SNMP Agent Software for Windows Operating Systems

Administration Guide

January 2003

05-1104-007

Publication Date: January 2003

Document Number: 05-1104-007

Intel Converged Communications
1515 Route 10
Parsippany, NJ 07054

For **Technical Support**, visit the Intel Telecom Support Resources website at:
http://developer.intel.com/design/telecom/support/

For **Products and Services Information**, visit the Intel Communications Systems Products website at:
http://www.intel.com/network/csp/

For **Sales Offices** and other contact information, visit the Intel Telecom Building Blocks Sales Offices page at:
http://www.intel.com/network/csp/sales/

Contents

Tables

About This Publication

The following topics provide information about this publication:

- Purpose
- Intended Audience
- How to Use This Publication
- Related Information

Purpose

This publication provides guidelines for using the SNMP agent software, which provides remote (or local) monitoring and limited control of Intel® Dialogic® boards and Intel® NetStructure™ boards over an Internet protocol (IP) network. **This publication was prepared for use with Intel® Dialogic® System Release Version 5.1.1 Feature Pack 1 on PCI and CompactPCI for Microsoft* Windows NT/2000/XP on Intel® Architecture.**

Intended Audience

This publication is written for the following audience:

- System Integrators
- Independent Software Vendors (ISVs)
- Original Equipment Manufacturers (OEMs)
- Telephony Equipment Manufacturers (TEMs)
- Network Equipment Providers

How to Use This Publication

This publication assumes that you are familiar with the Windows* operating system and Simple Network Management Protocol (SNMP), a standard IP network mechanism for exchanging management information between an SNMP agent and a network management station.

The information in this guide is organized as follows:

- Chapter 1, "Administration Overview" provides an overview of SNMP, the preparation needed to use the SNMP agent, and the administration tasks that can be performed using the SNMP agent software.

- Chapter 2 , “Stopping and Starting the System” provides information on stopping and starting the Intel® Dialogic® system and stopping and starting Intel Dialogic boards and Intel NetStructure boards.
- Chapter 3 , “Installing and Using the MIBs” provides information about installing the MIBs and using them to monitor DS-1 lines, ISDN interfaces, and the status of configured Intel Dialogic boards and Intel NetStructure boards.
- Chapter 4 , “Performance Monitoring” describes how to use the SNMP agent software to diagnose problems with the managed node or telephony application including traps, errors and lost messages.
- Glossary provides definitions for terms used in this guide.

Related Information

Refer to the following documents for more information about the Intel® Dialogic® system release software:

- The software installation and configuration guide for the Intel® Dialogic® system release provides information on installing and configuring the system release software and system administration tasks.
- For system requirements and other information that may affect the installation and configuration of the system release software, refer to the *Release Guide* and *Release Update*.
- Solutions to many problems and supplementary information can be found in the technical notes and other resources on the Intel® Telecom Support Resources web site at *http://developer.intel.com/design/telecom/support/*.

Administration Overview 1

This chapter provides an overview of SNMP, the preparation needed to use the SNMP agent, and the administration tasks that can be performed using the SNMP agent software.

1.1 Understanding SNMP and the SNMP Agent

The following sections describe how the SNMP agent software works with SNMP:

- What is SNMP?
- The Role of the SNMP Agent
- The Role of the SNMP Manager
- What is a MIB?
- What is a Trap?

1.1.1 What is SNMP?

SNMP stands for Simple Network Management Protocol (SNMP), a standard IP network mechanism for exchanging management information between an SNMP agent and an SNMP manager. SNMP was designed to manage routers, switches and other network devices. SNMP provides an industry standard protocol for fault and monitoring support.

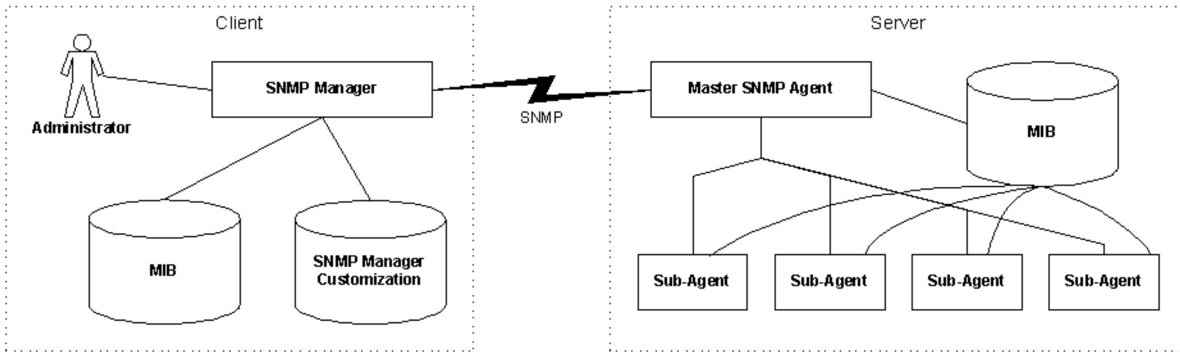
1.1.2 The Role of the SNMP Agent

The SNMP agent provides remote (or local) monitoring and limited control of Intel® Dialogic® boards and Intel® NetStructure™ boards over an IP network. The SNMP agent resides on an IP stack and responds to SNMP "gets" and "sets" from an SNMP manager, and the SNMP agent sends SNMP traps. The objects that an SNMP agent can manipulate are defined in **MIBs** (see What is a MIB?).

The node running the SNMP agent is called the **managed node**. The managed node is the system that contains the Intel Dialogic boards and Intel NetStructure boards you wish to administer remotely. The SNMP agent makes administrative functions available remotely over an IP network.

The **network management station** is the system that has the **SNMP manager** installed. The network management station is the system that exchanges administrative information and tasks with the SNMP agent on the managed node. The network makes this exchange possible and provides a user interface for viewing administrative information and performing management tasks.

Figure 1. General SNMP Architecture



1.1.3 The Role of the SNMP Manager

The SNMP manager is the management application that monitors and/or administers a remote system. The SNMP manager does the following:

- Receives and displays traps
- Sends SNMP “gets” and “sets” to the SNMP agent
- Provides a user interface.

A single SNMP manager can support multiple SNMP agents. For a fully supported network management application, you must use a third-party product such as HP OpenView* or any other SNMP-compliant network management application.

1.1.4 What is a MIB?

A MIB is a Management Information Base that resides on a managed node. The MIB definition sets the limits on what can be managed. The MIB defines

- Which variables or parameters will be accessed
- How each value will be identified
- How each value will be encoded
- How each value will be interpreted

A MIB contains *objects* (units of management information) divided into *scalars* and *tables*, which are identified by object identifiers (OIDs). These objects are exchanged between the managed node and the SNMP manager. Each MIB defines three types of OIDs:

- Read-Only OIDs: This type defines the information that the network management station can read from the managed node.
- Read-Write OIDs: This type defines the system configuration settings on the managed node that can be modified by the network management station.
- Traps: This type defines the information that the managed node sends to the network management station under specified conditions.

MIBs can be either proprietary (or enterprise) MIBs or standard MIBs. SNMP agent software currently supports both proprietary and standard MIBs. SNMP agent software provides the following MIB modules:

- Proprietary Hardware Information MIB Module
- Proprietary ISDN MIB Module
- Proprietary DS-1 MIB Module
- Standard DS-1 MIB Module
- Proprietary DM3 Extended Platform MIB

1.1.5 What is a Trap?

A trap is an unsolicited event sent by the agent to a manager. Related objects and their values (variable bindings) may be sent with a trap. Trap delivery is not guaranteed. Individual MIBs have objects which turn traps on and off.

1.2 Preparing to Use the SNMP Agent Software

Before you use the SNMP agent software, you will need to install the following: Windows SNMP service, the Intel Dialogic system release software (refer to the system release installation guide), and Intel Dialogic boards and/or Intel NetStructure boards (refer to the installation instructions shipped with each board). Optionally, you can also configure the SNMP agent software on the managed node.

This section contains the following information:

- Installing SNMP for Windows* 2000
- Configuring SNMP Service for Windows 2000
- Using the SNMP Configuration Utility

Note: If you uninstall and reinstall the Windows SNMP Service on the managed node, you will also have to uninstall and reinstall SNMP agent software.

1.2.1 Installing SNMP for Windows* 2000

The following steps describe how to install the SNMP service on a Windows 2000 machine:

1. Invoke the **Windows 2000 Control Panel**.
2. Double click the **Add/Remove Programs** icon.
3. Select **Add/Remove Windows Components**. The Windows Component Wizard is displayed.
4. Check the Management and Monitoring Tools box.
5. Click the **Details** button.

6. Check the Simple Network Management Protocol box and click **OK**. The Windows Component Wizard is redisplayed.

7. Click **Next** to proceed through the Windows Component Wizard until it has finished.

1.2.2 Configuring SNMP Service for Windows 2000

Follow the steps below to configure SNMP Service on a Windows 2000 machine:

1. Invoke the **Windows 2000 Control Panel**.

2. Double click the **Administrative Tools** icon.

3. Double click the **Services** icon.

4. Select **SNMP Service**.

5. Choose the **Agent** tab.

6. Enter **Contact** and **Location**.

7. Choose the **Traps** tab.

8. Enter **Community Name:** `public`.

   Note: In the interest of enhanced security, administrators may want to choose a less commonly used community string.

9. Click **Add to List**.

10. Under **Trap Destinations**, click **Add** and enter the **IP address** for each management node to receive trap notifications from this managed node.

11. Click **OK** when done.

12. Choose the **Security** tab.

13. Highlight the community name you set in step 8 and click **Edit**. Change **Community Rights** to READ CREATE.

14. Click **OK**.

15. Stop **SNMP Service** (if started) and restart.

16. Double click **SNMP Trap Service**.

17. Set **Startup Type** to 'Automatic'.

18. Click **OK**.

19. Stop **SNMP Trap Service** (if started) and restart.

1.2.3 Using the SNMP Configuration Utility

The SNMP configuration utility provides a graphical user interface for configuring SNMP agent software on the managed node. You can access it from the Start menu or by running *snmpconfig.exe* which is found in *\Program Files\Dialgoic\bin*. The SNMP configuration utility enables you to configure system polling frequency and traps.

Note: The Application Monitoring tab in the SNMP configuration utility is not supported in Intel® Dialogic® System Release Version 5.1.1 Feature Pack 1 on PCI and CompactPCI for Microsoft* Windows NT/2000/XP on Intel® Architecture.

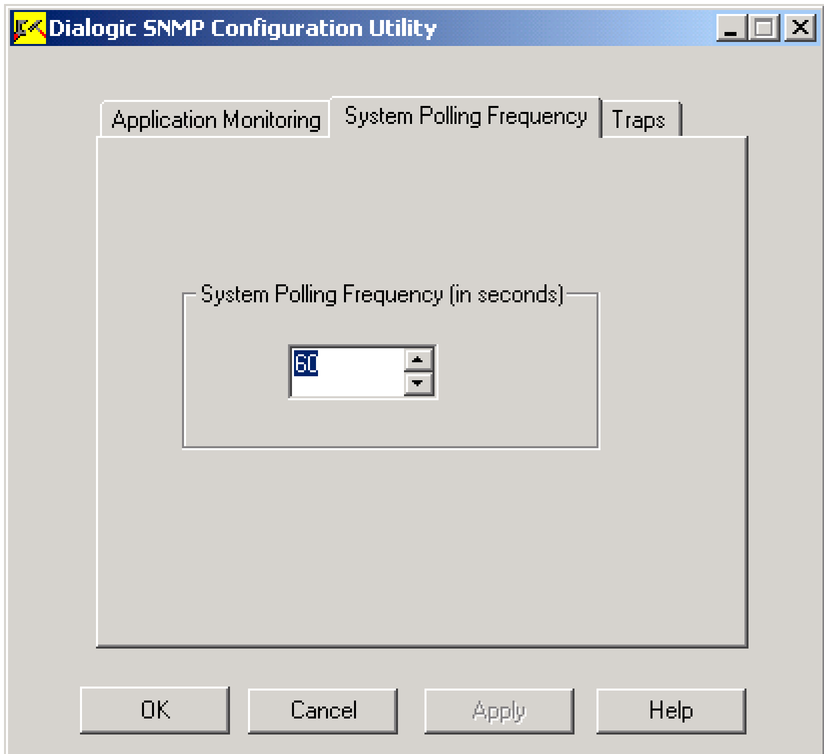
For more information, refer to the following sections:

- System Polling Frequency
- Traps

1.2.3.1 System Polling Frequency

The System Polling Frequency tab (Figure 2) enables you to determine the frequency with which the SNMP agents refresh the OIDs in the Common Module Group and Hardware Identification Group in the Hardware and Software (*dlghwinf.mib*) MIB.

Figure 2. SNMP Configuration Utility - System Polling Frequency Dialog



Use the System Polling Frequency Dialog to select or enter a polling frequency value between 0 and 32,000 seconds.

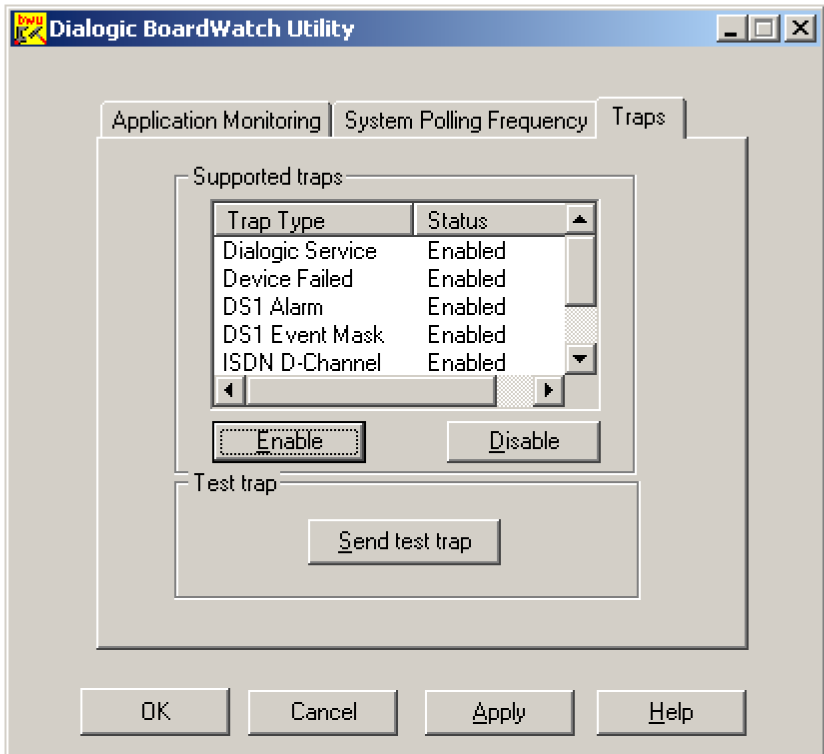
Note: The polling frequency you set with this tab has no effect on the frequency with which your network management application queries the SNMP agents.

1.2.3.2 Traps

The Traps tab (Figure 3) enables you to manage the traps that the SNMP agents send to the network management station(s). You can determine that traps will not be sent and send a test trap. You can enable or disable the following trap types:

- Dialogic Service
- Board Status Change
- DS1 Alarm
- DS1 Event Mask
- ISDN D-Channel
- ISDN B-Channel

Figure 3. SNMP Configuration Utility - Traps Dialog



The elements of the Traps dialog are

Trap Type	The Trap Type column lists all the currently available traps. For information about traps, see Using Traps to Monitor the System in the SNMP Agent Software Administration Guide.
Status	The Status column lists whether or not the trap is enabled or disabled. When a trap is disabled, the trap is not sent to the network management application even if the event that would normally trigger it occurs.
Enable	This button sets status of the selected trap to Enabled.
Disable	This button sets status of the selected trap to Disabled.
Send test trap	This button sends a test trap. When you click this button, the network management stations monitoring the managed node will receive dlgHiTestTrap with the name of the node on which this utility is running.

1.3 Using the SNMP Agent for Administrative Tasks

After you complete the preparations for using the SNMP agent software given in the software installation guide for the Intel Dialogic system release, you can perform the following administrative tasks:

- Reset the Intel Dialogic System
- Stop and start Intel Dialogic and Intel NetStructure boards
- Monitor DS-1 lines, ISDN interfaces, and the status of configured Intel Dialogic and Intel NetStructure boards
- Diagnose problems with the managed node or telephony application.

Details are given in Chapters 2, 3, and 4.

Stopping and Starting the System 2

This chapter provides information on using the dlgHiIdentServiceStatus OID in the *dlghwinf.mib* MIB to stop and start (reset) the Intel® Dialogic® system and using the SNMP agent software to stop and start Intel® Dialogic® boards and Intel® NetStructure™ boards.

Note: For information about starting the Intel Dialogic system for the first time, starting the Intel Dialogic System after initial startup, and stopping the system (without the use of SNMP), refer to the Administration Guide for the system release software.

2.1 Assumptions and Prerequisites

Before you can stop and restart the Intel Dialogic system, Intel Dialogic boards, and Intel NetStructure boards using the SNMP agent software, you must do the following:

- Install and configure Windows SNMP service. (Refer to Section 1.2, “Preparing to Use the SNMP Agent Software”, on page 9.)
- Install the Intel Dialogic system release software (refer to the Intel Dialogic system release software installation and configuration guide).
- Configure all the boards in the system (refer to the Intel Dialogic system release software installation and configuration guide).
- To start the Intel Dialogic system service using the SNMP agent software, you need to perform a one-time local startup of the Intel Dialogic system service on the managed node. Starting the Intel Dialogic system service results in downloading firmware with configuration parameter settings and initiating the host device drivers.

  Once you have started the Intel Dialogic system service, the Intel Dialogic system service can be stopped or started from the network management node.

Notes: ***1.*** For information about starting the Intel Dialogic System for the first time, starting the Intel Dialogic System after initial startup, and stopping the system (without the use of SNMP), refer to the administration chapter in the installation and configuration guide for the system release software.

2. For additional information about the Intel Dialogic Configuration Manager (DCM) and the Intel Dialogic system service, consult the DCM online help by pressing F1 when the DCM Main window is active.

2.2 Order of Procedures

The procedures in this chapter can be used in any order: one does not depend on the other. For information about starting the Intel Dialogic system for the first time, starting the Intel Dialogic system after initial startup, and stopping the system (without the use of SNMP), refer to the administration chapter in the installation and configuration guide for the system release software.

2.3 Stopping and Starting the System

Some situations require resetting all the Intel Dialogic boards and Intel NetStructure boards running in your system. Resetting the system requires interaction with more than the SNMP agent. To reset the system, follow these steps:

1. Re-route telephone activity away from the malfunctioning system to another system, or take each port, time slot, channel, or station line for all Intel Dialogic boards and Intel NetStructure boards in the system out of service as they become available.
2. Stop all telephony applications using Intel Dialogic hardware on the system.
3. Stop and restart all the Intel Dialogic boards and Intel NetStructure boards using the dlgHiIdentServiceStatus OID in the Hardware Information (HWINF) MIB.
   - To stop the system (all boards), set the OID to *stop-pending*
   - To start the system (all boards), set the OID to *start-pending*
4. Restart the telephony applications.
5. Re-route telephone activity back to the system and monitor the status to verify the functioning of the line or port.

Note: Shutting down the telephony application will disconnect any calls in progress. Be sure to wait for inactivity on any line before taking it off hook.

2.4 Stopping and Starting Boards

In order to use the SNMP agent software, you must start all the Intel Dialogic boards and Intel NetStructure boards on the managed node. Starting the boards results in downloading firmware with configuration parameter settings to the boards and initiating their host device drivers. Once you have used the SNMP agent to read OIDs from the managed node, you can start and stop the boards from the network management station as follows:

For starting/stopping an individual Intel NetStructure or Intel Dialogic board (excluding Intel Dialogic Springware boards, which cannot be individually stopped or started):

1. Open the network management application MIB browser.

2. Traverse the Hardware MIB tree looking for the dlgHiIdentAdminStatus OID for the desired Intel NetStructure board.
   – To start a stopped board, set dlgHiIdentAdminStatus object to *start-pending*.
   – To stop a started board, set dlgHiIdentAdminStatus object to *stop-pending*.

For starting/stopping all Intel NetStructure *and* Intel Dialogic boards:

1. Open the network management application MIB browser.

2. Traverse the Hardware MIB tree looking for the dlgHiIdentServiceStatus OID for the desired board.
   – To start stopped service, set dlgHiIdentServiceStatus object to *start-pending*.
   – To stop started service, set dlgHiIdentServiceStatus object to *stop-pending*.

When a board start/stop has completed, a dlgHiBoardStatusChange trap will be generated, indicating completion of the desired operation.

Installing and Using the MIBs 3

This chapter provides information about installing the MIBs and using them to monitor DS-1 lines, ISDN interfaces, and the status of configured Intel® Dialogic® and Intel® NetStructure™ boards.

3.1 Assumptions and Prerequisites

Before you use the MIBs, you must meet the following prerequisites:

- Install and configure Windows SNMP service. (Refer to Section 1.2, “Preparing to Use the SNMP Agent Software”, on page 9.)
- Install the Intel Dialogic system release software (refer to the Intel Dialogic system release software installation and configuration guide).
- Configure all the boards in the system (refer to the Intel Dialogic system release software installation and configuration guide).
- To start the Intel Dialogic system service using the SNMP agent software, you need to perform a one-time startup of the Intel Dialogic system service on the managed node. Starting the Intel Dialogic system service results in downloading firmware with configuration parameter settings and initiating the host device drivers.

  Once you have started the Intel Dialogic system service, the Intel Dialogic system service can be stopped or started from the network management node.

Notes: ***1.*** For information about starting the Intel Dialogic System for the first time, starting the Intel Dialogic System after initial startup, and stopping the system (without the use of SNMP), refer to the administration chapter in the installation and configuration guide for the system release software.

2. For information on stopping and starting boards and the system using SNMP, refer to Chapter 2, “Stopping and Starting the System”.

3. For additional information about DCM and the Intel Dialogic system service, consult the DCM online help by pressing F1 when the DCM Main window is active.

3.2 Order of Procedures

Once you have installed and compiled the MIBs (refer to Section 3.3, “Installing and Compiling the MIBs”, on page 20), you can use the other procedures in any order.

3.3 Installing and Compiling the MIBs

The Intel Dialogic MIBs are installed when you install the Intel Dialogic system release software.

In addition, some network management applications also require that you compile the MIBs. If compilation is required, you must compile the MIBs in the following order:

1. Hardware Information (*dlghwinf.mib*)
2. DM3 Extended Platform (*dlgdm3extplatinfo.mib*)
3. Springware Performance (*dlgsrprf.mib*)
4. DS-1 and ISDN (*dlgds1.mib*, *dlgisdn.mib*)

Note: For more information, refer to the documentation for your third-party network management application.

3.4 Using the Proprietary Hardware Information (HWINF) MIB Module

This section describes the proprietary Hardware Information (HWINF) MIB Module and provides a sample procedure demonstrating how to use it to monitor the status of Intel Dialogic and Intel NetStructure boards:

- Understanding the Proprietary Hardware Information (HWINF) MIB Module
- Using the Board Identification Table to Monitor Board Status

3.4.1 Understanding the Proprietary Hardware Information (HWINF) MIB Module

The proprietary Hardware Information MIB module provides a board inventory table and general parameters regarding the Intel Dialogic system services. This module permits the monitoring of board status changes and provides board status change notification using SNMP traps. It also provides the capability to start or stop individual Intel NetStructure boards and to start or stop all boards in the system.

The Hardware MIB module contains the following groups:

- MIB Revision Group (dlgHiMibRev)
- OS Common Group (dlgHiOsCommon)
- Board Identification Group (dlgHiIdent)

3.4.1.1 MIB Revision Group (dlgHiMibRev)

This group contains general revision and condition information and is comprised of:

- major revision number (dlgHiMibRevMajor): Indicates the major revision of the HWINF MIB module. This revision will always return ‘1’.
- minor revision number (dlgHiMibRevMinor): Indicates the minor revision of the HWINF MIB module. This revision will always return ‘2’.
- dlgHiMibCondition: Indicates the MIB status for this module. It will indicate *ok* if the MIB module can function properly or *failed* otherwise. When the agent starts up, this object initially indicates *failed* until the agent completes self-initialization.

3.4.1.2 OS Common Group (dlgHiOsCommon)

This group contains the following scalar objects and tables:

- Common Polling Frequency (dlgHiOsCommonPollFreq): This scalar object controls the global poll frequency. The current implementation of the SNMP agents does not support this variable, and setting its value to a new frequency has no effect on the run-time state of the agents.
- Common Module Table (dlgHiOsCommonModuleTable): This table contains a row entry for installed software modules used to control boards. It has read-only attribute columns that describe the module attributes (module name, module version, module date, and module purpose/description).
- Number of Modules (dlgHiOsCommonNumberofModules): Indicates the number of rows present in the Common Module Table.
- Log Enable (dlgHiOsLogEnable): This scalar object is not implemented. Network management stations (NMS) or other SNMP managers should handle logging any trap or status changes.
- Test Trap Enable (dlgHiOsTestTrapEnable): Setting this scalar object to ‘1’ causes the SNMP agent to send a test trap to all configured trap destinations (for instructions on how to configure trap destinations, see the Configuration Guide for the Intel NetStructure products). Reading this object always returns ‘0’ as result.

3.4.1.3 Board Identification Group (dlgHiIdent)

This group contains the following scalar objects and tables:

- Board Identification Table (dlgHiIdentTable)
- Board Identification Table Size (dlgHiIdentNumberOfDevices)
- Service Status Object
- Service Change Date
- Global Trap Mask

The following information describes each item:

Board Identification Table (dlgHiIdentTable)

The Board Identification Table contains a row entry for each board device installed on the managed node. It has read-only attribute columns that describe the board attributes (IRQ, firmware name, serial number, etc.) and the board operational status. It has mutable columns that describe the board administrative status and that allow the user to start, stop, or cause a board to perform self-diagnostics.

The following are the columns in the Board Identification Table:

- *board index (dlgHiIdentIndex)*: This column contains the indices for this table. All SNMP tables have one or more indices that uniquely identify the row in which they belong. Performing a **get** on this column should always return the same value as the row requested. For example,

  ```
  snmp get dlgHiIdentIndex.2
  ```

  returns 2
- *model name (dlgHiIdentModel)*: This column contains the string name of the board model for each board in the table.
- *family type (dlgHiIdentType)*: This column indicates the CT family type for each board.
- *functional description (dlgHiIdentFuncDesc)*: For each board, this column contains a short description of its purpose.
- *serial number (dlgHiIdentSerNum)*: This column contains the unique serial number assigned to each board by the Intel Dialogic system release software.
- *firmware name (dlgHiIdentFWName)*: This column contains the firmware file name for each board in the table.
- *firmware version (dlgHiIdentFWVers)*: This column contains the firmware version for each board in the table.
- *base memory address (dlgHiIdentMemBaseAddr)*: This column contains the memory address for each board's shared RAM space.
- *base I/O address (dlgHiIdentIOBaseAddr)*: This column contains the I/O address for each board that supports this style of interface.
- *interrupt request number (dlgHiIdentIRQ)*: This column contains each board's IRQ number.
- *board ID (dlgHiIdentBoardID)*: This column contains each board's board ID. For Intel NetStructure on DM3 architecture boards, the board ID is the logical board number. For Intel NetStructure IPT series boards, the board ID is the driver board ID. For Intel Dialogic Springware boards, the board ID is the ID selected from the on-board thumbwheel or the assigned board ID for PCI boards.
- *PCI slot number (dlgHiIdentPCISlotID)*: For each PCI board, this column contains the PCI slot number at which the board is installed. Non-PCI boards return –1 in this column.
- *PCI bus number (dlgHiIdentPCIBusID)*: For each PCI board, this column contains the PCI bus number at which the board is installed. Non-PCI boards return –1 in this column.
- *operational status (dlgHiIdentOperStatus)*: This column contains the operational status of each board. The operational status can indicate a *failed* or *ok* status. If this column indicates *ok*, then the board is working normally and has responded to all requests as expected. If this column indicates *failed*, then the board is not operating normally and something has caused it to enter a potentially dangerous state. Two conditions exist to cause *operational status* to indicate a *failed* status: the board has stopped responding to

routine ping messages or the board was instructed to start or stop using the administrative status column, and it has failed to do so. In general, a board that is indicating a *failed* status, should be restarted. If the problem persists, the board should be replaced.

- *administrative status (dlgHiIdentAdminStatus)*: Administrative status uses two groups of values. The first group is used to indicate the present condition of the board. The possible values are *started*, *stopped*, and *diagnose*. The state of the board is *diagnose* while the diagnostic is running.

Note: If a board is in the process of starting or stopping, it will always indicate the state from which it is coming from until the operation is complete.

The second group is made up of three values that are used to change the board state using `set` operations. These states are *start-pending*, *stop-pending*, and *diagnose*. When a board is in a stopped state, and its administrative status value is set to *start-pending*, the SNMP agent will attempt to start the board. When a board is in a started state, and its administrative status value is set to *stop-pending*, the SNMP agent will attempt to stop the board. If either operation fails, the board's *operational status* (see below) will indicate *failed* and the administrative status will stay at the old value. The *diagnose* state is used to cause a stopped board to run a power-on-self-test (POST). Once in the *diagnose* state, the board cannot be stopped or started until the POST has completed. When the diagnostics complete, the board is placed back into the stopped state, and the diagnostic results are stored in the Board Error Message column (see below).

Note: Only Intel NetStructure boards support these read/write states. Intel Dialogic Springware boards will reject any requests to stop and start individual boards.

All transitions in administrative status, except those that are the direct result of a `set`, cause a trap. So the transition from *stopped* to *diagnose* as the result of a `set` does not cause a trap. The transition from *diagnose* to *stopped* when the diagnostic finishes *does* cause a trap.

- *device change date (dlgHiIdentDeviceChangeDate)*: This column indicates the time and date (in 7-octet format) of when the administrative status of the board last changed.
- *specific group OID (dlgHiIdentSpecific)*: Not used in a meaningful way in this system release.
- *board error message (dlgHiIdentErrorMsg)*: This column is used to indicate the last error message for each board. Primarily, it is used to store the results in textual format of a `dm3post` operation (administrative status *diagnose* state). Refer to the system release software Diagnostics Guide for details on `dm3post`.

Board Identification Table Size (dlgHiIdentNumberOfDevices)

This scalar object is type INTEGER and returns the number of boards installed and configured on the managed node. This number corresponds directly with the number of rows present in the Board Identification Table (dlgHiIdentTable). A value of zero indicates that no boards have been installed or configured.

Service Status Object

This scalar object is used to control the entire Intel Dialogic system release software run-time state (i.e., the software program is stopped or started). When it indicates *stopped*, the Intel Dialogic run time has not been started. When it indicates *started*, the Intel Dialogic run time is loaded and running.

An SNMP manager may set the value of this object to change the run time state of the Intel Dialogic system release software. When the manager sets the value to *started*, the Intel

Dialogic system release software will start and initialize. All configured boards will be downloaded. When the manager sets the value to *stopped*, the Intel Dialogic system release software will be stopped and unloaded.

Service Change Date

This scalar object indicates the last time in which the system service was started or stopped though the Service Status Object (dlgHiIdentServiceStatus). The result is in 7-octet format described in the MIB specification.

Global Trap Mask

This scalar object is not implemented and setting its value has no effect.

3.4.2 Using the Board Identification Table to Monitor Board Status

The core of the Hardware MIB module is the Board Identification Table (dlgHiIdentTable), which contains a row for every board installed and configured on the managed node. Using this table, you can determine and verify which boards are configured on the node, and whether or not a board is working from a high-level point of view. Other MIB modules, such as the Intel Dialogic DS-1 MIB module and standard DS-1 MIB module, use the indices of the Board Identification Table to link subdevices, such as DS-1 trunk lines, to their host boards, such as an Intel NetStructure DM/V voice board.

An example of the output from the Board Identification Table follows:

dlgHiIdent-Index	dlgHiIdent-Model	...	dlgHiIdent-AdminStatus	dlgHiIdent-OperStatus
1	qs_t1	...	started(2)	ok(2)
2	qs_t1	...	stopped(3)	ok(2)
3	D/480JCT-2T1	...	started(2)	ok(2)
4	D/120JCT-LS	...	stopped(2)	failed(4)

In this example configuration, the managed node contains four boards indexed from 1 to 4, sequentially. The first three boards are operating normally, the second of which is an Intel NetStructure DM/V voice board in a stopped state. The fourth board is indicating that it has failed some operation and may need to be replaced.

The SNMP manager may poll this table periodically to determine if any board status changes have occurred and then notify the user accordingly, perhaps with a pop-up message or alert beep.

Suppose the user has stopped the second board and wishes to diagnose the board, check the results, and then restart the board after verifying the board is healthy. The following operations are performed to achieve this end:

1. An SNMP-get request is sent to the dlgHiIdentAdminStatus.2 object which returns *stopped(3)*, matching what is seen in the example table. Note that the ".2" suffix is added to specify which

row in the dlgHiIdentAdminStatus column the request is targeting. In this case, we are interested in the second board, which has an index of ‘2’.

2. An SNMP-set request is sent to dlgHiIdentAdminStatus.2 object with the value *diagnose(5)*. The agent will invoke the `dm3post` tool, which begins running diagnostics on the board. (For more information on the `dm3post` tool, refer to the Diagnostics Guide for the system release software.)

3. The SNMP manager now polls dlgHiIdentAdminStatus.2 until it changes from *diagnose(5)* back to *stopped(3)*. When this occurs, the POST diagnostics have completed.

4. To view the results of the diagnostics, the manager issues a SNMP-get request on dlgHiIdentErrorMsg.2. The manager again uses “.2” to indicate the second row in the table. When the error message is returned, it will contain a message detailing the results of the POST. If the diagnostic fails, the operational status is set to failed. When the diagnostic completes, a trap is always generated. If it completed in failure, the operational status in the trap will be *failed*. (For more information on the POST result messages, refer to the Diagnostics Guide for the system release software.)

5. Finally, the manager starts the board by issuing an SNMP-set request to dlgHiIdentAdminStatus.2 with the value *start-pending(5)*. After starting, the dlgHiIdentAdminStatus.2 object will indicate *started(2)* or the dlgHiIdentOperStatus.2 object will return a *failed* status indicating the start attempt failed.

3.5 Using the Proprietary ISDN MIB Module

This section describes the proprietary ISDN MIB module and describes how to use it to monitor ISDN interfaces:

- Understanding the Proprietary ISDN MIB Module
- Monitoring ISDN Interfaces

3.5.1 Understanding the Proprietary ISDN MIB Module

The proprietary ISDN MIB module provides management stations with an interface to review ISDN B- and D-channels, monitor their status, and report fault conditions using SNMP traps.

The ISDN MIB module contains the following tables and traps:

- D-Channel Table (dlgIsdnSigTable)
- Bearer Channel Table (dlgIsdnBearerTable)
- ISDN Traps

3.5.1.1 D-Channel Table (dlgIsdnSigTable)

The D-Channel table contains an entry for each D-channel for all ISDN-configured Intel Dialogic and Intel NetStructure boards. Primary Rate Interface (PRI) devices may have zero or one D-channel configured. If NFAS is not being used, then the D-Channel table contains one entry per trunk line, corresponding to each D-channel needed to provide control signals for each interface. If NFAS is configured, then the D-Channel table will contain fewer entries corresponding to the shared D-channel among multiple interfaces. The D-Channel table contains the following columns described in this section.

D-Channel Index (dlgIsdnSigIndex)

This column contains the table indices.

D-channel Name (dlgIsdnSigName)

This column contains the names of the D-channels using the Intel Dialogic R4 convention for naming devices. For PRI devices, the R4 names are dtiBx, where x is a positive integer.

ISDN Protocol (dlgIsdnSigProtocol)

The ISDN protocol used by each D-channel is stored in this column. For instance, 4ess, 5ess, and NTI are possible values for this column.

B-channel Count (dlgIsdnSigBchanCount)

This column contains the number of B-channels whose control signaling is performed by each D-channel.

Primary Channel Indicator (dlgIsdnSigPrimary)

When this boolean-typed column indicates true for an entry, then the corresponding D-channel is considered to be the primary ISDN D-channel for the managed node.

LAPD Operational Status (dlgIsdnSigLapdOperStatus)

Each D-channel may indicate a link-up or link-down status depending on the Layer-2 connection state of the LAPD layer. This column contains this state for each D-channel, and each entry contains one of the following three values:

- UNKNOWN: the managed node is unable to access this information
- LINK_DOWN: the D-channel layer-2 is inoperable
- LINK_UP: the D-channel layer-2 is operating normally

Board Index (dlgIsdnSigHiIdentIndex)

Each entry in this column contains an index into the Board Identification Table (dlgHiIdentTable) that indicates on which board its corresponding D-channel resides.

3.5.1.2 Bearer Channel Table (dlgIsdnBearerTable)

The Bearer Channel (B-channel) table contains one entry for each B-channel on every Intel Dialogic and Intel NetStructure -based ISDN board that is configured and started. This table allows the user to monitor the call state of each B-channel and to also trace a B-channel to its corresponding D-channel and its board or residence.

Bearer Channel Index (dlgIsdnBearerIndex)

This column contains the table indices.

Bearer Channel Name (dlgIsdnBearerName)

This column contains the name of each B-channel. This name is based on the Intel Dialogic convention for naming logical R4 devices. It has the form dtiBxTy, where x is the logical Intel Dialogic Springware board number and y is the time slot number of the B-channel.

Bearer Channel Status (dlgIsdnBearerStatus)

This column indicates the operational status for each B-channel. The possible values for B-channel status are:

- IN_SERVICE: the B-channel is ready to send and receive a message
- OUT_OF_SERVICE: the B-channel is unable to send and receive a message
- MAINTENANCE: the B-channel is ready to receive an incoming test call
- UNKNOWN: the state of the B-channel is undefined

Note: Do not use a `set` on the the dlgIsdnBearerStatus OID.

D-Channel Index

Each B-channel has an associated D-channel listed in the D-Channel table (section 1.1). The indices in this table refer to the indexed entries in the D-Channel table and can be used to trace each B-channel over to its associated D-channel.

Board Index

Each entry in this column contains an index into the Board Identification Table (dlgHiIdentTable) that indicates on which board its corresponding B-channel resides.

3.5.1.3 ISDN Traps

ISDN traps provide information on fault conditions.

D-Channel Status Change (dlgIsdnDChanged)

This trap is generated anytime the LAPD status of a D-channel changes value. The trap contains the following variables and associated current values:

- D-channel index (dlgIsdnSigIndex)
- D-channel name (dlgIsdnSigName)
- LAPD operational status (dlgIsdnSigLapdOperStatus)

Bearer Channel Status Change (dlgIsdnBChanged)

This trap is generated anytime the operational status of a B-channel changes value. The trap contains the following variables and associate current values:

- Bearer channel index (dlgIsdnBearerIndex)
- Bearer channel name (dlgIsdnBearerName)
- Bearer channel status (dlgIsdnBearerStatus)

Note: B Channel traps are not generated on Intel NetStructure boards because this could cause too many traps to be sent.

3.5.1.4 A Typical ISDN Configuration

The following tables show a typical managed node's ISDN tables:

D-Channel Table:

dlgIsdnSig-Index	dlgIsdnSig-Name	dlgIsdnSig-LapdOperStatus	dlgIsdnSig-HiIdentIndex
1	dtiB1	LINK_DOWN	1
2	dtiB2	LINK_UP	1
3	dtiB5	LINK_UP	3
4	dtiB6	LINK_UP	3

Bearer Channel Table:

dlgIsdn-BearerIndex	dlgIsdn-BearerName	dlgIsdn-BearerStatus	dlgIsdn-BearerHiIdentIndex
1	dtiB1T1	OUT_OF_SERVICE	1
2	dtiB1T2	OUT_OF_SERVICE	1
3	dtiB1T3	OUT_OF_SERVICE	1
4	dtiB1T4	OUT_OF_SERVICE	1
…	…	…	…
47	dtiB5T1	IN_SERVICE	3
48	dtiB5T2	IN_SERVICE	3
…	…	…	…

The first table shows four trunk lines configured with R4 virtual board device names dtiB1, dtiB2, dtiB5, and dtiB6. Each one has a D-channel configured; hence, there are four entries in the D-Channel table. The first D-channel is indicating a link-down status, which may indicate a potential fault on the trunk line. The potentially faulty trunk may be further diagnosed by referring to the DS-1 Alarm table and the DS-1 Configuration table for more detailed signal-level information. The first table also shows that the first two ISDN interfaces are configured on the board with index 1 in the Board Identification Table, and the second two interfaces are found at index 2 in the Board Identification Table.

The Bearer Channel table shown above provides greater depth of information regarding individual B-channels, each of which may be carrying voice data for a call conversation. From the table, it is obvious that the B-channels associated with the potentially faulty trunk line are out-of-service (indices 1-4 and beyond). The interfaces for the operational ISDN interfaces have been configured and they indicate this status in the entries 47, 48, and beyond.

3.5.2 Monitoring ISDN Interfaces

Using general SNMP monitoring methods, there are two ways to monitor the ISDN interfaces for fault detection. The first way requires that the management station poll the managed node for the ISDN tables at regular intervals. When a change in status has been detected, the management station can take appropriate action to notify the user of a detected fault.

The second way is more efficient, but less reliable. This method requires the management station to act on received ISDN traps (see Section 3.5.1.3, “ISDN Traps”, on page 27). Each time a B- or D-channel changes state, a trap is generated by the agent and sent to the management station. This trap contains the index and status of the channel that changed state. An important limitation of SNMP version 1 traps is that they are sent using an unreliable delivery method.

A third, more common approach that overcomes the unreliability of trap reception is to use trap-based polling. Trap-based polling requires the management station to poll the ISDN tables at regular intervals. This polling guarantees that a fault condition will be detected at each interval boundary. The reception of the ISDN traps optimizes the polling by forcing the end of an interval at the reception of a trap causing the fault to be detected immediately and the next polling interval to begin.

When providing fault detection capabilities on the management station, the most important table columns are the dlgIsdnSigLapdOperStatus and dlgIsdnBearerStatus columns. Because there are far more B-channels on a system than D-channels, it may only be practical to poll the dlgIsdnSigLapdOperStatus columns to detect faults. Inspecting the status of individual B-channels should only be performed when the B-channel is of particular interest, such as diagnosing a call connection.

3.6 Using the Proprietary Springware Performance Module

The proprietary Springware Performance Module (dlgsrprf.mib) contains Springware performance information maintained by the device driver. It includes the following groups:

- Current Interval Performance Information Group: contains the current interval’s performance data, including hardware interrupts, messages to and from the board, bulk data transfer counts, and lost or error message counts.

  The default interval of one hour can be changed. The performance counters are reset at the beginning of a new interval. A separate MIB variable indicates the duration of the current interval.

  This group contains the following:

  – dlgPsCurrentInterrupts - number of hardware interrupts generated by span or D4x based boards for the current interval
  – dlgPsCurrentDrvCommands - number of commands sent to the driver for the current interval
  – dlgPsCurrentFWCommands - number of commands sent to the firmware for the current interval
  – dlgPsCurrentUnSolEvents - number of unsolicited events from the firmware for the current interval
  – dlgPsCurrentBytesRead - number of bytes read from the firmware during record for the current interval
  – dlgPsCurrentBytesWritten - number of bytes written to the firmware during playback for the current interval

- dlgPsCurrentLostMsgToFW - number of lost messages (commands) to the firmware for the current interval. This value should always be zero. A non zero value typically indicates a firmware or hardware failure.
- dlgPsCurrentLostMsgFromFW - number of lost messages (events) from the firmware for the current interval. This value should typically be zero. A non zero value indicates the host cannot service all the messages being generated by the firmware.
- dlgPsCurrentFWErrorMsgs - number of error message packets generated by the firmware for the current interval. This should typically be zero.
- dlgPsCurrentDrvErrorMsgs - number of error message packets generated by the driver for the current interval. This should typically be zero.

• Total Performance Information Group: contains performance information cumulative from the time the agent started collecting statistics.

The information tracked is the same as for the Current Interval Performance Information Group. These counters are reset when the agent is restarted or when the device driver is restarted.

This group contains the following:

- dlgPsTotalInterrupts - the total number of hardware interrupts generated by the span or D4x based boards
- dlgPsTotalDrvCommands - the total number of commands sent to the driver
- dlgPsTotalFWCommands - the total number of commands sent to the firmware
- dlgPsTotalUnSolEvents - the total number of unsolicited events from the firmware
- dlgPsTotalBytesRead - the total number of bytes read from the firmware during record
- dlgPsTotalBytesWritten - the total number of bytes written to the firmware during playback
- dlgPsTotalLostMsgToFW - the total number of lost messages (commands) to the firmware. This value should always be zero. A non zero value typically indicates a firmware or hardware failure
- dlgPsTotalLostMsgFromFW - the total number of lost messages (events) from the firmware. This value should typically be zero. A non zero value indicates the host cannot service all the messages being generated by the firmware.
- dlgPsTotalFWErrorMsgs - the total number of error message packets generated by the firmware. This should typically be zero.
- dlgPsTotalDrvErrorMsgs - the total number of error message packets generated by the driver. This should typically be zero.

Note: The Intel Dialogic Springware Performance MIB (*dlgsrprf.mib*) can only be used with Intel Dialogic Springware boards.

3.7 Using the Proprietary DS-1 MIB Module

This section describes the proprietary DS-1 MIB module and provides a sample procedure demonstrating how to use it to monitor DS-1 lines:

- Understanding the Proprietary DS-1 MIB Module
- Monitoring DS-1 Lines

3.7.1 Understanding the Proprietary DS-1 MIB Module

The proprietary DS-1 MIB module provides management stations with an interface to review DS-1 trunk lines, monitor their status, and report fault conditions using an SNMP trap. It also provides a 24-hour historical fault metrics log in the current, interval, and total tables.

The DS-1 MIB module contains the following tables:

- DS-1 Trunk Configuration Table (dlgDsx1ConfigTable)
- DS-1 Alarm Table (dlgDsx1AlarmTable)
- Metrics Statistics Tables
  - Current Metrics Interval Table (dlgDsx1CurrentTable)
  - Historic Metrics Interval Table (dlgDsx1IntervalTable)
  - Total Metrics Interval Table

3.7.1.1 DS-1 Trunk Configuration Table (dlgDsx1ConfigTable)

The trunk configuration table is identified in the MIB as dlgDsx1ConfigTable. It contains one row entry for each DS-1 trunk line configured for all operational Intel Dialogic and Intel NetStructure boards in the managed node. When a board is in a stopped state or if a board is unconfigured, its trunk lines will not appear in this table. A trunk line can be verified for correct configuration by requesting its row from this table. A trunk line can be monitored for fault information by polling columns from this table.

Line Index Column (dlgDsx1LineIndex)

This column contains the indices for the table. In general, each line is assigned a unique index with the first line starting from ‘1’ and each successive line being assigned indices in ascending order. It is possible for indices to skip, or to not start at one, such as the case when boards are swapped in and out in CompactPCI-based platforms.

Board Index Column (dlgDsx1HiIdentIndex)

For each trunk line, this column contains the index of the board on which the trunk jack is located (near-side). This index corresponds to the indices in the dlgHiIdentTable (Board Identification Table).

Current Interval Elapsed Time Column (dlgDsx1TimeElapsed)

This column contains the time, in seconds, since the start of the current 15-minute metric interval. When this counter reaches 900, the metrics interval has ended, and it is archived in the Historic Interval table (refer to the section describing the Historic Metrics Interval Table (dlgDsx1IntervalTable)).

Valid Metric Intervals Column (dlgDsx1ValidIntervals)

For each trunk line, this column contains the number of elapsed 15-minute intervals since metrics tracking began. It contains a value between 0 and 96.

Line Type Column (dlgDsx1LineType)

This column contains a value that indicates the type of line the trunk is configured to be. The possible values are:

- D4: A T-1 bit rate introduced by AT&T
- E1-CRC: bit rate associated with E-1 lines

Line Coding Column (dlgDsx1LineCoding[1])

This column contains a value for each line that indicates the type of coding in use. The possible values are:

- JBZS
- B8ZS
- HDLC
- TRANS

Loopback Configuration Column (dlgDsx1LoopbackConfig[2])

Entries in this column indicate the loopback configuration for each line. The possible values are:

- Normal loopback
- Payload loopback
- Line loopback
- Local loopback

Line Status Column

This column can be used to determine the link state of each line in detail. The value contained in each entry is a bitmap that represents the possible conditions that indicate line faults. The bit numbering assumes bit 1 is the least-significant bit (LSB). The bitmap positions have the following meaning:

Bit	Description
1	The line is not indicating any faults
2	The far end is indicating loss-of-frame
3	The near end is sending an alarm indication signal (AIS)
4	The near end is indicating loss-of-frame
5	The near end is indicating loss-of-signal
6	The near end is configured for loopback
7	Time slot 16 of an E-1 line is showing an alarm indication signal
8	Far end is sending an E-1 time slot 16 loss-of-multiframe
9	Near end is sending an E-1 time slot 16 loss-of-multiframe
10	Unclassified failure (other)

Line Signal Mode Column

This column contains the signal mode for each line. The possible values for signal mode are:

- None
- Robbed bit (T-1)
- Bit-Oriented (E-1)
- ISDN

1. Do not try to set a different LineCoding. Dsx1LineCoding is not able to set values.
2. Do not try to set a different LoopbackConfig. Dsx1LoopbackConfig is not able to set values.

3.7.1.2 DS-1 Alarm Table (dlgDsx1AlarmTable)

The DS-1 Alarm table contains an entry for each trunk on every configured and started Intel Dialogic and Intel NetStructure board on a managed node. The Alarm Type column in this table can be polled to determine high-level alarm flag information for DS-1 trunk lines.

Alarm Table Index Column (dlgDsx1AlarmIndex)

This column is the table index. It corresponds directly to each index that exists in the DS-1 Trunk Configuration Table (see Section 3.7.1.1, “DS-1 Trunk Configuration Table (dlgDsx1ConfigTable)”, on page 31). For example, if trunk line A is located at index 3 in the configuration table, then it is also represented in the alarm table as index 3.

Alarm Type Column (dlgDsx1AlarmType)

This column contains the current alarm flag for each trunk line. The possible flags values are red, yellow, blue, E-1, and no-alarm. Under normal circumstances, a configured and operational line will indicate the “no-alarm” type. When the alarm type changes to any of the other values, the management station should take the steps necessary to alert the network administrator of a possible trunk fault.

Note: The dlgDsx1TransmitClockSource OID reports a 0 value when a query is issued. The description states that the only values that should be reported are 1,2,3. The dlgDsx1TransmitClockSource OID is not able to use the settable values of 1,2,3.

3.7.1.3 Metrics Statistics Tables

During the analysis of a recurring fault, the user may wish to consult current and historic fault metrics. These metrics may provide some insight into the cause of a faulty trunk. The metrics statistics tables present this information to the user.

Each metric measures an instance of a particular kind of fault. The metrics tracked are:

- Errored seconds
- Severely errored seconds
- Severely errored framing seconds
- Unavailable seconds
- Controlled slip seconds
- Path code violations
- Line error seconds
- Bursty error seconds
- Degraded minutes
- Line code violations

Each fault for each metric is counted for a fifteen-minute interval. The most current, incomplete interval is stored for each trunk line in the Current Metrics Interval table. After a fifteen-minute interval has expired, the metrics counters in the current table are archived in the Historic Metrics Interval table. A third table, called the Total Metrics Interval table, contains the summation of the metrics counters in all the archived intervals, for each line, in the Total Metrics interval table.

Current Metrics Interval Table (dlgDsx1CurrentTable)

This table contains one index column called dlgDsx1CurrentIndex. Indices in this column correspond one-to-one with the indexed entries of the DS-1 Trunk Configuration table (see Section 3.7.1.1, "DS-1 Trunk Configuration Table (dlgDsx1ConfigTable)", on page 31).

Historic Metrics Interval Table (dlgDsx1IntervalTable)

The historic metrics interval table contains archived 15-minute intervals for up to the last 24 hours of service. There are 96 15-minute intervals in 24 hours, so this table may contain up to 96 entries for each trunk line.

This table has two index columns. The first index corresponds one-to-one with the indexed entries of the DS-1 Trunk Configuration table. The second index identifies the interval number. The interval number ranges from 1 to 96, with 1 as the most recent interval and 96 as the least recent interval.

Total Metrics Interval Table

The Total Metrics Interval table contains the sum of all interval metrics for each trunk up to the last 24 hours. It has one index column called dlgDsx1TotalIndex. Each index in this column corresponds one-to-one with the indexed entries of the DS-1 Trunk Configuration table.

3.7.1.4 Trunk Line Status Trap

This trap is sent whenever an entry in the Line Status column changes value. The trap contains the index into the configuration table of the line affected and the line status bitmap value.

3.7.1.5 A Typical DS-1 Configuration

The following is an example of a typical DS-1 Trunk Configuration table:

dlgDsx1-LineIndex	dlgDsx1-LineType	...	dlgDsx1-LineStatus	dlgDsx1-SignalMode
1	dlgDsx1D4	...	1	dlgDsx1Isdn
2	dlgDsx1D4	...	1	dlgDsx1Isdn
3	dlgDsx1D4	...	96	dlgDsx1Isdn
4	dlgDsx1D4	...	96	dlgDsx1Isdn
5	dlgDsx1E1CRC	...	1	dlgDsx1BitOriented
6	dlgDsx1E1CRC	...	1	dlgDsx1BitOriented
...	...	...	...	...

The following is an example of a corresponding DS-1 Alarm Table:

dlgDsx1AlarmIndex	dlgDsx1AlarmType
1	No Alarm
2	No Alarm
3	Red Alarm
4	Red Alarm
5	No Alarm
6	No Alarm
…	…

These two tables contain 6 trunk lines: 4 configured as T-1 lines and 2 as E-1 lines. Two of the T-1 lines show a line status of 96, which is a bitmap that indicates near-end loss-of-frame and loss-of-signal faults. The indices for these two lines are 3 and 4. It is clear in the DS-1 Alarm table provided that the trunks represented by rows 3 and 4 are showing a red state, consistent with the fault indications in the configuration table. To further diagnose the problem, you can try analyzing the metrics counters in the current interval table to determine if the faults may be frame-related, bit-level, etc. The actual analyzing of each metric type to diagnose a difficult-to-diagnose line fault is beyond the scope of this document.

3.7.2 Monitoring DS-1 Lines

There are two options available to a management station that allows it to monitor trunk lines managed by this MIB module. The most high-level option is to monitor the DS-1 Alarm table for changes in the Alarm Type column. Using this method, a management station can determine high-level shifts in line service status. For instance, if a management station observes that a trunk line's alarm type changes from no-alarm to red, the conclusion can be made that a severe fault has been detected for that trunk.

A more detailed way of monitoring trunk line service status is to poll the Line Status column in the DS-1 Trunk Configuration table. Each entry in this column is a bitmap that indicates very specific fault conditions, but does not make any interpretation on what changes in these faults mean. This gives the management station flexibility in making its own policy decisions on what constitutes severe and non-severe alarm conditions.

A management station may use the Line Status trap as a trigger to poll these two columns. This can lead to a more efficient and timely alert to the user when a trunk fault is observed. For example, suppose a management station is configured to poll the Line Status column in the configuration table every five minutes. If the management station does not take advantage of the Line Status trap, a line fault would only be detected on the expiration of every five-minute interval. If the management station is configured to poll when it receives the line status trap, it can take action right away instead of waiting up to five minutes for its polling interval to expire.

Like tables in other media-specific MIBs, the interface number of the DS-1 line indexes the DS-1 MIB tables. These indices correspond to the indices of the same interfaces in the Interfaces Table. This correspondence between interface indices allows management software to quickly find the complete set of general (RFC 1213 MIB-2) and specific (RFC 2495 DS-1) information available for each DS-1 interface.

3.8 Using the Proprietary DM3 Extended Platform MIB

This section provides the following information:

- Understanding the DM3 Extended Platform MIB
- Collecting Information About Intel NetStructure Boards with a DM3 Architecture

3.8.1 Understanding the DM3 Extended Platform MIB

The proprietary DM3 Extended Platform (Dm3ExtPlatform) MIB module is a hardware-specific extension of the Board Identification Table (dlgHiIdentTable). This MIB contains detailed information about the configuration of hardware on Intel NetStructure boards with a DM3 architecture. This information is organized into three tables:

- Extended DM3 Board Information Table (dlgDm3ExtBrdLevelTable) - Each row in the Extended DM3 Board Information Table corresponds to a row in the Board Identification Table, so the indices of rows in the former table match indices of corresponding rows in the latter table. The Extended DM3 Board Information Table contains only configuration information that includes the following:
  - shelf ID
  - FCD configuration file name
  - PCD configuration name
  - PCD configuration version
- Extended DM3 Sub Assembly Information Table (dlgDm3ExtSubAssemblyTable) - The Extended DM3 Sub Assembly Information Table contains entries that represent board subassemblies, and each row can be mapped back to its parent board in the Extended DM3 Board Information Table. The Extended DM3 Sub Assembly Information Table contains only configuration information that includes the following:
  - subassembly type
  - serial number
  - index of parent board in the dlgDm3ExtBrdLevelTable table
- Extended DM3 Processor Information Table (dlgDm3ExtProcTable) - The Extended DM3 Processor Information Table contains rows that represent each processor on each subassembly of each board. Each entry in the Extended DM3 Processor Information Table can be mapped back to a parent subassembly in the Extended DM3 Sub Assembly Information Table. Each processor maps to a board by way of the subassembly table. The Extended DM3 Processor Information Table contains configuration and status information for each processor, and columns exist that provide the following information:
  - processor type
  - run-time kernel version
  - boot kernel version
  - operation status

In addition to its three tables, the DM3 Extended Platform (Dm3ExtPlatform) MIB module also defines a trap, dlgDm3epOperStatusChange, which is generated whenever the processor operation status value changes for a processor in the Extended DM3 Processor Information Table (dlgDm3ExtProcTable).

3.8.2 Collecting Information About Intel NetStructure Boards with a DM3 Architecture

The DM3 Extended Platform MIB is used to collect configuration information from a managed node regarding Intel NetStructure boards on the DM3 architecture and their subassemblies and processors. The dlgDm3epOperStatus column and the dlgDm3epOperStatusChange trap allow a management station to monitor boards for processor health information providing a gathering of finer-grained fault information from Intel NetStructure boards on the DM3 architecture than what the Board Identification Table provides through its operational status column.

Performance Monitoring 4

This chapter describes how to use the SNMP agent software to diagnose problems with the managed node or telephony application including traps, errors and lost messages.

4.1 Using Traps to Monitor the System

This section provides general information on using traps and a detailed description of each trap supported by this version of the SNMP agent software. This section contains the following subsections:

- Controlling the Occurrence of Traps
- Responding to Events That Trigger Traps

4.1.1 Controlling the Occurrence of Traps

A trap is a notification that is sent by the SNMP agent when an event occurs on the managed node. When your network management application receives a trap, it may indicate that your telephony application or the Intel Dialogic platform on the managed node entered a condition that requires diagnostic treatment.

The network management station may not receive all the traps described in the previous subsection. The traps your network management station receives are affected by the Intel Dialogic device driver's event mask. Currently, the only value supported for the event mask is 1 (allAlarmevtmsk), which indicates that all events applicable to T1 and E1 are enabled.

4.1.2 Responding to Events That Trigger Traps

This subsection describes all the traps supported by this version of the SNMP agent. These descriptions indicate the event that triggers the trap and include brief suggestions for responding to these events.

It is possible to use the SNMP agent software to monitor boards and to determine if a board has failed. If a board has failed, you can diagnose the board to determine if the hardware is good. If it is, the board can then be restarted. If the diagnosis indicates that the board is bad, then the board must be replaced. If you are using an SNMP manager that can be extended through programming, you can program the manager to diagnose the board when a fault is detected, and either restart the board or notify a person through email or other communication methods.

If there is a hardware fault on your board, the steps described under dlgHiBoardStatusChanged (below) can help you diagnose the issue. This process will not detect a failure in software, but will allow you to restart your system if the software has caused the fault and the hardware is good. Also, if there is a Signal Processor fault, it is possible the POST will not detect this. (Refer to the system release Diagnostics Guide for more information about POST.)

dlgHiBoardStatusChanged

MIB: Hardware Information MIB Module (*dlghwinf.mib*)

Condition: Change of a board state from *OK* to *failed* or from *failed* to *OK*

Remarks: This OID identifies the Intel Dialogic or Intel NetStructure board to which it applies by the board's device name. A trap is also sent if the administrative status changes other than by a set. If the board's state changes from *OK* to *failed*, perform the following steps:

1. You (or an automated program) should stop the board that has failed. The board is stopped through the use of the dlgHiIdentAdminStatus. This can be found in the dlghwinf.mib. You would set this value to stop pending state (7). This will stop the board.
2. Once the board is stopped, you can run the diagnose function. This is done by setting the dlgHiIdentAdminStatus to diagnose (5). While the board is in the diagnose state, it will not accept any other commands. The diagnose state runs the full power on self test (POST). For more information about POST, refer to the Diagnostics Guide for the system release.
3. Once the POST is complete, a trap will be sent to the SNMP manager. The dlgHiBoardStatus changed trap will be received. If there is an error during POST, it will be sent back in the trap.
4. If the POST did not return an error, then you can start the board. This is done by setting the dlgHiIdentAdminStatus to start-pending. Another trap will be sent up once the board has moved into the Started state.
5. If the POST did return an error, this means the board failed the POST and should be replaced. It should not be started.

Note: If this process is automated, certain safeguards should be put in place to make sure a board is not restarted too often. For example, if there is a software failure that causes the board to fail immediately after starting, it is possible that the process will continually restart the board. This will cause a large performance hit on the system. When automating the process, it is important to keep a counter on how often a board is restarted. This counter should be checked, and if a board is started too often within a specified amount of time, the restart for that board should be disabled. This will have to be implemented by the programmer as part of the process automation.

dlgHiTestTrap

MIB: Hardware Information MIB Module (*dlghwinf.mib*)

Condition: Setting of dlgHiIdentTestTrapEnable 0 to a value of 1.

Remarks: This trap can be sent by setting the dlgHiIdentTestTrapEnable OID to a value of 1 (using any MIB browser) to verify that the hardware agent is capable of communicating with the network management station.

dlgDsx1Alarm

MIB: Digital Service Level 1 (DS-1) Line Interface (*dlgds1.mib*)

Condition: Occurrence of new alarm for T-1 or E-1 interface (alarms include Red (101), Blue (102), Yellow (103), E-1 (104), and end of alarm (100)).

Remarks: In most cases, an alarm indicates a problem with either the network trunk or with the configuration of the Intel Dialogic board's front-end. For example, an alarm might indicate that the Intel Dialogic board is configured to provide clocking, which would conflict with the clocking provided by the network trunk.

dlgIsdnDChanged

MIB: ISDN Configuration and Statistics (*dlgisdn.mib*)

Condition: Change in the operational status of a D-channel's link access protocol (LAPD).

Remarks: This trap may or may not indicate a problem depending on its value for a given event.

dlgIsdnBChanged

MIB: ISDN Configuration and Statistics (*dlgisdn.mib*)

Condition: Change in the operational status of a B-channel.

Remarks: This trap may or may not indicate a problem depending on its value for a given event.

4.2 Resolving an Application Failure Using the SNMP Agent

If a telephony application fails, the remote monitoring and control facilities provided by the SNMP agent can help you to resolve the problem. When an application fails, take the following diagnostic steps:

1. Verify that all the Intel Dialogic and Intel NetStructure boards are still running.

   The dlgHiIdentServiceStatus OID in the Hardware MIB Module (*dlghwinf.mib*) MIB indicates the current status of the boards.

   If the boards are running, proceed to the next step. If the boards are not running, restart them using the dlgHiIdentServiceStatus OID.

2. Verify that the Intel Dialogic and Intel NetStructure boards used by the application are still running.

   The dlgHiIdentOperStatus OID in the Hardware Information MIB Module (*dlghwinf.mib*) MIB indicates the board's current status. The possible values for this OID are *OK*, *Failed*, *Degraded*, and *Other.*

3. If the board is not running, you may want to attempt to diagnose why the board stopped running before restarting it. Refer to the *DM3 Diagnostic Utilities Reference Guide* or go to http://developer.intel.com/design/telecom/support/. To reset the system, refer to the instructions given in Section 2.3, "Stopping and Starting the System", on page 16.

4.3 Monitoring Errors and Lost Messages

Note: The following information applies only to Intel Dialogic Springware boards.

Errors and lost messages are important indicators of a telephony application's stability. Errors are generated by the Intel Dialogic host drivers or board firmware when a request is made for a function that they cannot perform. Lost messages indicate malfunctioning of the driver or firmware. The presence of errors and lost messages indicates a potential problem that should be addressed.

You can monitor errors and lost messages with the OIDs specified in Table 1. All of the OIDs listed in Table 1 are contained in the Springware Performance MIB (*dlgsrprf.mib)*. They are refreshed periodically for an interval that can be specified with the dlgPsPollingInterval OID in the *dlgsrprf.mib*.

Table 1. OIDs for Critical Errors

OID	Description
dlgPsCurrentLostMsgToFW	Number of messages sent from the driver to the firmware that were lost.
dlgPsCurrentLostMsgFromFW	Number of messages sent from the firmware to the driver that were lost.
dlgPsCurrentFWErrorMsgs	Number of error messages generated by the firmware.
dlgPsCurrentDrvErrorMsgs	Number of error messages generated by the driver.

If errors or lost messages occur, resetting the system may clear the problem; for instructions, see Section 2.3, "Stopping and Starting the System", on page 16. If you continue to encounter errors or lost messages, take note of the frequency and types of errors in a given interval and report the problem to your telephony application developer.

Glossary

API: Application Programming Interface. A set of ready-to-use functions that provide the basis for a method of programming a user application.

Board: A Board is a physical board installed in the system (typically the managed node). A board may be made up of one or more devices, but each one of those devices shall have the same Board ID.

Community: An entity that contains one agent and one or more managers, and is named by the string of octets.

Device: A device is whatever the MIB Module creators choose it to be. It can be a board, or it can be a channel. That is up to the MIB Module. The Intel Dialogic agent is not concerned with what a device is.

Enterprise: Area for delegation of subtrees to other organizations.

Managed Node: The system that is being remotely monitored and has SNMP agents installed.

Management Station: (also called management node): The system that has the manager (management application) installed.

Manager: The management application which monitors and/or administers a remote system, such as HP OpenView* Network Node Manager.

Master Agent: The primary interface between the network manager and the subagents. The master agent acts as a request scheduler and dispatcher for all subscribed subagents. The subagents send traps to the master agent, which are then forwarded to the manager.

MIB: Management Information Base. Specification containing definitions of management information so that networked systems can be remotely monitored, configured and controlled.

NMS: Network Management Station. A dedicated workstation that gathers and stores network performance data. The NMS gets the data from network nodes (computers) running network agent software that enables them to collect the data.

OID: Object Identifier. SNMP uses an identification scheme found in ASN.1 to uniquely identify items used throughout SNMP. An identifier in this scheme is called an object identifier.

SNMP: Simple Network Management Protocol. A simple protocol which uses either UDP, TCP/IP, or IPX (depending on the operating system) to transmit messages between a manager and an agent to perform network management.

SNMP Agent: This SNMP subagent supports the Management Information Base (MIB) module and provides manageability to various Intel Dialogic applications or components within a system. The subagents interact with the Master Agent using SNMP.

SNMP Master Agent: Acts as a relay/multiplexor in its communication with subagents, and also as an agent in servicing requests from SNMP managers.

Trap: An event that is sent by the agent asynchronously. However, the manager does have control over whether traps are sent or not.

Index

E

F

G

H

I

L

M

N

O

P

S

T

U

V