requests Allowed	3	Requests Made 0
NODE	central/ok/norman	Requests Allowed	150	Requests Made 0
NODE	east/ma/capecod	Requests Allowed	299	Requests Made 132
NODE	east/pa/villanova	Requests Allowed	2	Requests Made 1
NODE	hemlock	Requests Allowed	150	Requests Made 0

### Screen 3-13. Node Usage Report

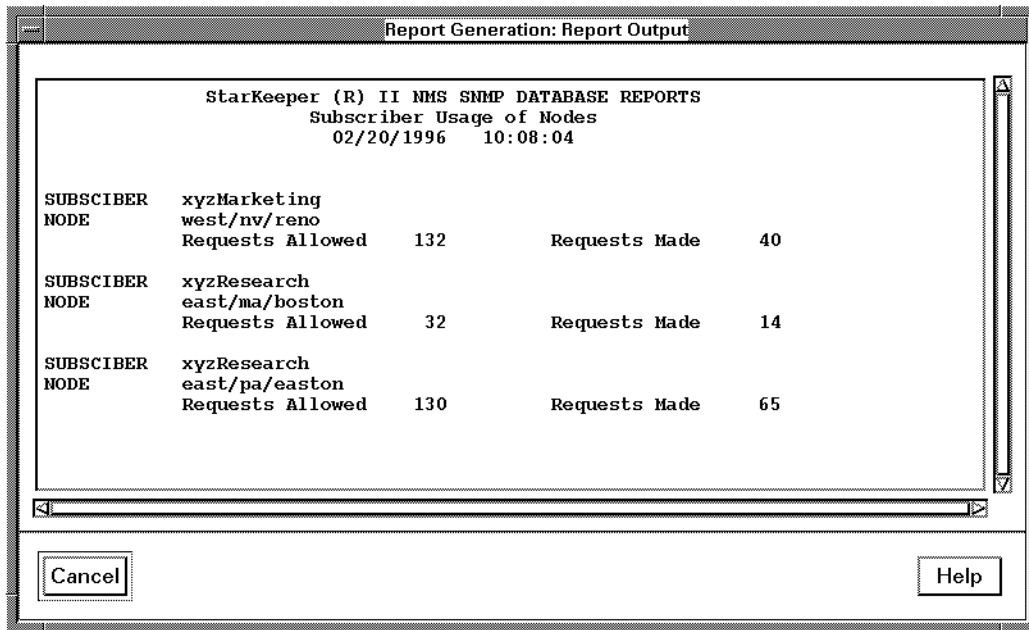
#### Procedure 3-17. Generating Subscriber Reports

1. In the *StarKeeper* II NMS SNMP Proxy Agent window, choose **Provisioning**, choose **Report Generation**, then choose **by Subscriber**.
2. Select a subscriber from the scroll box.
3. Select a report type:
  - Requests to all Nodes by Subscriber
  - SNMP PDUs Processed by a Subscriber
  - SNMP PDU Errors by a Subscriber
4. To generate the report, choose **Edit** from the menu bar, then choose **Display Report**.
5. After displaying the report, click on the **Cancel** button.
6. To exit the form, choose **Edit**, then choose **Exit**.

### Sample Subscriber Reports

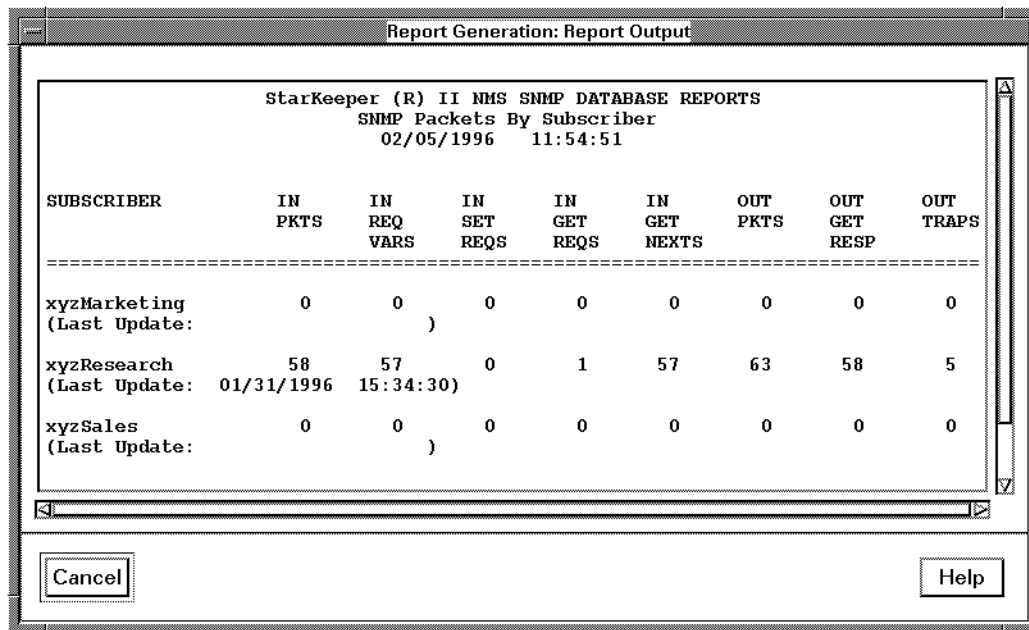
The three subscriber reports are described on the following pages.

**Subscriber Usage of Nodes Report**—This report lists the number of requests allowed and the number of requests made for an SNMP Manager.



Screen 3-14. Subscriber Usage of Nodes Report

**SNMP Packets by Subscriber Report**—This report lists SNMP usage by subscriber and totals for all subscribers. The following is an example of this report:



Screen 3-15. SNMP Packets by Subscriber Report

The counts listed for individual subscribers reflect the totals that have accumulated since the last time the usage count “Number of Requests Made” was re-initialized to zero.

The totals for all subscribers, shown in the row labeled *current\_totals* on the report, reflect the cumulative totals since the SNMP Proxy Agent was last started. The data for *current\_totals* is not the sum of the usage counts for each subscriber.

The following list explains the column headings that appear on the report:

- Subscriber—Subscriber Identifier.
- In Pkts—The number of packets received.
- In Req Vars—The total number of objects requested by the subscriber and received.
- In Set Reqs—Number of Set Requests received.
- In Get Reqs—Number of Get Requests received.
- In Get Nexts—Number of Get-Next Requests received.
- Out Pkts—Number of packets sent.
- Out Get Resp—Number of Get Responses sent.
- Out Traps—Number of traps sent.

**SNMP PDU Errors by Subscriber Report**— This report lists SNMP error statistics by subscriber and totals for all subscribers.

SUBSCRIBER	BAD ADDRS	BAD VERS	BAD COMM NAME	PARSE ERRS	TOO BIGS	NO SUCH NAME	GEN ERROR
xyzMarketing (Last Update: )	0	0	0	0	0	0	0
xyzResearch (Last Update: 01/31/1996 15:34:30)	0	0	0	0	0	1	0
xyzSales (Last Update: )	0	0	0	0	0	0	0

**Screen 3-16. SNMP PDU Errors by Subscriber Report**

The counts listed for individual subscribers reflect the totals that have accumulated since the last time you re-initialized the error counts to zero. For details, see “Usage Counts.”

The totals for all subscribers, shown in the row labeled *current\_totals* on the report, reflect the cumulative totals since the SNMP Proxy Agent was last started. The data for *current\_totals* is not the sum of the error counts for each subscriber.

The following list explains the column headings that appear on the report:

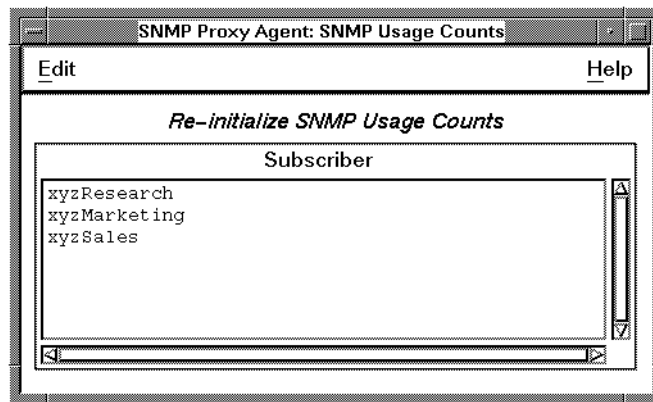
- Subscriber—Subscriber Identifier.
- Bad Addr—Number of times an invalid IP address was encountered.
- Bad Vers—Number of times a bad value was found in the version field for SNMP protocol. Version should be 1.
- Bad Comm Name—Number of times an invalid community string was used.
- Parse Errs—Number of times incoming packet could not be decoded.
- Too Bigs—Number of times response was too big to send back to SNMP Manager.

- No Such Name—Number of times the object requested was not known.
- Gen Err—Number of times other errors were encountered.

**Procedure 3-18. Re-initializing a Subscriber's Usage and Error Counts to Zero.**

---

1. In the *StarKeeper II* NMS SNMP Proxy Agent window, choose **Provisioning**, choose **Usage Counts**. The following window displays:



---

**Screen 3-17. Usage Counts: Re-initialize Usage Counts**

2. Select a subscriber from the scroll box.
3. Choose **Edit** from the menu bar, then choose **Re-initialize Usage Counts**.
4. To exit the form, choose **Edit**, then choose **Exit**.

## **SNMP Proxy Agent ASCII User Interface**

---

The SNMP Proxy Agent's ASCII user interface provides menus and forms to administer the SNMP database.

There are two types of menus:

- The Main Menu— lists the types of information contained in the database and provides an option to exit. When you select one of the Main Menu options, an Operations Ring Menu displays.
- An Operations Ring Menu— displays the types of actions that you can complete, such as add, change, delete, verify, and exit. When you select an option from an Operations Ring Menu, a form is displayed. A form contains fields where you can add, change, delete, or verify information in the SNMP database.

### Procedure 3-19. Displaying the Main Menu

---

1. Log in as **cnmsadm**.
2. Make sure that your TERM type is set correctly. For example, to set your TERM type to hp2621, enter the following:
  - **TERM=hp2621**
  - **export TERM**
3. At the prompt, enter **sksnmpcf**, and the Main Menu displays:

```

                SNMP Database Administration
                ADD, DELETE, CHANGE, VERIFY Operations

1. SYSTEM INFORMATION
2. NETWORK INTERFACES
3. CUSTOMER VIEWS
4. CUSTOMER VIEW MEMBERS
5. SUBSCRIBER INFORMATION
6. FLOW CONTROL AND USAGE
7. EXIT
    Please specify operation [1 - 7]: [ ]
    <Type Control-G to execute, Control-W for HELP>
    
```

---

### Screen 3-18. SNMP Database Administration Menu

4. To get help, press **(CONTROL) W**.
  5. To select an option, enter the number for that option and press **(CONTROL) G**.
- If you select any of the options 1-6, an Operations Ring Menu displays. For example, if you chose option 3 from the Main Menu, the Customer View Operations Ring Menu displays, as shown below:

```

CUSVIEWADM:  ADD DELETE CHANGE VERIFY EXIT
** one line help message appears here **
    
```

---

### Screen 3-19. Sample Operations Ring Menu

## Operations Ring Menu

Use the following procedure to choose an option from the operations ring menu.

### Procedure 3-20. Choosing an Option from the Operations Ring Menu

To choose an option from an Operations Ring Menu, do one of the following:

- Type the first letter of your choice.
- Move the highlight bar to the item you want and press **(ENTER)**.
  - To move the highlight bar to the right, use the **(SPACE)** or press **(CONTROL) L**.
  - To move the highlight bar to the left, press **(BACK SPACE)** or press **(CONTROL) H**.

## Entering Data on Forms

When you select an option from an Operations Ring Menu, a form is displayed. Forms enable you to enter, maintain, and verify data in the SNMP database.

When a form is displayed, the cursor is in the first field where information can be entered. The cursor will only visit fields that you can edit. Enter data using the keyboard. If you make an error, press **(BACK SPACE)** and type over the information or use the **(SPACE)** to erase characters.

Several keys allow you to move the cursor around forms and perform functions such as saving data and exiting. These keys are listed below.

**Table 3-1. Keys Used on Forms**

Key	Operation
<b>(CONTROL) J</b> or <b>(ENTER)</b>	Move forward to next field
<b>(CONTROL) K</b>	Move back to the previous field
<b>(CONTROL) L</b>	Move one space to the right within a field
<b>(CONTROL) H</b>	Move one space to the left within a field
<b>(BACK SPACE)</b>	Erase character
<b>(CONTROL) G</b>	Save additions or changes; also used for processing
<b>(DELETE)</b>	Return to previous menu from form
<b>(CONTROL) Z</b>	Exit application



**NOTE:**

With some terminal types, the arrow keys on your keyboard also enable you to move the cursor.

## Getting On-line Help

---

Each form and menu provides instructions for using the screen and keys.

In addition, as you move between fields on a form or between menu options on a menu, a one-line instructional help message automatically displays at the bottom of the screen.

To get a longer version of help for the Main Menu or for a form, press **(CONTROL) W**.

To page through the help, select `Screen` from the Help Operations Ring Menu. To return to the menu or form where you were last located, select `Resume`.

## Error Messages

---

When you make an invalid entry in a field on a form, an error message appears at the bottom of the screen.

The system prompts you to correct your entry before moving to the next field.

## Option 1: System Information

---

This option enables you to enter System Information and to print it out on your screen or printer. Enter System Information right after the SNMP Proxy Agent is installed.

To enter or verify System Information, select option 1, "System Information" from the SNMP Proxy Agent Main Menu.

The System Information Operations Ring Menu displays:



```
SYSTEMADM:  CHANGE  VERIFY  EXIT
```

---

**Screen 3-20. System Information Operations Ring Menu**

---

**Procedure 3-21. Changing System Information**

---

1. Select `CHANGE` from the Systems Information Operations Ring Menu and the "System Information Administration" form displays. Before you enter System Information, the modifiable fields on this form display the word **none**.

```
<Change> System Information
Type CTRL-G to save, DEL to return to menu.
----- Type Control-W for HELP -----

      System Information Administration

System Name:
[                               ]
System Description:
[                               ]
System Contact:
[                               ]
System Location:
[                               ]
System Object Id:
[                               ]
```

---

**Screen 3-21. System Information Administration Form**

---

2. Use the following field descriptions to fill in this form:
  - **System Name**  
The name of the host system that provides the SNMP Proxy Agent.
  - **System Description**  
The description of the services offered by the SNMP Proxy Agent.
  - **System Contact**  
The name of the system's administrative contact and information on how to contact this person.
  - **System Location**  
The location of the host system that provides the SNMP Proxy Agent.
  - **System Object Id**  
This field cannot be modified by the user. An entry is supplied automatically when the database is initialized.
3. Press `CONTROL G` to save the data. The system prompts you to confirm your changes.

4. Enter **y** for YES or **n** for NO.

You automatically return to the previous Operations Ring Menu.

---

### Procedure 3-22. Verifying System Information

---

To print the System Information on your screen or your printer, follow these steps:

1. From the System Information Operations Ring Menu, select **VERIFY**. The “Verify System Information” form displays.

```
VERIFY System Information
OUTPUT TO SCREEN (s) OR PRINTER (p) [+(SCREEN)]: [s]
<Type CTRL-G to process>
```

---

### Screen 3-22. Verify System Information Form

2. Enter **s** to display the output on your screen or **p** to send the report to the line printer. By default, the report is sent to the screen.
3. Press **CONTROL G** to process the report.
4. To exit the form and return to the previous Operations Ring Menu, press **DEL**.

---

## Option 2: Network Interfaces

Network interfaces are module ports on a BNS-2000 node, BNS-2000 VCS node, or BNS-2000 MPC (concentrator). An ICI trunk is also a network interface. Use this option to identify all the interfaces that the SNMP Proxy Agent supports. Each interface is identified by a unique Interface Index.

To enter network interface information, select option 2, “Network Interfaces” from the SNMP Proxy Agent Main Menu.

The Network Interfaces Operations Ring Menu displays:

```
IFADM:  ADD  DELETE  CHANGE  VERIFY  EXIT
```

---

### Screen 3-23. Network Interfaces Operations Ring Menu

The following form is used to add, delete, or change network interfaces:

```

                                <Operation> Network Interface
Enter network interface information, type CTRL-G to save, DEL to return to menu
----- Type Control-W for HELP -----

                                Network Interface Administration

                                Interface Index: [      ]

Node Name: [                                                              ]
Concentrator trunk module address: [ 0 ]
Node/Concentrator module address: [  ]

                                Port number: [  ]
                                Virtual Port number: [  ]

                                Interface type: [      ]
Interface provider contact: [                                                              ]
                                Interface location: [                                                              ]
Interface description: [                                                              ]
                                Interface subscriber: [                                                              ]

```

Screen 3-24. Network Interface Administration Form

### Fields on the Network Interface Administration Form

These fields appear on the “Network Interface Administration” form:

- **Interface Index**

The Interface Index identifies either a port on an SMDS or a Frame Relay module on a BNS-2000 node, BNS-2000 VCS node, BNS-2000 MPC (concentrator), or an ICI trunk on a BNS-2000 node. Each Interface Index must be unique.

Valid Interface Indices are 1 - 999999.

- **Node Name**

The name of the node where the interface is located. If you enter a node name that is not defined in the *StarKeeper II* NMS Core System database, the system displays an error message. This node name can be verified in the *StarKeeper II* NMS Core System database by using the Core System **cf** command.

If the interface is on a concentrator and the concentrator is connected to the node by a concentrator trunk module, enter the node name where the concentrator trunk module is located.

■ **Concentrator trunk module address**

For a Frame Relay interface located on a concentrator, enter the module address (slot number) of the concentrator trunk *on the node* that connects the node to the concentrator.

For an SMDS interface or a Frame Relay interface located on a node, leave the default entry, **0**, in this field to indicate that Concentrator trunk module address is not applicable.

■ **Node/Concentrator module address**

The module address (slot number) of the SMDS or Frame Relay interface. If the network interface is on the node, enter the interface's module address on the node. If the interface is on a concentrator, enter the interface's module address on the concentrator.

■ **Port number**

The port number for the SMDS or Frame Relay interface. The port number is 0 if the interface is an ICI trunk.

■ **Virtual Port number**

Applies to the M2 Frame Relay interface. The port number is 0 if no entry is made in this field.

■ **Interface type**

This field is automatically populated after you enter the Module address. If an entry is not automatically supplied, enter one of the following valid interface types.

For **SMDS Service**:

- ait1 (DS1 Access Interface module)
- ait3p (DS1 Access Interface module)
- ait3 (DS3 Access Interface module)
- aie1 (European Access Interface module)
- aie3 (European Access Interface module)
- trkt3i (ICI trunk module)

For **Frame Relay Service**:

- frm (Frame Relay R3.0 module)
- frm\_t1 (Frame Relay t1 module)
- frm\_e1 (Frame Relay e1 module)
- frm\_v35 (Frame Relay v35 module)
- m2\_frm\_t1 (Frame Relay M2 t1 module)
- m2\_frm\_e1 (Frame Relay M2 e1 module)

- **Interface provider contact**

The name of the network interface's administrative contact and information on how to contact this person.

- **Interface location**

A description of the physical location of the network interface.

- **Interface description**

This is a comment that describes the interface, such as SMDS-T1 or Frame Relay T1.

- **Interface subscriber**

The subscriber that uses this interface. You may enter a Subscriber Identifier or some other description for the subscriber. If there is no subscriber assigned to this interface, you can enter **none**.

### **Procedure 3-23. Adding a Network Interface**

---

Before you provision an Interface Index to a module port, make sure that the node name, concentrators, slot and port numbers are configured in the *StarKeeper II* NMS Core System database. The slot and port information is entered into the Core System database by using the Core System command, **skload**.

1. From the Network Interfaces Operations Ring Menu, select **ADD**. The "Network Interface Administration" form displays.
2. Fill in the Network Interface data.
3. Press **(CONTROL) G** to save the entries. The system prompts you to confirm your changes.
4. Enter **y** for **YES** or **n** for **NO**. The cursor returns to the top of the form and you can add another Interface Index.
5. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.



**NOTE:**

When you add a network interface, it is automatically assigned to Customer View, *MASTERVIEW*.

### **Procedure 3-24. Deleting a Network Interface**

---

1. From the Network Interfaces Operations Ring Menu, select **DELETE**. The "Network Interface Administration" form displays.
2. Enter the Interface Index that you want to delete. Press **(CONTROL) G** to search for the record.
3. Once the record is found, the system prompts you to confirm the deletion.

4. Enter **y** for YES or **n** for NO. The cursor returns to the top of the form and you can delete another Interface Index.
5. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.



**NOTE:**

When an Interface Index is deleted, the Interface Index is deleted from *MASTERVIEW* and from all other Customer Views that have this Interface Index as a member.

---

**Procedure 3-25. Changing a Network Interface**

---

1. From the Network Interfaces Operations Ring Menu, select *CHANGE*. The "Network Interface Administration" form displays.
2. Enter the Interface Index that you want to change. Press **(CONTROL) G** to search for the record.
3. Type in the new or corrected data into the appropriate fields.
4. Press **(CONTROL) G** to save the entries. The system prompts you to confirm your changes.
5. Enter **y** for YES or **n** for NO.
6. The cursor returns to the top of the form and you can change another Interface Index.
7. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.

---

**Procedure 3-26. Verifying a Network Interface**

---

To verify network interface information, you can generate a report on network interfaces three ways:

- **For a range of Interface Indices.** If you specify a range, the report includes the following information for each Interface Index from the first Interface Index up to and including the last Interface Index in the range: Node name, Concentrator trunk module address, Node/Concentrator module address, Port number, Interface provider contact, Interface location, Interface description, and Interface subscriber.
- **For a Node, Concentrator trunk module address, and Node/Concentrator module address.** This report lists all the Interface Indices that are assigned to the node name, Concentrator trunk module address, if any, and module address you specify.
- **For an Interface Type.** This report lists all Interface Indices of a specified type.

To verify the network interface information, follow these steps:

1. From the Network Interfaces Operations Ring Menu, select `VERIFY`. The "Verify Network Interfaces" form displays.

```

                VERIFY NETWORK INTERFACES
Enter choice and hit CTRL-G.  Enter criteria and hit CTRL-G to process.
----- Type Control-W for HELP -----

                Network Interfaces Administration

Selection Criteria (1,2,3): [ ]

1 - Initial Interface Index: [      ]
   Last Interface Index: [      ]

2 - Node name: [                                ]
   Concentrator trunk module address: [      ]
   Node/Concentrator module address: [      ]

3 - Interface Type: [      ]

OUTPUT TO SCREEN (s) OR PRINTER (p) [(SCREEN)]: [ ]

Enter your choice for selection criteria (1 or 2 or 3)

```

---

**Screen 3-25. Verify Network Interfaces Form**

2. Enter the selection criteria, **1**, **2**, or **3** and press `(CONTROL) G`. The cursor moves to the first appropriate field for your choice.
3. If you entered **1**, fill in the following fields:
 

**Initial Interface Index:** Enter the first Interface Index that you want to verify.

**Last Interface Index:** Enter the last Interface Index that you want to verify.
4. If you entered **2**, fill in the following fields:
 

**Node Name:** Enter the name of the node.

**Concentrator trunk module address:** Enter the address of the Concentrator trunk module. If there is none, enter **0**.

**Node/Concentrator module address:** Enter the module address. If the network interface is on the node, enter the node module address. If the interface is on a concentrator, enter the module address on the concentrator.
5. If you entered **3**, enter an interface type. For example, **ait1**, **frm\_e1**, or **frm\_v35**. The valid interface types are listed earlier in this chapter under "Fields on the Network Interface Administration Form."

6. Select the mode of output. Enter **s** to display the output on your screen or **p** to send the report to the line printer. By default, the report is sent to the screen.
7. Press **(CONTROL) G** to process the report.
8. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.

### Option 3: Customer Views

---

A Customer View identifies a set of Interface Indices that an SNMP Manger can manage. Select option 3 from the Proxy Agent Main Menu to identify names for Customer Views.

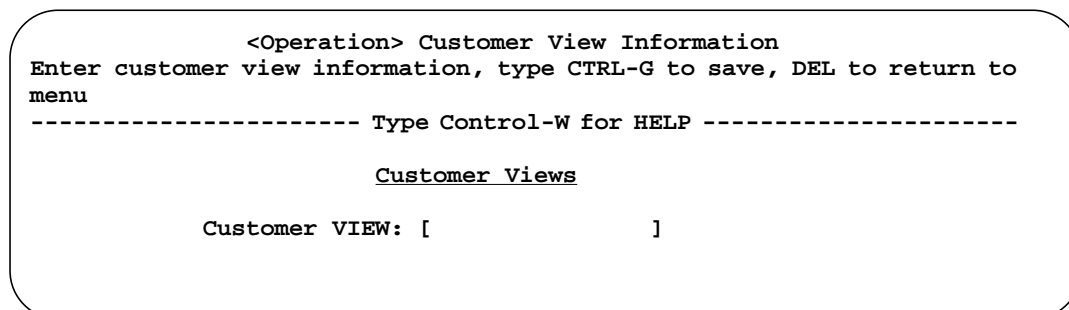
The Customer View Operations Ring Menu displays:




---

#### Screen 3-26. Customer View Operations Ring Menu

If you select add, delete, or change, the following form displays:




---

#### Screen 3-27. Customer Views Form

#### Fields on the Customer Views Form

These fields appear on the “Customer Views” form:

- **Customer View**

Each Customer View must be a unique ID in the SNMP database. Customer View IDs are alphanumeric character strings. Customer View IDs can also contain hyphens (-) and underline characters (\_). Spaces are not allowed. For example, valid Customer View IDs are a\_b\_c, a-b, a-b\_1\_2.

■ **Description**

A description of the Customer View.

**Procedure 3-27. Adding a Customer View**

---

1. From the Customer View Operations Ring Menu, select **ADD**. The “Customer Views” form displays.
2. Fill in the fields.
3. Press **(CONTROL) G** to save the entries. The system prompts you to confirm your additions.
4. Enter **y** for **YES** or **n** for **NO**.  
The cursor returns to the top of the form and you can add another Customer View.
5. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.

**Procedure 3-28. Deleting a Customer View**

---

1. From the Customer View Operations Ring Menu, select **DELETE**. The system displays the “Customer Views” form.
2. Enter the name of the Customer View that you want to delete.
3. Press **(CONTROL) G** to search for the record. Once the record is found, the system prompts you to confirm the deletion.
4. Enter **y** or **n**.
5. Once you type an entry, the system returns you to the top of the form so that you can delete another Customer View.
6. To return to the previous Operations Ring Menu, press **(DEL)**.



**NOTE:**

When a Customer View is deleted, all Interface Indices associated with this Customer View will be deleted from this Customer View.

**Procedure 3-29. Changing a Customer View**

---

1. Select **CHANGE** from the Customer View Operations Ring Menu. The “Customer Views” form displays.
2. Enter the name of the Customer View that you want to change. Press **(CONTROL) G** to search for the record.
3. Once the record is found, type in the changes.

4. Press **CONTROL G** to save your changes. The system prompts you to confirm your changes.
5. Enter **y** for YES or **n** for NO.  
The cursor returns to the top of the form and you can change another Customer View.
6. To exit the form and return to the previous Operations Ring Menu, press **DEL**.

---

**Procedure 3-30. Verifying a Customer View**

---

To see the names and descriptions for Customer Views, follow these steps:

1. Select **VERIFY** from the Customer View Operations Ring Menu. The “Verify Customer Views” form displays.

```
                VERIFY CUSTOMER VIEWS
Enter data. Hit CTRL-G to process

Customer VIEW: [all           ]

OUTPUT TO SCREEN (s) OR PRINTER (p) [+(SCREEN)]: [s]

                <Type CTRL-G to process>
```

---

**Screen 3-28. Verify Customer Views Form**

---

2. Enter one of the following entries in the Customer View field:
  - **all**—generates a report for all Customer Views. This is the default entry.
  - The exact name of a Customer View.
  - Part of the name of a Customer View. You can use the wildcard (\*) option when specifying the Customer View. For example, to verify all Customer Views that start with *mib123*, enter **mib123\***.
3. Select the mode of output. Enter **s** to display the output on your screen or **p** to send the report to the line printer. By default, the report is sent to the screen.
4. Press **CONTROL G** to process the report.
5. To exit the form and return to the previous Operations Ring Menu, press **DEL**.

### Option 4: Customer View Members

---

Once you create Interface Indices and Customer Views, you can assign Interface Indices to a Customer View. To add, delete, or verify Customer View Member information, select option 4, "Customer View Members" from the SNMP Proxy Agent Main Menu.

The Customer View Members Operations Ring Menu displays:

```
CUSVIEWMEMADM:  ADD  DELETE  VERIFY  EXIT
```

---

### Screen 3-29. Customer View Members Operations Ring Menu

The following form is shown for adding and deleting Customer View Members:

```
                <Operation> Customer View Members
Type CTRL-G to search, save; DEL to return to menu.
----- Type Control-W for HELP -----
Customer View:[          ]          Customer View Members
                                   [          ]
Description:                            [          ]
[          ]                            [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
                                   [          ]
```

---

### Screen 3-30. Customer View Members Form

#### Fields on the Customer View Members Form

These fields appear on the "Customer View Members" form:

- **Customer View**

The name of the Customer View that you will assign Customer View Members to or delete members from.

- **Description**

A description of the Customer View. This information is displayed, but cannot be modified on this screen.

- **Customer View Members**

A list of Interface Indices that belong to the Customer View.

### **Procedure 3-31. Adding a Customer View Member**

---

Before you can add members (Interface Indices) to a Customer View, the Customer View must have been defined previously using option 3, "Customer Views" on the SNMP Proxy Agent Main Menu.

1. Select **ADD** from the Customer View Members Operations Ring Menu. The "Customer View Members" form displays.
2. Enter the Customer View that you want to assign members to.  
If you entered a Customer View that does not exist, the system will prompt you to re-enter the Customer View.
3. When you have entered the Customer View, press **(CONTROL) G** to search for the record.
4. When the Customer View is found, the description for the Customer View displays. You are prompted to start assigning members to the specified Customer View.

If you want to find out which members are currently assigned to this Customer View, follow these steps:

- a. Press **(CONTROL) B**. The system displays a list of members (Interface Indices) that belong to the specified Customer View.
- b. Press **(ENTER)** to resume assigning members to the Customer View or press **Q** for quit.

To add a member, enter an Interface Index. You can enter only existing Interface Indices in the *Customer View Members* column.

The system will prompt you to re-enter a Customer View Member if the format of the Customer View Member is invalid or if it does not exist in the SNMP database.

5. When you finish assigning members to the Customer View, press **(CONTROL) G** to save the data.  
The system prompts you to confirm your additions.
6. Enter **y** for **YES** or **n** for **NO**.  
The cursor returns to the top of the form and you can add members to another Customer View.
7. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.

### Procedure 3-32. Deleting a Customer View Member

---

1. Select `DELETE` from the Customer View Member Operations Ring Menu. The “Customer View Members” form displays.
2. Enter the Customer View that you want to delete members from. Press `(CONTROL) G` to search for the record.
3. Once the record is found, you are prompted to enter the Customer View Members (Interface Indices) that you want to delete from the specified Customer View.
  - a. Press `(CONTROL) B` to view the Customer View Members currently assigned to this Customer View.
  - b. Press `(ENTER)` to resume entering Customer View Members or press `Q` for quit.

The system displays an error message if you try to delete a Customer View Member that does not exist or the format of the Customer View Member is invalid.

4. Once you have entered all the Customer View Members that you want to delete, press `(CONTROL) G` to process the delete operation.
5. The system prompts you to confirm the deletion. Enter `y` for `YES` or `n` for `NO`.

The cursor returns to the top of the form and you can specify another Customer View to delete members from.

6. To exit the form and return to the previous Operations Ring Menu, press `(DEL)`.

### Procedure 3-33. Verifying Customer View Members

---

There are two types of reports that you can generate to view customer members:

- A report based on Customer View lists all members of the Customer View(s) you specify.
- A report for a range of Interface Indices lists the Customer Views to which each Interface Index in the range belongs.

To verify Customer View member information, follow these steps:

1. Select `VERIFY` from the Customer View Members Operations Ring Menu. The “Verify Customer View Members” form displays:

```

                VERIFY CUSTOMER VIEW MEMBERS
Enter choice and criteria. Hit CTRL-G to process
-----Type Control-W for Help-----

                Customer View Member Administration

Selection Criteria (1,2): [ ]

1 - Customer View: [all           ]

2 - Initial Interface Index: [   ]
   Last Interface Index: [   ]

OUTPUT TO SCREEN (s) OR PRINTER (p) [+(SCREEN)]: [s]
```

---

### Screen 3-31. Verify Customer View Members Form

2. Enter the selection criteria, **1** or **2** and press `CONTROL G`. The cursor moves to the first appropriate field for your choice.
3. If you entered **1**, enter one of the following in the Customer View field:
  - **all**—generates a report for all Customer Views. This is the default entry.
  - The exact name of a Customer View.
  - Part of the name of a Customer View. You can use the wildcard (\*) option when specifying the Customer View. For example, to verify all Customer Views that start with *mib123*, you can enter **mib123\***.
4. If you entered **2**, fill in the following fields:

**Initial Interface Index:** Enter the first Interface Index that you want to verify.

**Last Interface Index:** Enter the last Interface Index that you want to verify.
5. Select the mode of output. Enter **s** to display the output on your screen or **p** to send the report to the line printer. By default, the report is sent to the screen.
6. Press `CONTROL G` to process the report.
7. To exit the form and return to the previous Operations Ring Menu, press `DEL`.

### Option 5: Subscriber Information

Each SNMP Manager must be identified by a unique Subscriber Identifier, its IP address, and a set of community strings and Customer Views.

To administer Subscriber Information, select option 5 from the SNMP Proxy Agent Main Menu. The Subscriber Information Operations Ring Menu displays:

```
SUBIDADM:  ADD  DELETE  CHANGE  VERIFY  EXIT
```

---

### Screen 3-32. Subscriber Information Operations Ring Menu

The following “Subscriber Information Administration” form is used to add, delete, and change Subscriber Information:

```

                                <Operation> Subscriber Information
Enter customer interface information, type CTRL-G to save, DEL to return
to menu
----- Type Control-W for HELP -----

                                Subscriber Information Administration

SUBSCRIBER IDENTIFIER: [          ]

Subscriber Description: [          ]
Subscriber Contact:    [          ]
IP Address:           [          ]

                                READ          WRITE          TRAP
Community String: [public      ][          ][public      ]
Customer View:    [NONE        ][          ][NONE        ]

Receive authentication traps: [n]
                             CNM Server: [n]
                             Service State: [i]
    
```

---

### Screen 3-33. Subscriber Information Administration Form

#### Subscriber Information Administration Form Fields

These fields appear on the “Subscriber Information Administration” form:

- **Subscriber Identifier**

A Subscriber Identifier is a name that identifies the SNMP Manager that will interface with the SNMP Proxy Agent.

Valid Subscriber Identifiers are alphanumeric character strings. The identifiers can also contain hyphens (-) and underline characters (\_). For example, **abc\_123** is a valid Subscriber Identifier. Spaces are not allowed.

■ **Subscriber Description**

A description of the SNMP Manager.

■ **Subscriber Contact**

The name of the administrative contact responsible for the SNMP Manager and information on how to contact this person.

■ **IP Address**

The IP address is the address of the host system where the subscriber's SNMP Manager is residing.

The IP address must be in the correct IP format. For example, 135.55.22.82 is a correctly formatted IP address.

The SNMP Proxy Agent uses this IP address to verify that an SNMP Manager is allowed access to the SNMP Proxy Agent and to send responses to requests.

■ **Community String (READ)**

The community string that this SNMP Manager will use in its SNMP queries to read (GET) data from the SNMP Proxy Agent.

The default Read Community String is **public**.

■ **Community String (WRITE)**

The community string that this subscriber's SNMP Manager will use in its SNMP queries to change (SET) data on the *StarKeeper* II NMS Core System.

■ **Community String (TRAP)**

The community string that the SNMP Proxy Agent will need to send traps (alarms) to this SNMP Manager.

The default Trap Community String is **public**.

■ **Customer View (READ)**

The Customer View that determines which interfaces the SNMP Manager can access.

The Customer View must have been defined previously using option 3, "Customer Views" on the SNMP Proxy Agent Main Menu.

■ **Customer View (WRITE)**

The Customer View that determines which interfaces the SNMP Manager can change.

The Customer View must have been defined previously using option 3, "Customer Views" on the SNMP Proxy Agent Main Menu.

- **Customer View (TRAP)**

The Customer View that determines which interfaces the SNMP Manager will receive traps for.

A trap will be sent to this SNMP Manager if an alarm occurs on any of the network interfaces that have been assigned to this Trap Customer View.

The Customer View must have been defined previously using option 3, "Customer Views" on the SNMP Proxy Agent Main Menu.



**NOTE:**

You can use the same Customer View for reading and for traps or you can specify different ones.

- **Receive Authentication Traps**

An authentication-failure trap is generated whenever this SNMP Manager initiates a request to the SNMP Proxy Agent with an invalid community string. If the SNMP Manager should receive authentication traps, enter **y**; if not, enter **n**.

- **CNM Server**

If the SNMP Manager is a Customer Network Management server, enter **y**. If the SNMP Manager is not a server, enter **n**.

The SNMP Proxy Agent sends authentication traps to CNM servers whenever an SNMP request is received from an invalid SNMP Manager. Also, only CNM servers can request the SNMP Proxy Agent for current SNMP usage statistics.

- **Service State**

Indicate if this SNMP Manager is in service (**i**) or out of service (**o**). If you want the SNMP Proxy Agent to respond to this SNMP Manager's requests, then enter **i**. If this SNMP Manager is currently not operating or if you do not want the SNMP Proxy Agent to accept SNMP requests, then enter **o** to indicate that this SNMP Manager is out of service.

### **Procedure 3-34. Adding Subscriber Information**

---

Follow these guidelines when you use this option:

- Each Subscriber Identifier must be unique.
- Each IP Address must be unique in the database. You cannot enter an IP Address if it already exists in the SNMP database.
- Customer Views must have been defined previously in the SNMP database by using SNMP Proxy Agent Main Menu options 3 and 4.
- The same Customer View can be assigned to Read and Trap Customer Views.

To add a subscriber, complete these steps:

1. Select **ADD** from the Subscriber Information Operations Ring Menu. The “Subscriber Information Administration” form displays.
2. Fill in the fields.
3. When you finish entering the data on the form, press **(CONTROL) G** to save the data. The system prompts you to confirm your additions.
4. Enter **y** for **YES** or **n** for **NO**. The cursor returns to the top of the form and you can add another Subscriber.
5. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.

---

### **Procedure 3-35. Deleting Subscriber Information**

---

1. Select **DELETE** from the Subscriber Information Operations Ring Menu. The “Subscriber Information Administration” form displays.
2. Enter the Subscriber Identifier that you want to delete. Press **(CONTROL) G** to search for the record.
3. Once the record is found, the system prompts you to confirm the deletion.
4. Enter **y** for **YES** or **n** for **NO**.  
The cursor returns to the top of the form and you can delete another subscriber.
5. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.

---

### **Procedure 3-36. Changing Subscriber Information**

---

1. Select **CHANGE** from the Subscriber Information Operations Ring Menu. The “Subscriber Information Administration” form displays.
2. Enter the Subscriber Identifier that you want to change. Press **(CONTROL) G** to search for the record.
3. Once the record is found, type the new or corrected data into the appropriate fields.
4. When you finish changing the data, press **(CONTROL) G** to save the data. The system prompts you to confirm your changes.
5. Enter **y** for **YES** or **n** for **NO**.  
The cursor returns to the top of the form and you can change information for another subscriber.
6. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.

### Procedure 3-37. Verifying Subscriber Information

---

You can verify Subscriber Information for a specified Subscriber Identifier or IP address. The report lists all of the associated subscriber information for the Subscriber Identifier or the IP address that you specify.

To verify Subscriber Information, complete the following steps:

1. Select `VERIFY` from the Subscriber Information Operations Ring Menu. The following form displays:

```

                VERIFY SUBSCRIBER INFORMATION
Enter choice and criteria. Hit CTRL-G to process
-----Type Control-W for Help-----

                Subscriber Information Administration

Selection Criteria (1,2): [ ]

1 - Subscriber Identifier: [           ]

2 - IP Address: [           ]

OUTPUT TO SCREEN (s) OR PRINTER (p) [+(SCREEN)]: [s]
```

---

### Screen 3-34. Verify Subscriber Information Form

2. Enter the selection criteria, **1** or **2** and press `(CONTROL) G`. The cursor moves to the first appropriate field for your choice.
3. If you entered **1**, enter one of the following in the Subscriber Identifier field:
  - **all**—generates a report including all subscribers. This is the default entry.
  - A Subscriber Identifier.
  - Part of a Subscriber Identifier. You can use the wildcard (\*) option when specifying the Subscriber Identifier. For example, to verify all Subscriber Identifiers that start with *subid1*, enter **subid1\***. To verify all Subscriber Identifiers that end with *subid1*, enter **\*subid1**.
4. If you entered **2**, enter one of the following in the IP Address field:
  - **all**—generates a report including all IP addresses. This is the default entry.
  - An IP address.

- Part of an IP address. You can use the wildcard (\*) option when specifying the IP address. For example, to verify all IP addresses that start with 135, enter 135\*. To verify all IP addresses that end with 201, enter \*201.
5. Select the mode of output. Enter **s** to display the output on your screen or **p** to send the report to the line printer. By default, the report is sent to the screen.
  6. Press **(CONTROL) G** to process the report.
  7. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.

### **Option 6: Flow Control and Usage**

---

Use this option to restrict the number of queries made to a node, to reinitialize SNMP usage counts, or to get reports on usage statistics.

When you select option 6 from the SNMP Proxy Agent Main Menu, the Flow Control and Usage Information Operations Ring Menu displays:



USAGEADM:    CHANGE    VERIFY    EXIT

---

#### **Screen 3-35. Flow Control and Usage Information Operations Ring Menu**

### **Changing Throttling Counts or Usage Counts**

Certain requests from an SNMP Manager require the SNMP Proxy Agent to get network management information from a node. You can change the number of allowed requests if you find the default setting is too high or too low.

If you want to restrict the number of queries to a node, you can adjust the throttling counts (number of allowed requests). You can restrict the number of queries made to a node for an individual subscriber or for all subscribers. There are two ways to adjust throttling counts:

- **Choice 1: Adjust Throttle Count for a Subscriber**— The default number of allowed requests per subscriber is 150 over a 15-minute interval. The throttle count (number of allowed requests) is reduced by 1 every time an SNMP Manager makes request requiring information from the node. This count is reset after a 15-minute interval.

If you want to prevent an SNMP Manager from accessing a node, use Choice 1 to make the throttle count 0.

- **Choice 2: Adjust Throttle Count for a Node**— You can adjust the throttle count by node.

By default, in a 15-minute interval, 150 requests are allowed to be made to a specified node. The SNMP Proxy Agent will not make any more requests to the node if this count is exhausted by all the subscribers. This count will be reinitialized to 150 after a 15-minute interval.

If you want to prevent all SNMP Managers from accessing a node, use Choice 2 to make the throttle count 0.

The following option lets you change usage counts.

- **Choice 3: Reinitializing Usage Counts**— Allows you to reset the SNMP usage and error counts that have been maintained in the SNMP database. All the SNMP counts are reset to 0.

**Procedure 3-38. Changing Throttling Counts or Reinitializing Usage Counts**

---

1. Select `CHANGE` from the Flow Control and Usage Operations Ring Menu. The “Flow Control and Usage Administration” form displays:

```

CHANGE FLOW CONTROL AND USAGE COUNTS
Enter choice and hit CTRL-G. Enter criteria and hit CTRL-G to process
----- Type Control-W for HELP -----

Flow Control and Usage Administration

Selection Criteria (1,2,3): [ ]

1 - Number of requests a Subscriber can make to a node
Subscriber Identifier: [          ]
Node Name: [                               ]
Number of Allowed Requests: [          ]

2 - Total number of requests all subscribers can make to a node
Node Name: [                               ]
Number of Allowed Requests: [          ]

3 - Re-initialize SNMP usage counts for a Subscriber
Subscriber Identifier: [          ]
    
```

---

**Screen 3-36. Flow Control and Usage Administration Form**

2. Enter the selection criteria, **1**, **2**, or **3** and press `CONTROL G`. The cursor moves to the first appropriate field for your choice.
3. If you entered **1**, fill in these fields:

- **Subscriber Identifier:** Enter the Subscriber Identifier for which you want to change the throttle count.
  - **Node Name:** Enter the name of the node that this SNMP Manager is going to get network management information from.
  - **Number of Allowed Requests:** Enter the number of allowed requests that this subscriber can make to the node. The number of allowed requests cannot be greater than the total number of allowed requests that all SNMP Managers can make to the specified node.
4. If you entered **2**, fill in these fields:
    - **Node Name:** Enter the name of the node for which you want to change the number of allowed requests.
    - **Number of Allowed Requests:** Enter the number of allowed requests that all subscribers (SNMP Managers) can make to the specified node.
  5. If you entered **3**, enter a Subscriber Identifier.
  6. Press **(CONTROL) G** to save your changes. The system prompts you to confirm your changes.
  7. Enter **y** for YES or **n** for NO.
  8. To exit the form and return to the previous Operations Ring Menu, press **(DEL)**.

### Verifying Flow Control and Usage Information

There are four usage reports available:

- **Usage Report 1— Total Number of Requests per Node:** This report gives you the total number of requests that have been made to each node by all SNMP Managers within the last fifteen minutes.
- **Usage Report 2— Total Number of Node Requests per SNMP Manager:** This report tells you the number of times an SNMP Manager queries a node within the last fifteen minutes.
- **Usage Report 3— SNMP PDU Counts:** This report gives you SNMP usage statistics by subscriber.
- **Usage Report 4—SNMP PDU Errors:** This report gives you SNMP error statistics by subscriber.

### Procedure 3-39. Generating Usage Reports

---

1. Select `VERIFY` from the Flow Control and Usage Operations Ring Menu. The "Verify Usage" form displays:

```

                                VERIFY USAGE
Enter choice and hit CTRL-G. Enter criteria and hit CTRL-G to process
----- Type Control-W for HELP -----
                                Usage Reports

Select Report (1,2,3,4): [ ]

  1 - Number of requests made by all Subscribers to a node
     Node Name: [ ]

  2 - Number of requests made by a Subscriber to a node
     Subscriber Identifier: [ ]

  3 - Report of SNMP PDUs processed by Subscribers
     Subscriber Identifier: [ ]

  4 - Report of SNMP PDU errors by Subscribers
     Subscriber Identifier: [ ]

OUTPUT TO SCREEN (s) OR PRINTER (p) [(SCREEN)]: [ ]

Enter your report choice ( 1 or 2 or 3 or 4)

```

---

### Screen 3-37. Verify Usage Form

2. Enter the selection criteria, **1**, **2**, **3**, or **4** and press `(CONTROL) G`. The cursor moves to the first appropriate field for your choice.
3. If you entered **1**, enter a node name or enter **all**.
4. If you entered **2**, **3**, or **4**, enter a Subscriber Identifier or enter **all**.
5. Select the mode of output. Enter **s** to display the output on your screen or **p** to send the report to the line printer. By default, the report is sent to the screen.
6. Press `(CONTROL) G` to process the report.
7. To exit the form and return to the previous Operations Ring Menu, press `(DEL)`.

### Sample Usage Reports

The following section describes the usage reports generated by selecting options 3 and 4 from the "Verify Usage" form.

### SNMP Packet Counts by Subscriber

The “SNMP Packets by Subscriber” report lists SNMP usage by subscriber and totals for all subscribers. This report is generated when you select Report 3 from the “Verify Usage” form.

The counts listed for individual subscribers reflect the totals that have accumulated since the last time you reinitialized the usage counts by using Choice 3 on the “Flow Control and Usage Administration” form.

The totals for all subscribers, shown in the row labeled *current\_totals* on the report, reflect the cumulative totals since the SNMP Proxy Agent was last started. The data for *current\_totals* is not the sum of the usage counts for each subscriber.

The following list explains the column headings that appear on the report:

- Subscriber—Subscriber Identifier.
- In Pkts—The number of packets received.
- In Req Vars—The total number of objects requested by the subscriber and received.
- In Set Reqs—Number of Set Requests received.
- In Get Reqs—Number of Get Requests received.
- In Get Nexts—Number of Get-Next Requests received.
- Out Pkts—Number of packets sent.
- Out Get Resp—Number of Get Responses sent.
- Out Traps—Number of traps sent.

```

StarKeeper (R) II NMS SNMP DATABASE REPORTS
SNMP Packets By Subscriber
printed: 01-01-98 16:03:19
-1-

```

SUBSCRIBER	IN PKTS	IN REQ VARS	IN SET REQS	IN GET REQS	IN GET NEXTS	OUT PKTS	OUT GET RESP	OUT TRAPS
customerA (Last update: 01/01/98 11:12:07)	2575	2588	0	8	2567	3026	2572	454
customerB (Last update: 01/01/98 11:13:53)	4033	5142	433	813	2787	3828	3598	230
customerC (Last update: 01/01/98 13:21:11)	1728	3416	119	554	1055	1686	1457	229
customerD (Last update: 01/01/98 11:12:06)	141	165	0	0	141	269	141	128
current_totals (Last update: 01/01/98 15:12:06)	306	559	40	126	140	306	264	42

### Screen 3-38. Report of SNMP PDUs Processed

#### SNMP PDU Errors by Subscriber

The “SNMP PDU Errors by Subscriber” report lists SNMP error statistics by subscriber and totals for all subscribers. This report is generated when you select Report 4 from the “Verify Usage” form.

The counts listed for individual subscribers reflect the totals that have accumulated since the last time you reinitialized the error counts by using Choice 3 on the “Flow Control and Usage Administration” form.

The totals for all subscribers, shown in the row labeled *current\_totals* on the report, reflect the cumulative totals since the SNMP Proxy Agent was last started. The data for *current\_totals* is not the sum of the error counts for each subscriber.

The following list explains the column headings that appear on the report:

- Subscriber—Subscriber Identifier.
- Bad Addr—Number of times an invalid IP address was encountered.

- Bad Vers—Number of times a bad value was found in the version field for SNMP protocol. Version should be 1.
- Bad Comm Name—Number of times an invalid community string was used.
- Parse Errs—Number of times incoming packet could not be decoded.
- Too Bigs—Number of times response was too big to send back to SNMP Manager.
- No Such Name—Number of times the object requested was not known.
- Gen Err—Number of times other errors were encountered.

```

StarKeeper (R) II NMS SNMP DATABASE REPORTS
SNMP PDU Errors By Subscriber
printed: 01-01-98 16:06:23
-1-

```

SUBSCRIBER	BAD ADDRS	BAD VERS	BAD COMM NAME	PARSE ERRS	TOO BIGS	NO SUCH NAME	GEN ERROR
customerA	0	0	3	0	0	1	0
(Last update: 01/01/98 11:12:07)							
customerB	0	0	2	0	0	319	0
(Last update: 01/01/98 11:13:53)							
customerC	0	0	152	0	0	113	0
(Last update: 01/01/98 13:21:11)							
customerD	0	0	0	0	0	0	0
(Last update: 01/01/98 11:12:06)							
current_totals	0	0	2	0	0	43	0
(Last update: 01/01/98 13:21:11)							

---

This chapter provides procedures to help you resolve problems encountered with the *StarKeeper II NMS SNMP Proxy Agent*.

## SNMP Proxy Agent Dependencies

The SNMP Proxy Agent relies on the proper functioning of the *StarKeeper II NMS Core System* and the TCP/IP networking product.

If there does not appear to be a problem with the SNMP Proxy Agent itself, refer to the appropriate documentation for *StarKeeper II NMS*, the TCP/IP networking product, UNIX, or the SNMP Manager.

## Monitoring the SNMP Proxy Agent Daemon Processes

You can monitor the SNMP Proxy Agent daemon processes by looking at the messages in the log file or if necessary, by using the trace facility, **sksnmpttrace**.

These daemon processes perform these SNMP operations:

- *sksnmpd* receives incoming SNMP requests and puts them on a message queue.
- *sksnmpps* picks up requests from the message queue for processing and then sends responses to the SNMP Managers.
- *sksnmpt* is responsible for sending status traps to the appropriate SNMP Managers. It determines when to send these traps based on the alarms it gets from the *StarKeeper II NMS Core System*.

## Displaying the Log File

---

The log file, `$SNMP_LOG/snmp_agent.log`, contains messages about certain critical events output by the daemon processes. Look at the log file to diagnose any problems with these processes. To browse through the log file you can use such UNIX tools as **pg**, **cat**, or **vi**.

**Appendix B, Error and Log Messages** describes the messages that may be sent to the log file.

## Using the Trace Facility

---

If you look at the log file and need more information to resolve a problem, use the trace facility, **sksnmpttrace**. The daemon processes send detailed trace messages to `$SNMP_LOG/snmp_agent.trace`.

The trace facility can be toggled on or off by using the **sksnmpttrace** command as described in **Appendix A, User Reference**.



### **CAUTION:**

*The trace facility is intended for use by StarKeeper II NMS support personnel. Due to the number of messages written to the trace file, it can become very large if left on for an extended period. Use **sksnmpttrace** only when necessary and remember to turn it off when you are done.*

## Troubleshooting the TCP/IP Subsystem

---

The SNMP Proxy Agent will encounter problems if the TCP/IP network is not functioning properly. This section introduces three commands, provided with your TCP/IP networking product, that can help you troubleshoot problems. Refer to your TCP/IP documentation for complete details.

### **ping**

---

Use **ping** to determine if a remote host on your network is up and accessible by your host. To invoke the command, enter:

**ping <host>**

<host> is the host name or IP address (in dot format) of the system. (The host name must be included in the `/etc/hosts` file).

The **ping** command sends a datagram to the remote system and waits for a response. On Hewlett-Packard systems, if the remote system is up, the **ping**

command produces one line of output for each response it gets. No output is produced if there is no response.



**NOTE:**

To help isolate a problem, first run **ping** on the local host to verify that the local network interface is working. Then, run **ping** on hosts and routers that are further and further away. Since **ping** generates traffic on the network, it is not advisable to use **ping** during normal operations or from automated scripts.

## **ifconfig**

---

Use the **ifconfig** command to display the current configuration of a network interface. For example, to display the configuration of an Ethernet interface on a Hewlett-Packard system, you invoke this command by entering:

### **ifconfig lan0**

The **ifconfig** command shows the operational state of the interface, if it supports broadcasting, and if broadcasting is currently running. It also shows the IP address of your host, the *netmask* being used, and the broadcast IP address.

## **netstat**

---

The **netstat** command is used to get certain statistics on the network. This command can be invoked with many options. The **-i** and **-s** are described below.

The **-i** option of **netstat** shows the status of the network interfaces on your host. Using this option, you can see how many packets are being sent and received by this host. A host with network traffic should show these counts continuously increasing.

The **-s** option of **netstat** shows the per-protocol statistics for the TCP/IP stack. This information is useful in identifying areas where the protocol stack is having problems. Since the SNMP protocol runs on UDP, the statistics for UDP and IP should be of primary interest to the SNMP Proxy Agent administrator.

## **Troubleshooting the *StarKeeper II* NMS Core System**

---

As noted earlier, the SNMP Proxy Agent relies heavily on the proper functioning of the *StarKeeper II* NMS Core System. Refer to the *StarKeeper II* NMS documentation for procedures to verify the status of the Core System.

## Troubleshooting by Symptom

---

This section provides tips on how to troubleshoot a problem, based on the symptoms.

### SNMP Proxy Agent Doesn't Come Up

---

- Verify that you have the correct release versions for the *StarKeeper* II NMS SNMP Proxy Agent and other related software especially the UNIX operating system, the *StarKeeper* II NMS Core System, and the TCP/IP networking product.
- Make sure you invoke the SNMP Proxy Agent using the **sksnmpstart** command, instead of invoking the daemon processes directly.
- Make sure you are invoking the SNMP Proxy Agent by logging in as **cnmsadm**.
- Before you try to start the SNMP Proxy Agent, verify that the SNMP Proxy Agent processes (*sksnmpd*, *sksnmpps*, *sksnmpt*) are not already up. To verify the operational state of the SNMP Proxy Agent, use **sksnmpstat**. To stop the SNMP processes, use **sksnmpstop**.
- Make sure the *StarKeeper* II NMS Core System and the TCP/IP network interface is up and working properly. Check the following things:
  - Check the */etc/hosts* file to make sure your local host is entered there with the correct IP address.
  - Check the */etc/services* file to make sure entries for *snmp* and *snmp-trap* are provided. If they are not, enter the following lines into the file:
    - snmp161/udp# port for snmp agent
    - snmp-trap162/udp# port for trap agent
- Make sure the *StarKeeper* II NMS SNMP Proxy Agent database exists and is made properly. The SNMP database is in `SNMP_DBS`.

### SNMP Proxy Agent Comes Down Abruptly

---

- Verify that the *StarKeeper* II NMS Core System and the TCP/IP network is still up and working. If not, diagnose and restart them, then restart the SNMP Proxy Agent.
- Look at the SNMP Proxy Agent log file, `SNMP_LOG/snmp_agent.log`, for indications of any other problems.
- If the log file doesn't help, restart the SNMP Proxy Agent and turn on the trace facility using **sksnmptrace**. If you need further assistance, contact your *StarKeeper* II NMS support personnel and provide them with the trace information.

### **SNMP Proxy Agent Doesn't Respond to Queries from any SNMP Managers**

---

- Verify that the SNMP Proxy Agent is up.
- Verify that the remote SNMP Managers have been properly entered in the SNMP database and their Service State in the SNMP database is in-service (i).
- Verify that *StarKeeper II* NMS has TCP/IP connectivity with the remote SNMP Manager. For details, refer to “Troubleshooting the TCP/IP Subsystem” earlier in this chapter.
- Verify that the SNMP Proxy Agent responds to the local tools **sksnmpget**, **sksnmpnext**, and **sksnmpwalk**. (Make sure your local system is entered in the SNMP database as a subscriber).
- If there is no response with these tools, then restart the SNMP Proxy Agent, and try again.

### **SNMP Proxy Agent Responds with NoSuchName Error**

---

An SNMP Get Request will receive a Get Response with a NoSuchName error due to two major reasons:

1. The Get Request refers to an invalid object. Note that only fully qualified object identifiers are allowed in a Get Request PDU.
2. The Get or Get-Next Request PDU includes objects that are not accessible to the SNMP Manager based on the assigned Customer View in the SNMP database.

If the SNMP Proxy Agent responds with **NoSuchName**, check the following things:

- Make sure that the SNMP Manager is issuing SNMP requests that refer to valid MIB objects only.
- Make sure that the *StarKeeper II* NMS SNMP Proxy Agent is running instead of the HP-UX® SNMP agent, */etc/snmpd*. If *snmpd* is running, stop it using the UNIX **kill** command and start the *StarKeeper II* NMS SNMP Proxy Agent.
- Make sure the SMDS and Frame Relay interfaces have been properly entered in the SNMP database.
- Verify that the SNMP Manager that is issuing the SNMP request is properly entered in the SNMP database.

- Make sure that the Customer View assigned to this SNMP Manager includes all the interfaces that they are allowed to access.
- If the SNMP Manager is a CNM Server, verify that this SNMP Manager is flagged as a CNM server in the SNMP database.

### **SNMP Proxy Agent Responds with TooBig Error**

This message indicates that the SNMP Manager is sending requests with too many objects, a response for which is too big.

To resolve this problem, the SNMP Manager should break up the request into a series of requests that include fewer objects.

### **SNMP Proxy Agent Responds with AccessDenied Error**

This message indicates that the SNMP Manager is sending SNMP Set Request PDUs to the SNMP Proxy Agent.

### **SNMP Proxy Agent Responds with GeneralError or BadValue Error**

This error message should rarely occur. Use the trace facility, **sksnmpttrace**, to find out which Request PDU generates the error and contact your *StarKeeper II* NMS support personnel for further assistance.

### **SNMP Proxy Agent Responds with ReadOnly Error**

This error indicates that the SNMP Manager is sending SNMP Set Request PDUs to the SNMP Proxy Agent.

### **SNMP Proxy Agent Sends Authentication-Failure Trap**

This error indicates one of the following conditions:

- An SNMP Manager that is not entered in the SNMP database is trying to access the SNMP Proxy Agent
- An SNMP Manager is issuing requests with an invalid community string.
- A Customer View is not assigned to the SNMP Manager that is issuing this request.

---

# SNMP Manager Configuration

# 5

---

## Prerequisites for SNMP Managers

Before an SNMP Manager can access the SNMP Proxy Agent, you must provide the administrator of the SNMP Manager with the following:

- The IP address of the *StarKeeper* II NMS Core System where the SNMP Proxy Agent resides.
- The community string that you have assigned to this SNMP Manager using the **sksnmpcf** or the **sksnmpxadm** command. Each request made by the SNMP Manager should include the assigned community string.
- A copy of the appropriate Lucent CNM MIB files.

To achieve optimum performance of the SNMP Proxy Agent, the SNMP Manager applications should follow the guidelines described in “Guidelines for SNMP Manager Applications Development” later in this chapter.

## Distribution of MIB Files

The MIB files are available in the `$SNMP_MIBS` directory on the *StarKeeper* II NMS Core System where the SNMP Proxy Agent resides. Copy the MIB files to an appropriate media, and give them to the administrator of the SNMP Manager.

The following MIB files must be distributed to SNMP Managers that need support for SMDS interfaces only:

- `luc_sys.mib`
- `luc_ds1.mib`
- `luc_ds3.mib`
- `luc_sip.mib`
- `luc_smads.mib`

The following MIB files must be distributed to SNMP Managers that need support for Frame Relay interfaces only:

- luc\_sys.mib
- luc\_fr.mib
- luc\_efr.mib

All of the MIB files listed need to be given to SNMP Managers that require both SMDS and Frame Relay support.

## **Guidelines for SNMP Manager Applications Development**

---

SNMP Manager applications should follow these guidelines to achieve optimum performance of the SNMP Proxy Agent:

- Increase the SNMP timeout value.  
In LAN environments, SNMP Managers expect quick responses from SNMP agents. However, responses from an SNMP Proxy Agent over a WAN, such as Frame Relay or SMDS, may take a longer period. Moreover, in a Customer Network Management scenario, the SNMP Proxy Agent may be handling a large number of SNMP Managers at the same time. To accommodate this, the SNMP timeout values used by the SNMP Manager should be adjusted to a larger value (at least 30 seconds).
- Traverse the SNMP database tables by row, not by column.  
To achieve better performance, SNMP Manager applications should traverse the SNMP database tables by row. Each query should include only those columns that are necessary. Even though the SNMP Proxy Agent supports measurements for several past intervals, SNMP Managers should query just for the interval(s) they need.
- SNMP Managers should avoid traversing entire tables as much as possible.
- Query for status and log objects only when necessary.  
Since a query for status and log objects often results in the SNMP Proxy Agent accessing the node in real-time, make these requests only when necessary. Frequent queries may result in longer response times and congestion on the node.

---

## User Reference



---

This appendix describes each of the SNMP Proxy Agent commands, the *crontab* for **cnmsadm**, and provides planning forms to help you organize and collect data needed in the SNMP database.

### SNMP Proxy Agent Commands

This appendix describes each of the SNMP Proxy Agent commands listed below. To use any of these commands, log in as **cnmsadm**.

- **sksnmpbdb**—backs up the SNMP database.
- **sksnmpcf**—displays the ASCII version of the SNMP Proxy Agent user interface.
- **sksnmpdbsync**—synchronizes the SNMP database with the *StarKeeper II* NMS Core System database.
- **sksnmpget**—retrieves variables from an SNMP agent.
- **sksnmphelp**—displays on-line help or software version.
- **sksnmpidb**— initializes or reinitializes the SNMP database.
- **sksnmpndsync**—synchronizes the SNMP database with node information.
- **sksnmpnext**— retrieves variables from an SNMP agent using an SNMP Get-Next Request PDU.
- **sksnmprdb**—restores the SNMP database.
- **sksnmpset**—sets variables on an SNMP agent using an SNMP Set Request PDU.
- **sksnmpstart**—starts the SNMP Proxy Agent.
- **sksnmpstat**—displays the operational state of the SNMP Proxy Agent and SNMP statistical counts.
- **sksnmpstop**—stops all of the SNMP Proxy Agent processes.

- **sksnmptrace**— toggles the trace facility on and off.
- **sksnmptrap**—retrieves SNMP Trap PDUs from remote SNMP trap-generating systems.
- **sksnmpupd**—applies SNMP Set Requests to the node, and to the *StarKeeper* II NMS Core System and SNMP Proxy Agent databases.
- **sksnmpwalk**—retrieves a class of variables from an SNMP agent using a series of SNMP Get-Next Request PDUs.
- **sksnmpxadm**—displays the Motif-based version of the SNMP Proxy Agent user interface.

**SKSNMPBDB(1)****SKSNMPBDB(1)****NAME**

**sksnmpbdb**—backs up the data in the SNMP database

**SYNOPSIS**

**sksnmpbdb**

**DESCRIPTION**

**sksnmpbdb** backs up the data from the SNMP database, *snmpdb*.

**sksnmpbdb** prompts you for the **root** password. The data in the SNMP database can be backed up either to a directory or to a tape.

If you choose to back up the SNMP database data to a directory, by default, it will be backed up in the directory *\$SNMPBKUPD*.

*SNMPBKUPD* is an environment variable that is defined in the environment file *\$SNMP\_ENV/snmp.vars*. *\$SNMPBKUPD* is set to */usr2/SNMP/rx.x/etc/backup*.

Note that **sksnmpbdb** will not back up SMDS configuration data or SMDS and Frame Relay data. The SMDS configuration and the SMDS and Frame Relay data will be restored after you or the cron process run the **sksnmpdbsync** and **sksnmpndsync** commands. When you run **sksnmpbdb**, it displays the following:

```
Backup Media:
1) Directory <$SNMPBKUPD>
2) Tape
3) Quit
Please specify backup media [+(1)]:
```

---

**Screen A-1. Backup Media Screen**

Enter **1** to back up your data to a directory.

Enter **2** to back up your data to a tape.

Enter **3** to exit.

## SKSNMPCF(1)

## SKSNMPCF(1)

### NAME

**sksnmpcf**—displays the SNMP Proxy Agent's ASCII-based user interface to the SNMP database.

### SYNOPSIS

**sksnmpcf**

### DESCRIPTION

The **sksnmpcf** command enables you to display the ASCII version of the menus and screens used to access the SNMP database. For details on how the user interface works, refer to **Chapter 3, Database Management**.

In order for forms to display properly, make sure the environment variable TERM is set to the correct terminal type. For example, **TERM=hp2621**.

**SKSNMPDBSYNC(1)****SKSNMPDBSYNC(1)****NAME**

**sksnmpdbsync**—retrieves Customer Network Management information for SMDS interfaces from the *StarKeeper* II NMS Core System database and stores it in the SNMP database.

**SYNOPSIS**

```
sksnmpdbsync <-c> | <-m [smds | frm | dlci ]>
```

**DESCRIPTION**

**sksnmpdbsync** retrieves SMDS configuration information from the *StarKeeper* II NMS Core System database for those Interface Indices and Customer Views that you have provisioned in the SNMP database.

Specify the **-c** option to synchronize the SMDS configuration in the SNMP database with the *StarKeeper* II NMS Core System database.

Specify the **-m** option to synchronize SMDS and Frame Relay measurement data in the SNMP database with the *StarKeeper* II NMS Core System database. One of the following options must be specified.

smds - synchronize SMDS measurements data

frm - synchronize Frame Relay measurement data

dlci - synchronize DLCI measurement data

**NOTE:**

SMDS configuration data are updated automatically in the SNMP database when **sksnmpdbsync** is invoked by cron. Run **sksnmpdbsync** manually only when you need immediate updates.

**EXAMPLE**

```
sksnmpdbsync -c
```

## SKSNMPGET(1)

## SKSNMPGET(1)

### NAME

**sksnmpget**—retrieves one or more individual variables from an SNMP agent.

### SYNOPSIS

**sksnmpget** <address> <community> <object\_id> [<object\_id>...]

### DESCRIPTION

**sksnmpget** is a tool used to retrieve one or more individual variables from an SNMP agent using a SNMP Get-Request PDU. The arguments are as follows:

<address>	The agent's IP address expressed either in the dot notation form, or as a host name that is listed in <i>/etc/hosts</i> .
<community>	The community string used to access the SNMP Proxy Agent.
<object_id>	One or more <b>fully</b> qualified variable(s) expressed as an <code>object_id</code> in the dot notation form or as a variable name as given in the MIB. For instance, <b>sksnmpget myhost public sysDescr.0</b> returns the variable <i>sysDescr.0</i> . Note that the call <b>sksnmpget myhost public sysDescr</b> would return an error from the agent since the variable is not a fully qualified SNMP variable.

### EXAMPLES

```
sksnmpget myhost public sysDescr.0
```

```
sksnmpget myhost public sysDescr.0 sysContact.0
```

## SKSNMPHELP(1)

## SKSNMPHELP(1)

### NAME

**sksnmphelp**—enables you to get on-line help.

### SYNOPSIS

**sksnmphelp** <name of command| VERSION>

### DESCRIPTION

The **sksnmphelp** command can be used to get general help information about the SNMP Proxy Agent, to get help information for a specific SNMP Proxy Agent command, or to display the software version.

To get general help, enter **sksnmphelp** without any arguments.

To get help information for a specific command, enter the command name as an argument.

To display the software version number, enter VERSION as the argument.

### EXAMPLES

**sksnmphelp**

**sksnmphelp sksnmpidb**

**sksnmphelp VERSION**

## SKSNMPIDB(1)

## SKSNMPIDB(1)

### NAME

**sksnmpidb**—initializes or reinitializes the SNMP database.

### SYNOPSIS

**sksnmpidb**

### DESCRIPTION

**sksnmpidb** creates and initializes the SNMP database, *snmpdb*. By default, **sksnmpidb** will create and initialize the SNMP database in directory *\$SNMP\_DBS (/usr2/SNMP/rx.x/snmpdbs)*.

The SNMP Proxy Agent must be stopped before you can reinitialize the SNMP database.

If a SNMP database already exists, **sksnmpidb** informs you that a database exists and asks if you want to remove the old database so that it can create a new one. If you proceed with the SNMP database initialization, the current SNMP database will be completely removed and a new SNMP database will be created in its place. All existing data will be deleted.

The SNMP database is automatically created during installation. The **sksnmpidb** command should only be used if you have to recreate the SNMP database.

If you want to back up the current database, use the **sksnmpbdb** command.

**SKSNMPNDSYNC(1)****SKSNMPNDSYNC(1)****NAME**

**sksnmpndsync**—synchronizes the SNMP database with the node.

**SYNOPSIS**

**sksnmpndsync** [-aoptions] [-i"ifindex1...n"][-nnode\_name][-fifindex\_file][-ofile]

**DESCRIPTION**

**sksnmpndsync** retrieves SMDS access class information and Frame Relay DLCI numbers from the network and inserts them into the SNMP database.

This command will run as a cron process to retrieve information from the nodes for any new Interface Indices that you have added to the SNMP database. Use the **sksnmpndsync** command to synchronize information immediately or when you make configuration changes to an Interface Index.

This command opens two files: 1) *sksnmp\_sync.log* and 2) *snmpsync.out* (can be renamed by using **-o** option). The file *snmpsync.log* contains all the activities generated by **sksnmpndsync**. The file *snmpsync.out* or the file specified by the **-o** option contains a list of *ifindices* that were not synchronized by this command. Generally failure to synchronize an *ifindex* occurs when connection to a specific node is not successful.

The following section describes each flag:

- aoptions** Three options are recognized here: 1) *frm*, 2) *smds*, and 3) *all*. Use this option to synchronize *ifindices* for the Frame Relay interfaces, SMDS interfaces and/or both. For example, **-a frm** synchronizes all the frame relay *ifindices* that are entered in the SNMP database. Other input options, i.e. **-i**, **-n**, or **-f**, are ignored when this option is used.
- i"ifindex1..n"** The **-i** option synchronizes one or more *ifindices* that are enclosed in double quotes and are separated by a space. For example, **-i"101 102 9999 2000"**.
- nnode\_name** The *node\_name* following **-n** must be entered exactly as the entry appears in the Node Name field on the "Network Interface Administration" form. This option causes all the *ifindices* that are provisioned on this node to be synchronized. This option could be used when a new node is added to the SNMP database. For example, **-a my\_rbo/myboc/myregion /co5551212** causes all the *ifindices* on the given node to be synchronized.

- f *ifindex\_file***      The *ifindex\_file* is the name of a file containing valid *ifindices* that require synchronization. Each line in this file lists one *ifindex* entry.
  
- o *ofile***              The file name following the **-o** option is the file that contains all the *ifindices* that were not synchronized. If this option is not given, the *ifindices* are written to a default file name, *snmpsync.out*.

## EXAMPLE

The following example causes all *ifindices* to be synchronized. The *ifindices* that are not synchronized are stored in file *failed.if*.

```
sksnmpndsync -a all -o failed.if
```

## WARNING

Since this command gets its information from the node, it could take some time to update all the *ifindices*. It is recommended that if you run this command manually, run it when there is little activity to the nodes by other *StarKeeper II* NMS applications, such as *Network Builder*.

**SKSNMPNEXT(1)****SKSNMPNEXT(1)****NAME**

**sksnmpnext**—retrieves one or more individual variables from an SNMP agent using an SNMP Get-Next Request PDU.

**SYNOPSIS**

**sksnmpnext** <address> <community> <object\_id> [<object\_id>...]

**DESCRIPTION**

**sksnmpnext** is a tool used to retrieve one or more individual variables from an SNMP agent using a SNMP Get-Next Request PDU. The arguments are as follows:

<address>	The agent's IP address expressed either in the dot notation form, or as a host name that is listed in <i>/etc/hosts</i> .
<community>	The community string used to access the SNMP Proxy Agent.
<object_id>	One or more variables expressed as an object_id in the dot notation form or as a variable name as given in the MIB. For instance, <b>sksnmpnext myhost public system</b> returns the variable <i>sysDescr.0</i> . Note that the call returns the fully qualified variable that is lexicographically greater than what was entered.

**EXAMPLE**

**sksnmpnext myhost public system**

**SKSNMPRDB(1)****SKSNMPRDB(1)****NAME**

**sksnmpbdb**—restores the data to the SNMP database, *snmpdb*.

**SYNOPSIS**

**sksnmpbdb**

**DESCRIPTION**

**sksnmpbdb** restores data to the SNMP database that has been backed up previously with the **sksnmpbdb** command. Before you run **sksnmpbdb**, stop the SNMP Proxy Agent by using the **sksnmpstop** command.

When you run **sksnmpbdb**, it prompts you for the **root** password and then displays the following:

```
Restore Media:
1) Directory
2) Tape
3) Quit
Please specify restore media [+(1)]:
```

---

**Screen A-2. Restore Media Screen**

Enter **1** to restore data from a directory, **2** to restore data from tape, or **3** to exit.

If you select 1 to restore data from a directory, the following prompt displays:

```
Enter path name of where backup tables are stored [q to quit +(/usr2/SNMP/
rx.x/etc/backup)]:
```

---

**Screen A-3. Restore from Directory Screen**

Enter a directory path or press **(ENTER)** to restore data from the default directory.

By default, **sksnmpbdb** restores the data from the directory *\$SNMPBKUPD*. *\$SNMPBKUPD* is set to */usr2/SNMP/rx.x/etc/backup*. *SNMPUPD* is an environment variable that is defined in the environment file *\$SNMP\_ENV/snmp.vars*.

**SKSNMPSET(1)****SKSNMPSET(1)****NAME**

**sksnmpset**—sets one or more individual variables on the SNMP agent's host using an SNMP Set Request PDU.

**SYNOPSIS**

**sksnmpset** <address> <community> [**<object\_id>** <type> <value>]

**DESCRIPTION**

**sksnmpset** is a tool used to set one or more individual variables by using an SNMP Set Request PDU. The arguments are as follows:

<address>	The agent's IP address expressed either in the dot notation form, or as a host name that is listed in <i>/etc/hosts</i> .
<community>	The community string used to access the SNMP Proxy Agent.
[<object_id> <type> <value>]	<p><b>&lt;object_id&gt;</b>—One or more variables expressed as an <b>object_id</b> in the dot notation form or as a variable name as given in the MIB.</p> <p><b>&lt;type&gt;</b>—Valid entries are: <b>i</b> (integer), <b>o</b> (octet), <b>d</b> (an object identifier in dot notation), <b>a</b> (IP address in dot notation), <b>c</b> (a counter), <b>g</b> (a gauge), and <b>t</b> (timeticks).</p> <p><b>&lt;value&gt;</b>—Action to be taken. <b>1</b>(add) or <b>2</b> (delete).</p>

**NOTE:**

Execute the **sksnmpupd** command to process the Set Requests issued by **sksnmpset**.

**EXAMPLES**

```
sksnmpset host public "lucentCNMsmdsIndScrSetAction.10.1.908.2224545" -i 1
```

This example is an add request for table `lucentCNMsmdsIndScrSetTable`. The request is to add individual screening address 19082224545 to Interface Index 10.

```
sksnmpset host public "lucentCNMsmdsIndScrSetAction.10.1.908.2224545" -i 2
```

This example is a delete request for table `lucentCNMsmdsIndScrSetTable`. The request is to delete individual screening address 19082224545 from Interface Index 10.

**SKSNMPSTART(1)****SKSNMPSTART(1)****NAME**

**sksnmpstart**—starts the SNMP Proxy Agent processes.

**SYNOPSIS**

**sksnmpstart** <number of servers>

**DESCRIPTION**

**sksnmpstart** starts the SNMP Proxy Agent daemon process *sksnmpd*. In turn, the *sksnmpd* process invokes *n* number of *sksnmpps* processes and the *sksnmpt* process.

You can start up a number of servers with **sksnmpstart**. The server process is *sksnmpps*. By default, if you execute **sksnmpstart** without specifying the number of servers, three servers will be started. It is useful to start up multiple servers if there are many SNMP Managers trying to access the SNMP Proxy Agent. The maximum number of servers that can be started is five.

To run **sksnmpstart**, you should be logged in as user **cnmsadm** and you must know the **root** password. **sksnmpstart** aborts if you fail to enter the correct **root** password.

Before **sksnmpstart** can start up *sksnmpd*, the *StarKeeper II* NMS Core System must be running. If the Core System is down, **sksnmpstart** issues a warning message indicating that the Core System is down and aborts. At this point, bring up the Core System and rerun **sksnmpstart**.

If *sksnmpd* is already running, **sksnmpstart** issues a warning message indicating that the SNMP Proxy Agent process is running and aborts.

If you want to restart the SNMP Proxy Agent process, stop all the SNMP Proxy Agent processes using **sksnmpstop** and then rerun **sksnmpstart**.

**NOTE:**

If you reboot the *StarKeeper II* NMS Core System, the SNMP Proxy Agent will automatically start up. You do not have to start it manually.

**EXAMPLES**

```
sksnmpstart
sksnmpstart 4
```

**SKSNMPSTAT(1)****SKSNMPSTAT(1)****NAME**

**sksnmpstat**—displays the operational status of the SNMP Proxy Agent, the current SNMP statistics of the SNMP Proxy Agent, and the SNMP statistics for an SNMP Manager.

**SYNOPSIS**

**sksnmpstat** <subscriber id>

**DESCRIPTION**

**sksnmpstat** displays the state of the SNMP Proxy Agent, the current SNMP statistics of the SNMP Proxy Agent, and if you enter a <subscriber id>, it displays statistics for an SNMP Manager.

The SNMP statistics that display when you use the **sksnmpstat** command also appear on usage reports. These statistics are described in “Sample Usage Reports” in **Chapter 3, Database Management**.

If the SNMP Proxy Agent is running, **sksnmpstat** indicates that the SNMP Proxy Agent is **UP**. If the SNMP Proxy Agent is not running, **sksnmpstat** indicates that the SNMP Proxy Agent is **DOWN**. If the SNMP Proxy Agent is down, look at the log file *\$SNMP\_LOG/snmp\_agent.log* to troubleshoot the problem.

**sksnmpstat** displays the current SNMP statistics of the SNMP Proxy Agent, such as the number of SNMP packets processed by the SNMP Proxy Agent and the number of SNMP errors encountered by the SNMP Proxy Agent. These statistics are collected by the SNMP Proxy Agent during run time. The SNMP statistical counts are reinitialized to zero when the SNMP Proxy Agent is restarted.

If you supply an SNMP Manager’s Subscriber Identifier as an option, **sksnmpstat** displays the SNMP statistics for that SNMP Manager. **sksnmpstat** displays such SNMP statistics as the number of SNMP packets sent to the Agent by that SNMP Manager and the number of SNMP errors encountered by the Agent for that SNMP Manager. These SNMP statistical counts are accumulated on an ongoing basis until you reinitialize them by using the **sksnmpcf** or the **sksnmpxadm** command.

**EXAMPLES**

```
sksnmpstat  
sksnmpstat customerA
```

## SKSNMPSTOP(1)

## SKSNMPSTOP(1)

### NAME

**sksnmpstop**—terminates the SNMP Proxy Agent processes.

### SYNOPSIS

**sksnmpstop**

### DESCRIPTION

**sksnmpstop** terminates the SNMP Proxy Agent daemon process *sksnmpd*. In turn, the *sksnmpd* process terminates all of the *sksnmpps* processes and the *sksnmpt* process.

To run **sksnmpstop**, you should be logged in as user **cnmsadm** and you must know the **root** password. **sksnmpstop** aborts if you fail to enter the correct **root** password.

## SKSNMPTRACE(1)

## SKSNMPTRACE(1)

### NAME

**sksnmptrace**—toggles the trace facility on and off.

### SYNOPSIS

**sksnmptrace** <on | off>

### DESCRIPTION

**sksnmptrace** is a tool that enables you to monitor the SNMP daemon processes. Trace messages are stored in *\$SNMP\_LOG/snmpagent.trace*.

<on> Tells the agent to log trace messages.

<off> Tells the agent to stop logging trace messages.

### EXAMPLES

**sksnmptrace on**

**sksnmptrace off**

## SKSNMPTRAP(1)

## SKSNMPTRAP(1)

### NAME

**sksnmptrap**—retrieves SNMP Trap PDUs from remote SNMP trap-generating systems.

### SYNOPSIS

**sksnmptrap**

### DESCRIPTION

**sksnmptrap** enables you to verify that the agent is sending out traps. When a network interface generates an alarm, the alarm is trapped by the daemon process *sksnmpt* and traps are sent to the appropriate SNMP Managers. **sksnmptrap** acts like an SNMP Manager in that it waits for traps to be generated and displays information when traps are received.

This tool must be run from **root** since it binds to the SNMP Trap port (udp/162) to listen for the traps. This tool decodes the incoming SNMP Trap PDUs and prints the decoded messages to standard output.

To run this command, log in to your system as **cnmsadm**, then enter **su root**. Execute the **sksnmptrap** command from the directory *\$SNMP\_BIN*. This command does not take any arguments.

**SKSNMPUPD(1)****SKSNMPUPD(1)****NAME**

**sksnmpupd**—executes pending Set Requests that have been submitted by SNMP Managers and updates the node, and the *StarKeeper II* NMS Core System and SNMP Proxy Agent databases.

**SYNOPSIS**

**sksnmpupd** [-d *dfile*] [-l] [-f]

**DESCRIPTION**

**sksnmpupd** modifies the node, *StarKeeper II* NMS Core System and SNMP Proxy Agent databases by processing Set Requests received from SNMP Managers and SNMP management applications.

When a customer submits a Set Request using their SNMP Manager or SNMP management application, the request is temporarily stored with other pending Set Requests in the `scr_reqts` table in the SNMP database. To update the node and the databases, you must execute the **sksnmpupd** command.

**sksnmpupd** logs all events and errors in log file `SNMP_LOG/snmp_set.log`. Refer to this log file when you run **sksnmpupd**. To browse through the log file, you can use such UNIX tools as **pg**, **cat**, or **vi**. Also, a detailed on-line report on each Set Request is produced if you use the **-l** option to **sksnmpupd**.

If errors occur, try to fix any error conditions prior to rerunning **sksnmpupd**. For example, if **sksnmpupd** tries to add an SMDS screening address to a port for a given node and the connection between the node and the *StarKeeper II* NMS Core System is down, the request will not be completed due to the down connection. Reactivate the connection and then rerun **sksnmpupd**.

The following section describes each flag:

- |                        |  |
|------------------------|--|
| <b>-d</b> <i>dfile</i> | Writes detailed debugging information to the file name specified. This option is intended for use by <i>StarKeeper II</i> NMS support personnel. |
| <b>-l</b>              | Displays an on-line report for each Set Request processed in the previous 48 hours.  |
| <b>-f</b>              | Displays an on-line report for all failed Set Requests.  |

When you run **sksnmpupd**, each Set Request is given a status of either Pending or Failed. **sksnmpupd** processes only Set Requests with a status of Pending. You can execute **sksnmpupd** up to 10 times to try to process a Set Request. If the request still can not be processed after the tenth try, the request is given a status of Failed. The failed request must be resubmitted by the SNMP Manager or SNMP management application.

Each Set Request starts off with an initial status of Pending. If a request is processed successfully, an appropriate message appears in the log file and the request is removed from the `scr_reqts` table.

If **sksnmpupd** can not update a node with the submitted Set Requests, then the *StarKeeper II* NMS Core System and SNMP Proxy Agent databases will also not be updated.

If **sksnmpupd** can update a node, but can not update the *StarKeeper II* NMS Core System and SNMP Proxy Agent databases, then run **skload** and **sksnmpdbsync** to keep the node and the databases synchronized.



**NOTE:**

The runtime of **sksnmpupd** depends on conditions such as the current load of the *StarKeeper II* NMS System and the availability of the AIEPs to the nodes. It is possible for **sksnmpupd** to take 5 minutes to run.

**EXAMPLE**

**sksnmpupd**

## SKSNMPWALK(1)

## SKSNMPWALK(1)

### NAME

**sksnmpwalk**—retrieves a class of variables from an SNMP agent using a series of SNMP Get-Next Request PDUs.

### SYNOPSIS

```
sksnmpwalk <address> <community> <object_id> [<object_id>...]
```

### DESCRIPTION

**sksnmpwalk** is a tool used to retrieve a class of variables from an SNMP agent using a series of SNMP Get-Next Request PDUs. The arguments are as follows:

<address>	The agent's IP address expressed either in the dot notation form, or as a host name that is listed in <i>/etc/hosts</i> .
<community>	The community string used to access the SNMP Proxy Agent.
<object_id>	One or more variables expressed as an <code>object_id</code> in the dot notation form or as a variable name as given in the MIB. For instance, <b>sksnmpwalk myhost public system</b> would traverse all the variables that form the system group of objects. The traversing of the variables stops when all of the classes being polled return a variable of a class different than what was requested.

### EXAMPLE

```
sksnmpwalk myhost public system
```

## SKSNMPXADM(1)

## SKSNMPXADM(1)

### NAME

**sksnmpxadm**—displays the SNMP Proxy Agent's Motif-based graphical user interface to the SNMP database.

### SYNOPSIS

```
sksnmpxadm [-d]
```

### DESCRIPTION

The **sksnmpxadm** command provides a graphical user interface that allows you to configure system information, network interfaces, customer views, customer view members, and subscribers in the SNMP database. For details on how to use the graphical user interface, refer to **Chapter 3, Database Management**.

Specify the **-d** option to display detailed debugging information in the window where **sksnmpxadm** was invoked. This option is intended for use by *StarKeeper II* NMS support personnel.

In order for forms to display properly, make sure the environment variable DISPLAY is set properly.

## **Crontab for cnmsadm**

---

The *crontab* for **cnmsadm** in */usr/spool/cron/crontab* will execute two commands, **sksnmpdbsync** and **sksnmpndsync**. These commands are invoked at various times during an hour and during a day. The times for invoking these commands by cron should not be altered.

**sksnmpdbsync** updates SMDS configuration information in the SNMP database. It extracts SMDS configuration information from the *StarKeeper II* NMS Core System database and assigns the information to the appropriate network interfaces and Customer Views.

**sksnmpndsync** updates Access Class information for SMDS network interfaces and DLCI information for Frame Relay network interfaces. **sksnmpndsync** goes to the nodes to get the information and updates the SNMP database.

Log files are generated when **sksnmpdbsync** and **sksnmpndsync** are invoked by cron. If there are any problems with SMDS or Frame Relay data updates in the SNMP database, browse through the logs to check for errors.

Log file *\$\$SNMP\_LOG/cnmcfgsynclog* contains log information when **sksnmpdbsync** updates configuration information.

Log file *\$\$SNMP\_LOG/ndsynchronlog* contains log information when **sksnmpndsync** updates Access Class and Frame Relay information.

## **Planning Forms**

---

This section provides planning forms to help you collect information to be entered into the SNMP database for an SNMP Manager.

For details on the fields that appear on these forms, refer to **Chapter 3, Database Management**.

There are four forms:

- **Customer Profile Worksheet**—Use this form to help develop a scenario for a customer. This form enables you to identify the network interfaces, Customer Views, and Subscriber Identifiers that a customer uses. This form provides a summary of customer data.
- **Customer View Form**—Use this form to record the name of each Customer View and its contents.
- **Subscriber Information Form**—Use this form to record the subscriber information.
- **Network Interfaces Form**—Use this form to gather data on the network interfaces that the SNMP Proxy Agent supports.

**SNMP PROXY AGENT —CUSTOMER PROFILE WORKSHEET**

CUSTOMER'S NAME: \_\_\_\_\_

**NETWORK INTERFACES**

Interface Index	Node Name	Concen. module	trunk address	Module address	Port	Virtual Port
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

**CUSTOMER VIEWS**

Customer View	Description	Members
[ ]	[ ]	[ ]
[ ]	[ ]	[ ]
[ ]	[ ]	[ ]
[ ]	[ ]	[ ]
[ ]	[ ]	[ ]
[ ]	[ ]	[ ]
[ ]	[ ]	[ ]

**SUBSCRIBER INFORMATION**

SUBSCRIBER IDENTIFIER:	[ ]
IP Address:	[ ]
READ	WRITE TRAP
Community String:	[ ] [ ] [ ] [ ]
Customer View:	[ ] [ ] [ ] [ ]
SUBSCRIBER IDENTIFIER:	[ ]
IP Address:	[ ]
READ	WRITE TRAP
Community String:	[ ] [ ] [ ] [ ]
Customer View:	[ ] [ ] [ ] [ ]
SUBSCRIBER IDENTIFIER:	[ ]
IP Address:	[ ]
READ	WRITE TRAP
Community String:	[ ] [ ] [ ] [ ]
Customer View:	[ ] [ ] [ ] [ ]



<b>SNMP PROXY AGENT —SUBSCRIBER INFORMATION FORM</b>			
CUSTOMER'S NAME: _____			
SUBSCRIBER IDENTIFIER:	[		]
Subscriber Description:	[		]
Subscriber Contact:	[		]
IP Address:	[		]
	READ	WRITE	TRAP
Community String:	[	]	[
Customer View:	[	]	[
Receive authentication traps: (y)es or (n)o			
CNM Server: (y)es or (n)o			
Service State: (i)n-service or (o)ut-of-service			
SUBSCRIBER IDENTIFIER:	[		]
Subscriber Description:	[		]
Subscriber Contact:	[		]
IP Address:	[		]
	READ	WRITE	TRAP
Community String:	[	]	[
Customer View:	[	]	[
Receive authentication traps: (y)es or (n)o			
CNM Server: (y)es or (n)o			
Service State: (i)n-service or (o)ut-of-service			





---

### SNMP Proxy Agent Error and Log Messages

---

This appendix provides a description of each message that may be written to the SNMP Proxy Agent log file, `$SNMP_LOG/snmp_agent.log`.

- **StarKeeper II NMS SNMP Agent started.**

This is an indication that the SNMP daemon `sksnmpd` has been invoked.

- **StarKeeper II NMS SNMP Server started.**

This is an indication that the SNMP server daemon `sksnmpls` has been invoked.

- **Sent Trap PDU type [n] to [a.b.c.d].**

This is an indication that the SNMP agent has sent a SNMP TRAP PDU to the specified destination.

- **Using default port number [x] for snmpd.**

This is an indication that the SNMP agent is using the specified default UDP port number to address the SNMP agent for receiving requests from SNMP Managers. In order to avoid this message in the future, make sure you update `/etc/services` with an entry for `snmp`.

- **Using default port number [x] for trapd.**

This is an indication that the SNMP agent is using the specified default UDP port number to address the SNMP Trap agent when sending traps to SNMP Managers. In order to avoid this message in the future, make sure you update `/etc/services` with an entry for `snmp-trap`.

- **StarKeeper II NMS SNMP Agent aborted.**

This is an indication that the SNMP daemon *sksnmpd* has been terminated.
- **StarKeeper II NMS SNMP Server aborted.**

This is an indication that the SNMP server daemon *sksnmpps* has been terminated.
- **Received a SIGTERM signal.**

This is an indication that the SNMP agent received a SIGTERM signal, and therefore is terminating.
- **Received a SIGCLD signal.**

This is an indication that the SNMP agent received a SIGCLD signal, and therefore is terminating.
- **Received a signal [x].**

This is an indication that the SNMP agent received an unexpected signal, and therefore is terminating.
- **Cannot open socket, errno=[n].**

This is an indication that the SNMP agent cannot create a socket. This may be due to either the TCP/IP network is not up or due to lack of general system resources.
- **Cannot bind socket, errno=[n].**

This is an indication that the SNMP agent cannot create a socket. This may be due to several reasons- another SNMP agent may be running, or the user has invalid permissions, or the TCP/IP network is not up or due to lack of general system resources.
- **Cannot create SNMP queue, errno=[n].**

This is an indication that the SNMP agent cannot create a queue, possibly due to lack of system resources, or the user has invalid permissions.
- **Cannot delete SNMP queue, errno=[n].**

This is an indication that the SNMP agent cannot delete a queue, possibly due to lack of system resources, or the user has invalid permissions.
- **Cannot get SNMP queue, errno=[n].**

This is an indication that the SNMP agent cannot get the queue id, possibly due to lack of system resources, or the user has invalid permissions.
- **Cannot write to SNMP queue, errno=[n].**

This is an indication that the SNMP agent cannot write a message to the queue, possibly due to lack of system resources, or the user has invalid permissions.

- **Cannot read from SNMP queue, errno=[n].**

This is an indication that the SNMP agent cannot receive a message from the queue, possibly due to lack of system resources, or the user has invalid permissions.

- **Cannot exec SNMP server, errno=[n].**

This is an indication that the SNMP daemon *sksnmpd* cannot spawn a new process, *sksnmps*, possibly due to lack of system resources, or the user has invalid permissions or environment.

- **Cannot send response PDU to [a.b.c.d], errno=[n]**

This is an indication that the SNMP agent cannot send an SNMP response to the specified destination, possibly due to lack of system resources.

- **Cannot send Trap PDU type [n] to [a.b.c.d], errno=[n].**

This is an indication that the SNMP agent cannot send an SNMP trap to the specified destination, possibly due to lack of system resources.

- **Not enough memory, errno=[n].**

This is an indication that the SNMP agent cannot allocate memory, possibly due to lack of system resources.

- **Cannot initialize SNMP database.**

This is an indication that the SNMP agent cannot initialize its database, possibly due to lack of system resources or because the user has invalid permissions or environment.

- **Cannot initialize SNMP MIB.**

This is an indication that the SNMP agent cannot initialize its MIB database, possibly due to lack of system resources or because the user has invalid permissions or environment.

- **Found duplicate object id [x.y.z] in SNMP MIB.**

This is an indication that the SNMP agent cannot initialize its MIB database, possibly due to a corrupt MIB file.

- **Cannot parse SNMP PDU from [a.b.c.d].**

This is an indication that the SNMP agent received a PDU from the specified address, and that the PDU cannot be parsed. The agent drops (ignores) such PDUs.

- **Cannot build response PDU to [a.b.c.d].**

This is an indication that the SNMP agent could not build a SNMP response PDU to the specified address, possibly due to lack of system resources.

- **Cannot update SNMP statistics for [a.b.c.d].**

This is an indication that the SNMP agent could not update the SNMP traffic statistics for the specified address, possibly due to lack of system resources or due to database inconsistencies.
- **Cannot get info for sending Trap PDU to [a.b.c.d].**

This is an indication that the SNMP agent could not get SNMP system information, possibly due to lack of system resources or due to database inconsistencies.
- **Received PDU from invalid IP Address: [a.b.c.d].**

This is an indication that the SNMP agent received a PDU from an unknown address. The agent drops (ignores) such PDUs.
- **Received PDU with invalid Community String [xxxx] from [a.b.c.d].**

This is an indication that the SNMP agent received an SNMP PDU from the specified address that included an unknown Community String. The agent drops (ignores) such PDUs. Valid Community Strings are registered in the SNMP database.
- **Received Set-Request PDU from [a.b.c.d].**

This is an indication that the SNMP agent received an unexpected Set-Request PDU from the specified address. The agent drops (ignores) such PDUs. Set-Request PDUs are not supported in this release of the SNMP agent.
- **Received invalid SNMP version [n] from [a.b.c.d].**

This is an indication that the SNMP agent received a PDU from the specified address, and that the PDU had a invalid value in the SNMP Version field. The agent drops (ignores) such PDUs.
- **Received Get-Response PDU from [a.b.c.d].**

This is an indication that the SNMP agent received an unexpected Get-Response PDU from the specified address. The agent drops (ignores) such PDUs. Get-Response PDUs are not supposed to be sent to SNMP agents.
- **Received Trap PDU from [a.b.c.d].**

This is an indication that the SNMP agent received an unexpected Trap PDU from the specified address. The agent drops (ignores) such PDUs. Trap PDUs are not supposed to be sent to SNMP agents.
- **Received invalid PDU [x] from [a.b.c.d].**

This is an indication that the SNMP agent received an invalid SNMP PDU from the specified address. The agent drops (ignores) such PDUs. Valid PDUs to SNMP agents are Get-Request, Get-Next-Request and Set-Request PDUs.

## **Error Messages from sksnmpndsync**

---

The following group of messages may appear when you run the **sksnmpndsync** command.

- **Could not open snmpdb**

This may be an indication that DBPATH is not set correctly

- **Could not get dlcis for ifindex=nnn**

This problem may result when the connection to the node is busy or the module pointed by ifindex is out of service.

- **Did not find access class for ifindex=[nnnn]**

This problem may arise if the SMDS Access Interface module is not in service.

- **No entry was found for ifindex=[nnn]**

This ifindex is not entered in the snmp database

- **Failed to get access classes from node=nodename**

This problem may result when the connection to the node is busy.

- **Could not get a queue**

This may be an indication that there is not any queue available. May need to clean the unused queues before issuing any console type command.

- **Received error message from node= nodename-Message:...**

This message identifies the type of error message that is received from the node.

- **Could not remove the queue=[nnn]**

This is an indication that the queue was not removed due to some abnormal termination.



---

# Supported SNMP MIB Objects and Traps

# C

---

## Overview

---

This appendix identifies the MIB objects and traps supported by the *StarKeeper II* NMS SNMP Proxy Agent. The design of these MIBs has been influenced by Bell Communication Research's Technical Advisory 1062, BellCore TRTSV00774, Internet RFCs, and *StarKeeper II* NMS features.

The included objects and traps are:

- standard MIB-II objects
- These include the *system* group of objects and a subset of the *interfaces* group of objects.
- standard MIB-II traps
- These include the *coldStart*, *authenticationFailure*, *linkUp* and *linkDown* traps.
- private MIB objects
- These objects are specific to the SNMP Proxy Agent and are classified into the following separate MIB modules:
  - CNM System MIB
  - CNM DS1 MIB
  - CNM DS3 MIB
  - CNM SIP MIB
  - CNM SMDS MIB
  - CNM Frame Relay MIB
  - CNM Enhanced Frame Relay MIB



**NOTE:**

Beginning with Release 9.0, *StarKeeper* II NMS provides new SNMP Proxy Agent MIB modules that use *Lucent* (*lucent*) in the object names. We recommend the use of the new MIBs, which use the “luc\_” prefix. However, the superseded “att\_” MIBs define the same objects, and the Proxy Agent supports the use of the previous MIBs.

## **Standard MIB-II Objects Supported by the SNMP Proxy Agent**

---

This section lists the MIB-II objects supported by the *StarKeeper* II NMS SNMP Proxy Agent. Depending on the object requested, the Proxy Agent may reply with the following:

The object's value

- A null value (0)
- A noSuchName error

Refer to *RFC1158: Management Information Base for Network Management of TCP/IP-based Internets: MIB-II* for details on the objects listed in this section.

### **MIB-II Objects That Return Their Value**

---

The SNMP Proxy Agent returns the following MIB-II objects with their value:

sysDescr	ifNumber
sysObjectID	ifIndex
sysUpTime	ifDescr
sysContact	ifType
sysName	ifMtu
sysLocation	ifSpeed
sysServices	
snmpInPkts	snmpInSetRequests
snmpOutPkts	snmpInGetResponses
snmpInBadVersions	snmpInTraps
snmpInBadCommunityNames	snmpOutTooBig
snmpInBadCommunityUses	snmpOutNoSuchNames
snmpInASNParseErrs	snmpOutBadValues
snmpInTooBig	snmpOutGenErrs
snmpInNoSuchNames	snmpOutGetRequests
snmpInBadValues	snmpOutGetNexts
snmpInReadOnly	snmpOutSetRequests
snmpInGenErrs	snmpOutGetResponses

snmplnTotalReqVars	snmplnTotalReqVars
snmplnTotalSetVars	snmpOutTraps
snmplnGetRequests	snmpEnableAuthenTraps
snmplnGetNexts	

### **MIB-II Objects That Return Null Value**

The Proxy Agent returns the following MIB-II objects with a null (0) value:

ifPhysAddress	ifOutOctets
ifAdminStatus	ifOutUcastPkts
ifOperStatus	ifOutNUcastPkts
ifLastChange	ifOutDiscards
ifInOctets	ifOutErrors
ifInUcastPkts	ifOutQLen
ifInNUcastPkts	ifSpecific
ifInDiscards	
ifInErrors	
ifInUnknownProtos	

### **MIB-II Objects That Return noSuchName Error**

The Proxy Agent returns all other objects that are in MIB-II (i.e. those that are not listed above) with a noSuchName error.

## **Standard MIB-II Traps Supported by the SNMP Proxy Agent**

The following table lists the standard MIB-II Traps supported by the Proxy Agent.

<b><i>StarKeeper II NMS SNMP MIB— Traps</i></b>	
<b>Trap</b>	<b>Event</b>
coldStartTrap	SNMP agent has been re-initialized
authenticationTrap	Received a SNMP PDU that fails authentication
linkDownTrap	An interface has gone down
linkUpTrap	An interface has come up

## Proxy Agent Behavior

---

### CNM System MIB

---

1. The values returned by the objects `lucentCNMsysDescr`, `lucentCNMsysObjectID`, `lucentCNMsysUpTime`, `lucentCNMsysContact`, `lucentCNMsysName`, `lucentCNMsysLocation`, and `lucentCNMsysServices` are the same as the values returned by the MIB II objects `sysDescr`, `sysObjectID`, `sysUpTime`, `sysContact`, `sysName`, `sysLocation`, and `sysServices` respectively.
2. The values returned by the objects `lucentCNMifNumber`, `lucentCNMifIndex`, `lucentCNMifDescr`, `lucentCNMifType`, `lucentCNMifMtu`, and `lucentCNMifSpeed` are the same as the values returned by the MIB II objects `ifNumber`, `ifIndex`, `ifDescr`, `ifType`, `ifMtu`, and `ifSpeed`.
3. The value returned by `lucentCNMifNumber` is the number of interfaces that are being managed by a subscriber. It does not reflect the total number of interfaces being managed by all subscribers.
4. Querying the objects in the `lucentCNMifStatusTable` may result in the SNMP Proxy Agent accessing the node (switch) in real-time. Whenever the value of the `lucentCNMifAdminStatus` or `lucentCNMifOperStatus` is "unknown," the SNMP Proxy Agent `lucent` attempts to update the object by querying the node in real-time. As a result, the responses to such SNMP queries may take a longer time than queries for other objects that do not require the Proxy Agent to access the node(s).
5. All the configuration data for this MIB is based on the information kept in the *StarKeeper* II NMS Core System and SNMP databases.

### CNM DS1 MIB

---

1. For SMDS interfaces, the value returned for `lucentCNMds1LineType` will always be "ds1ANSI-ESF."
2. For SMDS interfaces, the value returned for `lucentCNMds1ZeroCoding` will always be "ds1B8ZS."
3. The SNMP Proxy Agent supports the DS1 MIB for Enhanced Frame Relay T1 Interfaces also. However, these objects are accessible only to those customers that have subscribed to the full T1 rate. The interface speed would be defined as 1536000 bits per second through the Network Interfaces option of **`sksnmpcf`** or **`sksnmpxadm`**.
4. The SNMP Proxy Agent provides a maximum of 96 intervals of DS1 Error Counts for each SMDS/Frame Relay T1 interface. Each interval spans a maximum of 15 minutes.

The current interval (interval number 1) is always the latest 15-minute interval with respect to the current time. For example, if you query for data at 9:20 and specify interval number 1, the time period for this interval is 9:00-9:15. If you query for interval number 2 at 9:20, the time period for this interval is 8:45-9:00. All intervals are calculated with respect to the current time.

5. All the measurement and configuration data for this MIB is based on the information kept in the *StarKeeper* II NMS Core System and the SNMP databases.

### **CNM DS3 MIB**

---

1. For SMDS Interfaces, the value returned for `lucentCNMds3LineType` will always be "ds3ClearChannel."
2. For SMDS Interfaces, the value returned for `lucentCNMds3ZeroCoding` will always be "ds3B8ZS."
3. The SNMP Proxy Agent provides a maximum of 96 intervals of DS3 Error Counts for each SMDS interface. Each interval spans a maximum of 15 minutes.

The current interval (interval number 1) is always the latest 15-minute interval with respect to the current time. For example, if you query for data at 9:20 and specify interval number 1, the time period for this interval is 9:00-9:15. If you query for interval number 2 at 9:20, the time period for this interval is 8:45-9:00. All intervals are calculated with respect to the current time.

4. All the measurement and configuration data for this MIB is based on the information kept in the *StarKeeper* II NMS Core System database and the SNMP database.

### **CNM SIP MIB**

---

1. The SNMP Proxy Agent provides a maximum of 96 intervals of SIP Measurement Counts for each SMDS interface. Each interval spans a maximum of 15 minutes.

The current interval (interval number 1) is always the latest 15-minute interval with respect to the current time. For example, if you query for data at 9:20 and specify interval number 1, the time period for this interval is 9:00-9:15. If you query for interval number 2 at 9:20, the time period for this interval is 8:45-9:00. All intervals are calculated with respect to the current time.

2. All the SIP measurement data for this MIB is based on the information kept in the *StarKeeper* II NMS Core System database and the SNMP database.
3. Querying the objects in the `lucentCNMsipL3ErrorLogTable` results in the SNMP Proxy Agent accessing the node (switch) in real-time. As a result, the responses to such SNMP queries may take a longer time than queries for other objects that do not require the Proxy Agent to access the node(s).

## CNM SMDS MIB

---

1. The SNMP Proxy Agent provides a maximum of 96 intervals of SMDS Disagreement Counts for each SMDS interface. Each interval spans a maximum of 15 minutes.

The current interval (interval number 1) is always the latest 15-minute interval with respect to the current time. For example, if you query for data at 9:20 and specify interval number 1, the time period for this interval is 9:00-9:15. If you query for interval number 2 at 9:20, the time period for this interval is 8:45-9:00. All intervals are calculated with respect to the current time.

2. All the SMDS Configuration data, Address Tables, and the Disagreement counts for this MIB is based on the information kept in the *StarKeeper II* NMS Core System and the SNMP databases.
3. Querying the objects in the `lucentCNMsmdsDisagreeLogTable` results in the SNMP Proxy Agent accessing the node (switch) in real-time. As a result, the responses to such SNMP queries may take a longer time than queries for other objects that do not require the Proxy Agent to access the node(s).

## CNM FRM MIB

---

1. This MIB is supported only for Frame Relay R3.0 interfaces, that is FRM on *Datakit II* R3.0.
2. The SNMP Proxy Agent provides a maximum of 24 intervals of FRM Measurements for each FRM interface. Each interval spans a maximum of 60 minutes.

The current interval (interval number 1) is always the latest 1-hour period interval with respect to the current interval. For example, if you query for data at 9:20 and specify interval number 1, the time period for this interval is 8:00-9:00. If you query for interval number 2 at 9:20, the time period for this interval is 7:00-8:00. All other intervals (3-24) are calculated with respect to the current time.

3. All the FRM Configuration data and Measurement data for this MIB is based on the information kept in the *StarKeeper II* NMS Core System and the SNMP databases.
4. Whenever you add, delete or re-configure Frame Relay interfaces (ports) and DLCIs on the node (switch) make sure the new information is updated in the *StarKeeper II* NMS Core System and SNMP databases by running the **skload** and **sksnmpndsync** commands. Note that the **sksnmpndsync** command is also run by cron periodically.

## CNM Enhanced FRM MIB

---

1. This MIB supports all Frame Relay interfaces except Release 3.0. It supports FRM V35, FRM T1, FRM E1.

2. The SNMP Proxy Agent provides a maximum of 96 intervals (24 hours) of FRM measurement data for each Enhanced FRM interface. Each interval is updated every 15 minutes.

The current interval (interval number 1) always ends at the latest 15-minute interval with respect to the current time. The latest interval is updated every 15 minutes. The measurement count in an interval is *cumulative data from the top of the hour to the latest completed quarter hour* (15, 30, 45, 60). Therefore the length of an interval depends on the current time.

The previous intervals (2-96) each end at the preceding 15-minute interval. If *a*, *b*, *c*, and *d* are measurement counts, then for each hour, the first quarter hour will have a count of *a*, the second quarter hour will have a count of *a+b*, the third quarter hour will have a count of *a+b+c*, and the fourth quarter hour will have a count of *a+b+c+d*.

Refer to the following table for examples.

Time of Query	End Time for Interval 1	Data Accumulated	End Time for Interval 2	Data Accumulated
10:05	10:00	9:00 to 10:00	9:45	9:00 to 9:45
10:20	10:15	10:00 to 10:15	10:00	9:00 to 10:00
10:35	10:30	10:00 to 10:30	10:15	10:00 to 10:15
10:50	10:45	10:00 to 10:45	10:30	10:00 to 10:30

3. All the Enhanced FRM Configuration data and Measurement data for this MIB is based on the information kept in the *StarKeeper II* NMS Core System database and the SNMP database.
4. Whenever you add, delete or re-configure Frame Relay interfaces (ports) and DLCIs on the node (switch) make sure the new information is updated in the *StarKeeper II* NMS Core System and SNMP databases by running the **skload**, **cfg\_sync** and **sksnmpndsync** commands. Note that the **sksnmpndsync** command is also run via cron periodically.
5. For enhanced frame relay interfaces, the SNMP Proxy Agent provides peak measurements for the hour. Data is examined at 5-minute intervals and updated when a new peak for the hour is reached.

### General MIB Issues

- If you do a GET request and there is no data available for the requested interval, the SNMP Proxy Agent will return the message "No Such Name."
- If you do a GET-NEXT request and there is no data available for the requested interval, the SNMP Proxy Agent will return the next interval or the next object.
- The time period for a "15-minute interval" always starts on the quarter hour (0, 15, 30, or 45 minutes) and ends on the quarter hour (15, 30, 45, or 0 minutes).

- The time period for a “hourly interval” always starts at the top of the hour and ends at the end of the hour.
- The “LocalTime” for all measurement data is always the “END” time for that specified interval. For example if Local Time 10:15 is returned by `lucentCNMsipMeasLocalTime`, this indicates that the data is from 10:00 to 10:15.
- The “Time Stamp” for all measurement data is always in UNIX seconds and is always the “END” time for that specified interval. In the example above, 10:15 would be converted to UNIX seconds and returned by `lucentCNMsipMeasTimeStamp`.
- The SNMP Proxy Agent will not return anything to the SNMP Manager if a MIB instance in the *StarKeeper II* NMS database contains a value of -1. A -1 indicates that either the MIB instance does not exist or is not applicable for that ifindex. If a MIB table contains many -1 values and the SNMP Manager does a table walk of this MIB, the SNMP Manager may wait for a certain period of time before it gets a response from the SNMP Proxy Agent. The SNMP Proxy Agent will return a response once it finds a MIB object that does not have a -1 value. It is advisable for the SNMP Manager to limit the number of queries to the SNMP Proxy Agent until the SNMP Proxy Agent has completely processed all of the SNMP Manager requests.

## **Summary of Private CNM MIB Objects**

### **CNM System MIB**

The following tables list the objects specified in the CNM System MIB. These objects provide generic information about the overall SNMP Proxy Agent system and about the SMDS, ICI, and Frame Relay interfaces it supports.

<b>CNM SNMP MIB—Generic System Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMsysDescr	System description
lucentCNMsysObjectID	System object ID
lucentCNMsysUpTime	System up time
lucentCNMsysContact	System contact information
lucentCNMsysName	System name
lucentCNMsysLocation	System location
lucentCNMsysServices	System services
lucentCNMifNumber	Total number of interfaces supported
lucentCNMifConfigTable	Configuration table for all interfaces
lucentCNMifConfigEntry	An entry in the configuration table
lucentCNMifConfigIndex	Interface number
lucentCNMifDescr	Interface description
lucentCNMifType	Interface type
lucentCNMifMtu	Interface max. user data size
lucentCNMifSpeed	Interface line speed
lucentCNMifContact	Interface contact information
lucentCNMifLocation	Interface location
lucentCNMifSubscriber	Interface subscriber information
lucentCNMifStatusTable	Status table for all interfaces
lucentCNMifStatusEntry	An entry in the status table
lucentCNMifStatusIndex	Interface number
lucentCNMifAdminStatus	Expected state of the interface
lucentCNMifOperStatus	Actual operating state of the interface

**CNM DS1 MIB**

The following tables list the objects specified in the CNM DS1 MIB. They provide information about the DS1 (T1) layer of the SMDS and/or Frame Relay interfaces.

<b>CNM SNMP MIB— DS1 Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMds1ConfigTable	Configuration table for DS1 interfaces
lucentCNMds1ConfigEntry	An entry in the configuration table
lucentCNMds1ConfigIndex	Interface number
lucentCNMds1LineType	DS1 frame format supported by this interface
lucentCNMds1ZeroCoding	Zero code suppression/substitution used
lucentCNMds1ErrorsMaxIntervals	Max. number of intervals supported for DS1 errors
lucentCNMds1ErrorsIntervalLen	Time-length of each interval for DS1 errors
lucentCNMds1StatusTable	Status table for DS1 interfaces
lucentCNMds1StatusEntry	An entry in the status table
lucentCNMds1StatusIndex	Interface number
lucentCNMds1LineStatus	Most severe alarm condition outstanding on this interface
lucentCNMds1ErrorsTable	Errors table for DS1 interfaces
lucentCNMds1ErrorsEntry	An entry in the errors Table
lucentCNMds1ErrorsIndex	Interface number
lucentCNMds1ErrorsInterval	Interval number for which measurement is provided
lucentCNMds1ErrorsTimeStamp	Time stamp corresponding to end of specified interval
lucentCNMds1ErrorsLocalTime	Local time corresponding to end of specified interval
lucentCNMds1LCVs	Number of line code violations
lucentCNMds1LESs	Number of line errored seconds
lucentCNMds1LSESs	Number of line severely errored seconds
lucentCNMds1CVs	Number of code violations
lucentCNMds1ESs	Number of errored seconds

<b>CNM SNMP MIB— DS1 Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMds1SEsSs	Number of severely errored seconds
lucentCNMds1SEFSs	Number of severely errored framing seconds
lucentCNMds1UASs	Number of unavailable seconds
lucentCNMds1PctCU	Percent of channel io utilization
lucentCNMds1PctPkCU	Percentage of channel io utilization at 5-minute peak
lucentCNMds1ASs	Number of AIS active seconds
lucentCNMds1PctEFSSs	Percentage of error free seconds
lucentCNMds1FSSs	Number of frame slip seconds
lucentCNMds1FarLESs	Number of line errored seconds at far end
lucentCNMds1FarCVs	Number of code violations at far end
lucentCNMds1FarESs	Number of errored seconds at far end
lucentCNMds1FarSESs	Number of severely errored seconds at far end
lucentCNMds1FarSEFSs	Number of severely errored framing seconds at far end
lucentCNMds1FarPctEFSSs	Percentage of error free seconds at far end
lucentCNMds1FarFSSs	Number of frame slip seconds at far end
lucentCNMds1B6Ss	Percentage of seconds with 2 or more CRC errors
lucentCNMds1TotalErrorsTable	Protocol errors table for DS1 interfaces. All totals in this table are for 24-hour periods.
lucentCNMds1TotalErrorsEntry	An entry in the errors table
lucentCNMds1TotalErrorsIndex	Interface number
lucentCNMds1TotalLCVs	Number of line code violations
lucentCNMds1TotalLESs	Number of line errored seconds at near end
lucentCNMds1TotalLSESs	Number of severely errored seconds at near end
lucentCNMds1TotalCVs	Number of code violations at near end

<b>CNM SNMP MIB— DS1 Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMds1TotalESs	Number of errored seconds at near end
lucentCNMds1TotalSEsSs	Number of severely errored seconds at near end
lucentCNMds1TotalSEFSs	Number of severely errored framing seconds at near end
lucentCNMds1TotalUASs	Number of unavailable seconds
lucentCNMds1TotalASs	Number of AIS active seconds at near end
lucentCNMds1TotalFSSs	Number of frame slip seconds at near end
lucentCNMds1TotalFarLESs	Number of line errored seconds at near end
lucentCNMds1TotalFarCVs	Number of code violations at far end
lucentCNMds1TotalFarESs	Number of errored seconds at far end
lucentCNMds1TotalFarSEsSs	Number of severely errored seconds at far end
lucentCNMds1TotalFarSEFs	Number of severely errored framing seconds at far end
lucentCNMds1TotalFarFSSs	Number of frame slip seconds at far end
lucentCNMds1TotalB6Ss	Number of seconds with more than 2 CRC errors
lucentCNMds1plcpErrorsTable	PLCP errors table for DS1 interfaces
lucentCNMds1plcpErrorsEntry	An entry in the PLCP errors table
lucentCNMds1plcpErrorsIndex	Interface number
lucentCNMds1plcpErrorsInterval	Interval number for which measurement is provided
lucentCNMds1plcpErrorsTimeStamp	Time stamp corresponding to end of specified interval
lucentCNMds1plcpErrorsLocalTime	Local time corresponding to end of specified interval
lucentCNMds1plcpSEFs	Number of severely errored framing seconds
lucentCNMds1plcpUASs	Number of unavailable seconds
lucentCNMds1plcpStatusTable	Status table for DS1 interfaces at PLCP level
lucentCNMds1plcpStatusEntry	An entry in the status table

<b>CNM SNMP MIB— DS1 Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMds1plcpStatusIndex	Interface number
lucentCNMds1plcpLineStatus	Line status of interface at PLCP level

### **CNM DS3 MIB**

The following tables list the objects specified in the CNM DS3 MIB. They provide information about the DS3 (T3) layer of the SMDs interfaces.

<b>CNM SNMP MIB—DS3 Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMds3ConfigTable	Configuration table for DS3 interfaces
lucentCNMds3ConfigEntry	An entry in the configuration table
lucentCNMds3ConfigIndex	Interface number
lucentCNMds3LineType	DS3 frame format supported by this interface
lucentCNMds3ZeroCoding	Zero code suppression/substitution used
lucentCNMds3ErrorsMaxIntervals	Max. number of intervals supported for DS3 errors
lucentCNMds3ErrorsIntervalLen	Time-length of each interval for DS3 errors
lucentCNMds3StatusTable	Status table for DS3 interfaces
lucentCNMds3StatusEntry	An entry in the status table
lucentCNMds3StatusIndex	Interface number
lucentCNMds3LineStatus	Most severe alarm condition outstanding on this interface
lucentCNMds3ErrorsTable	Errors table for DS3 interfaces
lucentCNMds3ErrorsEntry	An entry in the errors table
lucentCNMds3ErrorsIndex	Interface number
lucentCNMds3ErrorsInterval	Interval number for which measurement is provided
lucentCNMds3ErrorsTimeStamp	Time stamp corresponding to end of specified interval
lucentCNMds3ErrorsLocalTime	Local time corresponding to end of specified interval

<b>CNM SNMP MIB—DS3 Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMds3LCVs	Number of line code violations
lucentCNMds3LESs	Number of line errored seconds
lucentCNMds3LSESs	Number of line severely errored seconds
lucentCNMds3CVs	Number of code violations
lucentCNMds3ESs	Number of errored seconds
lucentCNMds3SESs	Number of severely errored seconds
lucentCNMds3SEFSs	Number of severely errored framing seconds
lucentCNMds3UASs	Number of unavailable seconds

### **CNM SIP MIB**

The following tables list the objects specified in the CNM SIP MIB. They provide information about the Level 2 and Level 3 portion of the SMDS interfaces.

<b>CNM SNMP MIB — SIP Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMsipConfigTable	Configuration table for SIP interfaces
lucentCNMsipConfigEntry	An entry in the configuration table
lucentCNMsipConfigIndex	Interface number
lucentCNMsipMeasMaxIntervals	Max. number of intervals supported for SIP measurements
lucentCNMsipMeasIntervalLen	Time-length of each interval for SIP measurements
lucentCNMsipMeasTable	Measurements table for SIP interfaces
lucentCNMsipMeasEntry	An entry in the measurements table
lucentCNMsipMeasIndex	Interface number
lucentCNMsipMeasInterval	Interval number for which measurement is provided
lucentCNMsipMeasTimeStamp	Time stamp corresponding to end of specified interval
lucentCNMsipMeasLocalTime	Local time corresponding to end of specified interval

<b>CNM SNMP MIB — SIP Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMsipReceivedL3PDUs	Number of L3 PDUs received
lucentCNMsipSentL3PDUs	Number of L3 PDUs sent
lucentCNMsipReceivedGroupL3PDUs	Number of Group Addressed L3 PDUs received
lucentCNMsipSentGroupL3PDUs	Number of Group Addressed L3 PDUs sent
lucentCNMsipReceivedL2PDUs	Number of L2 PDUs received
lucentCNMsipSentL2PDUs	Number of L2 PDUs sent
lucentCNMsipL3ErrorLogTable	Protocol error log table for SIP interfaces
lucentCNMsipL3ErrorLogEntry	An entry in the error log table
lucentCNMsipL3ErrorLogIndex	Interface number
lucentCNMsipL3ErrorType	Protocol error type
lucentCNMsipL3ErrorSA	Source SMDS address
lucentCNMsipL3ErrorDA	Destination SMDS address
lucentCNMsipL3ErrorTimeStamp	Time stamp at the time of occurrence
lucentCNMsipL3ErrorLocalTime	Local time at the time of occurrence

### **CNM SMDS MIB**

The following tables list the objects specified in the CNM SMDS MIB. They provide SMDS subscriber information maintained by the SMDS interfaces.

<b>CNM SNMP MIB —SMDS Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMsmndsConfigTable	Configuration table for SMDS interfaces
lucentCNMsmndsConfigEntry	An entry in the configuration table
lucentCNMsmndsConfigIndex	Interface number
lucentCNMsmndsAccessClass	Access class supported
lucentCNMsmndsMCDUsIn	MCDUs allowed over interface from CPE to network
lucentCNMsmndsMCDUsOut	MCDUs allowed over interface from network to CPE

<b>CNM SNMP MIB —SMDS Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMsmdsIndivScreenMode	Allowed/Disallowed flag for individual address screening
lucentCNMsmdsGroupScreenMode	Allowed/Disallowed flag for group address screening
lucentCNMsmdsAddrIndexDescr	Describes how SMDS addresses are referenced to OIDs
lucentCNMsmdsDisagreeMaxIntervals	Max. no. of intervals supported for SMDS disagreements
lucentCNMsmdsDisagreeIntervalLen	Time-length of each interval for SMDS disagreements
lucentCNMsmdsAddrTable	Addresses table for SMDS interfaces
lucentCNMsmdsAddrEntry	An entry in the SNI Address table
lucentCNMsmdsAddrCountryIndex	Country code of the SMDS address
lucentCNMsmdsAddrAreaIndex	Area code of the SMDS address
lucentCNMsmdsAddrSubscriberIndex	Subscriber number of the SMDS address
lucentCNMsmdsAddressOnSNI	One of the SMDS addresses assigned to this interface
lucentCNMsmdsInterfaceIndex	Interface number
lucentCNMsmdsIndScrTable	Individual address screening table for SMDS interfaces
lucentCNMsmdsIndScrEntry	An entry in the screening table
lucentCNMsmdsIndScrIndex	Interface number
lucentCNMsmdsIndScrCountryIndex	Country code of the SMDS address
lucentCNMsmdsIndScrAreaIndex	Area code of the SMDS address
lucentCNMsmdsIndScrSubscriberIndex	Subscriber number of the SMDS address
lucentCNMsmdsIndivScreenAddress	One of the SMDS individual addresses used for screening
lucentCNMsmdsGrpScrTable	Group address screening table for SMDS interfaces
lucentCNMsmdsGrpScrEntry	An entry in the screening table
lucentCNMsmdsGrpScrIndex	Interface number
lucentCNMsmdsGrpScrCountryIndex	Country code of the SMDS address

<b>CNM SNMP MIB —SMDS Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMsmdsGrpScrAreaIndex	Area code of the SMDS address
lucentCNMsmdsGrpScrSubscriberIndex	Subscriber number of the SMDS address
lucentCNMsmdsGroupScreenAddress	One of the SMDS group addresses used for screening
lucentCNMsmdsMemGrpTable	Member-Group address table for SMDS interfaces
lucentCNMsmdsMemGrpEntry	An entry in the member-group table
lucentCNMsmdsMemGrpMemberCountryIndex	Country code of the SMDS member address
lucentCNMsmdsMemGrpMemberAreaIndex	Area code of the SMDS member address
lucentCNMsmdsMemGrpMemberSubscriberIndex	Subscriber number of the SMDS member address
lucentCNMsmdsMemGrpGroupCountryIndex	Country code of the SMDS group address
lucentCNMsmdsMemGrpGroupAreaIndex	Area code of the SMDS group address
lucentCNMsmdsMemGrpGroupSubscriberIndex	Subscriber number of the SMDS group address
lucentCNMsmdsMemberAddress	An SMDS individual address that is a member of a group
lucentCNMsmdsAssociatedGroup	An SMDS group address
lucentCNMsmdsGrpMemTable	Group-Member address table for SMDS interfaces
lucentCNMsmdsGrpMemEntry	An entry in the group-member table
lucentCNMsmdsGrpMemGroupCountryIndex	Country code of the SMDS group address
lucentCNMsmdsGrpMemGroupAreaIndex	Area code of the SMDS group address
lucentCNMsmdsGrpMemGroupSubscriberIndex	Subscriber number of the SMDS group address
lucentCNMsmdsGrpMemMemberCountryIndex	Country code of the SMDS member address
lucentCNMsmdsGrpMemMemberAreaIndex	Area code of the SMDS member address

<b>CNM SNMP MIB —SMDS Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMsmdsGrpMemMemberSubscriberIndex	Subscriber number of the SMDS member address
lucentCNMsmdsGroupAddress	An SMDS group address
lucentCNMsmdsGroupMember	An SMDS individual address that is a member of a group
lucentCNMsmdsDisagreeTable	Disagreements table for SMDS interfaces
lucentCNMsmdsDisagreeEntry	An entry in the disagreements table
lucentCNMsmdsDisagreeIndex	Interface number
lucentCNMsmdsDisagreeInterval	Interval number for which measurement is provided
lucentCNMsmdsDisagreeTimeStamp	Time stamp at the end of the measurement interval
lucentCNMsmdsDisagreeLocalTime	Local time at the end of the measurement interval
lucentCNMsmdsAccessClassExceededCounts	Access class exceeded
lucentCNMsmdsMCDUsExceededAtIngressCounts	MCDUs exceeded in ingress direction
lucentCNMsmdsMCDUsExceededAtEgressCounts	MCDUs exceeded in egress direction
lucentCNMsmdsSAScreenViolations	Source address screening violation
lucentCNMsmdsDAScreenViolations	Destination address screening violation
lucentCNMsmdsUnassignedSAs	Unassigned source address
lucentCNMsmdsDestinationSNIUnavailableCounts	Destination SNI unavailable
lucentCNMsmdsDisagreeLogTable	Disagreements log table for SMDS interfaces
lucentCNMsmdsDisagreeLogEntry	An entry in the disagreements log table
lucentCNMsmdsDisagreeLogIndex	Interface number
lucentCNMsmdsDisagreeLogType	Disagreement type
lucentCNMsmdsDisagreeLogSA	Source SMDS address
lucentCNMsmdsDisagreeLogDA	Destination SMDS address
lucentCNMsmdsDisagreeLogTimeStamp	Time stamp at the time of occurrence
lucentCNMsmdsDisagreeLogLocalTime	Local time at the time of occurrence

<b>CNM SNMP MIB —SMDS Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMsmdsXaSubscrTable	Subscription table containing ICI information for SNI interfaces
lucentCNMsmdsXaSubscrEntry	An entry in the subscription table
lucentCNMsmdsXaSubscrIndex	Interface number
lucentCNMsmdsXaSubscrPreselecCarrier	Preselected Carrier for this SNI
lucentCNMsmdsXaCICTable	Subscription table containing ICI information for ICI trunks
lucentCNMsmdsXaCICEntry	An entry in the subscription table
lucentCNMsmdsXaCICIndex	Interface number
lucentCNMsmdsXaCICCode	CIC Code
lucentCNMsmdsXaCICType	LEC or ICI
lucentCNMsmdsIndScrSetTable	Set table used for setting SMDS individual screening addresses
lucentCNMsmdsIndScrSetEntry	An entry in the Set table
lucentCNMsmdsIndScrSetIndex	Interface number
lucentCNMsmdsIndScrCountrySetIndex	Country code of the SMDS address
lucentCNMsmdsIndScrAreaSetIndex	Area code of the SMDS address
lucentCNMsmdsIndScrSubscriberSetIndex	Subscriber number of the SMDS address
lucentCNMsmdsIndScrSetAddress	One of the SMDS individual addresses that are being set (added or deleted)
lucentCNMsmdsIndScrSetAction	Set action is either Add or Delete
lucentCNMsmdsIndScrSetStatus	Status of set action (Pending or Failed)
lucentCNMsmdsIndScrSetTimeStamp	Time stamp showing the time that the set was submitted
lucentCNMsmdsIndScrSetLocalTime	Local time showing the time that the set was submitted
lucentCNMsmdsGrpScrSetTable	Set table used for setting SMDS group screening addresses
lucentCNMsmdsGrpScrSetEntry	An entry in the Set table
lucentCNMsmdsGrpScrSetIndex	Interface number
lucentCNMsmdsGrpScrCountrySetIndex	Country code of the SMDS address

<b>CNM SNMP MIB —SMDS Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMsmdsGrpScrAreaSetIndex	Area code of the SMDS address
lucentCNMsmdsGrpScrSubscriberSetIndex	Subscriber number of the SMDS address
lucentCNMsmdsGrpScrSetAddress	One of the SMDS group addresses that are being set (added or deleted)
lucentCNMsmdsGrpScrSetAction	Set action is either Add or Delete
lucentCNMsmdsGrpScrSetStatus	Status of set action (Pending or Failed)
lucentCNMsmdsGrpScrSetTimeStamp	Time stamp showing the time that the set was submitted
lucentCNMsmdsGrpScrSetLocalTime	Local time showing the time that the set was submitted

### **CNM Frame Relay MIB**

The following tables list the objects specified in the CNM Frame Relay MIB. They provide information about the Frame Relay R3.0 interfaces.

<b>CNM SNMP MIB — Frame Relay Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMfrConfigTable	Configuration table for Frame Relay R3.0 interfaces
lucentCNMfrConfigEntry	An entry in the configuration table
lucentCNMfrConfigIndex	Interface number
lucentCNMfrMeasMaxIntervals	Max. no. of intervals supported for port-level measurements
lucentCNMfrMeasIntervalLen	Time-length of each interval for port-level measurements
lucentCNMfrPVCMeasMaxIntervals	Max. no. of intervals supported for PVC measurements
lucentCNMfrPVCMeasIntervalLen	Time-length of each interval for PVC measurements
lucentCNMfrMeasTable	Measurements table for Frame Relay interfaces
lucentCNMfrMeasEntry	An entry in the measurements table

<b>CNM SNMP MIB — Frame Relay Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMfrMeasIndex	Interface number
lucentCNMfrMeasInterval	Interval number for which measurement is provided
lucentCNMfrMeasTimeStamp	Time stamp corresponding to end of measurement interval
lucentCNMfrMeasLocalTime	Local time corresponding to end of measurement interval
lucentCNMfrReceivedOctets	Number of octets received
lucentCNMfrSentOctets	Number of octets sent
lucentCNMfrReceivedFrames	Number of frames received
lucentCNMfrSentFrames	Number of frames sent
lucentCNMfrBadFrames	Number of bad frames received
lucentCNMfrReceiverOverruns	Number of receiver overruns
lucentCNMfrPVCMeasTable	Measurements table for Frame Relay PVCs
lucentCNMfrPVCMeasEntry	An entry in the measurements table
lucentCNMfrPVCMeasIfIndex	Interface number
lucentCNMfrPVCMeasIndex	DLCI number
lucentCNMfrPVCMeasInterval	Interval number for which measurement is provided
lucentCNMfrPVCMeasTimeStamp	Time stamp corresponding to end of measurement interval
lucentCNMfrPVCMeasLocalTime	Local time corresponding to end of measurement interval
lucentCNMfrCongestionAtIngress	Number of times frames were discarded due to congestion
lucentCNMfrCongestionAtEgress	Number of times frames were discarded due to congestion
lucentCNMfrPVCStatusTable	Status table for Frame Relay PVCs
lucentCNMfrPVCStatusEntry	An entry in the measurements table
lucentCNMfrPVCStatusIfIndex	Interface number

<b>CNM SNMP MIB — Frame Relay Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMfrPVCStatusIndex	DLCI number
lucentCNMfrPVCAdminStatus	Expected state of the Frame Relay PVC
lucentCNMfrPVCOperStatus	Actual operating state of the Frame Relay PVC

### **CNM Enhanced Frame Relay MIB**

The following tables list the objects specified in the CNM Enhanced Frame Relay MIB. They provide information about the Enhanced Frame Relay interfaces.

<b>CNM SNMP MIB—Enhanced Frame Relay Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMefrConfigTable	Configuration table for Enhanced Frame Relay interfaces
lucentCNMefrConfigEntry	An entry in the configuration table
lucentCNMefrConfigIndex	Interface number
lucentCNMefrConfigMgmtType	Identifies the PVC management scheme used on this port
lucentCNMefrConfigPollDirection	Polling direction for a valid PVC management scheme
lucentCNMefrConfigFullStatusPoll	Maximum number of Status Enquiry Intervals before issuance of a Full Status
lucentCNMefrConfigErrorThreshold	Maximum number of errored events before declaring the interface down
lucentCNMefrConfigMonitoredEvents	Used in conjunction with Error Threshold to declare link problem
lucentCNMefrConfigIntegrityTimer	Number of seconds between successive Status Enquiry
lucentCNMefrConfigPollVerifyTimer	Maximum second between Status Enquiries

<b>CNM SNMP MIB—Enhanced Frame Relay Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMefrConfigLMIFlowControl	Identifies that if the XON/XOFF in LMI management scheme is set
lucentCNMefrSupportedPVCs	Number of DLCIs supported on this Frame Relay interface
lucentCNMefrMeasMaxIntervals	Max. no. of intervals supported for port-level measurements
lucentCNMefrMeasIntervalLen	Time-length of each interval for port-level measurements
lucentCNMefrMaxFrameSize	Maximum frame size that can be received
lucentCNMefrAggregateCIR	Threshold for the sum of the CIRs
lucentCNMefrMinInterframeDelay	Minimum delay between transmission of frames
lucentCNMefrRate	Minimum pulse density for T1 interfaces
lucentCNMefrMeasTable	Measurements table for Enhanced Frame Relay interfaces
lucentCNMefrMeasEntry	An entry in the measurements table
lucentCNMefrMeasIndex	Interface number
lucentCNMefrMeasInterval	Interval number for which measurement is provided
lucentCNMefrMeasTimeStamp	Time stamp corresponding to end of measurement interval
lucentCNMefrMeasLocalTime	Local time corresponding to end of measurement interval
lucentCNMefrReceivedOctets	Number of octets received
lucentCNMefrSentOctets	Number of octets sent
lucentCNMefrReceivedFrames	Number of frames received
lucentCNMefrSentFrames	Number of frames sent
lucentCNMefrBadFrames	Number of bad frames received
lucentCNMefrReceiverOverruns	Number of receiver overruns

<b>CNM SNMP MIB—Enhanced Frame Relay Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMefrIngressUtil	Utilization level on the ingress side
lucentCNMefrEgressUtil	Utilization level on the egress side
lucentCNMefrPeakReceivedOctets	Peak number of bytes received for the hour
lucentCNMefrPeakReceivedFrames	Peak number of frames received for the hour
lucentCNMefrPeakSentOctets	Peak number of bytes sent for the hour
lucentCNMefrPeakSentFrames	Peak number of frames sent for the hour
lucentCNMefrPeakReceivedUtil	Percentage of utilization in the receive direction
lucentCNMefrPeakSentUtil	Percentage of utilization in the send direction
lucentCNMefrPeakRcvAvgFrmSize	Average frame size received in an hour
lucentCNMefrPeakSentAvgFrmSize	Average frame size sent in an hour
lucentCNMefrSentAvgFrmSize	Average frame size sent in a specified interval
lucentCNMefrReceivedAvgFrmSize	Average frame size received in a specified interval
lucentCNMefrReceivedCongestCount	No. of occurrences of buffer congestion in a specified interval
lucentCNMefrReceivedCongestSecs	No. of seconds of buffer congestion
lucentCNMefrSentCongestCount	No. of occurrences of transmitter buffer congestion
lucentCNMefrSentCongestSecs	No. of seconds of transmitter buffer congestion
lucentCNMefrSigUserLinkRelErrors	No. of user-side errored poll responses
lucentCNMefrSigNetLinkRelErrors	No. of network-side errored poll responses

<b>CNM SNMP MIB—Enhanced Frame Relay Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMefrSigNetUserProtErrors	No. of user-side local in-channel signaling protocol errors
lucentCNMefrSigNetProtErrors	No. of network-side local in-channel signaling protocol errors
lucentCNMefrPVCConfigTable	Configuration table for the PVCs on the Enhanced Frame Relay interfaces
lucentCNMefrPVCConfigEntry	An entry in the configuration table
lucentCNMefrPVCConfigIfIndex	Interface number
lucentCNMefrPVCConfigIndex	DLCI number on a frame relay interface
lucentCNMefrPVCServiceType	Identifies a DLCI as a unicast or multicast
lucentCNMefrLocalCIR	CIR on this PVC
lucentCNMefrLocalCommittedBurst	Committed Burst on this PVC
lucentCNMefrLocalExcessBurst	Excess Burst on this PVC
lucentCNMefrRemoteCIR	CIR on the remote end of this PVC
lucentCNMefrRemoteCommittedBurst	Committed Burst on the remote end of this PVC
lucentCNMefrRemoteExcessBurst	Excess Burst on the remote end of this PVC
lucentCNMefrMulticastGroup1	Identifies the first multicast DLCI for this PVC
lucentCNMefrMulticastGroup2	Identifies the second multicast DLCI for this PVC
lucentCNMefrMulticastGroup3	Identifies the third multicast DLCI for this PVC
lucentCNMefrMulticastGroup4	Identifies the forth multicast DLCI for this PVC
lucentCNMefrPVCMeasMaxIntervals	Max. no. of intervals supported for the PVCs
lucentCNMefrPVCMeasIntervallLen	Time-length of each interval for the PVCs

<b>CNM SNMP MIB—Enhanced Frame Relay Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMefrPVCMeasTable	Measurements table for Frame Relay PVCs
lucentCNMefrPVCMeasEntry	An entry in the measurements table
lucentCNMefrPVCMeasIfIndex	Interface number
lucentCNMefrPVCMeasIndex	DLCI number
lucentCNMefrPVCMeasInterval	Interval number for which measurement is provided
lucentCNMefrPVCMeasTimeStamp	Time stamp corresponding to end of measurement interval
lucentCNMefrPVCMeasLocalTime	Local time corresponding to end of measurement interval
lucentCNMefrPVCReceivedFrames	Number of frames received
lucentCNMefrPVCSentFrames	Number of frames sent
lucentCNMefrDiscardEligibilityFrames	No. of frames that exceeded Committed Burst Size
lucentCNMefrBurstSizeExceeded	No. of frames discarded due to enforcement of Committed Burst Size
lucentCNMefrCongestionAtIngress	Number of times frames were discarded due to congestion
lucentCNMefrCongestionAtEgress	Number of times frames were discarded due to congestion
lucentCNMefrPVCReceivedBytes	Number of bytes received
lucentCNMefrPVCSentBytes	Number of bytes sent
lucentCNMefrRecvDiscardEligFrames	No. of Discard Eligibility frames received
lucentCNMefrSentDiscardEligFrames	No. of Discard Eligibility frames sent
lucentCNMefrRecvForwardNoticFrames	No. of frames received with Forward Notification Bit for Congestion Control (FECN)
lucentCNMefrSentForwardNoticFramesNoticFrames	No. of frames sent with FECN
lucentCNMefrRecvBackwardNoticFrames	No. of frames received with Backward Notification Bit for Congestion Control (BECN)

<b>CNM SNMP MIB—Enhanced Frame Relay Information</b>	
<b>Object</b>	<b>Description</b>
lucentCNMefrSentBackwardNoticFrames	No. of frames sent with BECN
lucentCNMefrTotalRejectsSentRec	Total no. of rejected frames
lucentCNMefrPVCStatusTable	Status table for Frame Relay PVCs
lucentCNMefrPVCStatusEntry	An entry in the measurements table
lucentCNMefrPVCStatusIfIndex	Interface number
lucentCNMefrPVCStatusIndex	DLCI number
lucentCNMefrPVCAdminStatus	Expected state of the Frame Relay PVC
lucentCNMefrPVCOperStatus	Actual operating state of the Frame Relay PVC

## MIB Specification of Private CNM MIBs

### Format of Definitions

The next section contains the specification of all *StarKeeper II* NMS object types contained in the MIB. Following the conventions of the RFC 1212, the object types are defined using the following fields:

OBJECT-TYPE	A textual name, termed the OBJECT DESCRIPTOR, for the object type.
SYNTAX	The abstract syntax for the object type, presented using ASN.1. This must resolve to an instance of the ASN.1 type ObjectSyntax defined in the Structure of Management Information (SMI). SMI identifies the rules used to define the objects that can be accessed through a network management protocol.
ACCESS	A keyword, one of read-only, read-write, write-only, or not-accessible.
STATUS	A field describing the status of the object type.
DESCRIPTION	A textual description of the semantics of the object type. Implementations should ensure that their interpretation of the object type fulfills this definition since this MIB is intended for use in multi-vendor environments. As such it is vital that object types have consistent meaning across all machines.
::=	The OBJECT IDENTIFIER corresponding to the object type.

### CNM System MIB

```

LUCENT-CNM-SYSTEM-MIB DEFINITIONS ::= BEGIN

IMPORTS

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

```

#### Object Identifiers

```

lucent                OBJECT IDENTIFIER ::= { enterprises 1751 }
products              OBJECT IDENTIFIER ::= { lucent 1 }
mibs                  OBJECT IDENTIFIER ::= { lucent 2 }
lucent-cnmAgent       OBJECT IDENTIFIER ::= { products 5 }
lucent-cnm            OBJECT IDENTIFIER ::= { mibs 5 }

```

```
lucent-cnm-system      OBJECT IDENTIFIER ::= { lucent-cnm 1 }
lucent-cnm-interfaces  OBJECT IDENTIFIER ::= { lucent-cnm 2 }
```

### **System Group Of Objects (lucent-cnm-system)**

The System group of objects (lucent-cnm-system) provide information relevant to the entire system.

#### **lucentCNMsysDescr OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A printable ASCII string that is a textual description of this system. This value typically includes the full name and version identification of the system.

This identification can be used to distinguish this system from other systems for the purpose of fault identification/isolation, trouble reporting etc."

::= { lucent-cnm-system 1 }

#### **lucentCNMsysObjectID OBJECT-TYPE**

SYNTAX OBJECT IDENTIFIER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The vendor's authoritative identification of the network management subsystem contained in the entity.

This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining `what kind of box' is being managed. For example, if vendor `Flintstones, Inc.' was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its `Fred Router'.

This identification can be used to distinguish this system from other systems."

::= { lucent-cnm-system 2 }

#### **lucentCNMsysUpTime OBJECT-TYPE**

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time (in hundredths of a second) since the

SNMP interface portion of the system was last re-initialized.

This value may be used to determine the last time this system was re-initialized (i.e. the most recent reset for the SNMP interface)."

::= { lucent-cnm-system 3 }

**lucentCNMsysContact OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The textual identification of the contact person(s) or organization(s) for this system, along with information on how to contact them."

::= { lucent-cnm-system 4 }

**lucentCNMsysName OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"An administratively-assigned name for this system. By convention, this is the system's fully-qualified domain name.

This variable distinguishes this system from other systems in the network."

::= { lucent-cnm-system 5 }

**lucentCNMsysLocation OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The physical location of this system.

This variable distinguishes this system from others for location purposes."

::= { lucent-cnm-system 6 }

**lucentCNMsysServices OBJECT-TYPE**

SYNTAX INTEGER(0..127)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A value which indicates the set of services offered by the managed network elements.

The value is a sum. This sum initially takes the value zero, Then, for each layer, L, in the range 1 through 7, that this node performs transactions for, 2 raised to (L - 1) is added to the sum. For example, a node which performs primarily routing functions would have a value of 4 ( $2^{(3-1)}$ ). In contrast, a node which is a host offering application services would have a value of 72 ( $2^{(4-1)} + 2^{(7-1)}$ ). Note that in the context of the Internet suite of protocols, values should be calculated accordingly:

layer	functionality
1	physical (e.g., repeaters)
2	datalink/subnetwork (e.g., bridges)
3	internet (e.g., IP gateways)
4	end-to-end (e.g., IP hosts)
7	applications (e.g., mail relays)

For systems including OSI protocols, layers 5 and 6 may also be counted.

For SMDS and Frame Relay Service, this agent will return a value of 2 which indicates that the managed network elements offer a subnetwork-level service."

```
::= { lucent-cnm-system 7 }
```

### **Interfaces Group of Objects (lucent-cnm-interfaces)**

The Interfaces group of objects (lucent-cnm-interfaces) provides general information about all the interfaces supported by this system. It includes:

the Interfaces Configuration table  
the Interfaces Status table

#### **lucentCNMifNumber OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The number of network interfaces (regardless of their current state) that can be managed by a specific subscriber."

```
::= { lucent-cnm-interfaces 1 }
```

## Interfaces Configuration Table

---

This table provides configuration information on the interfaces managed by this system.

### **lucentCNMifConfigTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMifConfigEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "A list of entries containing general information  
    for all interfaces managed by this system."  
 ::= { lucent-cnm-interfaces 2 }

### **lucentCNMifConfigEntry OBJECT-TYPE**

SYNTAX LucentCNMifConfigEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "An entry containing general configuration  
    information for a particular interface."  
    INDEX{ lucentCNMifConfigIndex }  
 ::= { lucentCNMifConfigTable 1 }

### **LucentCNMifConfigEntry ::=**

SEQUENCE {  
    lucentCNMifConfigIndex  
        INTEGER,  
    lucentCNMifDescr  
        DisplayString,  
    lucentCNMifType  
        INTEGER,  
    lucentCNMifMtu  
        INTEGER,  
    lucentCNMifSpeed  
        Gauge,  
    lucentCNMifContact  
        DisplayString,  
    lucentCNMifLocation  
        DisplayString,  
    lucentCNMifSubscriber  
        DisplayString  
}

### **lucentCNMifConfigIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "A unique value assigned to each interface by  
    the administrator of this system."

```
::= { lucentCNMifConfigEntry 1 }
```

**lucentCNMifDescr OBJECT-TYPE**

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

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION
```

"A textual string containing information about this interface.

This string typically contains the name of the service provider, the service offered by this interface, and the name of the subscriber to this interface."

```
::= { lucentCNMifConfigEntry 2 }
```

**lucentCNMifType OBJECT-TYPE**

```
SYNTAX INTEGER {  
    other(1),  
    sip(31),  
    frame-relay(32)  
}
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION
```

"The type of interface being managed."

```
::= { lucentCNMifConfigEntry 3 }
```

**lucentCNMifMtu OBJECT-TYPE**

```
SYNTAX INTEGER
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION
```

"The size of the largest data PDU which can be sent/received on the interface, specified in octets.

For SMDS, this value represents the maximum L3 PDU size that is supported on this interface."

```
::= { lucentCNMifConfigEntry 4 }
```

**lucentCNMifSpeed OBJECT-TYPE**

```
SYNTAX Gauge
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION
```

"An estimate of the interface's current nominal bandwidth in bits per second."

```
::= { lucentCNMifConfigEntry 5 }
```

**lucentCNMifContact OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "A printable ASCII string that identifies the  
    contact person(s) or organization(s) responsible  
    for support of this interface."  
 ::= { lucentCNMifConfigEntry 6 }

**lucentCNMifLocation OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "A printable ASCII string that identifies the  
    physical location of the switching system that  
    terminates this interface."  
 ::= { lucentCNMifConfigEntry 7 }

**lucentCNMifSubscriber OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "A printable ASCII string that identifies the  
    contact person(s) or organization(s) that have  
    subscribed to this interface."  
 ::= { lucentCNMifConfigEntry 8 }

**Interfaces Status Table**

---

This table contains status information on the interfaces supported by this system.

**lucentCNMifStatusTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMifStatusEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "A list of entries containing status information  
    for all interfaces managed by this system."  
 ::= { lucent-cnm-interfaces 3 }

**lucentCNMifStatusEntry OBJECT-TYPE**

SYNTAX LucentCNMifStatusEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "An entry containing status information for a

```
        particular interface."
        INDEX{ lucentCNMifStatusIndex }
 ::= { lucentCNMifStatusTable 1 }

LucentCNMifStatusEntry ::=
    SEQUENCE {
        lucentCNMifStatusIndex
            INTEGER,
        lucentCNMifAdminStatus
            INTEGER,
        lucentCNMifOperStatus
            INTEGER
    }

lucentCNMifStatusIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "A unique value for each interface. The interface
        identified by a particular value of this index is
        the same interface as identified by the same value
        of an lucentCNMifConfigIndex object instance."
    ::= { lucentCNMifStatusEntry 1 }

lucentCNMifAdminStatus OBJECT-TYPE
    SYNTAX  INTEGER {
        up(1),
        down(2),
        testing(3),
        unknown(4)
    }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The desired state of this interface."
    ::= { lucentCNMifStatusEntry 2 }

lucentCNMifOperStatus OBJECT-TYPE
    SYNTAX  INTEGER {
        up(1),
        down(2),
        testing(3),
        unknown(4)
    }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The current operational state of this interface."
    ::= { lucentCNMifStatusEntry 3 }

END
```

## CNM Frame Relay MIB

---

```
LUCENT-CNM-FRAME-RELAY-MIB DEFINITIONS ::= BEGIN
IMPORTS
    enterprises, Gauge
        FROM RFC1155-SMI
    DisplayString
        FROM RFC1213-MIB
    OBJECT-TYPE
        FROM RFC-1212;

Object Identifiers

lucent          OBJECT IDENTIFIER ::= { enterprises 1751 }
products        OBJECT IDENTIFIER ::= { lucent 1 }
mibs            OBJECT IDENTIFIER ::= { lucent 2 }
lucent-cnmAgent OBJECT IDENTIFIER ::= { products 5 }
lucent-cnm      OBJECT IDENTIFIER ::= { mibs 5 }
lucent-cnm-fr   OBJECT IDENTIFIER ::= { lucent-cnm 7 }
```

### Frame Relay Group of Objects (lucent-cnm-fr)

---

The Frame Relay group of objects (lucent-cnm-fr) include:

- the Frame Relay Configuration table
- the Frame Relay Measurements table
- the Frame Relay PVC-level Measurements table
- the Frame Relay PVC-level Status tabl

### Frame Relay Configuration Table

---

This table provides configuration information the Frame Relay interfaces supported by this system.

#### **lucentCNMfrConfigTable OBJECT-TYPE**

```
SYNTAX SEQUENCE OF LucentCNMfrConfigEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A list of entries containing configuration
    information for all Frame Relay interfaces managed by
    this system."
::= { lucent-cnm-fr 1 }
```

#### **lucentCNMfrConfigEntry OBJECT-TYPE**

```
SYNTAX LucentCNMfrConfigEntry
ACCESS not-accessible
STATUS mandatory
```

## DESCRIPTION

"An entry containing configuration information for a particular Frame Relay interface."

```
INDEX { lucentCNMfrConfigIndex }
 ::= { lucentCNMfrConfigTable 1 }
```

**LucentCNMfrConfigEntry ::=**

```
SEQUENCE {
    lucentCNMfrConfigIndex
        INTEGER,
    lucentCNMfrMeasMaxIntervals
        INTEGER,
    lucentCNMfrMeasIntervalLen
        INTEGER,
    lucentCNMfrPVCMeasMaxIntervals
        INTEGER,
    lucentCNMfrPVCMeasIntervalLen
        INTEGER
}
```

**lucentCNMfrConfigIndex OBJECT-TYPE**

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

"A unique value for each Frame Relay interface. The interface identified by a particular value of this index is the same interface as identified by the same value of an lucentCNMifConfigIndex object instance."

```
::= { lucentCNMfrConfigEntry 1 }
```

**lucentCNMfrMeasMaxIntervals OBJECT-TYPE**

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

"This variable identifies the maximum number of measurement intervals supported for the measurements maintained by this Frame Relay interface in the lucentCNMfr

```
::= { lucentCNMfrConfigEntry 2 }
```

**lucentCNMfrMeasIntervalLen OBJECT-TYPE**

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

"This variable identifies the number of seconds that make up one complete measurement interval for

```
        for the measurements maintained by this Frame Relay
        interface in the lucentCNMfrMeasTable."
 ::= { lucentCNMfrConfigEntry 3 }
```

**lucentCNMfrPVCMeasMaxIntervals OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "This variable identifies the maximum number
        of measurement intervals supported for the
        measurements maintained by this Frame Relay
        interface in the lucentCNMfrPVCMeasTable."
 ::= { lucentCNMfrConfigEntry 4 }
```

**lucentCNMfrPVCMeasIntervalLen OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "This variable identifies the number of seconds
        that make up one complete measurement interval for
        for the measurements maintained by this Frame Relay
        interface in the lucentCNMfrPVCMeasTable."
 ::= { lucentCNMfrConfigEntry 5 }
```

**Frame Relay Measurements Table**

---

This table provides counts on the various measurement counts maintained by a Frame Relay interface during the specified measurement interval.

**lucentCNMfrMeasTable OBJECT-TYPE**

```
SYNTAX  SEQUENCE OF LucentCNMfrMeasEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
        "A list of entries containing measurement counts,
        maintained during the specified measurement interval,
        for all Frame Relay interfaces managed by this
        system."
 ::= { lucent-cnm-fr 2 }
```

**lucentCNMfrMeasEntry OBJECT-TYPE**

```
SYNTAX  LucentCNMfrMeasEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
        "An entry containing measurement counts, maintained
        during the specified measurement interval, for a
```

```

        particular Frame Relay interface."
        INDEX{ lucentCNMfrMeasIndex,
              lucentCNMfrMeasInterval }
 ::= { lucentCNMfrMeasTable 1 }

```

**LucentCNMfrMeasEntry ::=**

```

SEQUENCE {
    lucentCNMfrMeasIndex
        INTEGER,
    lucentCNMfrMeasInterval
        INTEGER,
    lucentCNMfrMeasTimeStamp
        INTEGER,
    lucentCNMfrMeasLocalTime
        DisplayString,
    lucentCNMfrReceivedOctets
        Gauge,
    lucentCNMfrSentOctets
        Gauge,
    lucentCNMfrReceivedFrames
        Gauge,
    lucentCNMfrSentFrames
        Gauge,
    lucentCNMfrBadFrames
        Gauge,
    lucentCNMfrReceiverOverruns
        Gauge
}

```

**lucentCNMfrMeasIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each Frame Relay interface.
    The interface identified by a particular value of
    this index is the same interface as identified by
    the same value of an lucentCNMifConfigIndex object
    instance."
 ::= { lucentCNMfrMeasEntry 1 }

```

**lucentCNMfrMeasInterval OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable identifies the measurement interval
    number for which measurement is provided. It is a
    number between 1 and XX, where 1 identifies the most
    recently completed measurement interval and XX is

```

the least recently completed measurement interval.

The value of XX is specified by the  
lucentCNMfrMeasMaxIntervals object given in the  
lucentCNMfrConfigTable.

The maximum length of each measurement interval is  
specified by the lucentCNMfrMeasIntervalLen object given  
in the lucentCNMfrConfigTable."

::= { lucentCNMfrMeasEntry 2 }

**lucentCNMfrMeasTimeStamp OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp corresponding to the end of the  
specified measurement interval, as measured in  
seconds from 00:00:00 UTC (Coordinated Universal  
Time) January 1, 1970. Any fraction is rounded up."

::= { lucentCNMfrMeasEntry 3 }

**lucentCNMfrMeasLocalTime OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp corresponding to the end of the  
specified measurement interval. Any fraction is  
rounded up. It is given as a printable ASCII string  
showing the local time at the end of the interval."

::= { lucentCNMfrMeasEntry 4 }

**lucentCNMfrReceivedOctets OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of bytes  
received by this Frame Relay interface during the  
specified measurement interval."

::= { lucentCNMfrMeasEntry 5 }

**lucentCNMfrSentOctets OBJECT-TYPE**

SYNTAX Gauge

ACCESS Read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of bytes  
sent by this Frame Relay interface during the

specified measurement interval."  
 ::= { lucentCNMfrMeasEntry 6 }

**lucentCNMfrReceivedFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of frames received by this Frame Relay interface during the specified measurement interval."  
 ::= { lucentCNMfrMeasEntry 7 }

**lucentCNMfrSentFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of frames received by this Frame Relay interface during the specified measurement interval."  
 ::= { lucentCNMfrMeasEntry 8 }

**lucentCNMfrBadFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of frames received by this interface that are faulty.  
  
A bad frame could be a frame with a non-integral number of octets, or an aborted frame, or a frame with a bad Frame Check Sequence (FCS)."  
 ::= { lucentCNMfrMeasEntry 9 }

**lucentCNMfrReceiverOverruns OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of receiver overruns occurred at this interface during the specified measurement interval.  
  
A receiver overrun occurs when frames are received faster than they can be stored and forwarded."  
 ::= { lucentCNMfrMeasEntry 10 }

## **Frame Relay PVC-Level Measurements Table**

This table provides counts on the various PVC-level measurement counts maintained by a Frame Relay interface during the specified measurement interval.

### **lucentCNMfrPVCMeasTable OBJECT-TYPE**

```
SYNTAX SEQUENCE OF LucentCNMfrPVCMeasEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A list of entries containing PVC-level measurement
    counts, maintained during the specified measurement
    interval, for all the Frame Relay interfaces managed
    by this system."
 ::= { lucent-cnm-fr 3 }
```

### **lucentCNMfrPVCMeasEntry OBJECT-TYPE**

```
SYNTAX LucentCNMfrPVCMeasEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "An entry containing measurement counts, maintained
    during the specified measurement interval, for a
    particular PVC on a specific Frame Relay interface."
INDEX { lucentCNMfrPVCMeasIfIndex,
        lucentCNMfrPVCMeasIfIndex,
        lucentCNMfrPVCMeasInterval }
 ::= { lucentCNMfrPVCMeasTable 1 }
```

### **LucentCNMfrPVCMeasEntry ::=**

```
SEQUENCE {
    lucentCNMfrPVCMeasIfIndex
        INTEGER,
    lucentCNMfrPVCMeasIndex
        INTEGER,
    lucentCNMfrPVCMeasInterval
        INTEGER,
    lucentCNMfrPVCMeasTimeStamp
        INTEGER,
    lucentCNMfrPVCMeasLocalTime
        DisplayString,
    lucentCNMfrCongestionAtIngress
        Gauge,
    lucentCNMfrCongestionAtEgress
        Gauge
}
```

### **lucentCNMfrPVCMeasIfIndex OBJECT-TYPE**

```
SYNTAX INTEGER
ACCESS read-only
```

```
STATUS    mandatory
DESCRIPTION
    "A unique value for each Frame Relay interface.
    The interface identified by a particular value of
    this index is the same interface as identified by
    the same value of an lucentCNMifConfigIndex object
    instance."
 ::= { lucentCNMfrPVCMeasEntry 1 }
```

**lucentCNMfrPVCMeasIndex OBJECT-TYPE**

```
SYNTAX    INTEGER
ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "The DLCI number on a Frame Relay interface that
    identifies a unique Data Link Connection."
 ::= { lucentCNMfrPVCMeasEntry 2 }
```

**lucentCNMfrPVCMeasInterval OBJECT-TYPE**

```
SYNTAX    INTEGER
ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "This variable identifies the measurement interval
    number for which measurement is provided. It is a
    number between 1 and XX, where 1 identifies the most
    recently completed measurement interval and XX is
    the least recently completed measurement interval."
 ::= { lucentCNMfrPVCMeasEntry 3 }
```

**lucentCNMfrPVCMeasTimeStamp OBJECT-TYPE**

```
SYNTAX    INTEGER
ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "The time stamp corresponding to the end of the
    specified measurement interval, as measured in
    seconds from 00:00:00 UTC (Coordinated Universal
    Time) January 1, 1970. Any fraction is rounded up."
 ::= { lucentCNMfrPVCMeasEntry 4 }
```

**lucentCNMfrPVCMeasLocalTime OBJECT-TYPE**

```
SYNTAX    DisplayString(SIZE(0..255))
ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "The time stamp corresponding to the end of the
    specified measurement interval. Any fraction is
    rounded up. It is given as a printable ASCII string
    showing the local time at the end of the interval."
```

```
::= { lucentCNMfrPVCMeasEntry 5 }
```

**lucentCNMfrCongestionAtIngress OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of times frames received by this PVC had to be discarded because buffers were unavailable or congestion control was being enforced during the specified measurement interval."

```
::= { lucentCNMfrPVCMeasEntry 6 }
```

**lucentCNMfrCongestionAtEgress OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of times frames sent by this PVC had to be discarded because buffers were unavailable or congestion control was being enforced during the specified measurement interval."

```
::= { lucentCNMfrPVCMeasEntry 7 }
```

## **Frame Relay PVC-Level Status Table**

---

This table provides status information on Data Link Connections on all Frame Relay interfaces supported by this system.

**lucentCNMfrPVCStatusTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMfrPVCStatusEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A list of entries containing status information about Data Link Connections on Frame Relay interfaces managed by this system."

```
::= { lucent-cnm-fr 4 }
```

**lucentCNMfrPVCStatusEntry OBJECT-TYPE**

SYNTAX LucentCNMfrPVCStatusEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"An entry containing status information about a particular Data Link Connection on a specific

```

        Frame Relay interface."
        INDEX{ lucentCNMfrPVCStatusIfIndex,
              lucentCNMfrPVCStatusIndex }
 ::= { lucentCNMfrPVCStatusTable 1 }

```

**LucentCNMfrPVCStatusEntry ::=**

```

SEQUENCE {
    lucentCNMfrPVCStatusIfIndex
        INTEGER,
    lucentCNMfrPVCStatusIndex
        INTEGER,
    lucentCNMfrPVCAdminStatus
        INTEGER,
    lucentCNMfrPVCOperStatus
        INTEGER
}

```

**lucentCNMfrPVCStatusIfIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each Frame Relay interface.
    The interface identified by a particular value of
    this index is the same interface as identified by
    the same value of an lucentCNMifConfigIndex object
    instance."
 ::= { lucentCNMfrPVCStatusEntry 1 }

```

**lucentCNMfrPVCStatusIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The DLCI number on a Frame Relay interface that
    identifies a unique Data Link Connection."
 ::= { lucentCNMfrPVCStatusEntry 2 }

```

**lucentCNMfrPVCAdminStatus OBJECT-TYPE**

```

SYNTAX  INTEGER {
    up(1),
    down(2),
    testing(3),
    unknown(4)
}
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The desired state of this Data Link Connection."
 ::= { lucentCNMfrPVCStatusEntry 3 }

```

**lucentCNMfrPVCOperStatus OBJECT-TYPE**

```

SYNTAX  INTEGER {
            up(1),
            down(2),
            testing(3),
            unknown(4)
        }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The current operational state of this Data Link
        Connection."
 ::= { lucentCNMfrPVCStatusEntry 4 }

```

**END****CNM SMDS MIB**

```

LUCENT-CNM-SMDS-MIB DEFINITIONS ::= BEGIN

```

**IMPORTS**

```

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

```

**Object Identifiers**

```

lucent                OBJECT IDENTIFIER ::= { enterprises 1751 }
products              OBJECT IDENTIFIER ::= { lucent 1 }
mibs                  OBJECT IDENTIFIER ::= { lucent 2 }
lucent-cnmAgent       OBJECT IDENTIFIER ::= { products 5 }
lucent-cnm             OBJECT IDENTIFIER ::= { mibs 5 }
lucent-cnm-smds       OBJECT IDENTIFIER ::= { lucent-cnm 6 }

```

**NOTE:**

All representations of SMDS addresses in this MIB module use, as a textual convention (i.e., this convention does not affect their encoding), the data type:

```

SMDSAddress ::= OCTET STRING(SIZE(8))

```

SMDSAddress is the 60-bit SMDS address, preceded by 4 bits with the following values:

```

    "1100" when representing an individual address
    "1110" when representing a group address

```

## **SMDS Group of Objects (lucent-cmn-smds)**

The SMDS Group of Objects (lucent-cmn-smds) include:

- the SMDS Configuration table
- the SMDS SNI Address table
- the SMDS Individual Address Screening table
- the SMDS Group Address Screening table
- the SMDS Member-Group Address table
- the SMDS Group-Member Address table
- the SMDS Disagreements table
- the SMDS Disagreements Log table

### **SMDS Configuration Table**

This table provides SMDS subscription parameters and configuration values maintained by all SMDS subscriber-network interfaces (SNIs) that are supported by this system.

#### **lucentCNMsmdsConfigTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmdsConfigEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A list of entries containing SMDS subscription parameters, and configuration values, for all the subscriber-network interfaces (SNIs) managed by this system."

::= { lucent-cnm-smds 1 }

#### **lucentCNMsmdsConfigEntry OBJECT-TYPE**

SYNTAX LucentCNMsmdsConfigEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"An entry containing SMDS subscription parameters, and configuration values for a specific subscriber network interface (SNI)."

INDEX{ lucentCNMsmdsConfigIndex }

::= { lucentCNMsmdsConfigTable 1 }

#### **LucentCNMsmdsConfigEntry ::=**

```
SEQUENCE {
    lucentCNMsmdsConfigIndex
        INTEGER,
    lucentCNMsmdsAccessClass
        INTEGER,
    lucentCNMsmdsMCDUsIn
        INTEGER,
```

```

    lucentCNMsmdsMCDUsOut
        INTEGER,
    lucentCNMsmdsIndivScreenMode
        INTEGER,
    lucentCNMsmdsGroupScreenMode
        INTEGER,
    lucentCNMsmdsAddrIndexDescr
        DisplayString,
    lucentCNMsmdsDisagreeMaxIntervals
        INTEGER,
    lucentCNMsmdsDisagreeIntervalLen
        INTEGER
}

```

**lucentCNMsmdsConfigIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each subscriber-network
    interface (SNI). The interface identified by a
    particular value of this index is the same
    interface as identified by the same value of an
    lucentCNMifConfigIndex object instance."
 ::= { lucentCNMsmdsConfigEntry 1 }

```

**lucentCNMsmdsAccessClass OBJECT-TYPE**

```

SYNTAX  INTEGER {
    noClass(1),
    accessClass1(2),
    accessClass2(3),
    accessClass3(4),
    accessClass4(5),
    accessClass5(6)
}
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The currently subscribed-to access class for this
    SNI. For a DS3-based access path, the value for this
    information indicates access class and the Sustained
    Information Rate. For a DS1-based access path, there
    is no access class enforcement and hence no
    information."
 ::= { lucentCNMsmdsConfigEntry 2 }

```

**lucentCNMsmdsMCDUsIn OBJECT-TYPE**

```

SYNTAX  INTEGER {
    mcdusIn1(1),
    mcdusIn16(2),
}

```

```
        mcdusIn32(3)
    }
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "The maximum number of SMDS data units that may be
    transferred concurrently over the SNI from the CPE
    to the SMDS network. Values can be 1, 16 or 32."
 ::= { lucentCNMsmdsConfigEntry 3 }
```

**lucentCNMsmdsMCDUsOut OBJECT-TYPE**

```
SYNTAX   INTEGER {
        mcdusOut1(1),
        mcdusOut16(2),
        mcdusOut32(3)
    }
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "The maximum number of SMDS data units that may be
    transferred concurrently over the SNI from the SMDS
    network to the CPE. Values can be 1, 16 or 32."
 ::= { lucentCNMsmdsConfigEntry 4 }
```

**lucentCNMsmdsIndivScreenMode OBJECT-TYPE**

```
SYNTAX   INTEGER {
        allowed(1),
        disallowed(2)
    }
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "The allowed/disallowed flag for the individual
    address screening table for this SNI. This indicates
    whether the individual address screening table
    includes 'allowed' addresses or 'disallowed'
    addresses (i.e. whether the screen is applied as an
    inclusionary or an exclusionary restriction)."
 ::= { lucentCNMsmdsConfigEntry 5 }
```

**lucentCNMsmdsGroupScreenMode OBJECT-TYPE**

```
SYNTAX   INTEGER {
        allowed(1),
        disallowed(2)
    }
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "The allowed/disallowed flag for the group address
    screening table for this SNI. This indicates whether
```

the group address screening table includes 'allowed' addresses or 'disallowed' addresses (i.e. whether the screen is applied as an inclusionary or an exclusionary restriction)."

```
 ::= { lucentCNMsmdsConfigEntry 6 }
```

**lucentCNMsmdsAddrIndexDescr OBJECT-TYPE**

```
SYNTAX  DisplayString(SIZE(0..255))
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
```

"This variable specifies how SMDS address tables should be indexed. A SMDS address is divided into three fields: Country Code, National Destination Code, and Subscriber Number. The National Destination Code is also referred to as the Area Code in the US. These three fields are used as indices into the tables that need to be indexed by a SMDS address.

This object will contain the following information, describing how this agent expects OIDs to be formed:

```
Country Code = 1
Area Code = next 3 digits (eg. 908)
Subscriber Number = remaining 7 digits
                    (eg. 5804357)
```

Therefore, for example, an OID associated with a SMDS address 0xC19085804357FFFF is derived to be 1.908.5804537

This information helps in specifying how this agent supports tables indexed by one or more SMDS addresses."

```
 ::= { lucentCNMsmdsConfigEntry 7 }
```

**lucentCNMsmdsDisagreeMaxIntervals OBJECT-TYP**

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

"This variable identifies the maximum number of measurement intervals supported for the disagreement counts maintained by this SNI in the lucentCNMsmdsDisagreeTable."

```
 ::= { lucentCNMsmdsConfigEntry 8 }
```

**lucentCNMsmdsDisagreeIntervalLen OBJECT-TYPE**

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

## DESCRIPTION

"This variable identifies the number of seconds that make up one complete measurement interval for the disagreement counts maintained by this SNI in the lucentCNMsmdsDisagreeTable.

```
::= { lucentCNMsmdsConfigEntry 9 }
```

**SMDS SNI Addresses Table**

---

This table lists the SMDS addresses assigned to the subscriber-network interfaces (SNIs) supported by this system.

**lucentCNMsmdsAddrTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmdsAddrEntry

ACCESS not-accessible

STATUS mandatory

## DESCRIPTION

"A table containing SMDS addresses assigned to each subscriber-network interface (SNI) that is managed by this system."

```
::= { lucent-cnm-smds 2 }
```

**lucentCNMsmdsAddrEntry OBJECT-TYPE**

SYNTAX LucentCNMsmdsAddrEntry

ACCESS not-accessible

STATUS mandatory

## DESCRIPTION

"An entry containing an SNI identifier, and one of the SMDS addresses assigned to that SNI."

```
INDEX { lucentCNMsmdsAddrCountryIndex,  
        lucentCNMsmdsAddrAreaIndex,  
        lucentCNMsmdsAddrSubscriberIndex }
```

```
::= { lucentCNMsmdsAddrTable 1 }
```

**LucentCNMsmdsAddrEntry ::=**

```
SEQUENCE {  
    lucentCNMsmdsAddrCountryIndex  
        INTEGER,  
    lucentCNMsmdsAddrAreaIndex  
        INTEGER,  
    lucentCNMsmdsAddrSubscriberIndex  
        INTEGER,  
    lucentCNMsmdsAddressOnSNI  
        SMDSAddress,  
    lucentCNMsmdsInterfaceIndex  
        INTEGER  
}
```

**lucentCNMsmdsAddrCountryIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The Country Code portion of the SMDS Address given by the lucentCNMsmdsAddressOnSNI object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsAddrEntry 1 }

**lucentCNMsmdsAddrAreaIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The Area Code (also known as the National Destination Code) portion of the SMDS Address given by the lucentCNMsmdsAddressOnSNI object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsAddrEntry 2 }

**lucentCNMsmdsAddrSubscriberIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The Subscriber Number portion of the SMDS Address given by the lucentCNMsmdsAddressOnSNI object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsAddrEntry 3 }

**lucentCNMsmdsAddressOnSNI OBJECT-TYPE**

SYNTAX SMDSAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A SMDS address assigned to a particular subscriber-network interface (SNI)."

```
 ::= { lucentCNMsmldsAddrEntry 4 }
```

**lucentCNMsmldsInterfaceIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The index that identifies the SNI that has been assigned a particular SMDS address."

```
 ::= { lucentCNMsmldsAddrEntry 5 }
```

### SMDS Individual Address Screening Table

This table provides the list of SMDS addresses that form the Individual Address Screening Table for a subscriber-network interface.

**lucentCNMsmldsIndScrTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmldsIndScrEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A list of entries containing SMDS addresses and SNI identifiers that form the individual address screening table for the SNIs."

```
 ::= { lucent-cnm-smlds 3 }
```

**lucentCNMsmldsIndScrEntry OBJECT-TYPE**

SYNTAX LucentCNMsmldsIndScrEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"An entry containing a SMDS address that belongs to the individual address screening table for a particular subscriber-network interface (SNI)."

```
 INDEX { lucentCNMsmldsIndScrIndex,
         lucentCNMsmldsIndScrCountryIndex,
         lucentCNMsmldsIndScrAreaIndex,
         lucentCNMsmldsIndScrSubscriberIndex }
 ::= { lucentCNMsmldsIndScrTable 1 }
```

**LucentCNMsmldsIndScrEntry ::=**

```
 SEQUENCE {
     lucentCNMsmldsIndScrIndex
         INTEGER,
     lucentCNMsmldsIndScrCountryIndex
         INTEGER,
     lucentCNMsmldsIndScrAreaIndex
         INTEGER,
     lucentCNMsmldsIndScrSubscriberIndex
```

```
        INTEGER,  
        lucentCNMsmndsIndivScreenAddress  
        SMDSAddress  
    }
```

**lucentCNMsmndsIndScrIndex OBJECT-TYPE**

```
SYNTAX  INTEGER  
ACCESS  read-only  
STATUS  mandatory  
DESCRIPTION  
    "A unique value for each SNI. The interface  
    identified by a particular value of this index is  
    the same interface as identified by the same value  
    of an lucentCNMifConfigIndex object instance."  
 ::= { lucentCNMsmndsIndScrEntry 1 }
```

**lucentCNMsmndsIndScrCountryIndex OBJECT-TYPE**

```
SYNTAX  INTEGER  
ACCESS  not-accessible  
STATUS  mandatory  
DESCRIPTION  
    "The Country Code portion of the SMDS Address  
    given by the lucentCNMsmndsIndivScreenAddress object.  
  
    This object is not accessible as it is used for  
    indexing purposes only. The attCNMsmndsAddrIndexDescr  
    object provides more information on the proper usage  
    and behaviour of such objects."  
 ::= { lucentCNMsmndsIndScrEntry 2 }
```

**lucentCNMsmndsIndScrAreaIndex OBJECT-TYPE**

```
SYNTAX  INTEGER  
ACCESS  not-accessible  
STATUS  mandatory  
DESCRIPTION  
    "The Area Code (also known as the National Destination  
    Code) portion of the SMDS Address given by the  
    lucentCNMsmndsIndivScreenAddress object.  
  
    This object is not accessible as it is used for  
    indexing purposes only. The attCNMsmndsAddrIndexDescr  
    object provides more information on the proper usage  
    and behaviour of such objects."  
 ::= { lucentCNMsmndsIndScrEntry 3 }
```

**lucentCNMsmndsIndScrSubscriberIndex OBJECT-TYPE**

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

DESCRIPTION

"The Subscriber Number portion of the SMDS Address given by the `lucentCNMsmdsIndivScreenAddress` object.

This object is used for indexing purposes only. The `lucentCNMsmdsAddrIndexDescr` object provides more information on the proper usage and behaviour of such objects."

::= { `lucentCNMsmdsIndScrEntry` 4 }

**`lucentCNMsmdsIndivScreenAddress` OBJECT-TYPE**

SYNTAX `SMDSAddress`

ACCESS `read-only`

STATUS `mandatory`

DESCRIPTION

"A SMDS address that belongs to the individual address screening table for a particular SNI."

::= { `lucentCNMsmdsIndScrEntry` 5 }

### SMDS Group Address Screening Table

This table provides the list of SMDS addresses that form the Group Address Screening Table for a subscriber-network interface.

**`lucentCNMsmdsGrpScrTable` OBJECT-TYPE**

SYNTAX `SEQUENCE OF LucentCNMsmdsGrpScrEntry`

ACCESS `not-accessible`

STATUS `mandatory`

DESCRIPTION

"A list of entries containing SMDS addresses and SNI identifiers that form the group address screening table for the SNIs."

::= { `lucent-cnm-smds` 4 }

**`lucentCNMsmdsGrpScrEntry` OBJECT-TYPE**

SYNTAX `LucentCNMsmdsGrpScrEntry`

ACCESS `not-accessible`

STATUS `mandatory`

DESCRIPTION

"An entry containing a SMDS address that belongs to the group address screening table for a particular subscriber-network interface (SNI)."

INDEX { `lucentCNMsmdsGrpScrIndex`,  
          `lucentCNMsmdsGrpScrCountryIndex`,  
          `lucentCNMsmdsGrpScrAreaIndex`,  
          `lucentCNMsmdsGrpScrSubscriberIndex` }

::= { `lucentCNMsmdsGrpScrTable` 1 }

**LucentCNMsmdsGrpScrEntry ::=**

```
SEQUENCE {
    lucentCNMsmdsGrpScrIndex
        INTEGER,
    lucentCNMsmdsGrpScrCountryIndex
        INTEGER,
    lucentCNMsmdsGrpScrAreaIndex
        INTEGER,
    lucentCNMsmdsGrpScrSubscriberIndex
        INTEGER,
    lucentCNMsmdsGroupScreenAddress
        SMDSAddress
}
```

**lucentCNMsmdsGrpScrIndex OBJECT-TYPE**

```
SYNTAX    INTEGER
ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "A unique value for each SNI. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMsmdsGrpScrEntry 1 }
```

**lucentCNMsmdsGrpScrCountryIndex OBJECT-TYPE**

```
SYNTAX    INTEGER
ACCESS    not-accessible
STATUS    mandatory
DESCRIPTION
    "The Country Code portion of the SMDS Address
    given by the lucentCNMsmdsGroupScreenAddress object.

    This object is not accessible as it is used for
    indexing purposes only. The attCNMsmdsAddrIndexDescr
    object provides more information on the proper usage
    and behaviour of such objects."
 ::= { lucentCNMsmdsGrpScrEntry 2 }
```

**lucentCNMsmdsGrpScrAreaIndex OBJECT-TYPE**

```
SYNTAX    INTEGER
ACCESS    not-accessible
STATUS    mandatory
DESCRIPTION
    "The Area Code (also known as the National Destination
    Code) portion of the SMDS Address given by the
    lucentCNMsmdsGroupScreenAddress object.

    This object is not accessible as it is used for
    indexing purposes only. The attCNMsmdsAddrIndexDescr
```

object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmndsGrpScrEntry 3 }

**lucentCNMsmndsGrpScrSubscriberIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"The Subscriber Number portion of the SMDS Address given by the lucentCNMsmndsGroupScreenAddress object.  
  
This object is not accessible as it is used for indexing purposes only. The attCNMsmndsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmndsGrpScrEntry 4 }

**lucentCNMsmndsGroupScreenAddress OBJECT-TYPE**

SYNTAX SMDSAddress  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"A SMDS address that belongs to the group address screening table for a particular SNI."  
 ::= { lucentCNMsmndsGrpScrEntry 5 }

**SMDS Member-Group Address Table**

---

This table identifies the group addresses that are associated with a particular individual address.

**lucentCNMsmndsMemGrpTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmndsMemGrpEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"A list of entries that identify the group addresses that have a particular individual address as a member."  
 ::= { lucent-cnm-smnds 5 }

**lucentCNMsmndsMemGrpEntry OBJECT-TYPE**

SYNTAX LucentCNMsmndsMemGrpEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"An entry identifying a group address that has a particular member individual address

```

        as its member."
INDEX { lucentCNMsmDsMemGrpMemberCountryIndex,
        lucentCNMsmDsMemGrpMemberAreaIndex,
        lucentCNMsmDsMemGrpMemberSubscriberIndex,
        lucentCNMsmDsMemGrpGroupCountryIndex,
        lucentCNMsmDsMemGrpGroupAreaIndex,
        lucentCNMsmDsMemGrpGroupSubscriberIndex }
 ::= { lucentCNMsmDsMemGrpTable 1 }

```

**LucentCNMsmDsMemGrpEntry ::=**

```

SEQUENCE {
    lucentCNMsmDsMemGrpMemberCountryIndex
        INTEGER,
    lucentCNMsmDsMemGrpMemberAreaIndex
        INTEGER,
    lucentCNMsmDsMemGrpMemberSubscriberIndex
        INTEGER,
    lucentCNMsmDsMemGrpGroupCountryIndex
        INTEGER,
    lucentCNMsmDsMemGrpGroupAreaIndex
        INTEGER,
    lucentCNMsmDsMemGrpGroupSubscriberIndex
        INTEGER,
    lucentCNMsmDsMemberAddress
        SMDSAddress,
    lucentCNMsmDsAssociatedGroup
        SMDSAddress
}

```

**lucentCNMsmDsMemGrpMemberCountryIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
    "The Country Code portion of the SMDS Address
    given by the lucentCNMsmDsMemberAddress object.

    This object is not accessible as it is used for
    indexing purposes only. The attCNMsmDsAddrIndexDescr
    object provides more information on the proper usage
    and behaviour of such objects."
 ::= { lucentCNMsmDsMemGrpEntry 1 }

```

**lucentCNMsmDsMemGrpMemberAreaIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
    "The Area Code (also known as the National Destination
    Code) portion of the SMDS Address given by the

```

lucentCNMsmdsMemberAddress object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsMemGrpEntry 2 }

**lucentCNMsmdsMemGrpMemberSubscriberIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The Subscriber Number portion of the SMDS Address given by the lucentCNMsmdsMemberAddress object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsMemGrpEntry 3 }

**lucentCNMsmdsMemGrpGroupCountryIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The Country Code portion of the SMDS Address given by the lucentCNMsmdsAssociatedGroup object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsMemGrpEntry 4 }

**lucentCNMsmdsMemGrpGroupAreaIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The Area Code (also known as the National Destination Code) portion of the SMDS Address given by the lucentCNMsmdsAssociatedGroup object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsMemGrpEntry 5 }

**lucentCNMsmmsMemGrpGroupSubscriberIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "The Subscriber Number portion of the SMDS Address given by the lucentCNMsmmsAssociatedGroup object.  
  
    This object is not accessible as it is used for indexing purposes only. The attCNMsmmsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmmsMemGrpEntry 6 }

**lucentCNMsmmsMemberAddress OBJECT-TYPE**

SYNTAX SMDSAddress  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "A SMDS individual address that is a member of a particular group."  
 ::= { lucentCNMsmmsMemGrpEntry 7 }

**lucentCNMsmmsAssociatedGroup OBJECT-TYPE**

SYNTAX SMDSAddress  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The SMDS group address that identifies a group."  
 ::= { lucentCNMsmmsMemGrpEntry 8 }

**SMDS Group-Member Address Table**

---

This table identifies the member individual addresses that belong to a particular group, identified by a group address.

**lucentCNMsmmsGrpMemTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmmsGrpMemEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "A list of entries that describe the group address table for this SMDS network."  
 ::= { lucent-cnm-smms 6 }

**lucentCNMsmmsGrpMemEntry OBJECT-TYPE**

SYNTAX LucentCNMsmmsGrpMemEntry  
ACCESS not-accessible

```

STATUS    mandatory
DESCRIPTION
    "An entry that identifies a member individual
    address that belongs to a particular group,
    identified by a group address."
INDEX     { lucentCNMsmndsGrpMemGroupCountryIndex,
            lucentCNMsmndsGrpMemGroupAreaIndex,
            lucentCNMsmndsGrpMemGroupSubscriberIndex,
            lucentCNMsmndsGrpMemMemberCountryIndex,
            lucentCNMsmndsGrpMemMemberAreaIndex,
            lucentCNMsmndsGrpMemMemberSubscriberIndex }
 ::= { lucentCNMsmndsGrpMemTable 1 }

```

**LucentCNMsmndsGrpMemEntry ::=**

```

SEQUENCE {
    lucentCNMsmndsGrpMemGroupCountryIndex
        INTEGER,
    lucentCNMsmndsGrpMemGroupAreaIndex
        INTEGER,
    lucentCNMsmndsGrpMemGroupSubscriberIndex
        INTEGER,
    lucentCNMsmndsGrpMemMemberCountryIndex
        INTEGER,
    lucentCNMsmndsGrpMemMemberAreaIndex
        INTEGER,
    lucentCNMsmndsGrpMemMemberSubscriberIndex
        INTEGER,
    lucentCNMsmndsGroupAddress
        SMDSAddress,
    lucentCNMsmndsGroupMember
        SMDSAddress
}

```

**lucentCNMsmndsGrpMemGroupCountryIndex OBJECT-TYPE**

```

SYNTAX    INTEGER
ACCESS    not-accessible
STATUS    mandatory
DESCRIPTION
    "The Country Code portion of the SMDS Address
    given by the lucentCNMsmndsMemberAddress object.

    This object is not accessible as it is used for
    indexing purposes only. The attCNMsmndsAddrIndexDescr
    object provides more information on the proper usage
    and behaviour of such objects."
 ::= { lucentCNMsmndsGrpMemEntry 1 }

```

**lucentCNMsmndsGrpMemGroupAreaIndex OBJECT-TYPE**

```

SYNTAX    INTEGER
ACCESS    not-accessible

```

STATUS mandatory  
DESCRIPTION  
"The Area Code (also known as the National Destination Code) portion of the SMDS Address given by the lucentCNMsmdsMemberAddress object.  
  
This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmdsGrpMemEntry 2 }

**lucentCNMsmdsGrpMemGroupSubscriberIndex OBJECT-TYPE**

SYNTAX INTEGERSYNTAX INTEGER  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"The Subscriber Number portion of the SMDS Address given by the lucentCNMsmdsMemberAddress object.  
  
This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmdsGrpMemEntry 3 }

**lucentCNMsmdsGrpMemMemberCountryIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"The Country Code portion of the SMDS Address given by the lucentCNMsmdsGroupMember object.  
  
This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmdsGrpMemEntry 4 }

**lucentCNMsmdsGrpMemMemberAreaIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"The Area Code (also known as the National Destination Code) portion of the SMDS Address given by the lucentCNMsmdsGroupMember object.  
  
This object is not accessible as it is used for

indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmdsGrpMemEntry 5 }

**lucentCNMsmdsGrpMemMemberSubscriberIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"The Subscriber Number portion of the SMDS Address given by the lucentCNMsmdsGroupMember object.  
  
This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmdsGrpMemEntry 6 }

**lucentCNMsmdsGroupAddress OBJECT-TYPE**

SYNTAX SMDSAddress  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"A SMDS group address."  
 ::= { lucentCNMsmdsGrpMemEntry 7 }

**lucentCNMsmdsGroupMember OBJECT-TYPE**

SYNTAX SMDSAddress  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"A SMDS individual address that is a member of a particular group, identified by a group address."  
 ::= { lucentCNMsmdsGrpMemEntry 8 }

**SMDS Disagreements Table**

---

This table provides the counts on the various SMDS disagreement errors encountered by all SNIs supported by this system.

**lucentCNMsmdsDisagreeTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmdsDisagreeEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"A list of entries containing the SMDS disagreement counts, maintained during the specified measurement interval, for all SNIs managed by this system."  
 ::= { lucent-cnm-smds 7 }

**lucentCNMsmdsDisagreeEntry OBJECT-TYPE**

```

SYNTAX   LucentCNMsmdsDisagreeEntry
ACCESS   not-accessible
STATUS   mandatory
DESCRIPTION
    "An entry containing the SMDS disagreement counts,
    maintained during the specified measurement interval,
    for a particular SNI."
INDEX    { lucentCNMsmdsDisagreeIndex,
            lucentCNMsmdsDisagreeInterval }
 ::= { lucentCNMsmdsDisagreeTable 1 }

```

**LucentCNMsmdsDisagreeEntry ::=**

```

SEQUENCE {
    lucentCNMsmdsDisagreeIndex
        INTEGER,
    lucentCNMsmdsDisagreeInterval
        INTEGER,
    lucentCNMsmdsDisagreeTimeStamp
        INTEGER,
    lucentCNMsmdsDisagreeLocalTime
        DisplayString,
    lucentCNMsmdsAccessClassExceededCounts
        Gauge,
    lucentCNMsmdsMCDUsExceededAtIngressCounts
        Gauge,
    lucentCNMsmdsMCDUsExceededAtEgressCounts
        Gauge,
    lucentCNMsmdsSAScreenViolations
        Gauge,
    lucentCNMsmdsDAScreenViolations
        Gauge,
    lucentCNMsmdsUnassignedSAs
        Gauge,
    lucentCNMsmdsDestinationSNIUnavailableCounts
        Gauge
}

```

**lucentCNMsmdsDisagreeIndex OBJECT-TYPE**

```

SYNTAX   INTEGER
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "A unique value for each SNI. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMsmdsDisagreeEntry 1 }

```

**lucentCNMsmdsDisagreeInterval OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"This variable identifies the measurement interval number for which measurement is provided. It is a number between 1 and XX, where 1 identifies the most recently completed measurement interval and XX is the least recently completed measurement interval.

The value of XX is specified by the lucentCNMsmdsDisagreeMaxIntervals object given in the lucentCNMsmdsConfigTable.

The maximum length of each measurement interval is specified by the lucentCNMsmdsDisagreeIntervalLen object given in the lucentCNMds1ConfigTable."

::= { lucentCNMsmdsDisagreeEntry 2 }

**lucentCNMsmdsDisagreeTimeStamp OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The time stamp corresponding to the end of the specified measurement interval, as measured in seconds from 00:00:00 UTC (Coordinated Universal Time) January 1, 1970. Any fraction is rounded up."

::= { lucentCNMsmdsDisagreeEntry 3 }

**lucentCNMsmdsDisagreeLocalTime OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The time stamp corresponding to the end of the specified measurement interval. Any fraction is rounded up. It is given as a printable ASCII string showing the local time at the end of the interval."

::= { lucentCNMsmdsDisagreeEntry 4 }

**lucentCNMsmdsAccessClassExceededCounts OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The counter associated with the number of L3 PDUs that were discarded, during the specified

```
        measurement interval, because the access class
        for this SNI has been exceeded."
 ::= { lucentCNMsmdsDisagreeEntry 5 }
```

**lucentCNMsmdsMCDUsExceededAtIngressCounts OBJECT-TYPE**

```
SYNTAX  Gauge
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The counter associated with the number of
        L3 PDUs that were discarded, during the specified
        measurement interval, because the MCDU was exceeded
        in the CPE to SMDS network direction."
 ::= { lucentCNMsmdsDisagreeEntry 6 }
```

**lucentCNMsmdsMCDUsExceededAtEgressCounts OBJECT-TYPE**

```
SYNTAX  Gauge
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The counter associated with the number of
        L3 PDUs that were discarded, during the specified
        measurement interval, because the MCDU was exceeded
        in the SMDS network to CPE direction."
 ::= { lucentCNMsmdsDisagreeEntry 7 }
```

**lucentCNMsmdsSAScreenViolations OBJECT-TYPE**

```
SYNTAX  Gauge
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The counter associated with the number of
        L3 PDUs that were discarded, during the specified
        measurement interval, because it violated the
        address screen based on source address screening
        for this SNI."
 ::= { lucentCNMsmdsDisagreeEntry 8 }
```

**lucentCNMsmdsDAScreenViolations OBJECT-TYPE**

```
SYNTAX  Gauge
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The counter associated with the number of
        L3 PDUs that were discarded, during the specified
        measurement interval, because it violated the
        address screen based on destination address screening
        for this SNI."
 ::= { lucentCNMsmdsDisagreeEntry 9 }
```

**lucentCNMsmdsUnassignedSAs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of  
    L3 PDUs that were discarded, during the specified  
    measurement interval, because the source address  
    was not assigned to this SNI."  
 ::= { lucentCNMsmdsDisagreeEntry 10 }

**lucentCNMsmdsDestinationSNIUnavailableCounts OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of  
    L3 PDUs that were discarded, during the specified  
    measurement interval, because the destination SNI  
    was not available."  
 ::= { lucentCNMsmdsDisagreeEntry 11 }

**SMDS Disagreements Log Table**

---

This table provides a log of the latest occurrences of the various SMDS disagreements encountered by all SNIs supported by this system.

**lucentCNMsmdsDisagreeLogTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmdsDisagreeLogEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "A list of entries identifying the latest  
    occurrence of SMDS disagreements encountered  
    by all SNIs managed by this system."  
 ::= { lucent-cnm-smds 8 }

**lucentCNMsmdsDisagreeLogEntry OBJECT-TYPE**

SYNTAX LucentCNMsmdsDisagreeLogEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "An entry identifying the latest occurrence of  
    a specific SMDS disagreement encountered by a  
    particular SNI."  
    INDEX{ lucentCNMsmdsDisagreeLogIndex,  
          lucentCNMsmdsDisagreeLogType }  
 ::= { lucentCNMsmdsDisagreeLogTable 1 }

```

LucentCNMsmdsDisagreeLogEntry ::=
    SEQUENCE {
        lucentCNMsmdsDisagreeLogIndex
            INTEGER,
        lucentCNMsmdsDisagreeLogType
            INTEGER,
        lucentCNMsmdsDisagreeLogSA
            SMDSAddress,
        lucentCNMsmdsDisagreeLogDA
            SMDSAddress,
        lucentCNMsmdsDisagreeLogTimeStamp
            TimeTicks,
        lucentCNMsmdsDisagreeLogLocalTime
            DisplayString
    }

```

**lucentCNMsmdsDisagreeLogIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each SNI. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMsmdsDisagreeLogEntry 1 }

```

**lucentCNMsmdsDisagreeLogType OBJECT-TYPE**

```

SYNTAX  INTEGER {
        sourceAddressScreenViolation(1),
        destinationAddressScreenViolation(2),
        invalidSourceAddressForSNI(3),
        destSNInotAvailable(4),
        accessClassExceeded(5),
        mcduExceededAtIngress(6),
        mcduExceededAtEgress(7)
    }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The type of SMDS disagreement. It could be one of:

    - Source Address Screening Violation,
    - Destination Address Screening Violation,
    - Source Address Not Assigned to a SNI,
    - Destination SNI Not Available,
    - Access Class Exceeded,
    - MCDUs Exceeded at Ingress,
    - MCDUs Exceeded at Egress

```

The SMDS Disagreements Measurement Table in this MIB describes these disagreement types."  
 ::= { lucentCNMsmdsDisagreeLogEntry 2 }

**lucentCNMsmdsDisagreeLogSA OBJECT-TYPE**

SYNTAX SMDSAddress  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The SMDS source address in the rejected L3 PDU."  
 ::= { lucentCNMsmdsDisagreeLogEntry 3 }

**lucentCNMsmdsDisagreeLogDA OBJECT-TYPE**

SYNTAX SMDSAddress  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The SMDS destination address in the rejected L3 PDU."  
 ::= { lucentCNMsmdsDisagreeLogEntry 4 }

**lucentCNMsmdsDisagreeLogTimeStamp OBJECT-TYPE**

SYNTAX Time Ticks  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The time stamp for the latest occurrence of the specified disagreement, as measured in seconds from 00:00:00 UTC (Coordinated Universal Time) January 1, 1970. Any fraction is rounded up."  
 ::= { lucentCNMsmdsDisagreeLogEntry 5 }

**lucentCNMsmdsDisagreeLogLocalTime OBJECT-TYPE**

SYNTAX Display String(SIZE(0..255))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The time stamp for the latest occurrence of the specified disagreement. It is given as a printable ASCII string showing the local time at the latest occurrence of this type of disagreement."  
 ::= { lucentCNMsmdsDisagreeLogEntry 6 }

**SMDS ICI Table**

---

This table provides SMDS ICI subscription parameters and configuration values maintained by all SMDS subscriber-network interfaces (SNIs) that are supported by this system.

**lucentCNMsmdsXaSubscrTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmdsXaSubscrEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "A list of entries containing SMDS ICI subscription parameters, and configuration values, for all the subscriber-network interfaces (SNIs) managed by this system."  
 ::= { lucent-cnm-smds 9 }

**lucentCNMsmdsXaSubscrEntry OBJECT-TYPE**

SYNTAX LucentCNMsmdsXaSubscrEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "An entry containing SMDS subscription parameters, and configuration values for a specific subscriber network interface (SNI)."  
INDEX { lucentCNMsmdsXaSubscrIndex }  
 ::= { lucentCNMsmdsXaSubscrTable 1 }

**LucentCNMsmdsXaSubscrEntry ::=**

SEQUENCE {  
    lucentCNMsmdsXaSubscrIndex  
        INTEGER,  
    lucentCNMsmdsXaSubscrPreselCarrier  
        INTEGER  
}

**lucentCNMsmdsXaSubscrIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "A unique value for each subscriber-network interface (SNI). The interface identified by a particular value of this index is the same interface as identified by the same value of an lucentCNMifConfigIndex object instance."  
 ::= { lucentCNMsmdsXaSubscrEntry 1 }

**lucentCNMsmdsXaSubscrPreselCarrier OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The preselected carrier for this SNI."  
 ::= { lucentCNMsmdsXaSubscrEntry 2 }

## SMDS CIC Table

This table provides SMDS ICI CIC Codes for all SMDS ICI trunks that are supported by this system.

### lucentCNMsmdsXaCICTable OBJECT-TYPE

SYNTAX SEQUENCE OF LucentCNMsmdsXaCICEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A list of entries containing SMDS ICI CIC parameters for all ICI trunks managed by this system."

::= { lucent-cnm-smds 10 }

### lucentCNMsmdsXaCICEntry OBJECT-TYPE

SYNTAX LucentCNMsmdsXaCICEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"An entry containing the SMDS ICI CIC parameters for a specific ICI trunk."

INDEX { lucentCNMsmdsXaCICIndex }

::= { lucentCNMsmdsXaCICTable 1 }

### LucentCNMsmdsXaCICEntry ::=

```
SEQUENCE {
    lucentCNMsmdsXaCICIndex
        INTEGER,
    lucentCNMsmdsXaCICCode
        INTEGER,
    lucentCNMsmdsXaCICType
        INTEGER
}
```

### lucentCNMsmdsXaCICIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A unique value for each ICI trunk. The interface identified by a particular value of this index is the same interface as identified by the same value of an lucentCNMifConfigIndex object instance."

::= { lucentCNMsmdsXaCICEntry 1 }

### lucentCNMsmdsXaCICCode OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION  
"The CIC Code assigned to this ICI trunk."  
 ::= { lucentCNMsmDsXaCICEntry 2 }

**lucentCNMsmDsXaCICType OBJECT-TYPE**

SYNTAX INTEGER {  
lec(1),  
ici(2)}  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The type of ICI trunk this trunk is provisioned  
as. It could be provisioned as LEC or ICI."  
 ::= { lucentCNMsmDsXaCICEntry 3 }

**SMDS Set Individual Address Screening Table**

This table provides the list of submitted requests to ADD or DELETE Individual SMDS Screening addresses from an SNI.

The entries in this table form the Set Individual Address Screening Table for a subscriber-network interface. These are the SMDS individual screening addresses that you have submitted to be ADDED to or DELETED from a subscriber-network interface. SMDS addresses that you have submitted to be ADDED or DELETED from an SNI are not updated in real time. These submitted SMDS addresses are queued in the Set Individual Address Screening table to be updated by the SNMP Agent administrator at an appropriate time.

The SNMP Agent administrator will run an update command that will take all the Individual Screening Addresses that have been submitted to this table and update the lucentCNMsmDsIndScrTable."

**lucentCNMsmDsIndScrSetTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmDsIndScrSetEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"A list of entries containing submitted requests to  
ADD or DELETE individual screening addresses from an  
SNI. These entries form the SET individual address  
screening table for the SNIs."  
 ::= { lucent-cnm-smDs 11 }

**lucentCNMsmDsIndScrSetEntry OBJECT-TYPE**

SYNTAX LucentCNMsmDsIndScrSetEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"An entry containing submit request information

```

        to ADD or DELETE an individual screening address for
        a particular subscriber-network interface (SNI)."
INDEX{ lucentCNMsmldsIndScrSetIndex,
       lucentCNMsmldsIndScrCountrySetIndex,
       lucentCNMsmldsIndScrAreaSetIndex,
       lucentCNMsmldsIndScrSubscriberSetIndex }
 ::= { lucentCNMsmldsIndScrSetTable 1 }

```

**LucentCNMsmldsIndScrSetEntry ::=**

```

SEQUENCE {
    lucentCNMsmldsIndScrSetIndex
        INTEGER,
    lucentCNMsmldsIndScrCountrySetIndex
        INTEGER,
    lucentCNMsmldsIndScrAreaSetIndex
        INTEGER,
    lucentCNMsmldsIndScrSubscriberSetIndex
        INTEGER,
    lucentCNMsmldsIndScrSetAddress
        SMDSAddress,
    lucentCNMsmldsIndScrSetAction
        INTEGER,
    lucentCNMsmldsIndScrSetStatus
        INTEGER,
    lucentCNMsmldsIndScrSetTimeStamp
        INTEGER,
    lucentCNMsmldsIndScrSetLocalTime
        DisplayString
}

```

**lucentCNMsmldsIndScrSetIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each SNI. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMsmldsIndScrSetEntry 1 }

```

**lucentCNMsmldsIndScrCountrySetIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
    "The Country Code portion of the SMDS Address
    given by the lucentCNMsmldsIndScrSetAddress object.

    This object is not accessible as it is used for

```

indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmdsIndScrSetEntry 2 }

**lucentCNMsmdsIndScrAreaSetIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"The Area Code (also known as the National Destination Code) portion of the SMDS Address given by the lucentCNMsmdsIndScrSetAddress object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmdsIndScrSetEntry 3 }

**lucentCNMsmdsIndScrSubscriberSetIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"The Subscriber Number portion of the SMDS Address given by the lucentCNMsmdsIndScrSetAddress object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."  
 ::= { lucentCNMsmdsIndScrSetEntry 4 }

**lucentCNMsmdsIndScrSetAddress OBJECT-TYPE**

SYNTAX SMDSAddress  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"An SMDS individual address that is to be ADDED to or DELETED from a particular SNI."  
 ::= { lucentCNMsmdsIndScrSetEntry 5 }

**lucentCNMsmdsIndScrSetAction OBJECT-TYPE**

SYNTAX INTEGER {  
                  add(1),  
                  delete(2)  
                  }  
ACCESS read-write  
STATUS mandatory

DESCRIPTION

"Action (ADD or DELETE) to be taken for this address. If you wish to ADD an individual screening address to an SNI, you need to set the Action to ADD.

If you wish to DELETE an individual screening address from an SNI, you need to set the Action to DELETE.

If you previously submitted an individual screening address that is to be ADDED to an SNI and you wish to cancel this request, you need to submit a DELETE request. The DELETE request will remove the ADD request from the Set Individual Screening Address table.

If you previously submitted an SMDS address that is to be DELETED from an SNI and you wish to cancel this request, you need to submit an ADD request. The ADD request will remove the DELETE request from the Set Individual Screening Address table."

::= { lucentCNMsmdsIndScrSetEntry 6 }

**lucentCNMsmdsIndScrSetStatus OBJECT-TYPE**

SYNTAX INTEGER {  
                  pending(1),  
                  failed(2)  
                  }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A value which indicates the status of the submitted request.

A pending (1) status indicates that the submitted request still needs to be processed by this SNMP agent system.

A failed (2) status indicates that the submitted request has been processed but for some reason the SMDS address that is to be added to or deleted from an SNI cannot be added or deleted. You need to either re-submit your request or contact the SNMP agent system administrator to find out the nature of the problem.

If a submitted request has been successfully processed, the submitted request is automatically removed from this table. You can verify the lucentCNMsmdsIndScrTable to make sure that the request has been processed."

::= { lucentCNMsmdsIndScrSetEntry 7 }

**lucentCNMsmdsIndScrSetTimeStamp OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The time stamp corresponding to when the request  
    was submitted, as measured in seconds from 00:00:00  
    UTC (Coordinated Universal Time) January 1, 1970."  
 ::= { lucentCNMsmdsIndScrSetEntry 8 }

**lucentCNMsmdsIndScrSetLocalTime OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The time stamp corresponding to when the request  
    was submitted. It is given as a printable ASCII  
    string showing the local time."  
 ::= { lucentCNMsmdsIndScrSetEntry 9 }

### **SMDS Set Group Address Screening Table**

---

This table provides the list of submitted requests to ADD or DELETE Group SMDS Screening addresses from an SNI.

The entries in this table form the Set Group Address Screening Table for a subscriber-network interface. These are the SMDS group screening addresses that you have submitted to be ADDED to or DELETED from a subscriber-network interface. SMDS addresses that you have submitted to be ADDED or DELETED from an SNI are not updated in real time. These submitted SMDS addresses are queued in the Set Group Address Screening table to be updated by the SNMP Agent administrator at an appropriate time.

The SNMP Agent administrator will run an update command that will take all the Group Screening Addresses that have been submitted to this table and update the lucentCNMsmdsGrpScrTable.

**lucentCNMsmdsGrpScrSetTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsmdsGrpScrSetEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "A list of entries containing submitted requests to  
    ADD or DELETE group screening addresses from an  
    SNI. These entries form the SET group address  
    screening table for the SNIs."  
 ::= { lucent-cnm-smds 12 }

**lucentCNMsmdsGrpScrSetEntry OBJECT-TYPE**

```
SYNTAX   LucentCNMsmdsGrpScrSetEntry
ACCESS   not-accessible
STATUS   mandatory
DESCRIPTION
    "An entry containing submit request information
    to ADD or DELETE a group screening address for
    a particular subscriber-network interface (SNI)."
```

INDEX { lucentCNMsmdsGrpScrSetIndex,  
          lucentCNMsmdsGrpScrCountrySetIndex,  
          lucentCNMsmdsGrpScrAreaSetIndex,  
          lucentCNMsmdsGrpScrSubscriberSetIndex }

```
::= { lucentCNMsmdsGrpScrSetTable 1 }
```

**lucentCNMsmdsGrpScrSetEntry ::=**

```
SEQUENCE {
    lucentCNMsmdsGrpScrSetIndex
        INTEGER,
    lucentCNMsmdsGrpScrCountrySetIndex
        INTEGER,
    lucentCNMsmdsGrpScrAreaSetIndex
        INTEGER,
    lucentCNMsmdsGrpScrSubscriberSetIndex
        INTEGER,
    lucentCNMsmdsGrpScrSetAddress
        SMDSAddress,
    lucentCNMsmdsGrpScrSetAction
        INTEGER,
    lucentCNMsmdsGrpScrSetStatus
        INTEGER,
    lucentCNMsmdsGrpScrSetTimeStamp
        INTEGER,
    lucentCNMsmdsGrpScrSetLocalTime
        DisplayString
}
```

**lucentCNMsmdsGrpScrSetIndex OBJECT-TYPE**

```
SYNTAX   INTEGER
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "A unique value for each SNI. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
```

```
::= { lucentCNMsmdsGrpScrSetEntry 1 }
```

**lucentCNMsmdsGrpScrCountrySetIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The Country Code portion of the SMDS Address given by the lucentCNMsmdsGrpScrSetAddress object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsGrpScrSetEntry 2 }

**lucentCNMsmdsGrpScrAreaSetIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The Area Code (also known as the National Destination Code) portion of the SMDS Address given by the lucentCNMsmdsGrpScrSetAddress object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsGrpScrSetEntry 3 }

**lucentCNMsmdsGrpScrSubscriberSetIndex OBJECT-TYPE**

SYNTAX INTEGER

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The Subscriber Number portion of the SMDS Address given by the lucentCNMsmdsGrpScrSetAddress object.

This object is not accessible as it is used for indexing purposes only. The attCNMsmdsAddrIndexDescr object provides more information on the proper usage and behaviour of such objects."

::= { lucentCNMsmdsGrpScrSetEntry 4 }

**lucentCNMsmdsGrpScrSetAddress OBJECT-TYPE**

SYNTAX SMDSAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

"An SMDS group address that is to be ADDED to or DELETED from a particular SNI."

```
::= { lucentCNMsmdsGrpScrSetEntry 5 }
```

**lucentCNMsmdsGrpScrSetAction OBJECT-TYPE**

```
SYNTAX  INTEGER {  
            add(1),  
            delete(2)  
        }
```

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Action (ADD or DELETE) to be taken for this address. If you wish to ADD a group screening address to an SNI, you need to set the Action to ADD.

If you wish to DELETE a group screening address from an SNI, you need to set the Action to DELETE.

If you previously submitted a group screening address that is to be ADDED to an SNI and you wish to cancel this request, you need to submit a DELETE request. The DELETE request will remove the ADD request from the Set Group Screening Address table.

If you previously submitted an SMDS address that is to be DELETED from an SNI and you wish to cancel this request, you need to submit an ADD request. The ADD request will remove the DELETE request from the Set Group Screening Address table."

```
::= { lucentCNMsmdsGrpScrSetEntry 6 }
```

**lucentCNMsmdsGrpScrSetStatus OBJECT-TYPE**

```
SYNTAX  INTEGER {  
            pending(1),  
            failed(2)  
        }
```

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A value which indicates the status of the submitted request.

A pending (1) status indicates that the submitted request still needs to be processed by this SNMP agent system.

A failed (2) status indicates that the submitted request has been processed but for some reason the SMDS address that is to be added to or deleted from an SNI cannot be added or deleted. You need to either re-submit

your request or contact the SNMP agent system administrator to find out the nature of the problem.

If a submitted request has been successfully processed, the submitted request is automatically removed from this table. You can verify the `lucentCNMsmndsGrpScrTable` to make sure that the request has been processed."

```
::= { lucentCNMsmndsGrpScrSetEntry 7 }
```

**lucentCNMsmndsGrpScrSetTimeStamp OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp corresponding to when the request was submitted, as measured in seconds from 00:00:00 UTC (Coordinated Universal Time) January 1, 1970."

```
::= { lucentCNMsmndsGrpScrSetEntry 8 }
```

**lucentCNMsmndsGrpScrSetLocalTime OBJECT-TYPE**

SYNTAX Display String(SIZE(0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp corresponding to when the request was submitted. It is given as a printable ASCII string showing the local time."

```
::= { lucentCNMsmndsGrpScrSetEntry 9 }
```

**END**

## **CNM SIP MIB**

---

```
LUCENT-CNM-SIP-MIB DEFINITIONS ::= BEGIN
```

### IMPORTS

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

### Object Identifiers

```
lucent                OBJECT IDENTIFIER ::= { enterprises 1751 }
products              OBJECT IDENTIFIER ::= { lucent 1 }
mibs                   OBJECT IDENTIFIER ::= { lucent 2 }
lucent-cnmAgent       OBJECT IDENTIFIER ::= { products 5 }
lucent-cnm             OBJECT IDENTIFIER ::= { mibs 5 }
lucent-cnm-sip        OBJECT IDENTIFIER ::= { lucent-cnm 5 }
```

All representations of SMDS addresses in this MIB module use, as a textual convention (i.e., this convention does not affect their encoding), the data type:

```
SMDSAddress ::= OCTET STRING(SIZE(8))
```

SMDSAddress is the 60-bit SMDS address, preceded by 4 bits with the following values:

```
"1100" when representing an individual address
"1110" when representing a group address
```

### **SIP Group of Objects (lucent-cnm-sip)**

---

The SIP group of objects (lucent-cnm-sip) include:

```
the SIP Configuration table
the SIP Measurements table
the SIP Level 3 Error Log table
```

### **SIP Configuration Table**

---

This table provides configuration information the SIP interfaces supported by this system.

#### **lucentCNMsipConfigTable OBJECT-TYPE**

```
SYNTAX SEQUENCE OF LucentCNMsipConfigEntry
ACCESS not-accessible
STATUS mandatory
```

DESCRIPTION  
    "A list of entries containing configuration information for all SIP interfaces managed by this system."  
 ::= { lucent-cnm-sip 1 }

**lucentCNMsipConfigEntry OBJECT-TYPE**

SYNTAX LucentCNMsipConfigEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "An entry containing configuration information for a particular SIP interface."  
INDEX { lucentCNMsipConfigIndex }  
 ::= { lucentCNMsipConfigTable 1 }

**LucentCNMsipConfigEntry ::=**

SEQUENCE {  
    lucentCNMsipConfigIndex  
        INTEGER,  
    lucentCNMsipMeasMaxIntervals  
        INTEGER,  
    lucentCNMsipMeasIntervalLen  
        INTEGER  
}

**lucentCNMsipConfigIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "A unique value for each SIP interface. The interface identified by a particular value of this index is the same interface as identified by the same value of an lucentCNMifConfigIndex object instance."  
 ::= { lucentCNMsipConfigEntry 1 }

**lucentCNMsipMeasMaxIntervals OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "This variable identifies the maximum number of measurement intervals supported for the measurements maintained by this SIP interface in the lucentCNMsipMeasTable."  
 ::= { lucentCNMsipConfigEntry 2 }

**lucentCNMsipMeasIntervalLen OBJECT-TYPE**

SYNTAX INTEGER

```

ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "This variable identifies the number of seconds
    that make up one complete measurement interval for
    the measurements maintained by this SIP interface
    in the lucentCNMsipMeasTable."
 ::= { lucentCNMsipConfigEntry 3

```

## **SIP Measurements Table**

This table provides counts on the various traffic measurements maintained by a SIP interface during the specified measurement interval.

### **lucentCNMsipMeasTable OBJECT-TYPE**

```

SYNTAX    SEQUENCE OF LucentCNMsipMeasEntry
ACCESS    not-accessible
STATUS    mandatory
DESCRIPTION
    "A list of entries containing traffic measurements,
    maintained during the specified measurement interval,
    for all SIP interfaces managed by this system."
 ::= { lucent-cnm-sip 2 }

```

### **lucentCNMsipMeasEntry OBJECT-TYPE**

```

SYNTAX    LucentCNMsipMeasEntry
ACCESS    not-accessible
STATUS    mandatory
DESCRIPTION
    "An entry containing traffic measurements,
    maintained during the specified measurement interval,
    for a particular SIP interface."
INDEX     { lucentCNMsipMeasIndex,
            lucentCNMsipMeasInterval }
 ::= { lucentCNMsipMeasTable 1 }

```

### **LucentCNMsipMeasEntry ::=**

```

SEQUENCE {
    lucentCNMsipMeasIndex
        INTEGER,
    lucentCNMsipMeasInterval
        INTEGER,
    lucentCNMsipMeasTimeStamp
        INTEGER,
    lucentCNMsipMeasLocalTime
        DisplayString,
    lucentCNMsipReceivedL3PDUs
        Gauge,

```

```
    lucentCNMsipSentL3PDUs
        Gauge,
    lucentCNMsipReceivedGroupL3PDUs
        Gauge,
    lucentCNMsipSentGroupL3PDUs
        Gauge,
    lucentCNMsipReceivedL2PDUs
        Gauge,
    lucentCNMsipSentL2PDUs
        Gauge
}
```

**lucentCNMsipMeasIndex OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each SIP interface. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMsipMeasEntry 1 }
```

**lucentCNMsipMeasInterval OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable identifies the measurement interval
    number for which measurement is provided. It is a
    number between 1 and XX, where 1 identifies the most
    recently completed measurement interval and XX is
    the least recently completed measurement interval.

    The value of XX is specified by the
    lucentCNMsipMeasMaxIntervals object given in the
    lucentCNMsipConfigTable.

    The maximum length of each measurement interval is
    specified by the lucentCNMsipMeasIntervalLen object
    given
    in the lucentCNMsipConfigTable."
 ::= { lucentCNMsipMeasEntry 2 }
```

**lucentCNMsipMeasTimeStamp OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The time stamp corresponding to the end of the
```

specified measurement interval, as measured in seconds from 00:00:00 UTC (Coordinated Universal Time) January 1, 1970. Any fraction is rounded up."  
 ::= { lucentCNMsipMeasEntry 3 }

**lucentCNMsipMeasLocalTime OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The time stamp corresponding to the end of the specified measurement interval. Any fraction is rounded up. It is given as a printable ASCII string showing the local time at the end of the interval."  
 ::= { lucentCNMsipMeasEntry 4 }

**lucentCNMsipReceivedL3PDUs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of individual-addressed and group-addressed L3 PDUs received by a SIP interface during the specified measurement interval."  
 ::= { lucentCNMsipMeasEntry 5 }

**lucentCNMsipSentL3PDUs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of individual-addressed and group-addressed L3 PDUs sent by a SIP interface during the specified measurement interval."  
 ::= { lucentCNMsipMeasEntry 6 }

**lucentCNMsipReceivedGroupL3PDUs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of group-addressed L3 PDUs received by a SIP interface during the specified measurement interval."  
 ::= { lucentCNMsipMeasEntry 7 }

**lucentCNMsipSentGroupL3PDUs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of  
    group-addressed L3 PDUs sent by a SIP interface  
    during the specified measurement interval."  
 ::= { lucentCNMsipMeasEntry 8 }

### **lucentCNMsipReceivedL2PDUs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of  
    L2 PDUs received by a SIP interface during the  
    specified measurement interval."  
 ::= { lucentCNMsipMeasEntry 9 }

### **lucentCNMsipSentL2PDUs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of  
    L2 PDUs sent by a SIP interface during the  
    specified measurement interval."  
 ::= { lucentCNMsipMeasEntry 10 }

## **SIP Level 3 Error Log Table**

---

This table provides a log of the latest occurrences of the various protocol errors encountered at the Level 3 layer of all SIP interfaces supported by this system.

### **lucentCNMsipL3ErrorLogTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMsipL3ErrorLogEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "A list of entries identifying the latest occurrence  
    of protocol errors encountered at the Level 3 layer  
    of all SIP interfaces managed by this system."  
 ::= { lucent-cnm-sip 3 }

### **lucentCNMsipL3ErrorLogEntry OBJECT-TYPE**

SYNTAX LucentCNMsipL3ErrorLogEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "An entry identifying the latest occurrence

```

of a specific protocol error encountered
at the Level 3 layer of a particular SIP interface."
INDEX{ lucentCNMsipL3ErrorLogIndex,
       lucentCNMsipL3ErrorLogType }
 ::= { lucentCNMsipL3ErrorLogTable 1 }

```

```

LucentCNMsipL3ErrorLogEntry ::=
SEQUENCE {
    lucentCNMsipL3ErrorLogIndex
        INTEGER,
    lucentCNMsipL3ErrorLogType
        INTEGER,
    lucentCNMsipL3ErrorLogSA
        SMDSAddress,
    lucentCNMsipL3ErrorLogDA
        SMDSAddress,
    lucentCNMsipL3ErrorLogTimeStamp
        INTEGER,
    lucentCNMsipL3ErrorLogLocalTime
        DisplayString
}

```

**lucentCNMsipL3ErrorLogIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each SIP interface. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMsipL3ErrorLogEntry 1 }

```

**lucentCNMsipL3ErrorLogType OBJECT-TYPE**

```

SYNTAX  INTEGER {
    daFieldFormatError(1),
    saFieldFormatError(2),
    baSizeFieldValueInvalid(3),
    heLengthInvalid(4),
    heElementLengthInvalid(5),
    heVersionElementInvalid(6),
    heCarrierSelectionElementInvalid(7),
    hePADInvalid(8),
    beTagMismatch(9),
    baSizeFieldNotEqualLengthField(10),
    incorrectLength(11),
    mriTimeout(12)
}
ACCESS  read-only
STATUS  mandatory

```

DESCRIPTION

"The type of error. It could be one of:

- Destination Address Field Format Error,
- Source Address Field Format Error,
- Invalid BAsize Field Value,
- Invalid Header Extension Length,
- Invalid Header Extension: Element Length,
- Invalid Header Extension: Carrier Selection Element Format,
- Invalid Header Extension: Carrier Selection Element Length,
- Invalid Header Extension: Version Element Position, Length or Value,
- Bntag Mismatch,
- BAsize Field Not Equal To Length Field,
- Incorrect Length,
- MSS Receive Interval Timeout."

::= { lucentCNMsipL3ErrorLogEntry 2 }

**lucentCNMsipL3ErrorLogSA OBJECT-TYPE**

SYNTAX SMDSAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The SMDS source address in the rejected L3 PDU."

::= { lucentCNMsipL3ErrorLogEntry 3 }

**lucentCNMsipL3ErrorLogDA OBJECT-TYPE**

SYNTAX SMDSAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The SMDS destination address in the rejected L3 PDU."

::= { lucentCNMsipL3ErrorLogEntry 4 }

**lucentCNMsipL3ErrorLogTimeStamp OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp for the latest occurrence of the specified error, as measured in seconds from 00:00:00 UTC (Coordinated Universal Time) January 1, 1970. Any fraction is rounded up."

::= { lucentCNMsipL3ErrorLogEntry 5 }

**lucentCNMsipL3ErrorLogLocalTime OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp for the latest occurrence of the specified error. It is given as a printable ASCII string showing the local time at the latest occurrence of this type of error."

::= { lucentCNMsipL3ErrorLogEntry 6 }

**END**

## CNM DS1 MIB

---

```
LUCENT-CNM-DS1-MIB DEFINITIONS ::= BEGIN

IMPORTS

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

Object Identifiers

lucent          OBJECT IDENTIFIER ::= { enterprises 1751 }
products        OBJECT IDENTIFIER ::= { lucent 1 }
mibs            OBJECT IDENTIFIER ::= { lucent 2 }
lucent-cnmAgent OBJECT IDENTIFIER ::= { products 5 }
lucent-cnm      OBJECT IDENTIFIER ::= { mibs 5 }
lucent-cnm-ds1 OBJECT IDENTIFIER ::= { lucent-cnm 3 }
```

### DS1 Group of Objects (lucent-cnm-ds1)

---

The DS1 group of objects (lucent-cnm-ds1) include:

```
the DS1 Configuration table
the DS1 Status table
the DS1 Error Counts table
```

### DS1 Configuration Table

---

This table provides configuration information on the DS1 interfaces supported by this system.

```
lucentCNMds1ConfigTable OBJECT-TYPE
SYNTAX SEQUENCE OF LucentCNMds1ConfigEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A list of entries containing configuration
    information for all DS1 interfaces managed by
    this system."
 ::= { lucent-cnm-ds1 1 }
```

```
lucentCNMds1ConfigEntry OBJECT-TYPE
SYNTAX LucentCNMds1ConfigEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
```

```

        "An entry containing configuration information
        for a particular DS1 interface."
        INDEX{ lucentCNMds1ConfigIndex }
 ::= { lucentCNMds1ConfigTable 1 }

```

**LucentCNMds1ConfigEntry ::=**

```

SEQUENCE {
    lucentCNMds1ConfigIndex
INTEGER,
    lucentCNMds1LineType
INTEGER,
    lucentCNMds1ZeroCoding
INTEGER,
    lucentCNMds1ErrorsMaxIntervals
INTEGER,
    lucentCNMds1ErrorsIntervalLen
INTEGER
}

```

**lucentCNMds1ConfigIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each DS1 interface. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMds1ConfigEntry 1 }

```

**lucentCNMds1LineType OBJECT-TYPE**

```

SYNTAX  INTEGER{
    other(1),
    ds1ESF(2),
    ds1D4(3),
    ds1ANSI-ESF(4),
    ds1E1(5),
    ds1E1-CRC(6)
}
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable indicates the variety of DS1 frame
    format supported by this interface.

    For T1 based SMDS, this variable will have the value
    ds1ANSI-ESF, denoting that this interface
    supports the DS1 Extended Super Frame format
    as specified by ANSI.

```

For E1 based SMDS, this variable will have the value ds1E1-CRC, denoting that this interface supports CCITT Recommendation G.704 (Table 4b)."

```
::= { lucentCNMds1ConfigEntry 2 }
```

**lucentCNMds1ZeroCoding OBJECT-TYPE**

```
SYNTAX INTEGER{
    ds1JammedBit(1),
    ds1B8ZS(2),
    ds1InvertedHDLC(3),
    ds1HDB3(4),
    ds1ZBTSI(5),
    other(6)
}
```

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable describes the variety of zero code suppression/substitution used on the DS1 interface.

For T1 based SMDS, this variable will have the value ds1B8ZS, which refers to the use of a specified pattern of normal bits and bipolar violations which are used to replace a sequence of eight zero bits.

For E1 based SMDS, this variable will have the value ds1HDB3."

```
::= { lucentCNMds1ConfigEntry 3 }
```

**lucentCNMds1ErrorsMaxIntervals OBJECT-TYPE**

```
SYNTAX INTEGER
```

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable identifies the maximum number of measurement intervals supported for the error counts maintained by this DS1 interface in the lucentCNMds1ErrorsTable."

```
::= { lucentCNMds1ConfigEntry 4 }
```

**lucentCNMds1ErrorsIntervalLen OBJECT-TYPE**

```
SYNTAX INTEGER
```

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable identifies the number of seconds that make up one complete measurement interval for the error counts maintained by this DS1 interface

```

        in the lucentCNMds1ErrorsTable."
 ::= { lucentCNMds1ConfigEntry 5 }

```

## **DS1 Status Table**

This table provides status information on the DS1 interfaces supported by this system.

### **lucentCNMds1StatusTable OBJECT-TYPE**

```

SYNTAX SEQUENCE OF LucentCNMds1StatusEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A list of entries containing status information
    for all DS1 interfaces managed by this system."
 ::= { lucent-cnm-ds1 2 }

```

### **lucentCNMds1StatusEntry OBJECT-TYPE**

```

SYNTAX LucentCNMds1StatusEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "An entry containing status information for a
    particular DS1 interface."
INDEX { lucentCNMds1StatusIndex }
 ::= { lucentCNMds1StatusTable 1 }

```

### **LucentCNMds1StatusEntry ::=**

```

SEQUENCE {
    lucentCNMds1StatusIndex
        INTEGER,
    lucentCNMds1LineStatus
        INTEGER
}

```

### **lucentCNMds1StatusIndex OBJECT-TYPE**

```

SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "A unique value for each DS1 interface. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMds1StatusEntry 1 }

```

**lucentCNMds1LineStatus OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "This variable indicates the most Line Status of this interface. This object is a bit map represented as a sum, therefore, it can represent multiple failures (alarms) simultaneously. The various bit positions are:  
  
    1No Alarm Present  
    2Yellow Alarm  
    4Near End Alarm-Indication-Signal (AIS)  
    8Near End Loss-Of-Frame (LOF)  
    16Near End Loss-Of-Signal (LOS)  
  
    For example, for an interface that has LOF and LOS outstanding, this object will have a value of 24."  
 ::= { lucentCNMds1StatusEntry 2 }

**DS1 Error Counts Table**

---

This table provides counts on the various protocol errors encountered by a DS1 interface during the specified measurement interval.

**lucentCNMds1ErrorsTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMds1ErrorsEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "A list of entries containing protocol error counts, maintained during the specified measurement interval, for all DS1 interfaces managed by this system."  
 ::= { lucent-cnm-ds1 3 }

**lucentCNMds1ErrorsEntry OBJECT-TYPE**

SYNTAX LucentCNMds1ErrorsEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "An entry containing protocol error counts, maintained during the specified measurement interval, for a particular DS1 interface."  
    INDEX{ lucentCNMds1ErrorsIndex,  
          lucentCNMds1ErrorsInterval }  
 ::= { lucentCNMds1ErrorsTable 1 }

```
LucentCNMds1ErrorsEntry ::=
    SEQUENCE {
        lucentCNMds1ErrorsIndex
            INTEGER,
        lucentCNMds1ErrorsInterval
            INTEGER,
        lucentCNMds1ErrorsTimeStamp
            INTEGER,
        lucentCNMds1ErrorsLocalTime
            DisplayString,
        lucentCNMds1LCVs
            Gauge,
        lucentCNMds1LEsSs
            Gauge,
        lucentCNMds1LSEsSs
            Gauge,
        lucentCNMds1CVs
            Gauge,
        lucentCNMds1ESs
            Gauge,
        lucentCNMds1SEsSs
            Gauge,
        lucentCNMds1SEFSs
            Gauge,
        lucentCNMds1UASs
            Gauge,
        lucentCNMds1PctCU
            Gauge,
        lucentCNMds1PctPkCU
            Gauge,
        lucentCNMds1ASs
            Gauge,
        lucentCNMds1PctEFsSs
            Gauge,
        lucentCNMds1FSSs
            Gauge,
        lucentCNMds1FarLEsSs
            Gauge,
        lucentCNMds1FarCVs
            Gauge,
        lucentCNMds1FarESs
            Gauge,
        lucentCNMds1FarSEsSs
            Gauge,
        lucentCNMds1FarSEFSs
            Gauge,
        lucentCNMds1FarPctEFsSs
            Gauge,
        lucentCNMds1FarFSSs
            Gauge,
```

```

        lucentCNMds1B6Ss
            Gauge
    }

```

**lucentCNMds1ErrorsIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each DS1 interface. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMds1ErrorsEntry 1 }

```

**lucentCNMds1ErrorsInterval OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable identifies the measurement interval
    number for which the measurement is provided. It is
    a number between 1 and XX, where 1 identifies the
    most recently completed measurement interval and XX
    is the least recently completed measurement interval.

    The value of XX is specified by the
    lucentCNMds1ErrorsMaxIntervals object given in the
    lucentCNMds1ConfigTable.

    The maximum length of each measurement interval is
    specified by the lucentCNMds1ErrorsIntervalLen object
    given in the lucentCNMds1ConfigTable."
 ::= { lucentCNMds1ErrorsEntry 2 }

```

**lucentCNMds1ErrorsTimeStamp OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The time stamp corresponding to the end of the
    specified measurement interval, as measured in
    seconds from 00:00:00 UTC (Coordinated Universal
    Time) January 1, 1970. Any fraction is rounded up."
 ::= { lucentCNMds1ErrorsEntry 3 }

```

**lucentCNMds1ErrorsLocalTime OBJECT-TYPE**

```

SYNTAX  DisplayString(SIZE(0..255))
ACCESS  read-only
STATUS  mandatory

```

DESCRIPTION

"The time stamp corresponding to the end of the specified measurement interval. Any fraction is rounded up. It is given as a printable ASCII string showing the local time at the end of the interval."

::= { lucentCNMds1ErrorsEntry 4 }

**lucentCNMds1LCVs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Line Code Violations encountered by a DS1 interface during the specified measurement interval.

A Line Code Violation is the occurrence of a received Bipolar Violation that is not part of a zero-substitution code.

The Bipolar Violation Rate is generally viewed as approximating the Bit Error Rate."

::= { lucentCNMds1ErrorsEntry 5 }

**lucentCNMds1LESS OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Line Errored Seconds (near end) encountered by a DS1 interface during the specified measurement interval.

A Line Errored Second is any second with at least one Line Code Violation."

::= { lucentCNMds1ErrorsEntry 6 }

**lucentCNMds1LSESS OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Line Severely Errored Seconds encountered by a DS1 interface during the specified measurement interval.

A Line Severely Errored Second is any second with 16 or more Line Code Violations monitored at the DS1 rate."

::= { lucentCNMds1ErrorsEntry 7 }

**lucentCNMds1CVs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Code Violations (near end) encountered by a DS1 interface during the specified measurement interval.

A Code Violation is the occurrence of a received CRC-6 code that is not identical to the corresponding locally-calculated code."

::= { lucentCNMds1ErrorsEntry 8 }

**lucentCNMds1ESSs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Errored Seconds (near end) encountered by a DS1 interface during the specified measurement interval.

An Errored Second is any second with at least one Code Violation."

::= { lucentCNMds1ErrorsEntry 9 }

**lucentCNMds1SESSs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Severely Errored Seconds (near end) encountered by a DS1 interface during the specified measurement interval.

A Severely Errored Second is any second with 15 or more Code Violations monitored at the DS1 rate."

::= { lucentCNMds1ErrorsEntry 10 }

**lucentCNMds1SEFSSs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Severely Errored Framing Seconds (near end) encountered by a DS1 interface during the specified measurement interval.

A Severely Errored Framing Second is any second during which one or more Severely Errored Framing (SEF) events occurred."  
 ::= { lucentCNMds1ErrorsEntry 11 }

**lucentCNMds1UAss OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of Unavailable Seconds encountered by a DS1 interface during the specified measurement interval.  
  
An Unavailable Second is any second during which the DS1 interface was unavailable to offer service."  
 ::= { lucentCNMds1ErrorsEntry 12 }

**lucentCNMds1PctCU OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The percentage of channel io utilization by a DS1 interface during the specified measurement interval."  
 ::= { lucentCNMds1ErrorsEntry 13 }

**lucentCNMds1PctPkCU OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The 5 minute peak percentage of channel io utilization by a DS1 interface during the specified measurement interval."  
 ::= { lucentCNMds1ErrorsEntry 14 }

**lucentCNMds1Ass OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"Number of seconds AIS was active (near end) for a DS1 interface during a specified measurement interval."  
 ::= { lucentCNMds1ErrorsEntry 15 }

**lucentCNMds1PctEFSS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory

DESCRIPTION

"Percentage of error free seconds (near end) for a DS1 interface during a specified measurement interval."  
 ::= { lucentCNMds1ErrorsEntry 16 }

**lucentCNMds1FSSs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of seconds with a frame slop (near end) for a DS1 interface during a specified measurement interval."  
 ::= { lucentCNMds1ErrorsEntry 17 }

**lucentCNMds1FarLESS OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Line Errored Seconds (far end) encountered by a DS1 interface during the specified measurement interval.  
  
A Line Errored Second is any second with at least one Line Code Violation."  
 ::= { lucentCNMds1ErrorsEntry 18 }

**lucentCNMds1FarCVs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Code Violations (far end) encountered by a DS1 interface during the specified measurement interval.  
  
A Code Violation is the occurrence of a received CRC-6 code that is not identical to the corresponding locally-calculated code."  
 ::= { lucentCNMds1ErrorsEntry 19 }

**lucentCNMds1FarESS OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of Errored Seconds (far end) encountered by a DS1 interface during the specified measurement interval."

An Errored Second is any second with at least one Code Violation."  
 ::= { lucentCNMds1ErrorsEntry 20 }

**lucentCNMds1FarSESS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of Severely Errored Seconds (far end) encountered by a DS1 interface during the specified measurement interval.  
  
A Severely Errored Second is any second with 15 or more Code Violations monitored at the DS1 rate."  
 ::= { lucentCNMds1ErrorsEntry 21 }

**lucentCNMds1FarSEFSS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of Severely Errored Framing Seconds (far end) encountered by a DS1 interface during the specified measurement interval.  
  
A Severely Errored Framing Second is any second during which one or more Severely Errored Framing (SEF) events occurred."  
 ::= { lucentCNMds1ErrorsEntry 22 }

**lucentCNMds1FarPctEFSS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"Percentage of error free seconds (far end) for a DS1 interface during a specified measurement interval."  
 ::= { lucentCNMds1ErrorsEntry 23 }

**lucentCNMds1FarFSSs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"Number of seconds (far end) with a frame slip for a DS1 interface during a specified measurement interval."  
 ::= { lucentCNMds1ErrorsEntry 24 }

**lucentCNMds1B6Ss OBJECT-TYPE**

```
SYNTAX Gauge
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Number of seconds with with more than two CRC errors
    for a DS1 interface during a specified measurement
    interval."
 ::= { lucentCNMds1ErrorsEntry 25 }
```

**DS1 Total Error Counts Table**

---

This table provides total counts on the various protocol errors encountered by a DS1 interface accumulated over the previous 24 hours.

**lucentCNMds1TotalErrorsTable OBJECT-TYPE**

```
SYNTAX SEQUENCE OF LucentCNMds1TotalErrorsEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A list of entries containing total counts on the
    various protocol errors encountered over the previous
    24 hours for all DS1 interfaces managed by this
    system."
 ::= { lucent-cnm-ds1 4 }
```

**lucentCNMds1TotalErrorsEntry OBJECT-TYPE**

```
SYNTAX LucentCNMds1TotalErrorsEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "An entry containing total counts on the various
    protocol errors encountered over the previous
    24 hours for a particular DS1 interface."
    INDEX{ lucentCNMds1TotalErrorsIndex }
 ::= { lucentCNMds1TotalErrorsTable 1 }
```

**LucentCNMds1TotalErrorsEntry ::=**

```
SEQUENCE {
    lucentCNMds1TotalErrorsIndex
        INTEGER,
    lucentCNMds1TotalLCVs
        Gauge,
    lucentCNMds1TotalLESS
        Gauge,
    lucentCNMds1TotalLSESS
        Gauge,
    lucentCNMds1TotalCVs
        Gauge,
```

```

    lucentCNMds1TotalESs
        Gauge,
    lucentCNMds1TotalSESSs
        Gauge,
    lucentCNMds1TotalSEFSs
        Gauge,
    lucentCNMds1TotalUASs
        Gauge,
    lucentCNMds1TotalASs
        Gauge,
    lucentCNMds1TotalFSSs
        Gauge,
    lucentCNMds1TotalFarLESSs
        Gauge,
    lucentCNMds1TotalFarCVs
        Gauge,
    lucentCNMds1TotalFarESs
        Gauge,
    lucentCNMds1TotalFarSESSs
        Gauge,
    lucentCNMds1TotalFarSEFSs
        Gauge,
    lucentCNMds1TotalFarFSSs
        Gauge,
    lucentCNMds1TotalB6Ss
        Gauge
}

```

**lucentCNMds1TotalErrorsIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION

```

"A unique value for each DS1 interface. The interface identified by a particular value of this index is the same interface as identified by the same value of an lucentCNMifConfigIndex object instance."

```
 ::= { lucentCNMds1TotalErrorsEntry 1 }
```

**lucentCNMds1TotalLCVs OBJECT-TYPE**

```

SYNTAX  Gauge
ACCESS  read-only
STATUS  mandatory
DESCRIPTION

```

"The counter associated with the total number of Line Code Violations encountered by a DS1 interface during the previous 24 hours.

A Line Code Violation is the occurrence of a received Bipolar Violation that is not part of a

zero-substitution code.

The Bipolar Violation Rate is generally viewed as approximating the Bit Error Rate."

::= { lucentCNMds1TotalErrorsEntry 2 }

**lucentCNMds1TotalLESS OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the total number of Line Errored Seconds (near end) encountered by a DS1 interface during the previous 24 hours.

A Line Errored Second is any second with at least one Line Code Violation."

::= { lucentCNMds1TotalErrorsEntry 3 }

**lucentCNMds1TotalLSESS OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the total number of Line Severely Errored Seconds encountered by a DS1 interface during the previous 24 hours.

A Line Severely Errored Second is any second with 16 or more Line Code Violations monitored at the DS1 rate."

::= { lucentCNMds1TotalErrorsEntry 4 }

**lucentCNMds1TotalCVs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the total number of Code Violations (near end) encountered by a DS1 interface during the previous 24 hours.

A Code Violation is the occurrence of a received CRC-6 code that is not identical to the corresponding locally-calculated code."

::= { lucentCNMds1TotalErrorsEntry 5 }

**lucentCNMds1TotalESS OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory  
DESCRIPTION  
"The counter associated with the total number of Errored Seconds (near end) encountered by a DS1 interface during the previous 24 hours.  
  
An Errored Second is any second with at least one Code Violation."  
 ::= { lucentCNMds1TotalErrorsEntry 6 }

**lucentCNMds1TotalSESS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the total number of Severely Errored Seconds (near end) encountered by a DS1 interface during the previous 24 hours.  
  
A Severely Errored Second is any second with 15 or more Code Violations monitored at the DS1 rate."  
 ::= { lucentCNMds1TotalErrorsEntry 7 }

**lucentCNMds1TotalSEFSS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the total number of Severely Errored Framing Seconds (near end) encountered by a DS1 interface during the previous 24 hours.  
  
A Severely Errored Framing Second is any second during which one or more Severely Errored Framing (SEF) events occurred."  
 ::= { lucentCNMds1TotalErrorsEntry 8 }

**lucentCNMds1TotalUASS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the total number of Unavailable Seconds encountered by a DS1 interface during the previous 24 hours.  
  
An Unavailable Second is any second during which the DS1 interface was unavailable to offer service."

```
::= { lucentCNMds1TotalErrorsEntry 9 }
```

**lucentCNMds1TotalASS OBJECT-TYPE**

```
SYNTAX Gauge
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Total number of seconds AIS was active (near end)
    for a DS1 interface during the previous 24 hours."
::= { lucentCNMds1TotalErrorsEntry 10 }
```

**lucentCNMds1TotalFSSs OBJECT-TYPE**

```
SYNTAX Gauge
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Total number of seconds with a frame slop (near end)
    for a DS1 interface during the previous 24 hours."
::= { lucentCNMds1TotalErrorsEntry 11 }
```

**lucentCNMds1TotalFarLESS OBJECT-TYPE**

```
SYNTAX Gauge
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The counter associated with the total number of Line
    Errored Seconds (far end) encountered by a DS1
    interface during the previous 24 hours.

    A Line Errored Second is any second with at least
    one Line Code Violation."
::= { lucentCNMds1TotalErrorsEntry 12 }
```

**lucentCNMds1TotalFarCVs OBJECT-TYPE**

```
SYNTAX Gauge
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The counter associated with the total number of Code
    Violations (far end) encountered by a DS1 interface
    during the previous 24 hours.

    A Code Violation is the occurrence of a received
    CRC-6 code that is not identical to the corresponding
    locally-calculated code."
::= { lucentCNMds1TotalErrorsEntry 13 }
```

**lucentCNMds1TotalFarESS OBJECT-TYPE**

```
SYNTAX Gauge
ACCESS read-only
```

STATUS mandatory  
DESCRIPTION  
"The counter associated with the total number of Errored Seconds (far end) encountered by a DS1 interface during the previous 24 hours.  
  
An Errored Second is any second with at least one Code Violation."  
 ::= { lucentCNMds1TotalErrorsEntry 14 }

**lucentCNMds1TotalFarSESSs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the total number of Severely Errored Seconds (far end) encountered by a DS1 interface during the previous 24 hours.  
  
A Severely Errored Second is any second with 15 or more Code Violations monitored at the DS1 rate."  
 ::= { lucentCNMds1TotalErrorsEntry 15 }

**lucentCNMds1TotalFarSEFSSs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the total number of Severely Errored Framing Seconds (far end) encountered by a DS1 interface during the previous 24 hours.  
  
A Severely Errored Framing Second is any second during which one or more Severely Errored Framing (SEF) events occurred."  
 ::= { lucentCNMds1TotalErrorsEntry 16 }

**lucentCNMds1TotalFarFSSs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"Total number of seconds (far end) with a frame slip for a DS1 interface during the previous 24 hours."  
 ::= { lucentCNMds1TotalErrorsEntry 17 }

**lucentCNMds1TotalB6Ss OBJECT-TYPE**

SYNTAX Gauge

```
ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "Total number of seconds with with more than two
    CRC errors for a DS1 interface during the previous
    24 hours."
 ::= { lucentCNMds1TotalErrorsEntry 18 }
```

## **DS1 PLCP Error Counts Table**

---

This table provides counts on the various plcp protocol errors encountered by a DS1 interface during the specified measurement interval.

### **lucentCNMds1plcpErrorsTable OBJECT-TYPE**

```
SYNTAX    SEQUENCE OF LucentCNMds1plcpErrorsEntry
ACCESS    not-accessible
STATUS    mandatory
DESCRIPTION
    "A list of entries containing plcp protocol error
    counts, maintained during the specified measurement
    interval, for all DS1 interfaces managed by this
    system."
 ::= { lucent-cnm-ds1 5 }
```

### **lucentCNMds1plcpErrorsEntry OBJECT-TYPE**

```
SYNTAX    LucentCNMds1plcpErrorsEntry
ACCESS    not-accessible
STATUS    mandatory
DESCRIPTION
    "An entry containing plcp protocol error counts,
    maintained during the specified measurement interval,
    for a particular DS1 interface."
    INDEX{ lucentCNMds1plcpErrorsIndex,
    lucentCNMds1plcpErrorsInterval }
 ::= { lucentCNMds1plcpErrorsTable 1 }
```

### **LucentCNMds1plcpErrorsEntry ::=**

```
SEQUENCE {
    lucentCNMds1plcpErrorsIndex
        INTEGER,
    lucentCNMds1plcpErrorsInterval
        INTEGER,
    lucentCNMds1plcpErrorsTimeStamp
        INTEGER,
    lucentCNMds1plcpErrorsLocalTime
        DisplayString,
    lucentCNMds1plcpSEFSS
        Gauge,
    lucentCNMds1plcpUASSs
```

```
    Gauge
}
```

**lucentCNMds1plcpErrorsIndex OBJECT-TYPE**

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

"A unique value for each DS1 interface. The interface identified by a particular value of this index is the same interface as identified by the same value of an `lucentCNMifConfigIndex` object instance."

```
::= { lucentCNMds1plcpErrorsEntry 1 }
```

**lucentCNMds1plcpErrorsInterval OBJECT-TYPE**

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

"This variable identifies the measurement interval number for which the measurement is provided. It is a number between 1 and XX, where 1 identifies the most recently completed measurement interval and XX is the least recently completed measurement interval.

The value of XX is specified by the `lucentCNMds1ErrorsMaxIntervals` object given in the `lucentCNMds1ConfigTable`.

The maximum length of each measurement interval is specified by the `lucentCNMds1ErrorsIntervalLen` object given in the `lucentCNMds1ConfigTable`."

```
::= { lucentCNMds1plcpErrorsEntry 2 }
```

**lucentCNMds1plcpErrorsTimeStamp OBJECT-TYPE**

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

"The time stamp corresponding to the end of the specified measurement interval, as measured in seconds from 00:00:00 UTC (Coordinated Universal Time) January 1, 1970. Any fraction is rounded up."

```
::= { lucentCNMds1plcpErrorsEntry 3 }
```

**lucentCNMds1plcpErrorsLocalTime OBJECT-TYPE**

```
SYNTAX  DisplayString(SIZE(0..255))
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
```

"The time stamp corresponding to the end of the specified measurement interval. Any fraction is rounded up. It is given as a printable ASCII string showing the local time at the end of the interval."  
 ::= { lucentCNMds1plcpErrorsEntry 4 }

**lucentCNMds1plcpSEFss OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of Severely Errored Framing Seconds encountered by a DS1 interface at the plcp level during the specified measurement interval.  
  
A Severely Errored Framing Second is any second during which one or more Severely Errored Framing (SEF) events occurred."  
 ::= { lucentCNMds1plcpErrorsEntry 5 }

**lucentCNMds1plcpUASs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of Unavailable Seconds encountered by a DS1 interface at the plcp level during the specified measurement interval.  
  
An Unavailable Second is any second during which the DS1 interface was unavailable to offer service."  
 ::= { lucentCNMds1plcpErrorsEntry 6 }

**DS1 PLCP Status Table**

---

This table provides status information on the DS1 interfaces supported by this system.

**lucentCNMds1plcpStatusTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMds1plcpStatusEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"A list of entries containing status information for all DS1 interfaces at the plcp level managed by this system."  
 ::= { lucent-cnm-ds1 6 }

```

lucentCNMds1plcpStatusEntry OBJECT-TYPE
    SYNTAX      LucentCNMds1plcpStatusEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "An entry containing status information for a
        particular DS1 interface at the plcp level."
    INDEX{ lucentCNMds1plcpStatusIndex }
    ::= { lucentCNMds1plcpStatusTable 1 }

LucentCNMds1plcpStatusEntry ::=
    SEQUENCE {
        lucentCNMds1plcpStatusIndex
            INTEGER,
        lucentCNMds1plcpLineStatus
            INTEGER
    }

lucentCNMds1plcpStatusIndex OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "A unique value for each DS1 interface. The interface
        identified by a particular value of this index is
        the same interface as identified by the same value
        of an lucentCNMifConfigIndex object instance."
    ::= { lucentCNMds1plcpStatusEntry 1 }

lucentCNMds1plcpLineStatus OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This variable indicates the most Line Status
        of this interface at the plcp level. This object
        is a bit map represented as a sum, therefore, it
        can represent multiple failures (alarms)
        simultaneously. The various bit positions are:

            1          No Alarm Present
            2          Yellow Alarm
            8          Near End Loss-Of-Frame (LOF)"
    ::= { lucentCNMds1plcpStatusEntry 2 }

END

```

## CNM DS3 MIB

---

```
Lucent-CNM-DS3-MIB DEFINITIONS ::= BEGIN

IMPORTS

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

Object Identifiers

lucent                OBJECT IDENTIFIER ::= { enterprises 1751 }
products              OBJECT IDENTIFIER ::= { lucent 1 }
mibs                  OBJECT IDENTIFIER ::= { lucent 2 }
lucent-cnmAgent       OBJECT IDENTIFIER ::= { products 5 }
lucent-cnm             OBJECT IDENTIFIER ::= { mibs 5 }
lucent-cnm-ds3        OBJECT IDENTIFIER ::= { lucent-cnm 4 }
```

### DS3 Group of Objects (lucent-cnm-ds3)

---

The DS3 group of objects (lucent-cnm-ds3) include:

- the DS3 Configuration table
- the DS3 Status table
- the DS3 Error Counts table

### DS3 Configuration Table

---

This table provides configuration information on DS3 interfaces supported by this system.

```
lucentCNMds3ConfigTable OBJECT-TYPE
    SYNTAX SEQUENCE OF LucentCNMds3ConfigEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of entries containing configuration
        information for all DS3 interfaces managed by
        this system."
    ::= { lucent-cnm-ds3 1 }
```

**lucentCNMds3ConfigEntry OBJECT-TYPE**

```
SYNTAX LucentCNMds3ConfigEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "An entry containing configuration information
    for a particular DS3 interface."
INDEX{ lucentCNMds3ConfigIndex }
 ::= { lucentCNMds3ConfigTable 1 }
```

**LucentCNMds3ConfigEntry ::=**

```
SEQUENCE {
    lucentCNMds3ConfigIndex
        INTEGER,
    lucentCNMds3LineType
        INTEGER,
    lucentCNMds3ZeroCoding
        INTEGER,
    lucentCNMds3ErrorsMaxIntervals
        INTEGER,
    lucentCNMds3ErrorsIntervalLen
        INTEGER
}
```

**lucentCNMds3ConfigIndex OBJECT-TYPE**

```
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "A unique value for each DS3 interface. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMds3ConfigEntry 1 }
```

**lucentCNMds3LineType OBJECT-TYPE**

```
SYNTAX INTEGER{
    other(1),
    ds3M23(2),
    ds3SYNTRAN(3),
    ds3CbitParity(4),
    ds3ClearChannel(5),
    e3other(6),
    e3Framed(7),
    e3Plcp(8)
}
ACCESS read-only
ACCESS read
DESCRIPTION
    "This variable indicates the variety of DS3 C-bit
```

application supported by this interface.

For T3 based SMDS, this variable will have the value `ds3ClearChannel`, denoting that this interface supports the clear-channel with C-bit usage unspecified as specified by ANSI.

For E3 based SMDS, this variable will have the value `e3Plcp`, denoting that this interface supports the ETSI T/NA(91)18 specification."

```
::= { lucentCNMds3ConfigEntry 2 }
```

**lucentCNMds3ZeroCoding OBJECT-TYPE**

```
SYNTAX  INTEGER{
        ds3other(1),
        ds3B3ZS(2),
        e3HDB3(3)
    }
```

```
ACCESS  read-only
```

```
ACCESS  read
```

```
DESCRIPTION
```

"This variable describes the variety of zero code suppression/substitution used on the DS3 interface its characteristics.

For T3 based SMDS, this variable will have the value `ds3B3ZS`.

For E3 based SMDS, this variable will have the value `e3HDB3`."

```
::= { lucentCNMds3ConfigEntry 3 }
```

**lucentCNMds3ErrorsMaxIntervals OBJECT-TYPE**

```
SYNTAX  INTEGER
```

```
ACCESS  read-only
```

```
STATUS  mandatory
```

```
DESCRIPTION
```

"This variable identifies the maximum number of measurement intervals supported for the error counts maintained by this DS3 interface in the `lucentCNMds3ErrorsTable`."

```
::= { lucentCNMds3ConfigEntry 4 }
```

**lucentCNMds3ErrorsIntervalLen OBJECT-TYPE**

```
SYNTAX  INTEGER
```

```
ACCESS  read-only
```

```
STATUS  mandatory
```

```
DESCRIPTION
```

"This variable identifies the number of seconds that make up one complete measurement interval for

```
        the error counts maintained by this DS3 interface
        in the lucentCNMds3ErrorsTable."
 ::= { lucentCNMds3ConfigEntry 5 }
```

### **DS3 Status Table**

---

This table provides status information on DS3 interfaces supported by this system.

#### **lucentCNMds3StatusTable OBJECT-TYPE**

```
SYNTAX  SEQUENCE OF LucentCNMds3StatusEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
        "A list of entries containing status information
        for all DS3 interfaces managed by this system."
 ::= { lucent-cnm-ds3 2 }
```

#### **lucentCNMds3StatusEntry OBJECT-TYPE**

```
SYNTAX  LucentCNMds3StatusEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
        "An entry containing status information for a
        particular DS3 interface."
        INDEX{ lucentCNMds3StatusIndex }
 ::= { lucentCNMds3StatusTable 1 }
```

#### **lucentCNMds3StatusEntry ::=**

```
SEQUENCE {
    lucentCNMds3StatusIndex
        INTEGER,
    lucentCNMds3LineStatus
        INTEGER
}
```

#### **lucentCNMds3StatusIndex OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "A unique value for each DS3 interface. The interface
        identified by a particular value of this index is
        the same interface as identified by the same value
        of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMds3StatusEntry 1 }
```

**lucentCNMds3LineStatus OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
ACCESS  read
DESCRIPTION
    "This variable indicates the most Line Status
    of this interface. This object is a bit map
    represented as a sum, therefore, it can represent
    multiple failures (alarms) simult
    various bit positions are:

            1      No Alarm Present
            2      Yellow Alarm
            4      Near End Alarm-Indication-Signal
            8      Near End Loss-Of-Frame
            16     Near End Loss-Of-Signal

    For example, for an interface that has LOF and
    LOS outstanding, this object will have a value
    of 24."
 ::= { lucentCNMds3StatusEntry 2 }
```

**DS3 Error Counts Table**

---

This table provides counts on the various protocol errors encountered by a DS3 interface during the specified measurement interval.

**lucentCNMds3ErrorsTable OBJECT-TYPE**

```
SYNTAX  SEQUENCE OF LucentCNMds3ErrorsEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
    "A list of entries containing protocol error counts,
    maintained during the specified measurement interval,
    for all DS3 interfaces managed by this system."
 ::= { lucent-cnm-ds3 3 }
```

**lucentCNMds3ErrorsEntry OBJECT-TYPE**

```
SYNTAX  LucentCNMds3ErrorsEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
    "An entry containing protocol error counts,
    maintained during the specified measurement interval,
    for a particular DS3 interface."
    INDEX{ lucentCNMds3ErrorsIndex,
           lucentCNMds3ErrorsInterval }
 ::= { lucentCNMds3ErrorsTable 1 }
```

**LucentCNMds3ErrorsEntry ::=**

```

SEQUENCE {
    lucentCNMds3ErrorsIndex
        INTEGER,
    lucentCNMds3ErrorsInterval
        INTEGER,
    lucentCNMds3ErrorsTimeStamp
        INTEGER,
    lucentCNMds3ErrorsLocalTime
        DisplayString,
    lucentCNMds3LCVs
        Gauge,
    lucentCNMds3LEsSs
        Gauge,
    lucentCNMds3LSEsSs
        Gauge,
    lucentCNMds3CVs
        Gauge,
    lucentCNMds3ESs
        Gauge,
    lucentCNMds3SEsSs
        Gauge,
    lucentCNMds3SEFSs
        Gauge,
    lucentCNMds3UASs
        Gauge
}

```

**lucentCNMds3ErrorsIndex OBJECT-TYPE**

```

SYNTAX INTEGER
ACCESS read-only
ACCESS read
DESCRIPTION

```

"A unique value for each DS3 interface. The interface identified by a particular value of this index is the same interface as identified by the same value of an lucentCNMifConfigIndex object instance."

```
 ::= { lucentCNMds3ErrorsEntry 1 }
```

**lucentCNMds3ErrorsInterval OBJECT-TYPE**

```

SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION

```

"This variable identifies the measurement interval number for which measurement is provided. It is a number between 1 and XX, where 1 identifies the most recently completed measurement interval and XX is the least recently completed measurement interval."

The value of XX is specified by the  
lucentCNMds3ErrorsMaxIntervals object given in the  
lucentCNMds3ConfigTable.

The maximum length of each measurement interval is  
specified by the lucentCNMds3ErrorsIntervalLen object  
given in the lucentCNMds3ConfigTable."

::= { lucentCNMds3ErrorsEntry 2 }

**lucentCNMds3ErrorsTimeStamp OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp corresponding to the end of the  
specified measurement interval, as measured in  
seconds from 00:00:00 UTC (Coordinated Universal  
Time) January 1, 1970. Any fraction is rounded up."

::= { lucentCNMds3ErrorsEntry 3 }

**lucentCNMds3ErrorsLocalTime OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp corresponding to the end of the  
specified measurement interval. Any fraction is  
rounded up. It is given as a printable ASCII string  
showing the local time at the end of the interval."

::= { lucentCNMds3ErrorsEntry 4 }

**lucentCNMds3LCVs OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

ACCESS read

DESCRIPTION

"The counter associated with the number of Line  
Code Violations encountered by a DS3 interface  
during the specified measurement interval.

For a B3ZS-coded signal, a Line Code Violation is  
the occurrence of a received bipolar violation that  
is not part of a zero-substitution code. For a B3ZS  
signal, an LCV may also include other error patterns  
such as: three consecutive zeros and incorrect parity.

The Bipolar Violation Rate is generally viewed as  
approximating the Bit Error Rate."

::= { lucentCNMds3ErrorsEntry 5 }

**lucentCNMds3LESS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
ACCESS read  
DESCRIPTION  
"The counter associated with the number of Line Errored Seconds encountered by a DS3 interface during the specified measurement interval.  
  
A Line Errored Second is any second with at least one Line Code Violation."  
::= { lucentCNMds3ErrorsEntry 6 }

**lucentCNMds3LSESS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
ACCESS read  
DESCRIPTION  
"The counter associated with the number of Line Severely Errored Seconds encountered by a DS3 interface during the specified measurement interval.  
  
A Line Severely Errored Second is any second with 45 or more Line Code Violations monitored at the DS3 rate."  
::= { lucentCNMds3ErrorsEntry 7 }

**lucentCNMds3CVs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
ACCESS read  
DESCRIPTION  
"The counter associated with the number of Code Violations encountered by a DS3 interface during the specified measurement interval.  
  
For SMDS, a DS3 Code Violation is an occurrence of a P-bit error. Excessive Coding Violations signifies a faulty transmission line."  
::= { lucentCNMds3ErrorsEntry 8 }

**lucentCNMds3ESS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
ACCESS read  
DESCRIPTION  
"The counter associated with the number of Errored Seconds encountered by a DS3 interface during the specified measurement interval."

For SMDS DS3 signals, this is the number of seconds during which at least one error occurred which is a P-bit error or a Severely Errored Framing event."  
 ::= { lucentCNMds3ErrorsEntry 9 }

**lucentCNMds3SESS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
ACCESS read  
DESCRIPTION  
"The counter associated with the number of Severely Errored Seconds encountered by a DS3 interface during the specified measurement interval.  
  
For SMDS DS3 signals, this is the number of seconds during which 44 or more P-bit errors occurred or at least one Severely Errored Framing event occurred.  
  
All Severely Errored Seconds are also counted as Errored Seconds."  
 ::= { lucentCNMds3ErrorsEntry 10 }

**lucentCNMds3SEFSS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
ACCESS read  
DESCRIPTION  
"The counter associated with the number of Severely Errored Framing Seconds encountered by a DS3 interface during the specified measurement interval.  
  
A Severely Errored Framing Second is any second during which one or more Severely Errored Framing (SEF) events occurred. A Severely Errored Framing event is declared when 3 or more errors in 16 or fewer consecutive F-bits occur within a DS3 M-frame."  
 ::= { lucentCNMds3ErrorsEntry 11 }

**lucentCNMds3UASS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
ACCESS read  
DESCRIPTION  
"The counter associated with the number of Unavailable Seconds encountered by a DS3 interface during the specified measurement interval.  
  
An Unavailable Second is any second during which the DS3 interface was unavailable to offer service."  
 ::= { lucentCNMds3ErrorsEntry 12 }

## DS3 PLCP Error Counts Table

This table provides counts on the various plcp protocol errors encountered by a DS3 interface during the specified measurement interval.

### lucentCNMds3plcpErrorsTable OBJECT-TYPE

```
SYNTAX SEQUENCE OF LucentCNMds3plcpErrorsEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A list of entries containing plcp level protocol
    error counts, maintained during the specified
    measurement interval, for all DS3 interfaces managed
    by this system."
 ::= { lucent-cnm-ds3 4 }
```

### lucentCNMds3plcpErrorsEntry OBJECT-TYPE

```
SYNTAX LucentCNMds3plcpErrorsEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "An entry containing plcp level protocol error counts,
    maintained during the specified measurement interval,
    for a particular DS3 interface."
INDEX { lucentCNMds3plcpErrorsIndex,
        lucentCNMds3plcpErrorsInterval }
 ::= { lucentCNMds3plcpErrorsTable 1 }
```

### LucentCNMds3plcpErrorsEntry ::=

```
SEQUENCE {
    lucentCNMds3plcpErrorsIndex
        INTEGER,
    lucentCNMds3plcpErrorsInterval
        INTEGER,
    lucentCNMds3plcpErrorsTimeStamp
        INTEGER,
    lucentCNMds3plcpErrorsLocalTime
        DisplayString,
    lucentCNMds3plcpSEFSs
        Gauge,
    lucentCNMds3plcpUASs
        Gauge
}
```

### lucentCNMds3plcpErrorsIndex OBJECT-TYPE

```
SYNTAX INTEGER
ACCESS read-only
ACCESS read
DESCRIPTION
    "A unique value for each DS3 interface. The interface
```

identified by a particular value of this index is the same interface as identified by the same value of an `lucentCNMifConfigIndex` object instance."  
 ::= { `lucentCNMds3plcpErrorsEntry` 1 }

**lucentCNMds3plcpErrorsInterval OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable identifies the measurement interval number for which measurement is provided. It is a number between 1 and XX, where 1 identifies the most recently completed measurement interval and XX is the least recently completed measurement interval.

The value of XX is specified by the `lucentCNMds3ErrorsMaxIntervals` object given in the `lucentCNMds3ConfigTable`.

The maximum length of each measurement interval is specified by the `lucentCNMds3ErrorsIntervalLen` object given in the `lucentCNMds3ConfigTable`."

::= { `lucentCNMds3plcpErrorsEntry` 2 }

**lucentCNMds3plcpErrorsTimeStamp OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp corresponding to the end of the specified measurement interval, as measured in seconds from 00:00:00 UTC (Coordinated Universal Time) January 1, 1970. Any fraction is rounded up."

::= { `lucentCNMds3plcpErrorsEntry` 3 }

**lucentCNMds3plcpErrorsLocalTime OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time stamp corresponding to the end of the specified measurement interval. Any fraction is rounded up. It is given as a printable ASCII string showing the local time at the end of the interval."

::= { `lucentCNMds3plcpErrorsEntry` 4 }

**lucentCNMds3plcpSEFSS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
ACCESS read  
DESCRIPTION

"The counter associated with the number of Severely Errored Framing Seconds encountered by a DS3 interface at the plcp level during the specified measurement interval.

A Severely Errored Framing Second is any second during which one or more Severely Errored Framing (SEF) events occurred. A Severely Errored Framing event is declared when 3 or more errors in 16 or fewer consecutive F-bits occur within a DS3 M-frame."

::= { lucentCNMds3plcpErrorsEntry 5 }

**lucentCNMds3plcpUASS OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
ACCESS read  
DESCRIPTION

"The counter associated with the number of Unavailable Seconds encountered by a DS3 interface at the plcp level during the specified measurement interval.

An Unavailable Second is any second during which the DS3 interface was unavailable to offer service."

::= { lucentCNMds3plcpErrorsEntry 6 }

**DS3 PLCP Status Table**

---

This table provides status information on the DS3 interfaces supported by this system.

**lucentCNMds3plcpStatusTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMds3plcpStatusEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION

"A list of entries containing status information for all DS3 interfaces at the plcp level managed by this system."

::= { lucent-cnm-ds3 5 }

**lucentCNMds3plcpStatusEntry OBJECT-TYPE**

SYNTAX LucentCNMds3plcpStatusEntry  
ACCESS not-accessible  
STATUS mandatory

```
DESCRIPTION
    "An entry containing status information for a
    particular DS3 interface at the plcp level."
    INDEX{ lucentCNMds3plcpStatusIndex }
 ::= { lucentCNMds3plcpStatusTable 1 }

LucentCNMds3plcpStatusEntry ::=
SEQUENCE {
    lucentCNMds3plcpStatusIndex
        INTEGER,
    lucentCNMds3plcpLineStatus
        INTEGER
}

lucentCNMds3plcpStatusIndex OBJECT-TYPE
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each DS3 interface. The interface
    identified by a particular value of this index is
    the same interface as identified by the same value
    of an lucentCNMifConfigIndex object instance."
 ::= { lucentCNMds3plcpStatusEntry 1 }

lucentCNMds3plcpLineStatus OBJECT-TYPE
SYNTAX  INTEGER
ACCESS  read-only
ACCESS  read
DESCRIPTION
    "This variable indicates the most Line Status
    of this interface at the plcp level. This object
    is a bit map represented as a sum, therefore, it
    can represent multiple failures (alarms)
    simultaneously. The various bit positions are:

        1          No Alarm Present
        2          Yellow Alarm
        8          Near End Loss-Of-Frame (LOF)"
 ::= { lucentCNMds3plcpStatusEntry 2 }

END
```

## **CNM Enhanced Frame Relay MIB**

```

LUCENT-CNM-ENHANCED-FRAME-RELAY-MIB DEFINITIONS ::= BEGIN

IMPORTS

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

Object Identifiers

lucent                OBJECT IDENTIFIER ::= { enterprises 1751 }
products              OBJECT IDENTIFIER ::= { lucent 1 }
mibs                  OBJECT IDENTIFIER ::= { lucent 2 }
lucent-cnmAgent       OBJECT IDENTIFIER ::= { products 5 }
lucent-cnm             OBJECT IDENTIFIER ::= { mibs 5 }
lucent-cnm-efr        OBJECT IDENTIFIER ::= { lucent-cnm 8 }

```

### **Frame Relay Group of Objects (lucent-cnm-efr)**

The Frame Relay group of objects (lucent-cnm-efr) include:

- the Frame Relay Interface-level Configuration table
- the Frame Relay Interface-level Measurements table
- the Frame Relay PVC-level Configuration table
- the Frame Relay PVC-level Measurements table
- the Frame Relay PVC-level Status table



**NOTE:**

This MIB, especially in the configuration tables, refers to "local" and "remote" devices. These terms are used to identify devices relative to the interface that is being managed. The "local" device is the Frame Relay equipment connected to the interface (port) being managed. The "remote" device refers to the Frame Relay equipment connected to the interface that will be at the other end of the call or calls associated with the PVCs being managed. Also, this MIB refers to originating and terminating endpoints. These terms are used to identify which DLCI initiates the call. It does not refer to the device that sends or receives data, that is independent of the call setup. The originating endpoints initiate the call, the terminating endpoint only acknowledges, receives, and completes the call setup process.

## Enhanced-Frame-Relay Interface-Level Configuration Table

---

This table provides configuration information the Frame Relay interfaces supported by this system.

### **lucentCNMefrConfigTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMefrConfigEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "A list of entries containing configuration information for all Frame Relay interfaces managed by this system."  
 ::= { lucent-cnm-efr 1 }

### **lucentCNMefrConfigEntry OBJECT-TYPE**

SYNTAX LucentCNMefrConfigEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
    "An entry containing configuration information for a particular Frame Relay interface."  
INDEX { lucentCNMefrConfigIndex }  
 ::= { lucentCNMefrConfigTable 1 }

### **lucentCNMefrConfigEntry ::=**

SEQUENCE {  
    lucentCNMefrConfigIndex  
        INTEGER,  
    lucentCNMefrMgmtType  
        INTEGER,  
    lucentCNMefrPollDirection  
        INTEGER,  
    lucentCNMefrFullStatusPoll  
        INTEGER,  
    lucentCNMefrErrorThreshold  
        INTEGER,  
    lucentCNMefrMonitoredEvents  
        INTEGER,  
    lucentCNMefrIntegrityTimer  
        INTEGER,  
    lucentCNMefrPollVerifyTimer  
        INTEGER,  
    lucentCNMefrLMIFlowControl  
        INTEGER,  
    lucentCNMefrSupportedPVCs  
        INTEGER,  
    lucentCNMefrMeasMaxIntervals  
        INTEGER,

```

    lucentCNMefrMeasIntervalLen
        INTEGER,
    lucentCNMefrMaxFrameSize
        INTEGER,
    lucentCNMefrAggregateCIR
        INTEGER,
    lucentCNMefrMinInterframeDelay
        INTEGER,
    lucentCNMefrRate
        INTEGER
}

```

**lucentCNMefrConfigIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each Frame Relay interface.
    The interface identified by a particular value of
    this index is the same interface as identified by
    the same value of an lucentCNMifConfigIndex object
    instance."
 ::= { lucentCNMefrConfigEntry 1 }

```

**lucentCNMefrMgmtType OBJECT-TYPE**

```

SYNTAX  INTEGER {
    auto-set(1),
    lmi(2),
    ansi(3),
    ccitt(4),
    none(5)
}
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable identifies the PVC management scheme
    being used to manage this Frame Relay interface.
    A value of auto-set(1) indicates that the interface
    automatically adopts the scheme being used by the
    access device. A value none(5) indicates that PVC
    management procedures are disabled on this interface."
 ::= { lucentCNMefrConfigEntry 2 }

```

**lucentCNMefrPollDirection OBJECT-TYPE**

```

SYNTAX  INTEGER {
    receive(1),
    send(2),
    both(3),
    off(4),
    none(5)
}

```

```
    }  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "This variable specifies the role the Frame Relay  
    interface performs in a bi-directional PVC  
    management scheme. A value of receive(1) means that  
    this interface simply responds to Status Enquiries  
    and a value of send(2) means that this interface  
    initiates Status Enquiries. A value of both(3) means  
    that this interface performs both functions."  
 ::= { lucentCNMefrConfigEntry 3 }
```

**lucentCNMefrFullStatusPoll OBJECT-TYPE**

```
SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "This variable identifies the maximum number  
    of Status Enquiry intervals that pass before  
    issuance of a Full Status Enquiry message.  
  
    This counter is referred to as N391 in the ANSI  
    standard.  
  
    This variable applies only if the PVC management  
    procedures are enabled. See lucentCNMefrMgmtType object  
    for more details."  
 ::= { lucentCNMefrConfigEntry 4 }
```

**lucentCNMefrErrorThreshold OBJECT-TYPE**

```
SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "This variable identifies the maximum number  
    of errored events that can occur before declaring  
    the Frame Relay interface down. An errored event  
    is the non-receipt of a Status Enquiry, or the  
    receipt of a Status Enquiry with an invalid sequence  
    number.  
  
    This counter is referred to as N392 in the ANSI  
    standard.  
  
    The value for this counter cannot exceed the  
    value of the monitored events counter (see the  
    lucentCNMefrMonitoredEvents object)."  
 ::= { lucentCNMefrConfigEntry 5 }
```

**lucentCNMefrMonitoredEvents OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable specifies a value 'm', that is used with the Error Threshold (lucentCNMefrErrorThreshold) to determine link problems. A link problem will be declared when the number of errors in the last 'm' monitored events exceeds the Error Threshold.

This counter is referred to as N393 in the ANSI standard."

::= { lucentCNMefrConfigEntry 6 }

**lucentCNMefrIntegrityTimer OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable identifies the number of seconds between successive Status Enquiry messages.

This timer is referred to as T391 in the ANSI standard.

This variable is applicable only if this interface is configured to be the sender of PVC management polls (see the lucentCNMefrPollDirection object)."

::= { lucentCNMefrConfigEntry 7 }

**lucentCNMefrPollVerifyTimer OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable specifies the maximum number of seconds the Frame Relay interface allows between the receipt of Status Enquiries. The expiration of this timer is considered to be an errored event.

This timer is referred to as T392 in the ANSI standard.

This variable is applicable only if this interface is configured to be the receiver of PVC management polls. Refer to the lucentCNMefrPollDirection object to verify this."

::= { lucentCNMefrConfigEntry 8 }

**lucentCNMefrLMIFlowControl OBJECT-TYPE**

```
SYNTAX  INTEGER {
        lmiFlowControlOn(1),
        lmiFlowControlOff(2)
    }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "If the PVC management scheme being used is LMI,
    then this variable indicates whether this Frame
    Relay interface uses XON/XOFF as a means of
    congestion avoidance.

    This technique is defined in Revision 1.0 of the
    LMI Specification 'Frame Relay Specification with
    Extensions'."
 ::= { lucentCNMefrConfigEntry 9 }
```

**lucentCNMefrSupportedPVCs OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable specifies the number of DLCIs
    currently supported on this Frame Relay interface."
 ::= { lucentCNMefrConfigEntry 10 }
```

**lucentCNMefrMeasMaxIntervals OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable identifies the maximum number
    of measurement intervals supported for the
    measurements maintained by this Frame Relay
    interface in the lucentCNMefrMeasTable."
 ::= { lucentCNMefrConfigEntry 11 }
```

**lucentCNMefrMeasIntervalLen OBJECT-TYPE**

```
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable identifies the number of seconds
    that make up one complete measurement interval for
    for the measurements maintained by this Frame Relay
    interface in the lucentCNMefrMeasTable."
 ::= { lucentCNMefrConfigEntry 12 }
```

**lucentCNMefrMaxFrameSize OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable identifies the maximum allowable frame size (in bytes) that this Frame Relay Interface can receive. The maximum allowable frame size includes the information field of a frame, but does not include the bytes the frame relay protocol uses in the frame header. The maximum frame size that an interface can receive should be greater than or equal to the frame size that the lucentached device transmits."

::= { lucentCNMefrConfigEntry 13 }

**lucentCNMefrAggregateCIR OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable specifies the threshold for the sum of the CIRs for calls being established on this Frame Relay Interface. Any call lucentempt that exceeds the selected threshold will be rejected. The threshold is based on the sum of the CIRs required for data transmitted from remote devices. If a CIR is not being used, the aggregate CIR should be 0, which means that any call lucentempt where the remote end of the PVC requires a nonzero CIR will be rejected.

A value for the MAXIMUM AGGREGATE CIR can be percentage of line speed (1-400%) or a value in bps from 1200-256000 or 0 for no CIR."

::= { lucentCNMefrConfigEntry 14 }

**lucentCNMefrMinInterframeDelay OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The variable specifies the minimum delay between the transmission of frames from the interface to the access device. This delay, which is produced by transmitting flag characters, is required by access devices that cannot process frames at the maximum line rate. The value 1 translates to no additional delay."

```
::= { lucentCNMefrConfigEntry 15 }
```

**lucentCNMefrRate OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable pertains only to a channelized T1 interface. It specifies 56KxN or clear channel operation (64KxN) is to be used for the time slot(s) associated with the virtual port. Each DS1 frame contains 24, 8-bit time slots. 56K coding forces one bit to be set to high for each time slot in order to maintain the minimum pulse density requirements of some DS1 equipment. The remaining seven bits are used for user data, which provides 56 Kbps per time slot. Clear channel operation uses the entire eight bits for user data, which yields 64 Kbps per time slot.

If the interface is a channelized e1, the value for rate will always be 64KxN."

```
::= { lucentCNMefrConfigEntry 16 }
```

### **Enhanced Frame Relay Measurements Table**

---

This table provides counts on the various measurement counts maintained by a Frame Relay interface during the specified measurement interval.

**lucentCNMefrMeasTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMefrMeasEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A list of entries containing measurement counts, maintained during the specified measurement interval, for all Frame Relay interfaces managed by this system."

```
::= { lucent-cnm-efr 2 }
```

**lucentCNMefrMeasEntry OBJECT-TYPE**

SYNTAX LucentCNMefrMeasEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"An entry containing measurement counts, maintained during the specified measurement interval, for a particular Frame Relay interface."

```
INDEX { lucentCNMefrMeasIndex,  
        lucentCNMefrMeasInterval }  
 ::= { lucentCNMefrMeasTable 1 }
```

**LucentCNMefrMeasEntry ::=**

```
SEQUENCE {  
    lucentCNMefrMeasIndex  
        INTEGER,  
    lucentCNMefrMeasInterval  
        INTEGER,  
    lucentCNMefrMeasTimeStamp  
        INTEGER,  
    lucentCNMefrMeasLocalTime  
        DisplayString,  
    lucentCNMefrReceivedOctets  
        Gauge,  
    lucentCNMefrSentOctets  
        Gauge,  
    lucentCNMefrReceivedFrames  
        Gauge,  
    lucentCNMefrSentFrames  
        Gauge,  
    lucentCNMefrBadFrames  
        Gauge,  
    lucentCNMefrReceiverOverruns  
        Gauge,  
    lucentCNMefrIngressUtil  
        Gauge,  
    lucentCNMefrEgressUtil  
        Gauge,  
    lucentCNMefrPeakReceivedOctets  
        Gauge,  
    lucentCNMefrPeakReceivedFrames  
        Gauge,  
    lucentCNMefrPeakSentFrames  
        Gauge,  
    lucentCNMefrPeakSentOctets  
        Gauge,  
    lucentCNMefrPeakReceivedUtil  
        Gauge,  
    lucentCNMefrPeakSentUtil  
        Gauge,  
    lucentCNMefrPeakRcvAvgFrmSize  
        Gauge,  
    lucentCNMefrPeakSentAvgFrmSize  
        Gauge,  
    lucentCNMefrSentAvgFrmSize  
        Gauge,  
    lucentCNMefrReceivedAvgFrmSize  
        Gauge,
```

```

    lucentCNMefrReceivedCongestCount
        Gauge,
    lucentCNMefrReceivedCongestSecs
        Gauge,
    lucentCNMefrSentCongestCount
        Gauge,
    lucentCNMefrSentCongestSecs
        Gauge,
    lucentCNMefrSigUserLinkRelErrors
        Gauge,
    lucentCNMefrSigNetLinkRelErrors
        Gauge,
    lucentCNMefrSigNetUserProtErrors
        Gauge,
    lucentCNMefrSigNetProtErrors
        Gauge
}

```

**lucentCNMefrMeasIndex OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A unique value for each Frame Relay interface.
    The interface identified by a particular value of
    this index is the same interface as identified by
    the same value of an lucentCNMifConfigIndex object
    instance."
 ::= { lucentCNMefrMeasEntry 1 }

```

**lucentCNMefrMeasInterval OBJECT-TYPE**

```

SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable identifies the measurement interval
    number for which measurement is provided. It is a
    number between 1 and XX, where 1 identifies the most
    recently completed measurement interval and XX is
    the least recently completed measurement interval.

    The value of XX is specified by the
    lucentCNMefrMeasMaxIntervals object given in the
    lucentCNMefrConfigTable.

    The maximum length of each measurement interval is
    specified by the lucentCNMefrMeasIntervalLen object
    given
    in the lucentCNMefrConfigTable."
 ::= { lucentCNMefrMeasEntry 2 }

```

**lucentCNMefrMeasTimeStamp OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The time stamp corresponding to the end of the specified measurement interval, as measured in seconds from 00:00:00 UTC (Coordinated Universal Time) January 1, 1970. Any fraction is rounded up."

::= { lucentCNMefrMeasEntry 3 }

**lucentCNMefrMeasLocalTime OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The time stamp corresponding to the end of the specified measurement interval. Any fraction is rounded up. It is given as a printable ASCII string showing the local time at the end of the interval."

::= { lucentCNMefrMeasEntry 4 }

**lucentCNMefrReceivedOctets OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The counter associated with the number of bytes received by this Frame Relay interface during the specified measurement interval."

::= { lucentCNMefrMeasEntry 5 }

**lucentCNMefrSentOctets OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The counter associated with the number of bytes sent by this Frame Relay interface during the specified measurement interval."

::= { lucentCNMefrMeasEntry 6 }

**lucentCNMefrReceivedFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The counter associated with the number of frames received by this Frame Relay interface during the specified measurement interval."

```
::= { lucentCNMefrMeasEntry 7 }
```

**lucentCNMefrSentFrames OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of frames sent by this Frame Relay interface during the specified measurement interval."

```
::= { lucentCNMefrMeasEntry 8 }
```

**lucentCNMefrBadFrames OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of frames received by this Frame Relay interface that are faulty. The interface discards bad frames received.

A bad frame could be a frame with a non-integral number of octets, or an aborted frame, or a frame with a bad Frame Check Sequence (FCS), or a frame that exceeds the maximum frame size."

```
::= { lucentCNMefrMeasEntry 9 }
```

**lucentCNMefrReceiverOverruns OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of receiver overruns that occurred at this Frame Relay interface during the specified measurement interval.

A receiver overrun occurs when frames are received faster than they can be stored and forwarded."

```
::= { lucentCNMefrMeasEntry 10 }
```

**lucentCNMefrIngressUtil OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the utilization level of the Frame Relay interface, derived as the total bytes received (in the ingress direction) divided by the line speed of the interface, over the measurement interval."

```
::= { lucentCNMefrMeasEntry 11 }
```

**lucentCNMefrEgressUtil OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the utilization level of the Frame Relay interface, derived as the total bytes sent (in the egress direction) divided by the line speed of the interface, over the measurement interval."

```
::= { lucentCNMefrMeasEntry 12 }
```

**lucentCNMefrPeakReceivedOctets OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of bytes received by the Frame Relay interface in the peak 5 minute sub-interval. The peak is the 5 minute sub-interval that contains the the largest bytes count received by the interface within the hour."

```
::= { lucentCNMefrMeasEntry 13 }
```

**lucentCNMefrPeakReceivedFrames OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of frames received by the Frame Relay interface in the peak 5 minute sub-interval. The peak is the 5 minute sub-interval that contains the largest frames count received by the interface within the hour."

```
::= { lucentCNMefrMeasEntry 14 }
```

**lucentCNMefrPeakSentOctets OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of bytes sent by the Frame Relay interface in the peak 5 minute sub-interval. The peak is the 5 minute sub-interval that contains the largest bytes count sent by the interface within the hour."

```
::= { lucentCNMefrMeasEntry 15 }
```

**lucentCNMefrPeakSentFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of frames sent by the Frame Relay interface in the peak 5 minute sub-interval. The peak is the five minute sub-interval that contains the largest frames count sent by the interface within the hour."  
 ::= { lucentCNMefrMeasEntry 16 }

**lucentCNMefrPeakReceivedUtil OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the percentage utilization of the Frame Relay interface in the receive direction in the peak 5 minute sub-interval. The peak is the five minute sub-interval that contains the largest percentage utilization by the interface in the receive direction within the hour."  
 ::= { lucentCNMefrMeasEntry 17 }

**lucentCNMefrPeakSentUtil OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the percentage utilization of the Frame Relay interface in the sent direction in the peak 5 minute sub-interval. The peak is the five minute sub-interval that contains the largest percentage utilization by the interface in the sent direction within the hour."  
 ::= { lucentCNMefrMeasEntry 18 }

**lucentCNMefrPeakRcvAvgFrmSize OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the average frame size received by the Frame Relay interface in the peak 5 minute sub-interval. The peak is the five minute sub-interval that contains the largest average frame size received by the interface within the hour."  
 ::= { lucentCNMefrMeasEntry 19 }

**lucentCNMefrPeakSentAvgFrmSize OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the average frame size sent by the Frame Relay interface in the peak 5 minute sub-interval. The peak is the five minute sub-interval that contains the largest average frame size sent by the interface within the hour."  
 ::= { lucentCNMefrMeasEntry 20 }

**lucentCNMefrSentAvgFrmSize OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the average frame size sent by the Frame Relay interface during the specified measurement interval."  
 ::= { lucentCNMefrMeasEntry 21 }

**lucentCNMefrReceivedAvgFrmSize OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the average frame size received by the Frame Relay interface during the specified measurement interval."  
 ::= { lucentCNMefrMeasEntry 22 }

**lucentCNMefrReceivedCongestCount OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of times this interface went into buffer congestion during the measurement interval. This count is only applicable to M2-FRM interfaces."  
 ::= { lucentCNMefrMeasEntry 23 }

**lucentCNMefrReceivedCongestSecs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of seconds the interface was in buffer congestion. This count

is applicable only to M2-FRM interfaces."  
 ::= { lucentCNMefrMeasEntry 24 }

**lucentCNMefrSentCongestCount OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of times this interface went into buffer congestion during the measurement interval. Transmitter congestion usually indicates a speed mismatch, that is, more data is entering the module from the backplane than can be transmitted through the port. This count is only applicable to M2-FRM interfaces."  
 ::= { lucentCNMefrMeasEntry 25 }

**lucentCNMefrSentCongestSecs OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of seconds this interface was in buffer congestion. This count is applicable only to M2-FRM interfaces."  
 ::= { lucentCNMefrMeasEntry 26 }

**lucentCNMefrSigUserLinkRelErrors OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of times a poll (status enquiry message) was expected but was not received within the polling verification timer, or a poll was received with an invalid sequence number for this interface. This count is applicable only to M2-FRM interfaces."  
 ::= { lucentCNMefrMeasEntry 27 }

**lucentCNMefrSigNetLinkRelErrors OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of times a poll response (status message) was expected but was not received within the link integrity verification polling timer, or a poll response was received with an invalid sequence number for this interface. This

count is applicable only to M2-FRM interfaces."  
 ::= { lucentCNMefrMeasEntry 28 }

**lucentCNMefrSigNetUserProtErrors OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of user-side local in-channel signaling protocol errors for this interface. This count is applicable only to M2-FRM interfaces."  
 ::= { lucentCNMefrMeasEntry 29 }

**lucentCNMefrSigNetProtErrors OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The counter associated with the number of network-side local in-channel signaling protocol for this interface. This count is applicable to the M2-FRM interfaces."  
 ::= { lucentCNMefrMeasEntry 30 }

**Enhanced Frame Relay PVC Level Configuration Table**

---

This table provides PVC-level configuration information for the Frame Relay interfaces supported by this system.

**lucentCNMefrPVCConfigTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMefrPVCConfigEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"A list of entries containing configuration information for all Frame Relay interfaces managed by this system."  
 ::= { lucent-cnm-efr 3 }

**lucentCNMefrPVCConfigEntry OBJECT-TYPE**

SYNTAX LucentCNMefrPVCConfigEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"An entry containing configuration information for a particular Frame Relay interface."  
INDEX { lucentCNMefrPVCConfigIfIndex,

```

        lucentCNMefrPVCConfigIndex }
 ::= { lucentCNMefrPVCConfigTable 1 }

```

**lucentCNMefrPVCConfigEntry ::=**

```

SEQUENCE {
    lucentCNMefrPVCConfigIfIndex
        INTEGER,
    lucentCNMefrPVCConfigIndex
        INTEGER,
    lucentCNMefrPVCSERVICEType
        INTEGER,
    lucentCNMefrLocalCIR
        INTEGER,
    lucentCNMefrLocalCommittedBurst
        INTEGER,
    lucentCNMefrLocalExcessBurst
        INTEGER,
    lucentCNMefrRemoteCIR
        INTEGER,
    lucentCNMefrRemoteCommittedBurst
        INTEGER,
    lucentCNMefrRemoteExcessBurst
        INTEGER,
    lucentCNMefrMulticastGroup1
        INTEGER,
    lucentCNMefrMulticastGroup2
        INTEGER,
    lucentCNMefrMulticastGroup3
        INTEGER,
    lucentCNMefrMulticastGroup4
        INTEGER,
    lucentCNMefrPVCMeasMaxIntervals
        INTEGER,
    lucentCNMefrPVCMeasIntervalLen
        INTEGER
}

```

**lucentCNMefrPVCConfigIfIndex OBJECT-TYPE**

```

SYNTAX    INTEGER
ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "A unique value for each Frame Relay interface.
    The interface identified by a particular value of
    this index is the same interface as identified by
    the same value of an lucentCNMifConfigIndex object
    instance."
 ::= { lucentCNMefrPVCConfigEntry 1 }

```

**lucentCNMefrPVCConfigIndex OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"The DLCI number on a Frame Relay interface that identifies a unique Data Link Connection."  
::= { lucentCNMefrPVCConfigEntry 2 }

**lucentCNMefrPVCSERVICEType OBJECT-TYPE**

SYNTAX INTEGER {  
    unicast(1),  
    multicast(2)  
}  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"This variable specifies whether the DLCI is a multicast or PVC (unicast) DLCI. A PVC DLCI is used to connect a specific endpoint to another endpoint, whereas a multicast DLCI allows an endpoint to transmit single frames to this Frame Relay interface and have them delivered to multiple destinations."  
::= { lucentCNMefrPVCConfigEntry 3 }

**lucentCNMefrLocalCIR OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"This variable specifies the Committed Information Rate (CIR) for data transmitted on this PVC from the local device through the network via this Frame Relay interface. The maximum value for the CIR is limited to the line speed of this interface."  
::= { lucentCNMefrPVCConfigEntry 4 }

**lucentCNMefrLocalCommittedBurst OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"This variable specifies the Committed Burst (Bc) for data transmitted on this PVC from the local device through the network via this Frame Relay interface."  
::= { lucentCNMefrPVCConfigEntry 5 }

**lucentCNMefrLocalExcessBurst OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "This variable specifies the Excess Burst (Be)  
    for data transmitted on this PVC from the local  
    device through the network via this Frame Relay  
    interface."  
 ::= { lucentCNMefrPVCConfigEntry 6 }

**lucentCNMefrRemoteCIR OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "This variable specifies the Committed Information  
    Rate (CIR) for data transmitted on this PVC from  
    the remote device, through the network, to the local  
    device via this Frame Relay interface. The maximum  
    value for the CIR is limited to the line speed of  
    this interface."  
 ::= { lucentCNMefrPVCConfigEntry 7 }

**lucentCNMefrRemoteCommittedBurst OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "This variable specifies the Committed Burst (Bc)  
    for data transmitted on this PVC from the remote  
    device, through the network, to the local device  
    via this Frame Relay interface."  
 ::= { lucentCNMefrPVCConfigEntry 8 }

**lucentCNMefrRemoteExcessBurst OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "This variable specifies the Excess Burst (Be)  
    for data transmitted on this PVC from the remote  
    device, through the network, to the local device  
    via this Frame Relay interface."  
 ::= { lucentCNMefrPVCConfigEntry 9 }

**lucentCNMefrMulticastGroup1 OBJECT-TYPE**

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory

DESCRIPTION

"A multicast (group) DLCI that includes this DLCI as one of its members. Any messages addressed to this multicast DLCI will be sent to this PVC and to other PVCs that belong in this group.

A unicast DLCI can be a member of at most 4 groups. This variable identifies the first multicast DLCI that includes this DLCI as its member."

::= { lucentCNMefrPVCConfigEntry 10 }

**lucentCNMefrMulticastGroup2 OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A multicast (group) DLCI that includes this DLCI as one of its members. Any messages addressed to this multicast DLCI will be sent to this PVC and to other PVCs that belong in this group.

A unicast DLCI can be a member of at most 4 groups. This variable identifies the second multicast DLCI that includes this DLCI as its member."

::= { lucentCNMefrPVCConfigEntry 11 }

**lucentCNMefrMulticastGroup3 OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A multicast (group) DLCI that includes this DLCI as one of its members. Any messages addressed to this multicast DLCI will be sent to this PVC and to other PVCs that belong in this group.

A unicast DLCI can be a member of at most 4 groups. This variable identifies the third multicast DLCI that includes this DLCI as its member."

::= { lucentCNMefrPVCConfigEntry 12 }

**lucentCNMefrMulticastGroup4 OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A multicast (group) DLCI that includes this

DLCI as one of its members. Any messages addressed to this multicast DLCI will be sent to this PVC and to other PVCs that belong in this group.

A unicast DLCI can be a member of at most 4 groups. This variable identifies the fourth multicast DLCI that includes this DLCI as its member."

::= { lucentCNMefrPVConfigEntry 13 }

**lucentCNMefrPVCMeasMaxIntervals OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable identifies the maximum number of measurement intervals supported for the measurements maintained by this Frame Relay interface in the lucentCNMefrPVCMeasTable."

::= { lucentCNMefrPVConfigEntry 14 }

**lucentCNMefrPVCMeasIntervalLen OBJECT-TYPE**

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable identifies the number of seconds that make up one complete measurement interval for the measurements maintained by this Frame Relay interface in the lucentCNMefrPVCMeasTable."

::= { lucentCNMefrPVConfigEntry 15 }

**Frame Relay PVC Level Measurements Table**

This table provides counts on the various PVC-level measurement counts maintained by a Frame Relay interface during the specified measurement interval.

**lucentCNMefrPVCMeasTable OBJECT-TYPE**

SYNTAX SEQUENCE OF LucentCNMefrPVCMeasEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A list of entries containing PVC-level measurement counts, maintained during the specified measurement interval, for all the Frame Relay interfaces managed by this system."

::= { lucent-cnm-efr 4 }

**lucentCNMefrPVCMeasEntry OBJECT-TYPE**

SYNTAX LucentCNMefrPVCMeasEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"An entry containing measurement counts, maintained during the specified measurement interval, for a particular PVC on a specific Frame Relay interface."

INDEX { lucentCNMefrPVCMeasIfIndex,  
          lucentCNMefrPVCMeasIndex,  
          lucentCNMefrPVCMeasInterval }

::= { lucentCNMefrPVCMeasTable 1 }

**LucentCNMefrPVCMeasEntry ::=**

SEQUENCE {  
    lucentCNMefrPVCMeasIfIndex  
        INTEGER,  
    lucentCNMefrPVCMeasIndex  
        INTEGER,  
    lucentCNMefrPVCMeasInterval  
        INTEGER,  
    lucentCNMefrPVCMeasTimeStamp  
        INTEGER,  
    lucentCNMefrPVCMeasLocalTime  
        DisplayString,  
    lucentCNMefrPVCReceivedFrames  
        Gauge,  
    lucentCNMefrPVCSentFrames  
        Gauge,  
    lucentCNMefrDiscardEligibilityFrames  
        Gauge,  
    lucentCNMefrBurstSizeExceeded  
        Gauge,  
    lucentCNMefrCongestionAtIngress  
        Gauge,  
    lucentCNMefrCongestionAtEgress  
        Gauge,  
    lucentCNMefrPVCReceivedBytes  
        Gauge,  
    lucentCNMefrPVCSentBytes  
        Gauge,  
    lucentCNMefrRecvDiscardEligFrames  
        Gauge,  
    lucentCNMefrSentDiscardEligFrames  
        Gauge,  
    lucentCNMefrRecvForwardNoticFrames  
        Gauge,  
    lucentCNMefrSentForwardNoticFrames  
        Gauge,  
    lucentCNMefrRecvBackwardNoticFrames

```
        Gauge,  
    lucentCNMefrSentBackwardNoticFrames  
        Gauge,  
    lucentCNMefrTotalRejectsSentRec  
        Gauge,  
    lucentCNMefrPVCExpectedState  
        INTEGER,  
    lucentCNMefrPVCOperationalState  
        INTEGER,  
    lucentCNMefrPVCCallStatus  
        INTEGER,  
    lucentCNMefrPVCLinkStatus  
        INTEGER  
}
```

**lucentCNMefrPVCMeasIfIndex OBJECT-TYPE**

```
SYNTAX  INTEGER  
ACCESS  read-only  
STATUS  mandatory  
DESCRIPTION  
    "A unique value for each Frame Relay interface.  
    The interface identified by a particular value of  
    this index is the same interface as identified by  
    the same value of an lucentCNMifConfigIndex object  
    instance."  
 ::= { lucentCNMefrPVCMeasEntry 1 }
```

**lucentCNMefrPVCMeasIndex OBJECT-TYPE**

```
SYNTAX  INTEGER  
ACCESS  read-only  
STATUS  mandatory  
DESCRIPTION  
    "The DLCI number on a Frame Relay interface that  
    identifies a unique Data Link Connection."  
 ::= { lucentCNMefrPVCMeasEntry 2 }
```

**lucentCNMefrPVCMeasInterval OBJECT-TYPE**

```
SYNTAX  INTEGER  
ACCESS  read-only  
STATUS  mandatory  
DESCRIPTION  
    "This variable identifies the measurement interval  
    number for which measurement is provided. It is a  
    number between 1 and XX, where 1 identifies the most  
    recently completed measurement interval and XX is  
    the least recently completed measurement interval."  
 ::= { lucentCNMefrPVCMeasEntry 3 }
```

**lucentCNMefrPVCMeasTimeStamp OBJECT-TYPE**

```
SYNTAX  INTEGER
```

ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The time stamp corresponding to the end of the specified measurement interval, as measured in seconds from 00:00:00 UTC (Coordinated Universal Time) January 1, 1970. Any fraction is rounded up."  
::= { lucentCNMefrPVCMeasEntry 4 }

**lucentCNMefrPVCMeasLocalTime OBJECT-TYPE**

SYNTAX DisplayString(SIZE(0..255))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The time stamp corresponding to the end of the specified measurement interval. Any fraction is rounded up. It is given as a printable ASCII string showing the local time at the end of the interval."  
::= { lucentCNMefrPVCMeasEntry 5 }

**lucentCNMefrPVCReceivedFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of frames received by this Frame Relay PVC during the specified measurement interval."  
::= { lucentCNMefrPVCMeasEntry 6 }

**lucentCNMefrPVCSentFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of frames sent by this Frame Relay PVC during the specified measurement interval."  
::= { lucentCNMefrPVCMeasEntry 7 }

**lucentCNMefrDiscardEligibilityFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of ingress frames received by this PVC that exceed the Committed Burst Size but don't exceed the maximum throughput configured for the PVC (the Committed Burst Size plus the Excess Burst Size).

In addition, all ingress frames with the Discard Eligible (DE) Indicator bit set are not included in this count."

::= { lucentCNMefrPVCMeasEntry 8 }

**lucentCNMefrBurstSizeExceeded OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of frames that were discarded due to exceeding the maximum throughput configured for the PVC (the Committed Burst Size plus the Excess Burst Size). If the Committed Information Rate (CIR) has not been configured for the PVC, then this field is not applicable."

::= { lucentCNMefrPVCMeasEntry 9 }

**lucentCNMefrCongestionAtIngress OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of times frames received by this PVC had to be discarded because buffers were unavailable or congestion control was being enforced during the specified measurement interval."

::= { lucentCNMefrPVCMeasEntry 10 }

**lucentCNMefrCongestionAtEgress OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of times frames sent by this PVC had to be discarded because buffers were unavailable or congestion control was being enforced during the specified measurement interval."

::= { lucentCNMefrPVCMeasEntry 11 }

**lucentCNMefrPVCReceivedBytes OBJECT-TYPE**

SYNTAX Gauge

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The counter associated with the number of bytes received from the line."

```
 ::= { lucentCNMefrPVCMeasEntry 12 }
```

**lucentCNMefrPVCSentBytes OBJECT-TYPE**

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

```
    "The counter associated with the number of bytes
    transmitted to the line."
```

```
 ::= { lucentCNMefrPVCMeasEntry 13 }
```

**lucentCNMefrRecvDiscardEligFrames OBJECT-TYPE**

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

```
    "The counter associated with the number of frames
    received from the line that have the DE (Discard
    Eligibility) bit set."
```

```
 ::= { lucentCNMefrPVCMeasEntry 14 }
```

**lucentCNMefrSentDiscardEligFrames OBJECT-TYPE**

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

```
    "The counter associated with the number of frames
    transmitted to the line that have the DE (Discard
    Eligibility) bit set."
```

```
 ::= { lucentCNMefrPVCMeasEntry 15 }
```

**lucentCNMefrRecvForwardNoticFrames OBJECT-TYPE**

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

```
    "The counter associated with the number of frames
    received from the line that have the FECN (Forward
    Notification Bit for Congestion Control) set."
```

```
 ::= { lucentCNMefrPVCMeasEntry 16 }
```

**lucentCNMefrSentForwardNoticFrames OBJECT-TYPE**

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

```
    "The counter associated with the number of frames
    trasmitted to the line that have the FECN (Forward
    Notification Bit for Congestion Control) set."
```

```
 ::= { lucentCNMefrPVCMeasEntry 17 }
```

**lucentCNMefrRecvBackwardNoticFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of frames  
    received from the line that have the BECN (Backward  
    Notification Bit for Congestion Control) set."  
 ::= { lucentCNMefrPVCMeasEntry 18 }

**lucentCNMefrSentBackwardNoticFrames OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the number of frames  
    trasmitted to the line that have the BECN (Backward  
    Notification Bit for Congestion Control) set."  
 ::= { lucentCNMefrPVCMeasEntry 19 }

**lucentCNMefrTotalRejectsSentRec OBJECT-TYPE**

SYNTAX Gauge  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The counter associated with the sum of rejects  
    received and rejects sent."  
 ::= { lucentCNMefrPVCMeasEntry 20 }

**lucentCNMefrPVCExpectedState OBJECT-TYPE**

SYNTAX INTEGER {  
    in(1),  
    out(2),  
    unknown(4)  
}  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The expected or administrative service state  
    of the DLCI (in, out)."  
 ::= { lucentCNMefrPVCMeasEntry 21 }

**lucentCNMefrPVCOperationalState OBJECT-TYPE**

SYNTAX INTEGER {  
    in(1),  
    out(2),  
    rfs(3),  
    unknown(4)  
}

```

ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The actual service state of the DLCI (up, down or
    rfs - ready for service)."
```

::= { lucentCNMefrPVCMeasEntry 22 }

**lucentCNMefrPVCCallStatus OBJECT-TYPE**

```

SYNTAX INTEGER {
    up(1),
    down(2),
    unknown(4)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Indicates the state that the DLCI thinks the
    call is in (up, down)."
```

::= { lucentCNMefrPVCMeasEntry 23 }

**lucentCNMefrPVCLinkStatus OBJECT-TYPE**

```

SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "This variable indicates the various states the
    link was in during the given measurement interval.
    This object is a bit map represented as a sum,
    therefore it can represent multiple states
    simultaneously. The various bit positions are:
```

0	normal
1	local DLCI receive problem
2	local DLCI send problem
3	local DLCI receive and send problem
4	remote dlci problem
5	local receive and remote dlci problem
6	local send and remote dlci problem
7	local receive, send and remote dlci problem
8	call is not up

```

For example, for an interface that has a receive
and send problem, this object will have a value
of 3."
 ::= { lucentCNMefrPVCMeasEntry 24 }
```

## Frame Relay PVC Level Status Table

This table provides status information on Data Link Connections on all Frame Relay interfaces supported by this system.

### lucentCNMefrPVCStatusTable OBJECT-TYPE

```
SYNTAX SEQUENCE OF LucentCNMefrPVCStatusEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A list of entries containing status information
    about Data Link Connections on Frame Relay interfaces
    managed by this system."
 ::= { lucent-cnm-efr 5 }
```

### lucentCNMefrPVCStatusEntry OBJECT-TYPE

```
SYNTAX LucentCNMefrPVCStatusEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "An entry containing status information about
    a particular Data Link Connection on a specific
    Frame Relay interface."
INDEX { lucentCNMefrPVCStatusIfIndex,
        lucentCNMefrPVCStatusIndex }
 ::= { lucentCNMefrPVCStatusTable 1 }
```

### LucentCNMefrPVCStatusEntry ::=

```
SEQUENCE {
    lucentCNMefrPVCStatusIfIndex
        INTEGER,
    lucentCNMefrPVCStatusIndex
        INTEGER,
    lucentCNMefrPVCAdminStatus
        INTEGER,
    lucentCNMefrPVCOperStatus
        INTEGER
}
```

### lucentCNMefrPVCStatusIfIndex OBJECT-TYPE

```
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "A unique value for each Frame Relay interface.
    The interface identified by a particular value of
    this index is the same interface as identified by
    the same value of an lucentCNMifConfigIndex object
    instance."
 ::= { lucentCNMefrPVCStatusEntry 1 }
```

**lucentCNMefrPVCStatusIndex OBJECT-TYPE**

```
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The DLCI number on a Frame Relay interface that
    identifies a unique Data Link Connection."
 ::= { lucentCNMefrPVCStatusEntry 2 }
```

**lucentCNMefrPVCAdminStatus OBJECT-TYPE**

```
SYNTAX INTEGER {
    up(1),
    down(2),
    testing(3),
    unknown(4)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The desired state of this Data Link Connection."
 ::= { lucentCNMefrPVCStatusEntry 3 }
```

**lucentCNMefrPVCOperStatus OBJECT-TYPE**

```
SYNTAX INTEGER {
    up(1),
    down(2),
    testing(3),
    unknown(4)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The current operational state of this Data Link
    Connection."
 ::= { lucentCNMefrPVCStatusEntry 4 }
```

**END**



---

## Database Schema

# D

---

This appendix describes the fields and tables in the SNMP Proxy Agent database.

### dis\_logs

This table stores SMDS disagreement log information.

Column	Data Type	Length	Description
ifindex	integer	4	Interface Index
srcaddr	char	16	SMDS source address
dstaddr	char	16	SMDS destination address
logtime	integer	4	Time when disagreement occurred
dislogtype	integer	4	Type of disagreement
update	integer	4	Time when disagreement is recorded in database

### grpmem

This table stores the group members and the member addresses and Customer Views that are associated to the group members. This table is populated by **sksnmpdbsync**.

Column	Data Type	Length	Description
grp_addr	char	11	SMDS group address

Column	Data Type	Length	Description
mem_addr	char	11	SMDS individual address
mibview	char	15	Customer View

## **grpscr**

---

This table stores the SMDS group screening addresses for SMDS interfaces. This table is populated by **sksnmpdbsync**.

Column	Data Type	Length	Description
ifindex	integer	4	Interface Index
grp_addr	char	11	SMDS group address

## **ifdlci**

---

This table stores a list of all the DLCI numbers associated with an *ifindex*. The *ifindex* must be a Frame Relay module port. This table is populated when you execute the **sksnmpndsync** command.

Column	Data Type	Length	Description
ifindex	smallint	2	Interface Index
ifdlci	smallint	2	DLCI number

## **ifstatus**

---

This table stores ifindex status information.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ifindex	integer	4	Interface Index
ifdci	smallint	2	DLCI number
exp_state	smallint	2	Expected state
oper_state	smallint	2	Operating state
update	integer	4	Time when state was updated

## **iftype**

---

This table stores a list of all the interface types supported by the SNMP Proxy Agent. This table is populated during the initialization of the SNMP database.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
iftypeid	smallint	2	Integer value of interface type
iftype	char	10	Interface type

## **ifindex**

---

This table stores a list of all the Interface Indices configured by the administrator. This table is modified when you use the Network Interface Administration form.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ifindex	integer	4	Interface index [1-999999]
nodename	char	50	Node Name
module	smallint	2	Module (slot) number
link	smallint	2	Slot number of Concentrator Trunk module
port	smallint	2	Port number
vport	smallint	2	Virtual port number
iftypeid	smallint	2	Interface Type id
ifmtu	integer	4	Interface MTU
ifspeed	decimal	16	Interface Speed
ifcontact	char	30	Interface Contact
iflocation	char	35	Interface Location
ifdescr	char	40	Interface Description
ifsubscr	char	30	Interface Subscriber
ifacc_class	smallint	2	Access Class of Interface

## **indaddr**

---

This table stores SMDS individual addresses for SMDS interfaces. This table is populated by **sksnmpdbsync**.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ind_addr	char	11	Individual address
ifindex	integer	4	Interface Index

## **indscr**

---

This table stores SMDS individual screening addresses for SMDS interfaces. This table is populated by **sksnmpdbsync**.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ifindex	integer	4	Interface Index
ind_addr	char	11	Individual address

## **ip**

---

This table stores a list of all the Subscriber Identifiers configured by the administrator. This table is modified when you use the Subscriber Information Administration form.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
subscrid	char	15	Subscriber Identifier
ipaddr	char	15	IP Address
rdcomstr	char	15	READ Community String
wrcomstr	char	15	WRITE Community String
trcomstr	char	15	TRAP Community String
rdmibvw	char	15	READ Customer View
wrmibvw	char	15	WRITE Customer View
trmibvw	char	15	TRAP Customer View
authtrap	char	1	Receive authentication traps
ipdescr	char	40	Description of Subscriber
cuscontact	char	30	Subscriber contact
bccnms	smallint	2	Is this a CNM Server?
servestate	smallint	2	Service state

## **ip\_throt**

---

This table stores the node usage count of all requests made to a node by a Subscriber.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ipaddr	char	15	IP address of Subscriber
nodename	char	50	Name of node
allow_attempts	smallint	2	Number of requests allowed
curr_tries	smallint	2	Number of requests made
update_time	integer	4	Time when request was made

## **memgrp**

---

This table stores SMDS individual/group members for SMDS interfaces. This table is populated by **sksnmpdbsync**.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
mem_addr	char	11	Individual address
grp_addr	char	11	Group address
mibview	char	15	Customer View address

## **mibview**

---

This table stores a list of all the Customer Views created by the administrator. This table is modified when you use the Customer View Information form.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
mibview	char	15	Customer View name
viewdescr	char	50	Description of Customer View

## **mibviewmem**

This table stores a list of all the Customer Views and their members. Customer View members are interface indices. This table is modified when you use the Customer View Member form.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
mibview	char	15	Customer View name
ifindex	integer	4	Interface Index

## **node\_throt**

This table stores the usage count that all subscribers made to a node.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
nodename	char	50	Name of node
allow_attempts	smallint	2	Number of requests allowed
curr_tries	smallint	2	Number of requests made
update_time	integer	4	Time when request was made

## **pe\_logs**

This table stores SMDS protocol error log information.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ifindex	integer	4	Interface Index
srcaddr	char	16	SMDS source address
dstaddr	char	16	SMDS destination address
logtime	integer	4	Time when protocol error occurred
pelogtype	integer	4	Type of protocol error
update	integer	4	Time when protocol error is recorded in database

**scr\_log**

---

This table stores the results from processing each submitted Set Request. This table is populated by **sksnmpupd**.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ifindex	integer	4	Interface Index
scr_addr	char	11	SMDS group or individual screening address
ip_addr	char	15	IP address of SNMP Manager that submitted request
action	smallint	2	Add (1) or delete (2)
result	smallint	2	Result of update
reason	smallint	2	Reason if result is a failure
ticks	integer	4	Time in UNIX seconds when request was updated

**scr\_reqts**

---

This table stores submitted Set Requests for SMDS individual and group screening addresses. These Set Requests are processed by **sksnmpupd**.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ifindex	integer	4	Interface Index
scr_addr	char	11	SMDS screening address to be added or deleted
ip_addr	char	15	IP address of SNMP Manager that submitted request
action	smallint	2	Add (1) or delete (2)
status	smallint	2	Pending (1) or failed (2)
reason	smallint	2	Reason submitted request did not get processed
addr_type	smallint	2	Group (1) or individual (2) screening address
retry_count	smallint	2	Number of retries made to process this request
lock	smallint	2	Lock off (0) or lock on (1). Used for updates.
ticks	integer	4	Time in UNIX seconds when request was submitted

**snicfg**

---

This table stores SMDS configuration for SMDS interfaces. This table is populated by **sksnmpdbsync**.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ifindex	integer	4	Interface Index
mcd_u_ingress	smallint	2	MCDU on ingress
mcd_u_egress	smallint	2	MCDU on egress
ind_allow_flag	char	1	Individual screen allow flag
grp_allow_flag	char	1	Group screen allow flag

**snmpcounts**

---

The table stores all the SNMP counts related to an IP address.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
ipaddr	char	15	IP Address
in_pkts	integer	4	Received packets
bad_addrs	integer	4	Bad IP address
bad_versions	integer	4	Bad value for SNMP version number
bad_comm_names	integer	4	Bad community string
bad_comm_uses	integer	4	Wrong use of community string
parse_errs	integer	4	Packet could not be decoded
in_tot_req_vars	integer	4	Number of requested objects
in_tot_set_vars	integer	4	Number of set objects received
in_set_reqs	integer	4	Number of set requests received
in_get_reqs	integer	4	Number of get requests received
in_get_nexts	integer	4	Number of get-next requests received
in_get_resp	integer	4	Number of get responses received
in_traps	integer	4	Number of traps received
out_pkts	integer	4	Number of packets sent
too_bigs	integer	4	Response too big to send
no_such_names	integer	4	Object not none
bad_values	integer	4	Not supported
gen_errs	integer	4	Other general errors
out_get_responses	integer	4	Number of get responses sent
out_traps	integer	4	Number of traps sent

**snmpsys**

---

This table stores the system information that describes the SNMP Proxy Agent service. This table is modified when you use the System Information Administration form.

<b>Column</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>
sysname	char	60	Name of system
sysdesc	char	80	System description
syscontact	char	60	System Contact
syslocation	char	60	System location
sysobjid	char	60	System Object Identifier
inittime	integer	4	Time SNMP Agent started
syservices	integer	4	Services offered by SNMP Agent

---

# Glossary

The definitions in this section appear in alphabetical order. Cross references in the entries are printed in bold type.

---

## A

### agent

The **SNMP** agent is software that translates requests from SNMP Manager stations into requests that network management systems (NMS) and switch nodes can understand. The agent also translates the retrieved data back to SNMP format. The end customers use the data sent back to the SNMP Manager to monitor the network end of their system.

---

## B

### BNS-2000

Lucent Technologies data switch product supporting Frame Relay and SMDS and other data services.

---

## C

### CNM

Customer Network Management. Refers to all the activities that customers perform to manage their communications networks.

### community string

A password. Each SNMP Manager has two community strings. The Read Community String is used by the SNMP Manager when issuing SNMP requests. The Trap Community String is used by the SNMP Proxy Agent to send traps to the SNMP Manager.

### Core System

The *StarKeeper II* NMS Core System processor. A processor equipped with *StarKeeper II* NMS Core System

processes. This processor may not contain any workstation software.

### Customer View

A set of Interface Indices that an SNMP Manager can manage. A Read Customer View defines the set of network interfaces that an SNMP Manager manages. A Trap Customer View defines the set of interfaces that an SNMP Manager receives traps for.

---

## H

### HP VUE

Hewlett-Packard Visual User Interface.

---

## I

### Interface Index

A unique numeric identifier assigned to each network interface that the SNMP Proxy Agent supports. An Interface Index can be any number from 1 to 999999.

### Internet Protocol (IP)

The network **protocol** that supplies a connectionless-mode network service in the **Internet** suite of protocols.

---

## L

### LAN

Local Area Network. Any of one of a number of technologies that provides high-speed, low-latency data transfer over a limited geographic area.

---

## M

### Manager

The system responsible for managing the end-user side of a network via SNMP. The manager controls the agent system by executing requests for information, setting variables, and asynchronously receiving SNMP traps.

### MIB

Management Information Base. A collection of objects that can be accessed via a network management protocol. The rules used to define MIBs are found in the Structure of Management Information (SMI).

---

## N

### Network Builder

An optional graphics-based *StarKeeper II* NMS application package used for configuration management and analysis. The application provides a Forms Interface to configure nodes, groups, trunks, and Session Maintenance.

---

## O

### objects

Part of a **MIB**. The MIB is a collection of objects that specify an entity such as a port for management purposes. Within the MIB structure, the objects are like leaves in a tree. The path from the root through the hierarchical tree structure identifies the object.

---

## P

### PDU

Protocol Data Unit. A data object exchanged by **protocol** machines, usually containing both protocol control information and user data.

### private MIB

These **MIBs**, also known as enterprise-specific MIBs, allow vendors to identify network management objects that are unique to their networking subsystems. Vendors can register their models, MIBs, and products to be shared.

### proxy

The **SNMP** proxy agent is a technique where one machine, the proxy, acts on behalf of a second machine by translating SNMP requests from the SNMP Manager into whatever interactions are needed to manage or monitor the second non-SNMP machine.

### protocol

A strict procedure for the initiation and the maintenance of data communications.

---

## R

### RFC

Request for Comments. The document series describing the Internet suite of **protocols** and related experiments. RFCs describe proposed and accepted **Internet protocol** standards.

---

## S

### SMDS

Switched Multimegabit Data Service.

### SNI

Subscriber Network Interface.

### SNMP

Simple Network Management Protocol. The application protocol that offers network management service in the **Internet** suite of protocols.

### SNMP Manager

See **Manager**.

### subscriber

A customer who subscribes to SMDS or Frame Relay services. Each SNMP Manager is registered in the SNMP database as a subscriber and is identified by a Subscriber Identifier and a set of Customer Views and community strings.

---

## T

### TCP

Transmission Control Protocol. The transport protocol that provides connection-oriented transport service in the Internet suite of protocols.

### TCP/IP

Refers to the Internet suite of protocols. A collection of computer-communication protocols originally developed under the Defense Advanced Research Agency (DARPA) sponsorship. The Internet suite of protocols is currently the de facto solution for open networking.

### trap

An SNMP trap **PDU** is an unsolicited message from an SNMP agent to an SNMP manager indicating that an unusual event has occurred. These traps function as alarms and can be defined by the SNMP manager.

---

## U

### UDP

User Datagram Protocol. The transport protocol offering a connectionless-mode transport service in the Internet suite of protocols.



---

# Index

---

## Symbols

`$SNMP_LOG/snmp_agent.log`, 2-4, 4-2

---

## A

Administrative Tasks, 2-1–2-6  
ASCII menu options  
  Customer View Members, 3-49  
  Customer Views, 3-46  
  Flow Control and Usage, 3-58  
  Network Interfaces, 3-40  
  Subscriber Information, 3-53  
  System Information, 3-38  
ASCII user interface, 3-35–3-46  
  cursor movement, 3-37  
  displaying Main Menu, 3-36  
  on-line help, 3-38

---

## B

Backing up the SNMP Database, 2-4

---

## C

CNM Service  
  provisioning for, 2-5  
`cnmsadm`  
  crontab for, A-24  
  logging in, 2-2  
Customer Profile Worksheet, 3-4, A-27  
Customer View Form, A-30

---

## D

Daemon Processes  
  monitoring, 4-1  
  `sksnmpd`, 4-1  
  `sksnmpd`, 1-8  
  `sksnmpps`, 4-1  
  `sksnmpps`, 1-8  
  `sksnmpt`, 4-1  
  `sksnmpt`, 1-9

## Database

SNMP database, 1-6  
StarKeeper II NMS Core System database, 1-6

---

## F

### Field descriptions

CNM Server, 3-25, 3-55  
Community String (READ), 3-24, 3-54  
Community String (TRAP), 3-24, 3-54  
Community String (WRITE), 3-24, 3-54  
Conc., 3-14  
Concentrator trunk module address, 3-42  
Contact, 3-12, 3-15  
Customer View, 3-19, 3-46, 3-49  
Customer View (READ), 3-24, 3-54  
Customer View (TRAP), 3-24, 3-55  
Customer View (WRITE), 3-24, 3-54  
Customer View Description, 3-20, 3-47  
Customer View Members, 3-50  
Description, 3-12, 3-15, 3-50  
Interface description, 3-43  
Interface Index, 3-14, 3-41  
Interface location, 3-43  
Interface provider contact, 3-43  
Interface subscriber, 3-43  
Interface Type, 3-42  
IP Address, 3-23, 3-54  
Location, 3-12, 3-15  
Mod Type, 3-14  
Mod., 3-14  
MTU, 3-15  
Node Address, 3-14  
Node Name, 3-41  
Node/Concentrator module address, 3-42  
Nodename, 3-15  
Object Id, 3-12  
Port number, 3-14, 3-42  
Receive Authentication Traps, 3-25, 3-55  
Requests Allowed, 3-28, 3-29  
Requests Made, 3-28, 3-29  
Service State, 3-25, 3-55  
Speed, 3-15  
Subscriber, 3-15  
Subscriber Contact, 3-24, 3-54  
Subscriber Description, 3-23, 3-54  
Subscriber Identifier, 3-23, 3-53  
System Contact, 3-39  
System Description, 3-39  
System Location, 3-39  
System Name, 3-12, 3-39  
System Object Id, 3-39  
Virtual port number, 3-42  
Vport, 3-14

### Forms

See also *Planning forms*

---

## G

Get Request PDU, 1-4  
Get Response PDU, 1-4  
Get-Next Request PDU, 1-4  
Glossary, GL-1–GL-3

---

## H

Help  
  for commands, A-7  
HP VUE, 3-7, 3-10

---

## I

**ifconfig** command, 4-3  
Initializing the SNMP Database, 2-4  
Installing the SNMP Proxy Agent, 2-2  
Internet Protocol, 1-1

---

## L

Licensing requirements, 2-1  
Log file  
  \$SNMP\_LOG/snmp\_set.log, A-20  
  cleaning up, 2-6  
  displaying \$SNMP\_LOG/snmp\_agent.log, 2-4  
  displaying \$SNMP\_LOG/snmp\_agent.log, 4-2  
  displaying \$SNMP\_LOG/snmp\_set.log, 2-5  
Logging In, 2-2

---

## M

MASTerview, 3-5  
Menu Interface  
  starting, A-4, A-23  
MIB files  
  for Frame Relay interfaces, 5-2  
  for SMDS interfaces, 5-1  
MIBs  
  supported, C-1  
Motif menu options  
  Customer Views, 3-8, 3-18  
  Exit, 3-9  
  Flow Control, 3-8, 3-27  
  Help, 3-8

---

Network Interfaces, 3-8, 3-13  
  Provisioning, 3-8  
  Report Generation, 3-8, 3-30  
  Subscriber, 3-8, 3-22  
  System Information, 3-8, 3-11  
  Usage Counts, 3-8  
Motif user interface, 3-7–3-35  
  accessing, 3-7  
  Cursor Movement, 3-10  
  help facility, 3-11

---

## N

**netstat** command, 4-3  
Network Interface Administration Form, 3-13, 3-14  
Network Interfaces  
  administering, 3-13  
Network Interfaces Form, A-29

---

## O

On-line Help  
  for ASCII user interface, 3-38  
  for commands, A-7  
  from MOTIF user interface, 3-11

---

## P

**ping** command, 4-2  
Planning forms  
  Customer Profile Worksheet, A-27  
  Customer Profile Worksheet sample, 3-4  
  Customer View Form, A-30  
  Network Interfaces Form, A-29  
  Subscriber Information Form, A-31

---

## R

Read Community String, 3-2  
Read Customer View, 3-3  
Reports, 3-60  
  Node Usage Report, 3-30  
  SNMP Packets by Subscriber Report, 3-32  
  SNMP PDU Errors by Subscriber Report, 3-34  
  Subscriber Usage of Nodes Report, 3-31  
Restoring the SNMP Database, 2-4

---

---

## S

Set Request, 2-5, 3-2, A-1, A-13, A-20  
   executing, A-20  
 Set Request PDU, 1-4  
 skload command, 2-5  
**skload** command, 3-16  
**sksnmpbdb** command, A-3  
 sksnmpcf command, 2-5  
**sksnmpcf** command, 3-36, A-4  
**sksnmpdbsync** command, 3-6, A-5  
 sksnmpget command, 2-3  
**sksnmpget** command, A-6  
**sksnmphelp** command, A-7  
**sksnmpidb** command, A-8  
**sksnmpndsync** command, 3-5, A-9  
**sksnmpnext** command, A-11  
**sksnmpnext** command, 2-3  
**sksnmpbdb** command, A-12  
 sksnmpset command, 2-3  
**sksnmpset** command, A-13  
**sksnmpstart** command, 2-3, A-15  
**sksnmpstat** command, 2-2, A-16  
**sksnmpstop** command, 2-3, A-17  
**sksnmptrace** command, A-18  
**sksnmptrap** command, A-19  
 sksnmpupd command, 2-5  
**sksnmpupd** command, A-20  
**sksnmpwalk** command, A-22  
**sksnmpwalk** command, 2-3  
**sksnmpxadm** command, 3-7  
 sksnmpxadm command, 2-5  
**sksnmpxadm** command, 3-7, A-23  
 SNMP Database, 1-6  
   backing up, 2-4, A-3  
   configuring, 2-3  
   description of contents, 1-6  
   initializing, 2-4, A-8  
   reinitializing, A-8  
   restoring, 2-4, A-12  
   synchronizing, 2-3, 3-5  
   System Information, 3-2  
   tables, D-1–D-12  
 SNMP Get Requests, 3-2  
 SNMP Get-Next Requests, 3-2  
 SNMP Manager  
   prerequisites for accessing SNMP Proxy Agent, 5-1  
 SNMP Proxy Agent  
   administration and maintenance tasks, 2-1–2-5  
   Architecture, 1-8  
   commands, A-1–A-22  
   features, 1-2–1-4  
   installation, 2-2  
   prerequisites for installation, 2-1  
   software processes, 1-6–1-8  
   starting, 2-2, A-15

  stopping, 2-3, A-17  
   supported interfaces, 1-4  
   testing, 2-3  
   troubleshooting, 2-4, 4-1–4-6  
   verifying status, A-16  
 SNMP requests  
   processing of, 1-6  
   restricting number of queries, 3-6  
 SNMP Set Requests, 3-2  
 SNMP traffic statistics, 3-6  
 snmpdb  
   See *SNMP Database*  
*StarKeeper II NMS Core System Database*, 1-6  
 Starting the SNMP Proxy Agent, 2-2  
 Stopping the SNMP Proxy Agent, 2-3  
 Subscriber  
   Read Customer View Dialog, 3-26  
   Subscriber Information Form, A-31  
 System Information  
   administering, 3-11  
 System Information Administration Form, 3-12

---

## T

TCP/IP Subsystem  
   troubleshooting, 4-2  
 Testing the SNMP Proxy Agent, 2-3  
 Trace Facility  
   **sksnmptrace** command, A-18  
   turning on and off, A-18  
   using, 4-2  
 Trap Community String, 3-3  
 Trap Customer View, 3-3  
 Trap PDU, 1-4  
 Trap PDUs, A-19  
 Troubleshooting, 2-4, 4-1–4-6

---

## U

User Datagram Protocol, 1-1  
 User interface  
   ASCII, 3-35–3-46  
   Motif, 3-7–3-35

---

## W

Write Community String, 3-2  
 Write Community View, 3-3

