



# Dialogic® System Configuration Guide

---

*August 2007*

Copyright © 2007, Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation.

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. **Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Dialogic, Diva, Eicon, Eicon Networks, Eiconcard and SIPcontrol, among others, are either registered trademarks or trademarks of Dialogic. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement. Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. Other names of actual companies and products mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Publication Date: August 2007

Document Number: 05-2519-001

# Contents

---

	<b>Revision History</b> .....	9
	<b>About This Publication</b> .....	10
	Purpose .....	10
	Applicability .....	10
	Intended Audience .....	10
	How to Use This Publication .....	11
	Related Information .....	11
<b>1</b>	<b>Configuration Overview</b> .....	13
	1.1 Major Configuration Steps .....	13
	1.2 The Configuration Process .....	14
	1.3 Configuration/Administration Tools .....	15
	1.3.1 Command Line Interface (CLI) .....	15
	1.3.2 Simple Network Management Protocol (SNMP) .....	15
	1.4 System Performance Tuning .....	16
<b>2</b>	<b>Configuration Details</b> .....	17
	2.1 Configuration File Sets .....	17
	2.2 Media Loads .....	18
	2.2.1 Features Supported .....	18
	2.2.2 Flexible Routing Configuration .....	19
	2.3 CT Bus (TDM) Clocking .....	19
	2.3.1 Primary Clock Fallback .....	19
	2.4 Dialogic® HMP Software Clocking and Fallback .....	20
	2.5 Transparent Data Routing for 3G-324M .....	21
	2.5.1 Transparent Mode Overview .....	21
	2.5.2 Enabling Transparent Mode on a Channel .....	22
	2.5.3 Enabling Transparent Mode on a System .....	22
	2.6 Preserving Data in User Configuration Files .....	23
	2.7 Modifying the FCD File By Editing the CONFIG File .....	24
	2.7.1 Editing the CONFIG File .....	24
	2.7.2 Generating the FCD File .....	24
<b>3</b>	<b>Command Line Interface (CLI) Basics</b> .....	27
	3.1 CLI Overview .....	27
	3.2 Accessing the CLI .....	27
	3.3 Default Usernames and Passwords for CLI Users .....	28
	3.4 CLI Commands by User Role .....	28
	3.5 CLI Help .....	29
	3.6 CLI Account Management Commands .....	30
	3.7 CLI Show Command .....	31
	3.8 CLI Configure (Conf) Command .....	31
	3.9 Troubleshooting .....	32
<b>4</b>	<b>Command Line Interface (CLI) Reference</b> .....	33

## Contents

4.1	CLI Reference Overview . . . . .	33
4.2	CLI Configure (Conf) Command Reference . . . . .	33
4.3	CLI Show Command Reference . . . . .	35
<b>5</b>	<b>Configuration Procedures Using CLI . . . . .</b>	<b>37</b>
5.1	Assumptions and Prerequisites . . . . .	37
5.2	Order of Procedures . . . . .	38
5.3	Showing Current Configuration Settings . . . . .	38
5.4	Stopping Media Services . . . . .	38
5.5	Activating a License . . . . .	39
5.6	Configuring PSTN Network Interfaces . . . . .	40
5.7	Configuring the TDM Bus . . . . .	42
5.8	Configuring System-Level Bridging and Streaming . . . . .	43
5.9	Configuring Device-Level Bridging and Streaming . . . . .	44
5.10	Restarting Media Services . . . . .	46
<b>6</b>	<b>SNMP Basics . . . . .</b>	<b>47</b>
6.1	SNMP Overview . . . . .	47
6.2	MIBs . . . . .	48
6.3	System Requirements . . . . .	50
6.3.1	Requirements for Remote Management Stations . . . . .	50
6.3.2	Requirements for Local Managed Nodes . . . . .	52
6.4	Starting and Stopping the SNMP Agent . . . . .	52
6.4.1	Stopping the SNMP Agent . . . . .	52
6.4.2	Starting the SNMP Agent . . . . .	52
6.4.3	Changing the SNMP Listening Port . . . . .	52
6.4.4	Changing the Trap Listening Port . . . . .	53
6.5	Username and Password . . . . .	53
6.5.1	SNMP Version 1 and 2c . . . . .	53
6.5.2	SNMP Version 3 . . . . .	54
6.6	Troubleshooting . . . . .	54
<b>7</b>	<b>SNMP MIB Reference . . . . .</b>	<b>57</b>
7.1	SNMP MIB Reference Overview . . . . .	57
7.2	Public MIB 2 Module Reference . . . . .	58
7.3	Public DS1 Module Reference . . . . .	59
7.4	Private Bridge Device Module Reference . . . . .	60
7.5	Private CT Bus Device Module Reference . . . . .	60
7.6	Private External Interface Module Reference . . . . .	60
7.7	Private Hardware Device Module Reference . . . . .	60
7.8	Private Hardware DM3 Device Module Reference . . . . .	61
7.9	Private Hardware DM3 Trunk Module Reference . . . . .	61
7.10	Private License Module Reference . . . . .	61
7.11	Private Media Control Protocol Module Reference . . . . .	61
7.12	Private RTP Configuration Module Reference . . . . .	61
7.13	Private System Environment Module Reference . . . . .	61
<b>8</b>	<b>Configuration Procedures Using SNMP . . . . .</b>	<b>63</b>
8.1	Assumptions and Prerequisites . . . . .	63
8.2	Order of Procedures . . . . .	64
8.3	Configuring Traps . . . . .	64

8.4	Stopping Media Services . . . . .	66
8.5	Activating a License . . . . .	66
8.6	Configuring PSTN Network Interfaces . . . . .	67
8.7	Configuring the TDM Bus . . . . .	67
8.8	Configuring System-Level Bridging and Streaming . . . . .	68
8.9	Configuring Device-Level Bridging and Streaming . . . . .	68
8.10	Restarting Media Services . . . . .	70
<b>9</b>	<b>System Performance Tuning . . . . .</b>	<b>71</b>
9.1	Tuning a System with Dialogic® HMP Software . . . . .	71
9.1.1	High Channel Density Recommendations . . . . .	71
9.1.2	Optimizing NIC Driver Settings . . . . .	72
9.1.3	Disabling High Precision Event Timer (HPET) . . . . .	72
9.2	Tuning a System with Dialogic® HMP Interface Boards . . . . .	72
9.2.1	Disabling Driver Status Messages . . . . .	73
<b>10</b>	<b>Echo Cancellation . . . . .</b>	<b>75</b>
10.1	Echo Cancellation Background Information . . . . .	75
10.1.1	Overview . . . . .	75
10.1.2	Acoustic Echo . . . . .	75
10.1.3	Electrical Echo . . . . .	76
10.2	Echo Cancellation Features . . . . .	77
10.3	Using Echo Cancellation . . . . .	78
10.4	Configuring Echo Cancellation . . . . .	78
10.4.1	Echo Cancellation on IP Media Devices . . . . .	79
10.4.2	Echo Cancellation on Voice Devices . . . . .	80
10.4.3	Echo Cancellation on Conferencing Devices . . . . .	80
10.4.4	Echo Cancellation on CSP/Voice Devices . . . . .	82
10.5	Echo Cancellation Sample Scenarios . . . . .	82
10.5.1	DTI and IP Gateway Example . . . . .	83
10.5.2	DTI and IP Conferencing Example . . . . .	83
10.5.3	DTI and CSP/Voice Example . . . . .	84
10.5.4	DTI and Voice Example . . . . .	84
10.5.5	IP and IP Connection Example . . . . .	84
10.5.6	IP and CSP/Voice Example . . . . .	84
10.5.7	IP and Voice Example . . . . .	85
<b>11</b>	<b>CONFIG File Details . . . . .</b>	<b>87</b>
11.1	CONFIG File Formatting Conventions . . . . .	87
11.2	CONFIG File Sections . . . . .	88
11.3	[NFAS] Section . . . . .	88
11.4	[CHP] Section . . . . .	89
11.5	[TSC] Section . . . . .	90
<b>12</b>	<b>CONFIG File Parameter Reference for Dialogic® HMP Software . . . . .</b>	<b>93</b>
12.1	[0x2c] Echo Cancellation Parameters . . . . .	93
12.2	[0x3b] Conferencing Parameters . . . . .	95
12.3	[IPVSC] IP Media Parameters . . . . .	96
12.4	[decoder] Speed Control Parameter . . . . .	100
12.5	[sigDet] Parameters . . . . .	100

## Contents

<b>13</b>	<b>CONFIG File Parameter Reference for Dialogic® HMP Interface Boards</b> . . . . .	<b>103</b>
13.1	[0x2c] Echo Cancellation Parameters. . . . .	104
13.2	[lineAdmin.x] Parameters (Digital Voice) . . . . .	105
13.3	[NFAS] Parameters. . . . .	113
13.4	[NFAS.x] Parameters . . . . .	114
13.5	[CCS] Parameters. . . . .	116
13.6	[CHP] Parameters. . . . .	121
13.7	[CHP] ISDN Protocol Variant Definitions . . . . .	122
13.8	[TSC] Parameters. . . . .	130
13.9	[TSC] defineBSet Parameters . . . . .	130
13.10	[sigDet] Parameters . . . . .	137

# Figures

---

1	Clock Fallback . . . . .	20
2	General SNMP Architecture . . . . .	48
3	Electrical Echo and the Hybrid. . . . .	76
4	T1/E1 Gateway Example. . . . .	83
5	DTI and IP Conferencing Example . . . . .	83
6	DTI and CSP/Voice Example. . . . .	84
7	DTI and Voice Example. . . . .	84
8	IP and IP Connection Example . . . . .	84
9	IP and CSP/Voice Example . . . . .	85
10	IP and Voice Example . . . . .	85

## Tables

---

1	CLI Usernames and Passwords . . . . .	28
2	CLI Users and Command Privileges . . . . .	29
3	Account Management Commands . . . . .	30
4	Supported MIBs . . . . .	49
5	Version 1 and 2c Community Names . . . . .	53
6	Version 3 Usernames and Passwords . . . . .	54
7	MIB 2 Conformance Table . . . . .	58
8	DS1 MIB Conformance Table . . . . .	59
9	Supported Traps . . . . .	65



# Revision History

---

This revision history summarizes the changes made in each published version of this document.

Document No.	Publication Date	Description of Revisions
05-2519-001	August 2007	Initial version of document.

# About This Publication

---

The following topics provide more information about this publication:

- [Purpose](#)
- [Applicability](#)
- [Intended Audience](#)
- [How to Use This Publication](#)
- [Related Information](#)

## Purpose

This document provides information for configuring Dialogic® Host Media Processing (HMP) Software and Dialogic® HMP Interface Boards (Dialogic® DNI boards) in a Linux system. Configuration procedures are included, as well as descriptions of configuration files and configuration parameters. This document also provides system performance tuning guidelines and recommendations.

## Applicability

This document (05-2519-001) is originally published for Dialogic® Host Media Processing Software Release 3.1LIN.

This document may also be applicable to other software releases (including service updates) on Linux operating systems. Check the Release Guide for your software release to determine whether this document is supported.

## Intended Audience

This publication is intended for the following audience:

- System, application, and technology developers
- Toolkit vendors
- Value Added Resellers (VARs) and system integrators
- System and network administrators

This document assumes that you are familiar with the Linux operating system and have experience using the C programming language.

## How to Use This Publication

The information in this document is organized as follows:

- [Chapter 1, “Configuration Overview”](#) describes the major configuration steps and introduces the tools available for configuration.
- [Chapter 2, “Configuration Details”](#) provides more information about configuration topics such as configuration file sets, CT Bus clocking, and Dialogic® HMP Software clocking and fallback.
- [Chapter 3, “Command Line Interface \(CLI\) Basics”](#) provides an overview of the CLI including how to access the tool, default usernames and passwords, and more.
- [Chapter 4, “Command Line Interface \(CLI\) Reference”](#) provides a reference to commonly used commands in the CLI.
- [Chapter 5, “Configuration Procedures Using CLI”](#) contains detailed procedural information for configuring a system using the CLI tool.
- [Chapter 6, “SNMP Basics”](#) provides an overview of SNMP including system requirements, default usernames and passwords, and more.
- [Chapter 8, “Configuration Procedures Using SNMP”](#) contains detailed procedural information for configuring a system using SNMP.
- [Chapter 9, “System Performance Tuning”](#) provides recommendations and guidelines for tuning system performance.
- [Chapter 10, “Echo Cancellation”](#) describes echo cancellation support in Dialogic® HMP Software and Dialogic® DNI boards. Background echo cancellation information and configuration information are included.
- [Chapter 11, “CONFIG File Details”](#) provides more information about *.config* (CONFIG) files such as file formatting conventions.
- [Chapter 12, “CONFIG File Parameter Reference for Dialogic® HMP Software”](#) describes parameters in the CONFIG files associated with the Dialogic® HMP Software. Included are a description, a list of values, and configuration guidelines.
- [Chapter 13, “CONFIG File Parameter Reference for Dialogic® HMP Interface Boards”](#) describes parameters in the CONFIG files associated with the Dialogic® DNI boards. Included are a description, a list of values, and configuration guidelines.

## Related Information

Refer to the following sources for more information:

- For information on installation and obtaining licenses, see the *Dialogic® Host Media Processing Software Release 3.1LIN Installation Guide*.
- For information about system requirements, release features, and release documentation, see the *Dialogic® Host Media Processing Software Release 3.1LIN Release Guide*.
- For details on compatibility issues, restrictions and limitations, known problems, and late-breaking updates or corrections to the release documentation, see the *Dialogic® Host Media Processing Software Release 3.1LIN Release Update*.

## ***About This Publication***

- For technical support, see <http://www.dialogic.com/support/>.
- For product information, see <http://www.dialogic.com>.

This chapter provides an overview of major configuration tasks to be performed and tools available to configure the system.

- Major Configuration Steps ..... 13
- The Configuration Process ..... 14
- Configuration/Administration Tools ..... 15
- System Performance Tuning ..... 16

## 1.1 Major Configuration Steps

In a system running Dialogic® Host Media Processing (HMP) Software only, follow these major configuration steps:

1. Showing current configuration settings
2. Stopping media services (**required**)
3. Activating a license (**required**)
4. Modifying the FCD File By Editing the CONFIG File
5. Configuring the Global Call CDP File
6. Restarting media services (**required**)

In a system running Dialogic® Host Media Processing (HMP) Software with Dialogic® HMP Interface Boards (Dialogic® DNI boards), additional steps may be required for your environment. Consider the following:

1. Showing current configuration settings
2. Stopping media services (**required**)
3. Activating a license (**required**)
4. Configuring PSTN network interfaces
5. Configuring the TDM bus
6. Configuring system-level bridging and streaming
7. Configuring device-level bridging and streaming
8. Configuring echo cancellation
9. Modifying the FCD File By Editing the CONFIG File
10. Configuring the Global Call CDP File
11. Restarting media services (**required**)

# 1.2 The Configuration Process

After the Dialogic® HMP Software is installed and the appropriate licensing is obtained, you can begin the configuration process. For information on installation and obtaining licenses, see the *Dialogic® Host Media Processing Software Release 3.1LIN Installation Guide*.

Two tools are available to perform configuration tasks as described in [Section 1.3, “Configuration/Administration Tools”](#), on page 15.

### Showing current configuration settings

Default configuration settings are provided with the software according to the license obtained; for example, Dialogic® HMP Software only or Dialogic® HMP Software with Dialogic® DNI boards. You can modify these default settings as needed to suit your environment.

View current configuration settings before performing any modifications; for command line interface (CLI) tool, see [Section 5.3, “Showing Current Configuration Settings”](#), on page 38.

### Stopping media services

Media services, which include services required to operate Dialogic® HMP Software and Dialogic® DNI boards, must be stopped before configuring most components or subsystems. For CLI, see [Section 5.4, “Stopping Media Services”](#), on page 38; for SNMP, see [Section 8.4, “Stopping Media Services”](#), on page 66.

### Activating a license

You must activate a license before using the software. For CLI, see [Section 5.5, “Activating a License”](#), on page 39; for SNMP, see [Section 8.5, “Activating a License”](#), on page 66.

### Configuring PSTN network interfaces

Change the default settings such as the protocol for each PSTN network interface for the Dialogic® DNI boards in your system. For CLI, see [Section 5.6, “Configuring PSTN Network Interfaces”](#), on page 40; for SNMP, see [Section 8.6, “Configuring PSTN Network Interfaces”](#), on page 67.

### Configuring the TDM bus

Change the default settings for the TDM bus such as TDM bus clock source, primary board, secondary board, and more. For CLI, see [Section 5.7, “Configuring the TDM Bus”](#), on page 42; for SNMP, see [Section 8.7, “Configuring the TDM Bus”](#), on page 67.

### Configuring system-level bridging and streaming

Change the default settings for bridging and streaming at the system level such as system law and idle pattern. For CLI, see [Section 5.8, “Configuring System-Level Bridging and Streaming”](#), on page 43. For SNMP, see [Section 8.8, “Configuring System-Level Bridging and Streaming”](#), on page 68.

### Configuring device-level bridging and streaming

Change the default settings for bridging and streaming per device such as maximum voice streams, maximum data streams, and more. For CLI, see [Section 5.9, “Configuring Device-Level Bridging and Streaming”](#), on page 44. For SNMP, see [Section 8.9, “Configuring Device-Level Bridging and Streaming”](#), on page 68.

### Configuring echo cancellation

Echo cancellation is typically required when Dialogic® HMP devices (for example, IP media, voice, and conferencing) receive media streams from the PSTN via DTI devices on certain boards (Dialogic® DNI300TEPHMP and DNI1200TEPHMP boards). You can enable or

disable echo cancellation, configure echo cancellation tail length, and more; see [Chapter 10, “Echo Cancellation”](#).

Modifying the FCD file by editing the CONFIG file

It is sometimes necessary to adjust the parameters within the FCD file; this is done by editing the associated CONFIG file. The files in a configuration file set (*.pcd*, *.fcd*, and *.config* files) are located in the *data* directory. For more information, see [Section 2.7, “Modifying the FCD File By Editing the CONFIG File”](#), on page 24 and [Section 2.6, “Preserving Data in User Configuration Files”](#), on page 23.

Configuring the Dialogic® Global Call CDP file

If you are using the Dialogic® Global Call protocols, see the *Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide*.

Restarting media services

After performing all configuration procedures for your system, you must restart media services to activate new configuration settings. For CLI, see [Section 5.10, “Restarting Media Services”](#), on page 46. For SNMP, see [Section 8.10, “Restarting Media Services”](#), on page 70.

## 1.3 Configuration/Administration Tools

Two tools are available to configure and monitor the system and, if applicable, the boards in the system. They both have access to the same configuration and monitoring data.

**Note:** These tools replace *config.sh*, which is no longer available.

- [Command Line Interface \(CLI\)](#)
- [Simple Network Management Protocol \(SNMP\)](#)

### 1.3.1 Command Line Interface (CLI)

CLI is a text command-line based tool that provides multi-user access. The CLI agent is included with Dialogic® software. Users enter commands at the command prompt to execute configuration tasks.

For more information on the CLI, see [Chapter 3, “Command Line Interface \(CLI\) Basics”](#), [Chapter 4, “Command Line Interface \(CLI\) Reference”](#), and [Chapter 5, “Configuration Procedures Using CLI”](#).

A CLI script, *cte.pl*, that automates repetitive tasks is included in the demos directory; see the *readme.txt* file for more information on this script.

### 1.3.2 Simple Network Management Protocol (SNMP)

SNMP is a MIB-based tool that provides multi-user access. The SNMP agent software is included with Dialogic® software. Users can use an SNMP MIB browser to modify appropriate MIBs for configuration. SNMP MIB browser is not included with Dialogic® software.

## **Configuration Overview**

For more information on SNMP, see [Chapter 6, “SNMP Basics”](#), [Section 7, “SNMP MIB Reference”](#), on page 57, and [Chapter 8, “Configuration Procedures Using SNMP”](#).

## **1.4 System Performance Tuning**

This guide provides recommendations and guidelines on tuning system performance. For more information, see [Chapter 9, “System Performance Tuning”](#).



The following topics provide more information about the configuration process:

- Configuration File Sets ..... 17
- Media Loads ..... 18
- CT Bus (TDM) Clocking..... 19
- Dialogic® HMP Software Clocking and Fallback..... 20
- Transparent Data Routing for 3G-324M..... 21
- Preserving Data in User Configuration Files..... 23
- Modifying the FCD File By Editing the CONFIG File..... 24

## 2.1 Configuration File Sets

The set of files associated with a specific configuration all have the same name; only the extensions (*.pcd*, *.fcd* and *.config*) differ. A set of these files with the same name are used for a specific board type. The board type can include a single board or a group of similar boards. Depending on the board type and the protocol that the board will use, a specific FCD and PCD file are downloaded to that board. If the FCD file needs to be modified, the CONFIG file in that same set is used.

The files associated with configuration file sets include:

### CONFIG File

The CONFIG file (*.config*), located in the *data* directory under INTEL\_DIALOGIC\_DIR (the environment variable for the directory in which the software is installed), contains the modifiable parameter settings used to configure board components. For additional information about CONFIG files, see [Chapter 11, “CONFIG File Details”](#).

### Feature Configuration Description (FCD) File

An FCD file (*.fcd*), located in the *data* directory under INTEL\_DIALOGIC\_DIR, must be downloaded to each board in the system. The purpose of the FCD file is to adjust the settings of the components that make up each product. For example, the FCD file may contain instructions to set certain country codes, or may send messages that configure the Telephony Service Provider (TSP) component to operate with a particular network protocol.

The FCD file defines a simple message form that the downloader parses and sends to a specific component. These parameters are sent to a component within a message and can be thought of as configurable *features* of a component. The FCD file is created automatically from the associated CONFIG file during the board initialization process. For information about changing FCD file parameters, see [Section 2.7, “Modifying the FCD File By Editing the CONFIG File”](#), on page 24.

**Note:** The FCD file should not be edited directly. If parameters require modification, the changes are made by editing the associated CONFIG file. Also, an FCD file should not be copied from another directory to the *data* directory.

## Configuration Details

### Product Configuration Description (PCD) File

A PCD file (*.pcd*), located in the *data* directory under INTEL\_DIALOGIC\_DIR, must be downloaded to each board in the system. The purpose of the PCD file is to determine the software components your system will use. It defines the product by mapping download object files to specific processors, configuring the kernel for each processor and setting the number of component instances to run on each processor.

**Note:** The PCD file should not be modified by the user.

An example of a configuration file set for a Dialogic® DNI601TEPHMP board using QSIG E1 on both trunks is as follows:

- *ghmpl1\_hmpdsb\_2\_qsige1.config*
- *ghmpl1\_hmpdsb\_2\_qsige1.fcd*
- *ghmpl1\_hmpdsb\_2\_qsige1.pcd*

## 2.2 Media Loads

Media loads are pre-defined sets of features. A media load consists of a configuration file set (PCD, FCD, and CONFIG files) and an associated firmware load that are downloaded to each board.

- [Features Supported](#)
- [Flexible Routing Configuration](#)

### 2.2.1 Features Supported

This section describes the features supported by the media loads for the following Dialogic® HMP Interface Boards (Dialogic® DNI boards): DNI300TEPHMP, DNI601TEPHMP, and DNI1200TEPHMP.

The media loads and features supported are:

#### NETWORKONLY

Dialogic® DNI300TEPHMP and DNI1200TEPHMP boards support a media load called NETWORKONLY. This media load supports network interface functionality only. Media processing functionality such as tone detection and tone generation, call progress analysis, and echo cancellation are provided by the Dialogic® HMP Software.

#### HMPL1

Dialogic® DNI601TEPHMP boards support a media load called HMPL1. This media load supports network interface functionality as well as some media processing functionality required for call control signaling, namely tone detection, tone generation, and call progress analysis. In addition, echo cancellation is performed on media received from the T1/E1 interface prior to that media being sent to the CT Bus and/or the Dialogic® HMP Software. Therefore, echo cancellation capabilities of other resources, such as IP media and conferencing, are not required for connections between these resources and T1/E1 interfaces on the DNI601TEPHMP boards. Other media processing resources are provided by the Dialogic® HMP Software.

## 2.2.2 Flexible Routing Configuration

Dialogic® DNI boards support flexible routing configuration.

### Flexible routing configuration

With flexible routing, the resource devices (voice/fax) and network interface devices are independent, which allows exporting and sharing of the resources. All resources have access to the TDM bus. Each voice resource channel device and each network interface time slot device can be independently routed on the TDM bus.

## 2.3 CT Bus (TDM) Clocking

The system provides clocking and clock fallback to maintain timing in the event that the current clock source fails. The following provides reference information about the type of clock fallback:

- [Primary Clock Fallback](#)

### 2.3.1 Primary Clock Fallback

For the following discussion, refer to [Figure 1, “Clock Fallback”](#), on page 20 for an illustration of the CT Bus clocking concepts.

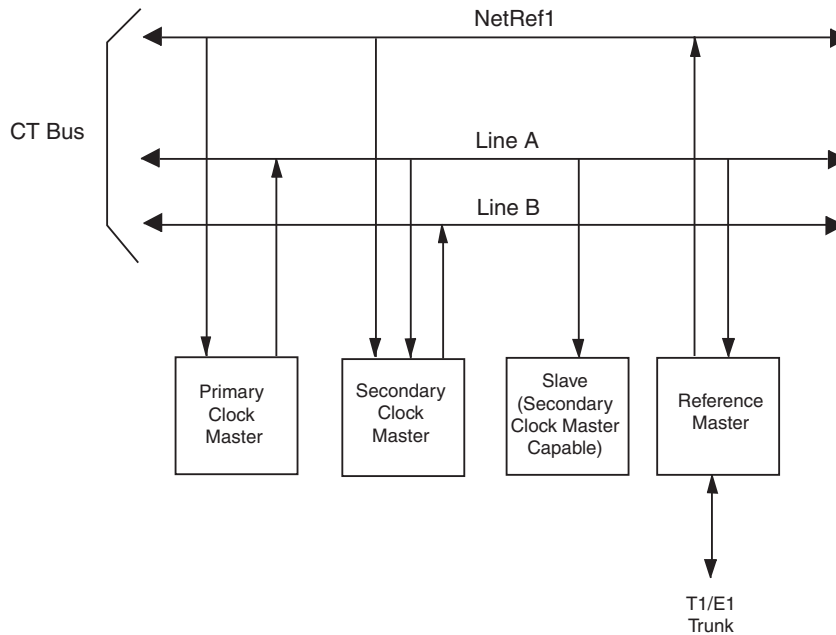
The Primary Clock Master is a device (board) that provides timing to all other devices attached to the bus. The Primary Clock Master drives bit and framing clocks for all of the other boards (slaves) in the system via CT Bus Line A or Line B. This bus clocking is synchronized to either the board's internal oscillator or, preferably, to the NetRef1 line which provides a timing reference (8 kHz) derived from a T1 or E1 interface signal.

The timing reference is provided by the Reference Master board. A T1 or E1 trunk on the Reference Master board is the source for the T1 or E1 interface signal from which the 8 kHz timing reference is derived. The timing reference is sent from the Reference Master board to the NetRef1 line.

In addition, a Secondary Clock Master can be defined as a backup for the same purpose. This board, like the Primary Clock Master, is capable of driving the bit and framing clocks for all of the other boards in the system. The Secondary Clock Master uses whichever CT Bus line (A or B) is not defined for the Primary Master Clock. If the system senses a failure of the Primary Clock Master, the system will cause the clock source to fall back to the Secondary Clock Master. The Secondary Clock Master, like the primary, also provides clocking that is synchronized to either the board's internal oscillator or, preferably, to the NetRef1 line.

## Configuration Details

Figure 1. Clock Fallback



In the case where the Primary Clock Master has failed, and the clock source falls back to the Secondary Clock Master, the system selects a new Secondary Clock Master, assuming that a board in the system meets the criteria for a clock master.

If the Primary Clock Master fails and no Secondary Clock Master has been defined, the system will automatically choose another board to be Primary Clock Master, if another board in the system is clock master capable.

Both the Primary and Secondary Clock Masters can be configured by the user; default values are assigned by the system. For instructions on modifying the clock source, see [Section 5.7, “Configuring the TDM Bus”](#), on page 42 when using the CLI, and see [Section 8.7, “Configuring the TDM Bus”](#), on page 67 when using SNMP.

## 2.4 Dialogic® HMP Software Clocking and Fallback

Dialogic® HMP Interface Boards include a bridge device that can stream data between the boards connected to the CT Bus and Dialogic® HMP Software, and additionally is capable of providing clocking to HMP software. The clocking provided to HMP software from an HMP interface board is derived from CT Bus clocking. In an HMP system that contains interface boards, one board is selected as the Primary Clock Master for HMP software. Additional interface boards serve as backup clocking sources for HMP software, via a fallback list, should the Primary Clock Master fail.

In a Dialogic® HMP Software system with no interface boards or no active boards, clocking for HMP software is derived from an alternate system clock source. This alternate system clock source also serves as the last clock source for HMP software on the fallback list.

Clocking is configured as part of the bridge device configuration. When using the CLI, see [Section 5.9, “Configuring Device-Level Bridging and Streaming”](#), on page 44 for more information. When using SNMP, see [Section 8.9, “Configuring Device-Level Bridging and Streaming”](#), on page 68.

## 2.5 Transparent Data Routing for 3G-324M

Transparent data routing (transparent mode) for 3G-324M is described in the following topics:

- [Transparent Mode Overview](#)
- [Enabling Transparent Mode on a Channel](#)
- [Enabling Transparent Mode on a System](#)

### 2.5.1 Transparent Mode Overview

Transparent data routing or transparent mode is used when connecting a DTI device supporting 3G-324M to a 3G-324M device. When these two devices are connected, H.223 data is passed between them transparently.

**Note:** Dialogic® DNI601TEPHMP boards do not support transparent mode and therefore do not support 3G-324M; Dialogic® DNI300TEPHMP and DNI1200TEPHMP boards do support this functionality.

The DTI device, which represents the T1/E1 bearer channel interface, passes data between a bearer channel on the T1/E1 interface and the physical CT Bus. Data is then passed between timeslots on the physical CT Bus and the HMP soft CT Bus as needed using the bridging capability provided by the Dialogic® DNI boards.

When voice or audio is passed, it may be processed such as being converted to linear format or to a common system law format (A-law, mu-Law). When data other than voice or audio such as H.223 data supporting 3G-324M is passed, no processing should occur; that is, data should be passed transparently.

Use the `gc_SetConfigData()` function in the Global Call API library to set the DTI device or Global Call device to pass data transparently. See [Section 2.5.2, “Enabling Transparent Mode on a Channel”](#), on page 22 for more information.

With the command line interface (CLI) utility, use the **conf hardware bridge** command to configure the bridge device to support the desired number of data streams for your environment. The default setting for data streams is 0 for Dialogic® DNI boards. The default setting for voice streams is 253 for Dialogic® DNI boards. As an example, for an application supporting 120 3G-324M channels, ensure that a minimum of 120 data streams have been configured. See [Section 5.9, “Configuring Device-Level Bridging and Streaming”](#), on page 44 for more information when using

## Configuration Details

the CLI. See [Section 8.9, “Configuring Device-Level Bridging and Streaming”](#), on page 68 for more information when using SNMP.

A DTI channel must be configured to operate in transparent mode when supporting 3G-324M. DTI channels may be individually configured in real time per call via the `gc_SetConfigData()` function. Dialogic® HMP Software will automatically select and use bridge data streams when connecting DTI devices to 3G-324M devices.

### 2.5.2 Enabling Transparent Mode on a Channel

The `gc_SetConfigData()` function, which supports the Global Call Real Time Configuration Management (RTCM) feature, allows the configuration of transparent mode on a PSTN channel basis at runtime. When transparent mode is **enabled**, there is no processing on the front end data coming in from the PSTN network before passing it to the TDM bus, nor is there any processing of data being transmitted to the PSTN network from the TDM bus. The default setting is transparent mode **disabled**.

The `gc_SetConfigData()` function uses a `GC_PARM_BLK` structure that contains the configuration element. The `GC_PARM_BLK` is populated using the `gc_util_insert_parm_val()` function.

To enable/disable transparent mode on a channel, the following setID/parmID pair is used:

- `CCSET_DM3FW_PARM` is the setID
- `CCPARAM_TRANSPARENTMODE` is the parmID, with values:
  - `CCDM3FW_TRANSPARENTMODE_DISABLE`
  - `CCDM3FW_TRANSPARENTMODE_ENABLE`

The size of the parameter is a char or UINT8.

Once the `GC_PARM_BLK` has been populated with the desired values, the `gc_SetConfigData()` function can be issued to perform the configuration. Use the target type `GCTGT_CCLIB_CHAN`.

To configure transparent mode on a system wide basis, you can add a parameter to the `Hmp.Uconfig` user configuration file. For more information, see [Section 2.5.3, “Enabling Transparent Mode on a System”](#), on page 22.

### 2.5.3 Enabling Transparent Mode on a System

To configure transparent mode on a system wide basis, add the following section to the `Hmp.Uconfig` file. Set the parameter to 1 to enable transparent mode on all PSTN channels and to 0 to disable transparent mode on all PSTN timeslots. Default is disabled.

```
[0x5a]
SetParm=0x1221, 1 ! 0 = normal, 1 = 3G clear channel mode
```

For more information on the `Hmp.Uconfig` file, see [Section 2.6, “Preserving Data in User Configuration Files”](#), on page 23.

To configure transparent mode on a channel basis, see [Section 2.5.2, “Enabling Transparent Mode on a Channel”](#), on page 22.

## 2.6 Preserving Data in User Configuration Files

Configuration settings unique to your environment can be preserved and re-applied whenever any Dialogic® HMP Software license is changed or reactivated.

Previously, customized settings in the `<licensename>.config` file (the default `.config` file created during license generation) were lost every time a new Service Update was installed or a new Dialogic® HMP Software license was activated. These customized settings then had to be re-entered.

A new file called the user configuration file, `Hmp.Uconfig`, is introduced. This file, which has the same format as the `<licensename>.config` file, contains only the parameters and parameter values that are customized for your environment.

Rather than editing the generated `<licensename>.config` file, you create a separate `Hmp.Uconfig` file in the data directory under `INTEL_DIALOGIC_DIR`, the environment variable for the directory in which the HMP software is installed. The contents of the `Hmp.Uconfig` file are not overwritten but are merged into the `<licensename>.config` file whenever a new Service Update is installed or a new HMP license is activated. You should also make a copy of the `Hmp.Uconfig` file and save it in a safe location as a backup.

If you do not want the customized parameters in the `Hmp.Uconfig` file to be merged into the generated `<licensename>.config` file, you should either remove the `Hmp.Uconfig` file from the data directory or rename it.

### Example

The following is a sample `Hmp.Uconfig` file, where the default AGC setting has been changed and a new parameter has been added:

```
[encoder]
SetParm=0x400,0    !AGC Enabled (1=Enable, 0=Disable)

[0xe]
SetParm=0xb17,4    !QFC3_ PrmResponseTimeout default 3 seconds
```

The following is a sample excerpt of a merged `<licensename>.config` file. It shows the new AGC setting in the `[encoder]` section followed by the default value for AGC, commented by “!<sup>^</sup>”. It also shows a new parameter in `[0xe]`, delimited by the “!<sup>^</sup><add>”.

```
[encoder]
SetParm=0x400,0    !AGC Enabled (1=Enable, 0=Disable)
!^SetParm=0x400,1  !AGC Enabled (1=Enable, 0=Disable)

[0xe]
...
```

## Configuration Details

```
!<add>
SetParm=0xb17,4    !QFC3_ PrmResponseTimeout default 3 seconds
!</add>
```

## 2.7 Modifying the FCD File By Editing the CONFIG File

If the default settings in the FCD files are not appropriate for your configuration, you can modify the FCD file parameters using the CONFIG file and the *fdgen* utility. Modifications can be made at any time prior to starting the system. For each FCD file to be modified, the procedure includes:

1. [Editing the CONFIG File](#)
2. [Generating the FCD File](#)

**Note:** It is recommended that you use the procedure described in [Section 2.6, “Preserving Data in User Configuration Files”](#), on page 23 for modifying FCD file parameters. You create a user configuration file called *Hmp.Uconfig*.

### 2.7.1 Editing the CONFIG File

To edit the CONFIG file:

1. From the command prompt, go to the */usr/dialogic/data* directory and locate the CONFIG file.
2. Using a text editor (for example, *vi*), open the CONFIG file that corresponds to the FCD file you want to modify. By default, the CONFIG file will have the same file name as the FCD file, but with a *.config* extension.
3. Edit the CONFIG file as necessary.
4. For more information on CONFIG file sections and formatting conventions, see [Chapter 11, “CONFIG File Details”](#). For details about CONFIG file parameters, see [Chapter 12, “CONFIG File Parameter Reference for Dialogic® HMP Software”](#) and [Chapter 13, “CONFIG File Parameter Reference for Dialogic® HMP Interface Boards”](#).
5. Save and close the CONFIG file.

Proceed with [Section 2.7.2, “Generating the FCD File”](#), on page 24.

### 2.7.2 Generating the FCD File

In order to generate an FCD file, the corresponding CONFIG file must be converted to an FCD file. The *fdgen* utility converts the CONFIG file into a format that can be read directly by the downloader.

1. From the command prompt, go to the */usr/dialogic/data* directory.



2. Execute *fcngen* as follows:

```
../bin/fcngen -f <input file>.config -o <output file>.fcd
```

OR

```
../bin/fcngen <input file>.config
```

The resulting FCD file is created in the *data* directory. The *-f* and *-o* options are optional. If this command is run with the input file filename only, the default output FCD file will have the same filename as the input file, but with a *.fcd* extension.

**Configuration Details**

# Command Line Interface (CLI) Basics

---

## 3

This chapter provides basic information on accessing and using the Command Line Interface (CLI):

- CLI Overview ..... 27
- Accessing the CLI ..... 27
- Default Usernames and Passwords for CLI Users. .... 28
- CLI Commands by User Role ..... 28
- CLI Help ..... 29
- CLI Account Management Commands ..... 30
- CLI Show Command ..... 31
- CLI Configure (Conf) Command ..... 31
- Troubleshooting ..... 32

### 3.1 CLI Overview

CLI is one of two interfaces that can be used to configure Dialogic® software and Dialogic® hardware. The other is Simple Network Management Protocol (SNMP). They both have access to the same configuration and monitoring data. SNMP is MIB-based, and CLI is text command line-based. For information about using SNMP for configuration management, see [Chapter 6, “SNMP Basics”](#).

When the CLI is running, events are displayed on the CLI terminal.

### 3.2 Accessing the CLI

The CLI agent provides multi-user access for remote and/or local configuration management and monitoring of the Dialogic software and Dialogic hardware. The CLI agent is included with the Dialogic software and is automatically started when the software is started using a system reboot or the `dlstart` command.

Up to three users can access the CLI at the same time. Each login session (except for the “root” account) is controlled by a session timer, so when users forget to log off the system, CLI will do it for them.

## Command Line Interface (CLI) Basics

The CLI agent is a server process that monitors Telnet port 23 by default. (You can specify a different port as needed during the Dialogic software installation.) Users can access the CLI agent using the telnet command from any system supporting the Telnet protocol. For example:

```
> telnet 192.168.12.34
```

If you are already logged on to the system, you can use the loopback IP address to access the CLI:

```
> telnet 127.0.0.1
```

Once you successfully Telnet to the system, you will be presented with a product banner, followed by a login prompt.

After you successfully log in, you can use CLI commands to view and modify software and board settings and parameters, depending on your access level. The CLI command prompt is:

```
CLI>
```

### 3.3 Default Usernames and Passwords for CLI Users

CLI supports three levels of users. Each level has different capabilities and command access within the CLI.

**Table 1. CLI Usernames and Passwords**

Role	Username	Password	Description
Account manager	acctmgr	acctmgr	Allowed to add, delete, and update all user information.
Craftsperson	craft	craft	Allowed to read data except account manager specific data.
Administrator	admin	admin	Allowed to read and write data except account manager specific data.

*Note:* Currently, you must use the “root” username and “public” password to access the CLI.

### 3.4 CLI Commands by User Role

The CLI consists of basic commands used to configure and monitor the software and hardware, for example:

- **show** - To view information
- **conf** - To configure information

The basic commands are then followed by one or more subcommands that describe the type of data or specific data item you wish to view or modify. The following lists the available commands as they apply to a user's role.

**Table 2. CLI Users and Command Privileges**

Command	Account Manager	Craftsperson	Administrator	Description
adduser	Yes	No	No	Add users to the system
clear	Yes	Yes	Yes	Clear display
conf	No	No	Yes	Change parameters
deleteuser	Yes	No	No	Delete users from system
editpword	Yes	Yes	Yes	Change your password
edituser	Yes	No	No	Modify user account data
keepalive	Yes	Yes	Yes	Specify number of seconds to wait between CLI commands before auto logout for this session
list	Yes	No	No	List CLI user accounts
logout	Yes	Yes	Yes	Log off
man	Yes	Yes	Yes	Help
prompt	Yes	Yes	Yes	Change the display prompt for the current session
show	Yes	Yes	Yes	Show parameters and statistics
who	Yes	Yes	Yes	Display users currently logged on to CLI agent

### 3.5 CLI Help

The CLI agent provides context-level help at any point during command input by typing **-h** followed by a carriage return. For example:

```
CLI> show hardware -h
help - expected
      dm3
      dev
      pstn
      bridge
```

For the list of available commands based on your login permission, enter **?** at the command prompt. For example:

```
Login :admin
Password :*****
CLI> ?
adduser      clear      conf      deleteuser  editpword
...
```

## Command Line Interface (CLI) Basics

The keyboard up/down arrows allow you to scroll through the input history buffer, and thus repeat the last line(s) you typed. Using the left and right arrow keys, you can edit the input line for updated values without having to retype the entire line.

### 3.6 CLI Account Management Commands

This section covers the CLI account management commands used to provision and maintain CLI users and permissions.

As stated earlier, there are three levels of users in the CLI. The account manager user is responsible for provisioning and managing all other users with the CLI. The account manager user can add new users, delete users, update user passwords or context (user role), and set user aging parameters. Aging parameters are used to automatically remove passwords and user accounts after a configurable time period and are provided to increase system security.

The commands to create, modify, and delete users have the form:

**[add/edit/delete]user** *[username]* *<parameters>*

where the command prefix is one of **add**, **edit**, or **delete**; *username* identifies the user; and the *parameters* define the managed user and optional settings. The *username* and *parameters* can be entered in any order.

The following lists account management commands.

**Table 3. Account Management Commands**

Command	Parameters	Description	Examples
adduser	[-g <password aging>] [-a <account aging>] [-i <context>] username password	Add user to the system. Default password aging: 60 Default account aging: 180 Contexts are ACCTMGRCTX for account manager, CRAFTCTX for craftsperson, ADMINCTX for administrator.	CLI> adduser -i ADMINCTX foo bar Creates user "foo" with password "bar" and permissions "ADMINCTX"
deleteuser	username	Deletes user "username" account	CLI> deleteuser foo Deletes user "foo"
editpword	password	Change current user password.	CLI> editpword foo foobar Updates current user password to "foobar"
edituser	[-g <password aging>] [-a <account aging>] [-i <context>] [-p <new password>] username	Edits account information for the specified user.	CLI> edituser foo -p bar Changes user "foo" password to "bar"

## 3.7 CLI Show Command

The **show** command is used to view any/all configuration parameters and statistics. It has the form:

```
show <subcommand>
```

where *subcommand* can be a terminating command that results in data output, such as **show system**, or a submenu category such as **show tcp**, which requires additional arguments before output is displayed (for example, **show tcp brief**).

The initial command and subcommand are case sensitive and require complete command entry, for example:

```
CLI> show interface
```

Parameters following the command and subcommand will match to the shortest unique command. For instance, the following are valid inputs to the **interface** subcommand:

```
CLI> show interface e
CLI> show interface Ethernet
CLI> show interface de
CLI> show interface ds
```

If the user enters a non-unique parameter, a help message is displayed with the list of valid parameters. For example:

```
CLI> show interface d
ambiguous option - expected
      desc          Display link/administration status for interfaces
      ds1           DS1 interface information
      ethernet      Ethernet interface information
      media-ethernet Media ethernet interface information
      <cr>          Dump all interface information
```

See [Chapter 4, “Command Line Interface \(CLI\) Reference”](#) for more information on CLI commands.

## 3.8 CLI Configure (Conf) Command

The **conf** command is used to update configuration parameters. It has the general form:

```
conf <subcommand> <parameter> <value>
```

where *subcommand* identifies the group being configured, such as **interface** or **system**. Following the command and subcommand are a simple parameter/value pair or additional subcategories followed by a parameter/value setting.

## Command Line Interface (CLI) Basics

For example, the following is a simple parameter/value setting that updates the system group, field name, to “My NetMedia Box”:

```
CLI> conf system name 'My NetMedia Box'
```

The same rules for the **show** command entry also apply to the **conf** command. The command and subcommand are case sensitive and must be entered as shown. Subcategories and parameters are case insensitive and match on the shortest, most unique input.

See [Chapter 4, “Command Line Interface \(CLI\) Reference”](#) for more information on CLI commands.

## 3.9 Troubleshooting

This section provides information on troubleshooting.

### Restarting CLI Agent

In case of CLI failure, you may need to start the CLI agent manually.

```
dlservices cli start
```

### Capturing CLI Session in Log File

To help troubleshoot configuration issues, it is sometimes helpful to capture your entire CLI session in a log file. To capture the CLI session, you can use the **script** operating system command. For example:

```
bash> script <optional filename>
bash> telnet localhost
cli> ... [perform CLI operations]
cli> logout
bash> exit
bash> more typescript <or optional filename>
```

If you don't provide an argument for script, the default filename is typescript. The script command writes to the buffer until the exit command is issued.



# Command Line Interface (CLI) Reference

---

# 4

This chapter provides reference information for commonly used commands in the Command Line Interface (CLI):

- CLI Reference Overview . . . . . 33
- CLI Configure (Conf) Command Reference . . . . . 33
- CLI Show Command Reference . . . . . 35

## 4.1 CLI Reference Overview

CLI is a command line interface used to configure Dialogic® software and Dialogic® hardware. For basic CLI information such as how to access CLI, see [Chapter 3, “Command Line Interface \(CLI\) Basics”](#).

Information on all supported commands is available via the online help. You can access help by typing `?` at the command prompt. You can also access context-level help at any point during command input by typing `-h` followed by a carriage return.

## 4.2 CLI Configure (Conf) Command Reference

The **conf** command is used to update configuration parameters. Its subcommands are listed as follows:

hardware *<subcommand>*

configures hardware

- dm3 (or hib) – parameters for Dialogic® DM3 devices including Dialogic® HMP Interface Boards. See hardware dm3 for list of values.
- dev – device parameters. See hardware dev for list of values.
- pstn – PSTN network interface parameters. See hardware pstn for list of values.
- bridge – bridging (host streaming interface or HSI) parameters. See hardware bridge for list of values.

hardware bridge *<subcommand>*

configures bridging and streaming parameters

- sys – bridging and streaming system-level parameters. See hardware bridge sys for list of values.
- dev – bridging and streaming device-level parameters. See hardware bridge dev for list of values.

hardware bridge dev *<device ID>* *<subcommand>*

configures bridging and streaming parameters at the device level

## Command Line Interface (CLI) Reference

- dev-enable – enable bridge device
- nbr-userdef – user-defined clock master fallback for Dialogic® HMP Software
- bh-datastreams – maximum board to host data streams
- hb-datastreams – maximum host to board data streams
- bh-voicestreams – maximum board to host voice streams
- hb-voicestreams – maximum host to board voice streams

hardware bridge sys <subcommand>

configures bridging and streaming parameters at the system level

- a-law – idle pattern for a-law
- mu-law – idle pattern for mu-law
- nbrhsi-buffer – number of host streaming interface (HSI) hold buffers

hardware dev <device ID> <subcommand>

configures hardware device parameters

- admin value – set administration status of Dialogic® HMP Interface Boards; that is enabling or disabling a board. Values are: 1 (enabled) and 2 (disabled).

hardware dm3 <DM3 device ID> <subcommand>

configures parameters for a Dialogic® DM3 device including Dialogic® HMP Interface Boards

- logical-id value – logical device number
- fcd value – FCD file
- pcd value – PCD file
- media-load value – media load. Note that this parameter cannot be modified.
- pcmencoding value – PCM encoding

**Note:** It is recommended that you do not modify the FCD file and PCD file values via this command. These values are updated when you configure the protocol for the network interface.

hardware pstn <device ID> <trunk ID> <subcommand>

configures PSTN network interface parameters

- protocol – protocol

interface <subcommand>

configures an interface

- ethernet – ethernet interface information

ip <subcommand>

configures global IP configuration subcommands

- ttl value – default packet time to live

license <subcommand>

configures media license parameters

- directory value – directory location of license file
- activate value – license (file name) to activate on next media start
- oem-function value – function to be invoked for OEM licensing
- oem-library value – library to be loaded for OEM licensing

msml <subcommand>

configures remote media control parameters

- access-list – trusted server list settings. See msml access-list for a list of values.
- http-caching – enable http caching
- ip value – set remote media control ip address

- no – disable MSML remote media control features
- port *value* – set remote media control port
- protocol *value* – enable MSML remote media control protocol
- transport-protocol *value* – set remote media control protocol transport
- schema-validation – enable schema validation

msml access-list <*subcommand*> or *value*

configures trusted server list settings. This command enables the list.

- <trusted ip address> – add this IP address to the trusted server list, if not already added
- no <trusted ip address> – delete this IP address from the trusted access server list

system <*subcommand*>

configures global system and media parameters

- contact *value* – set contact information
- ipmedia *value* – start/stop the IP media subsystem
- location *value* – set system location
- name *value* – set system name
- pstn apply – apply user defined values to system
- hmp-rtp-address *value* – set IP address used by Dialogic® HMP Software to send RTP messages
- tdm bus *value* – Alias for tdm bus is ctbus. Set TDM bus values. See system tdm bus for list of values.

system tdm bus <*subcommand*>

Alias for tdm bus is ctbus. Configures TDM bus values.

- daemon-mode *value* – set Clock Daemon Mode
- primary-line *value* – set TDM bus primary line
- primary-board *value* – set TDM bus primary line board
- secondary-board *value* – set TDM bus secondary line board
- primary-line-clock *value* – set TDM bus primary line clock (clock source)
- netref-provider *value* – set TDM bus NetRef provider board
- netref-clock-rate *value* – set TDM bus NetRef clock rate
- netref-interface *value* – set TDM bus NetRef line interface
- netref-fallback-provider *value* – set TDM bus NetRef fallback provider board
- netref-fallback-interface *value* – set NetRef fallback entry interface
- netref-fallback-enable – set NetRef fallback status

## 4.3 CLI Show Command Reference

The CLI **show** command is used to view any/all configuration parameters and statistics. Its parameters are listed as follows:

arp

shows ARP table

controller

shows controller status and information

- ds1 – ds1 interface

hardware

shows hardware information

## **Command Line Interface (CLI) Reference**

- dm3 (or hib) – parameters for Dialogic® DM3 device including Dialogic® HMP Interface Boards
- dev – device parameters
- pstn – PSTN trunk parameters
- bridge – bridging (host streaming interface or HSI) device parameters

### **icmp**

shows ICMP information

### **interface**

shows interface information

- desc – display link/administration status for interfaces
- ds1 – DS1 interface information
- ethernet – ethernet interface information
- media-ethernet – media ethernet interface information

### **ip**

shows global IP configuration subcommands

- addr – IP address table
- arp – ARP table
- interface – interface settings
- routes – route table entries
- traffic – all IP traffic data

### **license**

shows license information

### **msml**

shows MSML properties

### **system**

shows system information

- tdm bus – Alias for tdm bus is ctbus. Shows TDM bus (or CT Bus) information.

### **tcp**

shows status of TCP connections

- brief – brief
- statistics – statistics

### **udp**

shows status of UDP connections

- brief – brief
- statistics – statistics

### **version**

shows version information such as system name (if defined), form factor, model (or software release), and vendor.

# Configuration Procedures Using CLI 5

---

This chapter provides detailed procedures for each major step in the configuration process using the Command Line Interface (CLI) software to configure the Dialogic® Host Media Processing (HMP) Software and Dialogic® HMP Interface Boards (Dialogic® DNI boards). Some steps may not apply to your system configuration.

- Assumptions and Prerequisites ..... 37
- Order of Procedures ..... 38
- Showing Current Configuration Settings ..... 38
- Stopping Media Services ..... 38
- Activating a License ..... 39
- Configuring PSTN Network Interfaces ..... 40
- Configuring the TDM Bus ..... 42
- Configuring System-Level Bridging and Streaming ..... 43
- Configuring Device-Level Bridging and Streaming ..... 44
- Restarting Media Services ..... 46

## 5.1 Assumptions and Prerequisites

The following assumptions and prerequisites exist regarding the configuration procedures using the CLI:

- All required software, including prerequisites, have been installed and Dialogic® services have been started according to the procedures in the software installation guide supplied with your release.
- The release was installed in the default directory under INTEL\_DIALOGIC\_DIR, the environment variable for the directory in which the software is installed. Command instructions, directory paths and environment variable are shown relative to the default installation directory.
- You have accessed the CLI agent and logged in with the appropriate user account (such as administrator). For more information on the CLI, see [Chapter 3, “Command Line Interface \(CLI\) Basics”](#) and [Chapter 4, “Command Line Interface \(CLI\) Reference”](#).
- If applicable, the Dialogic® Global Call protocols have been installed. The Dialogic® Global Call protocols are provided as part of the release. For information about country dependent parameters associated with a protocol, see the *Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide*.

## Configuration Procedures Using CLI

- If a Dialogic® DNI board is removed from the system and reinserted, you will need to reconfigure the board using the CLI or SNMP utility.

## 5.2 Order of Procedures

The following is a suggested order for performing configuration procedures. Procedures that are required when configuring any system are noted as such. Additional procedures may be required depending on your system. A system may consist of Dialogic® HMP Software only or Dialogic® HMP Software with Dialogic® DNI boards.

1. [Showing Current Configuration Settings](#)
2. [Stopping Media Services](#) (**required**)
3. [Activating a License](#) (**required**)
4. [Configuring PSTN Network Interfaces](#)
5. [Configuring the TDM Bus](#)
6. [Configuring System-Level Bridging and Streaming](#)
7. [Configuring Device-Level Bridging and Streaming](#)
8. [Restarting Media Services](#) (**required**)

## 5.3 Showing Current Configuration Settings

Default configuration settings are provided with the software according to the license obtained for your system; for example, a system with Dialogic® HMP Software only or a system with Dialogic® HMP Software and Dialogic® DNI boards.

Use the **show** command to view configuration settings. See [Section 4.3, “CLI Show Command Reference”](#), on page 35 for a list of all show commands.

You can modify these settings as needed to suit your environment.

## 5.4 Stopping Media Services

One of the assumptions in this chapter is that all required software, including prerequisites, have been installed and Dialogic® services have been started according to the procedures in the software installation guide supplied with your release. As part of this process, media services (which include boards) are automatically started.

Media services must be stopped before configuring most components or subsystems. Therefore, it is recommended that this procedure always be performed. Attempting to configure components or subsystems in a system where media services are not stopped will result in this message:

```
Media service is not in the proper state for operation
```

Follow this procedure to stop media services. This procedure also stops all boards that are enabled.

1. Retrieve the system run status found in the system group using the **show** command. Verify that the run status is active.

```
CLI> show system
```

2. To stop media services, use the **conf** command to update the system ipmedia setting to stop.

```
CLI> conf system ipmedia stop
updated - This action can take a few minutes to complete. Please wait.
```

The updated message is displayed and indicates that the command has been accepted.

Stopping the media services results in a number of events being generated by the system and displayed to any user logged on the CLI (these include system events and possibly bridge device events). Shutdown event notification messages are displayed. Once the media services are stopped, the following message is displayed:

```
ipmedia stop complete
```

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 5.10, “Restarting Media Services”](#), on page 46.

## 5.5 Activating a License

You can activate the verification license supplied with the software or another license you have obtained using this procedure:

1. Verify that the media services and boards are stopped. See [Section 5.4, “Stopping Media Services”](#), on page 38.

2. Retrieve the current license information for your system.

```
show license
```

Information about active license, selected license, license directory, and more is displayed.

3. Specify the path to the license file; that is, the directory where you downloaded the license. The default directory for license files is */usr/dialogic/data*.

```
conf license directory <license directory>
```

## Configuration Procedures Using CLI

4. Specify the license file name and activate the license. The license will be activated on the next media services restart.

```
conf license activate <license file name>
```

Configuring a license file from the CLI involves the license file validation. It may take a few seconds to complete. Upon successful completion, a message similar to the following is displayed:

```
CLI> License verification is in progress. Please wait  
updated - will be effective on the next ipmedia start
```

In case of an invalid license, the following error message is displayed:

```
License activation failed. Reverted to last selected license
```

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 5.10, “Restarting Media Services”](#), on page 46.

## 5.6 Configuring PSTN Network Interfaces

To configure PSTN network interfaces for Dialogic® HMP Interface Boards, follow this procedure:

1. Verify that the media services and boards are stopped. See [Section 5.4, “Stopping Media Services”](#), on page 38.

**Note:** You can configure PSTN network interface settings while media services are active. The configuration is **updated**; however, the configuration is only **applied** (1) after media services are stopped, (2) the **conf system pstn apply** command is issued (see [Step 6](#)), and (3) media services are restarted. Thus, it is recommended that you stop media services before performing the procedure.

2. Retrieve the device ID for the hardware in your system.

```
CLI> show hardware
```

All Dialogic® devices are displayed including device ID, technology, device name, slot ID and more.

3. Retrieve the current settings for the PSTN network interfaces.

```
CLI> show hardware pstn
```

Information about device ID, trunk ID, interface ID, protocol, and more is displayed.



4. Modify a current PSTN network interface setting for a specific device and trunk; for example, modify protocol.

```
CLI> conf hardware pstn <device ID> <trunk ID> protocol <protocol value>
```

You can leave out the <protocol value> to display all supported protocols. Sample command and sample output are:

```
CLI> conf hardware pstn 2 1 protocol
missing option - expected

<integer>          1 - 4ESS
<integer>          2 - 5ESS
<integer>          3 - CAS
<integer>          4 - DMS
<integer>          5 - E1CC
<integer>          6 - NET5
<integer>          7 - NI2
<integer>          8 - NTT
<integer>          9 - QSIGE1
<integer>         10 - QSIGT1
<integer>         11 - R2MF
<integer>         12 - T1CC
<integer>         13 - DASS2
<integer>         14 - DPNSS
```

**Note:** Depending on the board, you may assign the same protocol or different protocols to each trunk on the board, with some limitations. For the Dialogic® DNI300TEPHMP board, you may assign an ISDN protocol or the CAS protocol to the trunk. For the Dialogic® DNI601TEPHMP board, you may assign an ISDN, CAS, or R2MF protocol to either trunk. For the Dialogic® DNI1200TEPHMP board, you may assign an ISDN protocol or the CAS protocol to any trunk.

The following example shows that the protocol is changed from 1 to 9 (QSIGE1) for device 2 trunk 1.

```
CLI> conf hardware pstn 2 1 protocol 9
updated
```

The updated message is displayed.

5. Verify the new settings using the **show** command.

```
CLI> show hardware pstn
```

Information about device ID, trunk ID, interface ID, protocol, and more is displayed in two columns: User Defined Table and Resolved Table. The protocol value will show QSIGE1 (the value “9” is translated) in the User Defined Table and 4ESS (the value “1” is translated) in the Resolved Table. The user defined value takes effect after the setting has been applied.

6. Apply the new settings using the **conf** command. This step validates that the protocol is supported on this board. The setting takes effect on the next media services restart.

```
CLI> conf system pstn apply
updated
```

## Configuration Procedures Using CLI

7. Verify that the User Defined Table and Resolved Table values match, indicating that the new value has been applied.

```
CLI> show hardware pstn
```

Information about device ID, trunk ID, interface ID, protocol, and more is displayed in two columns: User Defined Table and Resolved Table. In this example, after you have restarted media services, the protocol value will show QSIG1 (the value “9” is translated) in the User Defined Table and QSIG1 (the value “9” is translated) in the Resolved Table. Any errors are displayed in the Error Description for each device.

8. Repeat steps 4 through 7 for each PSTN network interface setting to be modified.

The FCD file, PCD file, and media load values are updated appropriately after you configure the protocol for the PSTN network interface.

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 5.10, “Restarting Media Services”](#), on page 46.

## 5.7 Configuring the TDM Bus

To modify default values for the TDM bus, follow this procedure:

1. Verify that the media services and boards are stopped. See [Section 5.4, “Stopping Media Services”](#), on page 38.
2. Retrieve the device ID for the hardware in your system.

```
CLI> show hardware
```

All Dialogic® devices are displayed including device ID, technology, device name, slot ID and more.

3. Retrieve the current settings for the TDM bus.

```
CLI> show system tdm bus
```

Information about primary line, primary board, secondary board, clock source, and more is displayed.

4. Modify a current TDM bus setting for a specific device and trunk; for example, modify primary board to be device ID 3.

```
CLI> conf system tdmbus primary-board 3
updated
```

The updated message is displayed. The setting takes effect on the next media services restart.

As another example, modify TDM bus primary clock source to line B (2).

```
CLI> conf system tdmbus primary-line-clock 2
updated
```

The updated message is displayed. The setting takes effect on the next media services restart.

5. Repeat step 4 as needed to modify other TDM bus parameters.
6. Verify the new settings using the **show** command.

```
CLI> show system tdmbus
```

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 5.10, “Restarting Media Services”](#), on page 46.

## 5.8 Configuring System-Level Bridging and Streaming

Dialogic® HMP Interface Boards have a bridge device that enables communication and media streaming between Dialogic® HMP Software and the boards on the CT Bus. The **conf hardware bridge sys** command is used to set bridging and streaming characteristics at the system level, while the **conf hardware bridge dev** command is used to set bridging and streaming characteristics at the board (device) level. The media stream connections are managed by the bridge controller.

To modify default values for bridging and streaming at the system level, follow this procedure:

1. Verify that the media services and boards are stopped. See [Section 5.4, “Stopping Media Services”](#), on page 38.
2. Retrieve the current settings for bridging and streaming.

```
CLI> show hardware bridge
```

Information about system-level and device-level bridging is displayed.

3. If desired, modify the idle pattern for system A-law that applies equally to all system TDM buses (CT bus and HMP soft CT bus). While these are not configured directly for bridge devices, this information is included as part of the configuration that is forwarded on to the bridge device during the startup process and must be provided to the bridge controller in the same manner.

```
CLI> conf hardware bridge sys a-law
```

Valid values: 84, 85, 213. Default is 84.

## Configuration Procedures Using CLI

4. If desired, modify the idle pattern for system mu-law that applies equally to all system TDM buses (CT bus and HMP soft CT bus). While these are not configured directly for bridge devices, this information is included as part of the configuration that is forwarded on to the bridge device during the startup process and must be provided to the bridge controller in the same manner.

```
CLI> conf hardware bridge sys mu-law
```

Valid values: 127, 255. Default is 127.

5. If desired, modify the number of hold buffers used by the host streaming interface. The greater the number of hold buffers specified, the greater the round trip media stream latency incurred between the host and the board.

```
CLI> conf hardware bridge sys nbrhsi-buffer
```

Valid values: 3 to 32. Default is 3.

6. Verify the new settings using the **show** command.

```
CLI> show hardware bridge
```

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 5.10, “Restarting Media Services”](#), on page 46.

## 5.9 Configuring Device-Level Bridging and Streaming

Dialogic® HMP Interface Boards have a bridge device that enables communication and media streaming between Dialogic® HMP Software and the boards on the CT Bus. The **conf hardware bridge sys** command is used to set bridging and streaming characteristics at the system level, while the **conf hardware bridge dev** command is used to set bridging and streaming characteristics at the board (device) level. The media stream connections are managed by the bridge controller.

To modify default values for bridging and streaming per device, follow this procedure:

1. Verify that the media services and boards are stopped. See [Section 5.4, “Stopping Media Services”](#), on page 38.
2. Retrieve the current settings for bridging and streaming.

```
CLI> show hardware bridge
```

Information about system-level and device-level bridging is displayed.

3. If desired, modify the setting for bridge device enabled or disabled.

```
CLI> conf hardware bridge dev <device ID> dev-enable <value>
```

Valid values: 1 (enabled), 2 (disabled). Default is enabled.

4. If desired, modify the setting for Dialogic® HMP Software Clock Master Fallback number (user-defined) of the bridge device. The bridge device assigned the lowest number is the

primary master. The bridge device assigned the next lowest number is the secondary master, and so on. The user defined (requested) fallback numbers are sent to the bridge controller. Since duplicates are allowed in the fallback number, the actual fallback number assigned (resolved) by the bridge controller may not be the same as the requested one. The assigned fallback number is called “Resolved”.

```
CLI> conf hardware bridge dev <device ID> nbr-usrdef <value>
```

Valid values: 0 to 15. Default is 3.

The setting for the Dialogic® HMP Software Clock Master Fallback number (resolved) of the bridge device is updated to reflect the resolved fallback number. The resolved setting cannot be modified directly.

5. If desired, modify the maximum board to host data streams. This parameter specifies the maximum number of data streams from the board to the host for the device.

```
CLI> conf hardware bridge dev <device ID> bh-datastreams <value>
```

Valid values: 0 to 253.

Valid values and default value are board-dependent. For Dialogic® DNI boards, default value is 0.

**Note:** This value combined with the associated voice streams value (in [Step 6](#)) must not exceed the maximum streams for the device. For Dialogic® DNI boards, maximum streams is 253. Examples: bh-datastreams=0, bh-voicestreams must not exceed 253; bh-datastreams=126, bh-voicestreams must not exceed 127.

6. If desired, modify the maximum board to host voice streams. This parameter specifies the maximum number of voice streams from the board to the host for the device.

```
CLI> conf hardware bridge dev <device ID> bh-voicestreams <value>
```

Valid values: 0 to 253.

Valid values and default value are board-dependent. For Dialogic® DNI boards, default value is 253.

**Note:** See note in [Step 5](#) on maximum streams for the device.

7. If desired, modify the maximum host to board data streams. This parameter specifies the maximum number of data streams from the host to the board for the device.

```
CLI> conf hardware bridge dev <device ID> hb-datastreams <value>
```

Valid values: 0 to 253.

Valid values and default value are board-dependent. For Dialogic® DNI boards, default value is 0.

**Note:** This value combined with the associated voice streams value (in [Step 8](#)) must not exceed the maximum streams for the device. For Dialogic® DNI boards, maximum streams is 253. Examples: hb-datastreams=0, hb-voicestreams must not exceed 253; hb-datastreams=126, hb-voicestreams must not exceed 127.

## Configuration Procedures Using CLI

8. If desired, modify the maximum host to board voice streams. This parameter specifies the maximum number of voice streams from the host to the board for the device.

```
CLI> conf hardware bridge dev <device ID> hb-voicestreams <value>
```

Valid values: 0 to 253.

Valid values and default value are board-dependent. For Dialogic® DNI boards, default value is 253.

**Note:** See note in [Step 7](#) on maximum streams for the device.

9. Verify the new settings using the **show** command. Setting takes effect on the next media services restart.

```
CLI> show hardware bridge
```

The streams parameters mentioned in [Step 5](#) through [Step 8](#) are used to support transparent mode. For more information on transparent mode, see [Section 2.5, “Transparent Data Routing for 3G-324M”](#), on page 21.

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 5.10, “Restarting Media Services”](#), on page 46.

## 5.10 Restarting Media Services

Media services, which include boards, must be stopped before performing any configuration procedures. To activate new configuration settings, restart media services.

Follow this procedure to restart media services:

1. Retrieve the system run status found in the system group using the **show** command. Verify that the run status is inactive.

```
CLI> show system
```

2. To start media services, use the **conf** command to update the system ipmedia setting to start.

```
CLI> conf system ipmedia start  
updated
```

The updated message is displayed and indicates that the command has been accepted.

Starting or restarting the media services results in a number of events being generated by the system and displayed to any user logged on the CLI (these include system events and possibly bridge device events). Restart event notification messages are displayed. Once the media services are started, the following message is displayed:

```
ipmedia start complete
```

This chapter provides information about using the Simple Network Management Protocol (SNMP) agent software for local or remote monitoring and configuration of the Dialogic® system software and hardware. It is assumed that the reader of this chapter has SNMP knowledge:

- [SNMP Overview](#) ..... 47
- [MIBs](#) ..... 48
- [System Requirements](#) ..... 50
- [Starting and Stopping the SNMP Agent](#) ..... 52
- [Username and Password](#) ..... 53
- [Troubleshooting](#) ..... 54

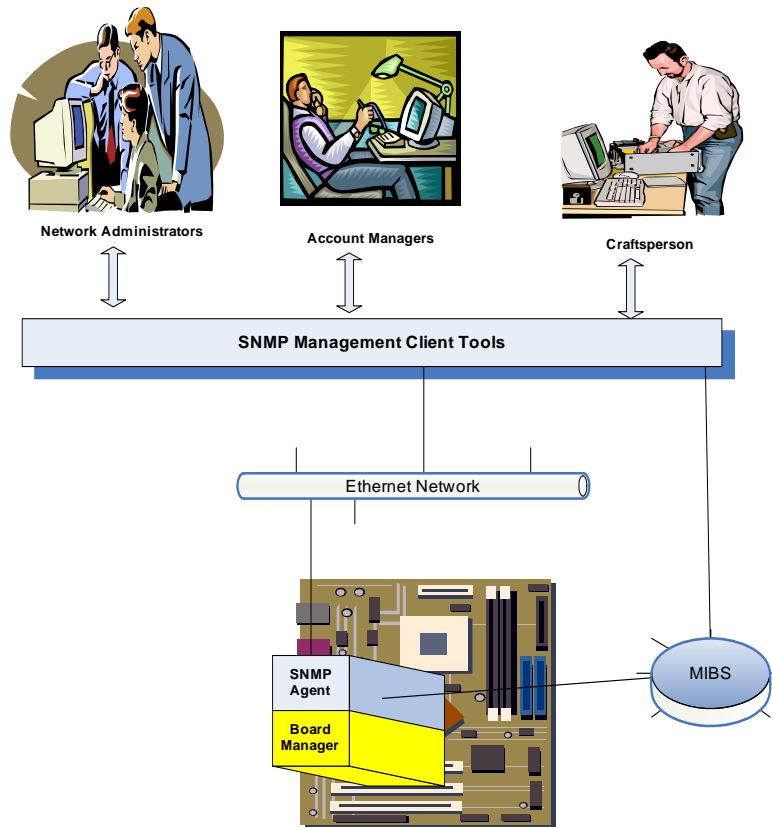
## 6.1 SNMP Overview

SNMP is a standard IP network management mechanism for exchanging management information between SNMP agents that typically reside on a managed device. The main job of SNMP is to manage network devices.

The SNMP agent provides remote (or local) monitoring and configuration. There are two types of security that the SNMP agent is governed by, namely, V2 and V3 securities. Security is discussed in [Section 6.5, “Username and Password”](#), on page 53.

Figure 2 shows the three types of SNMP users that exist in the SNMP framework: craftsperson, administrator, and account manager. These various user roles are covered in [Section 6.5, “Username and Password”](#), on page 53. It also shows that the SNMP framework relies on the IP network and therefore, the network has to be properly set up before using SNMP.

Figure 2. General SNMP Architecture



## 6.2 MIBs

This section lists, defines, and gives the location of the supported MIBs. For specific information about each MIB, refer to the MIBs reference in [Chapter 7, “SNMP MIB Reference”](#).

A MIB is a specification containing definitions of management information so that networked systems can be remotely monitored, configured, and controlled.

A MIB 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, as well as how each value will be identified, encoded, and 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 OID, read-write OID, and traps.



MIBs can be either private (enterprise) MIBs or public (standard) MIBs. SNMP agent software currently supports both public and private MIBs. You can find all supported MIBs in the following directory in a given system: `/usr/dialogic/mibs/`.

**Note:** Any public MIB can be obtained from the Internet or other sources. Public MIBs have been included in the MIB directory as a convenience. Also, note that all groups in a given MIB module may not be fully supported. For more information, see the MIBs reference in [Chapter 7, “SNMP MIB Reference”](#).

Table 4 lists the MIB modules that the SNMP agent software supports. The table does not include all the MIB modules that are included by the described MIB files. For all MIBs, view `/usr/dialogic/mibs/`.

**Table 4. Supported MIBs**

MIB Module	File Name ( <code>/usr/dialogic/mibs</code> )	Private/ Public	Description
AGENT-CONFIG-MIB	AGENT-CONFIG-MIB.mib	Public	Trap sink configuration
AGENT-SNMP-CONFIG-MIB	AGENT-SNMP-CONFIG-MIB.mib	Public	Trap sink configuration
RFC1213	RFC1213-MIB.mib	Public	MIB-2 - IP based information
RFC2495	RFC2495DS1-MIB.mib	Public	DS1 trunks
snmpUsmMIB	SNMP-USER-BASED-SM-MIB.mib	Public	SNMP V3 Security - user authentication
snmpVacmMIB	SNMP-VIEW-BASED-ACM-MIB.mib	Public	SNMP V3 Security- user authorization.
Intel-Common-MIB	Intel-Common-Mib.mib	Private	Identifiers' branch typically contains objects, i.e., Intel branch project. Use for setting OID branch. <b>Note:</b>
Intel-Prod-Tel-MCPD-IPMS-Root	Intel-Prod-Tel-MCPD-IPMS-Root.mib	Private	Identifiers' branch typically contains objects,
ipmsBridge	ipms_bridge.mib	Private	Configuration and status information for bridging and streaming
ipmsctBus	ipms_ct_bus.mib	Private	Configuration of TDM bus; device list information
ipmsExtIf	ipms_ext_if.mib	Private	Configuration and status information for all other types of connections not covered by standard publicly defined MIB, such as interface IP address assignment
ipmsHwDev	ipms_hardware_dev.mib	Private	Media service related hardware information and configuration
ipmsHwDM3Dev	ipms_hardware_dm3_dev.mib	Private	Dialogic <sup>®</sup> DM3 device information and configuration
ipmsHwDM3Trunk	ipms_hardware_dm3trunk.mib	Private	Media service related DM3 trunk information and configuration
ipmsLicense	ipms_license.mib	Private	IP media server license information and configuration

Table 4. Supported MIBs (Continued)

MIB Module	File Name (/usr/dialogic/mibs)	Private/ Public	Description
ipmsMediaCtlProtocol	ipms_media_ctl_protocol.mib	Private	IP media server protocol control information and configuration
ipmsRtpCfg	ipms_rtp_cfg.mib	Private	RTP information and configuration
ipmsSysEnv	ipms_sys_env.mib	Private	Configuration of the IP media server environment variable and media process control

## 6.3 System Requirements

This section lists the software requirements for using the SNMP agent with the Dialogic® software release. It mainly concentrates on the SNMP client requirements for the management side.

There are two ways to manage the system:

- First, you can manage the system using SNMP client tools. Since SNMP supports remote management, the client manager station exists as separate computer equipment, and the SNMP client tools must be installed in this equipment.
- Second, you can use Secure Shell (ssh) program to log in to the system and execute SNMP command line interface (CLI). This approach may not be suitable for managing a system that includes multiple Dialogic® DNI boards since the ssh operation is required for each managed board. Nevertheless, for managing a board at a time, this approach can be used.

### 6.3.1 Requirements for Remote Management Stations

The Remote Management Station can be based on either Linux/Unix or Windows® operating system. However, this document focuses on the Linux OS.

If you already have a management SNMP platform, such as your own browsers and SNMP development environment, then you need only load the appropriate Dialogic® software release MIBs. This can be done by either copying the MIBs from the distribution or from the */usr/dialogic/mibs* directory after the Dialogic® software has been installed.

If the Remote Management Station does not contain any SNMP client support, there are many possibilities for purchasing a commercial toolset or obtaining a free version. For example, the Net-SNMP client library is freely available from Open Source and can be downloaded from <http://net-snmp.sourceforge.net/>. Some SNMP management commercial tools include AdventNet (<http://www.adventnet.com/>), HP Openview (<http://h20229.www2.hp.com>), and MG-Soft (<http://www.mg-soft.com>).

These are only potential approaches; there are no restrictions imposed in the selection of any SNMP client toolset. Note that many Linux operating systems already come with the Net-SNMP toolset.

## Basic SNMP Client Toolset

A basic SNMP client toolset (or libraries) is composed of the following:

- MIB compiler
- SNMP library such as C/Java binding. These libraries can be used for writing client management applications using the given APIs.
- A set of command line tools such as get, sets, table gets, and trap manager. These command line tools are simple client executables that can be easily called from either the command line or can be embedded in a script language to build very powerful remote or local configuration scripts.
- Other tools may offer GUIs such as SNMP browsers and trap displays.

## Examples of SNMP Client Toolsets

The free Net-SNMP is an example of an available client toolset and contains only the following:

- MIB compiler
- SNMP library such as C binding. These libraries can be used for writing client management applications using the given APIs.
- Command line tools

For the Red Hat Advance Server libraries, only the following SNMP client libraries are needed (any version greater than 5.1 is acceptable):

- net-snmp-libs-5.1.2-11.EL4.6
- net-snmp-5.1.2-11.EL4.6

In addition, you will need to build the following Net-SNMP CLI clients. This SNMP CLI allows you to script configuration.

- snmpbulkget
- snmpget
- snmpset
- snmptable
- snmptranslate
- snmptrapd
- snmpvacm
- snmpbulkwalk
- snmpdf
- snmpgetnext
- snmpstatus
- snmpstest
- snmptrap
- snmpusm

- [snmpwalk](#)

Consult the Net-SNMP site for obtaining the libraries and the Net-SNMP CLI.

### 6.3.2 Requirements for Local Managed Nodes

Net-SNMP client libraries are required to use the SNMP CLI. You must install the OS; therefore, you also must install the Net-SNMP client environment.

## 6.4 Starting and Stopping the SNMP Agent

This section describes how to stop and start the SNMP agent software independently from the media application software, and also includes other related topics.

- [Stopping the SNMP Agent](#)
- [Starting the SNMP Agent](#)
- [Changing the SNMP Listening Port](#)
- [Changing the Trap Listening Port](#)

### 6.4.1 Stopping the SNMP Agent

The SNMP agent software is automatically started at system reboot and when the media application software suite is started using **dlstart**. It is automatically stopped when the media application suite is terminated using **dlstop**. You can bring the SNMP agent up and down as often as needed while the Dialogic<sup>®</sup> software is up.

Use the **dlservices** command to stop the SNMP agent:

```
# dlservices snmp stop
Shutting SNMP Agent Service : [ OK ]
```

### 6.4.2 Starting the SNMP Agent

To restart the SNMP agent, use the **dlservices** command:

```
# dlservices snmp start
Starting SNMP Agent Service : [ OK ]
```

### 6.4.3 Changing the SNMP Listening Port

The default SNMP listening port is 161. It can be changed during the software installation process. To change it at any other time, use **dlservices**. For example:

```
# dlservices snmp port 166
```

This changes the SNMP listening port to the specified port, 166 in this example. Port changes take effect on the next **dlstart** or **dlservices snmp start**.

No range checking is done, so you are responsible for providing a usable address for the SNMP listening port if you are not using the default address.

## 6.4.4 Changing the Trap Listening Port

The default trap listening port is 162. To change this port, use the **dlservices** command:

```
# dlservices snmp trapport <value>
```

## 6.5 Username and Password

The SNMP agent handles both SNMP protocol Version 1/2c and Version 3. The Dialogic® system software has been preconfigured with various default users and passwords.

The following subsections explain each default user role and the respective username and passwords. Username and passwords can be changed, added, and deleted.

- [SNMP Version 1 and 2c](#)
- [SNMP Version 3](#)

### 6.5.1 SNMP Version 1 and 2c

When using SNMP version 1 or 2c, the following SNMP User Community exists (Table 5).

**Table 5. Version 1 and 2c Community Names**

Community Name	Access
Craftsperson	Read only
Administrator	Read and write

## 6.5.2 SNMP Version 3

When using SNMP version 3, the following SNMP User Roles exist (Table 6).

**Table 6. Version 3 Usernames and Passwords**

Role	Username	Password	Auth Protocol	Context	Privacy Protocol	Role Description
Account Manager	AccountManager	AccountManager	MD5	auth	none	Allowed to add, delete, and change usernames and passwords. Allowed to add trap sink.
Administrator	Administrator	Administrator	MD5	auth	none	Allowed to read and write all supported MIBs except for Account Manager operations.
Craftsperson	Craftsperson	Craftsperson	MD5	auth	none	Allowed to read all supported MIBs except for Account Manager operations.
Account Manager (SHA)	AccountManagerSHA1	AccountManagerSHA1	SHA	auth	none	Allowed to add, delete, and change usernames and passwords. Allowed to add trap sink.
Administrator (SHA)	AdministratorSHA1	AdministratorSHA1	SHA	auth	none	Allowed to read and write all supported MIBs except for Account Manager operations.
Craftsperson (SHA)	CraftspersonSHA1	CraftspersonSHA1	SHA	auth	none	Allowed to read all supported MIBs except for Account Manager operations.

**Note:** All users use security level = authNoPriv

## 6.6 Troubleshooting

System log files store information, warnings, and errors, and are used to help to identify certain system problems.

The system log file can be found in the `/var/log/` directory. The `ngmit.log` file represents the log file for the OA&M SNMP log information generated by the OA&M framework.

Example of a log message in `ngmit.log` file:

```
Jun  2 15:57:35 mymachine Board Manager: From Component: MIB2-OS-Adaptor :
Dsl empty
```

Notice the format of the log message. All messages have the following generic format:

<Date> <machine name> <OA&M component> <Error Message>

**Note:** The syslogd service must be running in order for logging to work. Also, the */etc/syslog.conf* file must contain the following configuration lines for NGMIT:

```
# NGMIT Logging and Tracing  
local1.* /var/log/ngmit.log
```

The configuration entries are automatically configured for the system when the Dialogic® software has been properly installed.

## ***SNMP Basics***



This chapter provides reference information about MIB modules that the SNMP agent software supports. It is assumed that the reader of this chapter has SNMP knowledge:

- [SNMP MIB Reference Overview](#) . . . . . 57
- [Public MIB 2 Module Reference](#). . . . . 58
- [Public DS1 Module Reference](#) . . . . . 59
- [Private Bridge Device Module Reference](#) . . . . . 60
- [Private CT Bus Device Module Reference](#) . . . . . 60
- [Private External Interface Module Reference](#) . . . . . 60
- [Private Hardware Device Module Reference](#). . . . . 60
- [Private Hardware DM3 Device Module Reference](#) . . . . . 61
- [Private Hardware DM3 Trunk Module Reference](#). . . . . 61
- [Private License Module Reference](#) . . . . . 61
- [Private Media Control Protocol Module Reference](#) . . . . . 61
- [Private RTP Configuration Module Reference](#) . . . . . 61
- [Private System Environment Module Reference](#). . . . . 61

## 7.1 SNMP MIB Reference Overview

A MIB is a specification containing definitions of management information so that networked systems can be remotely monitored, configured, and controlled.

MIBs can be either private (enterprise) MIBs or public (standard) MIBs. SNMP agent software currently supports both public and private MIBs. The user can find all supported MIBs in the following directory in a given system: `/usr/dialogic/mibs/`.

**Note:** Any public MIB can be obtained from the Internet or other sources. Public MIBs have been included in the MIB directory as a convenience to the user. Also, note that all groups in a given MIB module may not be fully supported.

For more information about SNMP, see [Chapter 6, “SNMP Basics”](#) and [Chapter 8, “Configuration Procedures Using SNMP”](#).

## 7.2 Public MIB 2 Module Reference

Table 7 describes which groups in the given MIB modules are implemented, and describes any deviations from the MIB if taken. It serves as the Module Conformance. Each MIB entry is not described because the MIB itself should be self-descriptive.

**Table 7. MIB 2 Conformance Table**

RFC1213					
MIB Group	MIB Attribute	Read	Write	Supported	Comments
System(1.3.6.1.2.1.1)	sysDesc (1)	Yes	Yes	Yes	
	sysObjectID (2)	Yes	No	Yes	
	sysUpTime (3)	Yes	No	Yes	
	sysContact(4)	Yes	Yes	Yes	
	sysName(5)	Yes	Yes	Yes	
	sysLocation(6)	Yes	Yes	Yes	
	sysServices(7)	Yes	Yes	Yes	
	sysORLastChange(8)	No	No	No	Never supported
	SysORTable(9)	No	No	No	Never supported
interfaces(1.3.6.1.2.1.2)	ifNumber(1)	Yes	No	Yes	
	ifTable(2)	Yes	Yes	Yes	Table has only one write value
	The only column from this table that is writable is ifAdminStatus (.1.7)	Yes	Yes	Yes	
at (1.3.6.1.2.1.3)	Not supported	No	No	No	Deprecated and never supported
ip (1.3.6.1.2.1.4)	ipForwarding(1)	Yes	No	Yes	Write not supported in backend
	ipDefaultTTL(2)	Yes	No	Yes	Write not supported in backend
	ipInReceives(3)	Yes	No	Yes	
	ipInHdrErrors(4)	Yes	No	Yes	
	ipInAddrErrors(5)	Yes	No	Yes	
	ipForwDatagrams(6)	Yes	No	Yes	
	ipInUnknownProtos(7)	Yes	No	Yes	
	ipInDiscards(8)	Yes	No	Yes	
	ipInDelivers(9)	Yes	No	Yes	
	ipOutRequests(10)	Yes	No	Yes	
	ipOutDiscards(11)	Yes	No	Yes	
	ipOutNoRoutes(12)	Yes	No	Yes	
	ipReasmTimeout(13)	Yes	No	Yes	
	ipReasmReqds(14)	Yes	No	Yes	
	ipReasmOKs(15)	Yes	No	Yes	
	ipReasmFails(16)	Yes	No	Yes	
	ipFragOKs(17)	Yes	No	Yes	
	ipFragFails(18)	Yes	No	Yes	

Table 7. MIB 2 Conformance Table (Continued)

RFC1213					
MIB Group	MIB Attribute	Read	Write	Supported	Comments
	ipFragCreates(19)	Yes	No	Yes	
	ipAddrTable (20)	Yes	No	Yes	All cols supported
	ipRouteTable(21)	Yes	No	Yes	Support for table fields is read-only
	ipNetToMediaTable	Yes	No	Yes	Read-only
icmp (1.3.6.1.2.1.5)	All	Yes	No	Yes	All attributes are read-only and all are supported
tcp (1.3.6.1.2.1.6)	All scalar are read only	Yes	No	Yes	
	tcpConnTable(13) - all read is supported	Yes	No	Yes	tcpConnState (13.1.1) write is not supported
udp (1.3.6.1.2.1.7)	All scalar are supported	Yes	No	Yes	
	udpTable (5.1)	Yes	No	Yes	
egp (1.3.6.1.2.1.8)	Not supported	No	No	No	Never, not supported by backend
snmp (1.3.6.1.2.1.11)	All scalar are supported	Yes	No	Yes	

## 7.3 Public DS1 Module Reference

Table 8 describes which groups in the given MIB modules are implemented, and describes any deviations from the MIB if taken. It serves as the Module Conformance. Each MIB entry is not described because the MIB itself should be self-descriptive.

On Dialogic® HMP Software, this MIB module is read only. To configure DS1 information, use the private **ipmsHwDM3Trunk** MIB module.

Table 8. DS1 MIB Conformance Table

Module	MIB Attribute	Read	Write	Supported	Comments
dsx1 configTable (1.3.6.1.2.1.10.18.6)	dsx1LineType (.1.5)	Yes	No	Yes	
	dsx1LineCoding(.1.6)	Yes	No	Yes	
	dsx1CircuitIdentifier(.1.8)	Yes	No	Yes	
	dsx1SendCode(1.7)	Yes	No	Yes	
	dsx1LineLength(1.15)	Yes	No	Yes	
	dsx1LoopbackConfig(1.9)	Yes	No	Yes	
	dsx1TransmitClockSource	Yes	No	Yes	
	dsx1CurrentTable(.18.7)	Yes	No	Yes	
	dsx1IntervalTable(18.8)	Yes	No	Yes	
	dsx1IntervalTable(18.9)	Yes	No	Yes	

**Table 8. DS1 MIB Conformance Table (Continued)**

Module	MIB Attribute	Read	Write	Supported	Comments
	dsx1LineStatusChangeTrapEnable	Yes	No	Yes	
	Not the far ends, and fact tables are not supported	No	No	No	
	Also dsx1ChanMappingTable not supported	No	No	No	

## 7.4 Private Bridge Device Module Reference

The **ipmsBridge** MIB module describes the configuration and status of the bridge device which is used to stream data between the boards connected to the CT Bus and Dialogic® HMP Software. Consult the MIB for specific definitions.

## 7.5 Private CT Bus Device Module Reference

The **ipmsctBus** MIB module describes the configuration of the TDM bus and device list, including primary line board, primary line clock source, secondary line board, and more. Consult the MIB for specific definitions.

## 7.6 Private External Interface Module Reference

The **ipmsExtIf** (1.3.6.1.4.1.343.2.15.2.1.3) MIB module allows you to correlate all Ethernet interfaces available in the hardware. Each interface is uniquely identified by its interface index field. The interface index is associated with the MIB 2 interface table.

This MIB module contains two main tables. The **ipmsExtIfEntry** table associates the various types of Ethernet interfaces to their corresponding types. On Dialogic® HMP Software, the possible Ethernet type is **baseInterface(1)**.

The other table mainly allows the user to set the Ethernet IP configuration such as IP address and mask.

## 7.7 Private Hardware Device Module Reference

The **ipmsHwDev** MIB module describes basic hardware device information such as device ID, device name, device type, and device serial number. Consult the MIB for specific definitions.

## 7.8 Private Hardware DM3 Device Module Reference

The **ipmsHwDM3Dev** MIB module describes DM3 device information such as DM3 device ID, DM3 device logical ID, media load, PCD file, FCD file, and PCM encoding. Consult the MIB for specific definitions.

## 7.9 Private Hardware DM3 Trunk Module Reference

The **ipmsHwDM3Trunk** MIB module describes network interface information for the DM3 device such as PSTN protocol, line type, and line coding. Consult the MIB for specific definitions.

## 7.10 Private License Module Reference

The **ipmsLicense** MIB module describes the IP media server licensing information such as license file directory and license file name. Consult the MIB for specific definitions.

## 7.11 Private Media Control Protocol Module Reference

The **ipmsMediaCtlProtocol** MIB module describes the IP media server protocol control information such as MSML remote access protocol stack. Consult the MIB for specific definitions.

## 7.12 Private RTP Configuration Module Reference

The **ipmsRtpCfg** MIB module describes the configuration of the RTP. Consult the MIB for specific definitions.

## 7.13 Private System Environment Module Reference

The **ipmsSysEnv** MIB module (1.3.6.1.4.1.343.2.15.2.1.1) contains basic information about the system environment such as the vendor, serial number, form factor, run status, and the attribute to start and stop the media service. Consult the MIB for specific definitions.

**SNMP MIB Reference**

# Configuration Procedures Using SNMP

---

## 8

This chapter describes the procedures in the configuration process when using the Simple Network Management Protocol (SNMP).

- Assumptions and Prerequisites ..... 63
- Order of Procedures ..... 64
- Configuring Traps ..... 64
- Stopping Media Services ..... 66
- Activating a License ..... 66
- Configuring PSTN Network Interfaces ..... 67
- Configuring the TDM Bus ..... 67
- Configuring System-Level Bridging and Streaming ..... 68
- Configuring Device-Level Bridging and Streaming ..... 68
- Restarting Media Services ..... 70

## 8.1 Assumptions and Prerequisites

The following assumptions and prerequisites exist regarding the configuration procedures using SNMP:

- All required software, including prerequisites, have been installed and Dialogic® services have been started according to the procedures in the software installation guide supplied with your release.
- The release was installed in the default directory under INTEL\_DIALOGIC\_DIR, the environment variable for the directory in which the software is installed. Command instructions, directory paths and environment variable are shown relative to the default installation directory.
- You have prior knowledge of SNMP and are familiar with SNMP browsers or command line tools.
- You have accessed the SNMP agent and logged in as administrator. For more information on SNMP, see [Chapter 6, “SNMP Basics”](#) and [Chapter 7, “SNMP MIB Reference”](#).
- If applicable, the Dialogic® Global Call protocols have been installed. The Dialogic® Global Call protocols are provided as part of the release. For information about country dependent parameters associated with a protocol, see the *Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide*.
- Default configuration settings are provided with the software according to the license obtained for your system. Modify these settings as needed to suit your environment.

## Configuration Procedures Using SNMP

- When a private SNMP table is changed, for example the **ipmsLicenseFileList** table in the **ipmsLicense** MIB, it may take several seconds for the changes to be reflected in the table.

**Note:** If a Dialogic® HMP Interface board is removed from the system and reinserted, you will need to reconfigure the board using the CLI or SNMP utility.

## 8.2 Order of Procedures

The following is a suggested order for performing configuration procedures. Procedures that are required when configuring any system are noted as such. Additional procedures may be required depending on your system. A system may consist of Dialogic® Host Media Processing (HMP) Software only or Dialogic® HMP Software with Dialogic® HMP Interface Boards (Dialogic® DNI boards).

1. [Configuring Traps](#)
2. [Stopping Media Services](#) (required)
3. [Activating a License](#) (required)
4. [Configuring PSTN Network Interfaces](#)
5. [Configuring the TDM Bus](#)
6. [Configuring System-Level Bridging and Streaming](#)
7. [Configuring Device-Level Bridging and Streaming](#)
8. [Restarting Media Services](#) (required)

## 8.3 Configuring Traps

The purpose of the trap configuration is to register with the SNMP agent all manager stations that are interested in receiving SNMP traps (events).

There are two ways that you can configure the V3 trap forwarding table (the configuration of this table also applies to V1/V2c).

- You can add and remove entries in the `/usr/dialogic/conf/snmp/v3managertable.txt`. This file contains the persistence information that is read by the SNMP agent upon boot up in order to find out the manager stations that have registered for traps.

For example:

```
192.168.99.90 SNMP_VERSION_2c public noAuthUser USM NOAUTHNOPRIV 6
noAuth 5 0 ACTIVE
```



```
192.168.99.90 SNMP_VERSION_3 Administrator Administrator USM
NOAUTHNOPRIV 6 noAuth 5 0 ACTIVE
```

**Note:** Every time you change the *v3managertable.txt* file, the SNMP agent must be restarted in order for the changes to take place. For more information, see [Section 6.4, “Starting and Stopping the SNMP Agent”](#), on page 52.

**Note:** Make sure that the security values for the trap browser or the Manager Application receiving the trap application, which has been configured to work with SNMP V3, match the V3 user security parameters set in */usr/dialogic/conf/snmp/v3managertable.txt* file.

- You can manipulate the trap forwarding table using SNMP. In this case, the changes take effect immediately and no restart of the SNMP agent is required.

Supported traps are listed and described in Table 9.

**Table 9. Supported Traps**

Trap Name	Private/ Public	Description
Dsx1LineStatusChange	Public	Sent when the value of an instance dsx1LineStatus changes. It can be used by a network management system to trigger polls. When the line status change results from a higher level line status change (i.e., ds3), then no traps for the ds1 are sent.
coldStart	Public	Signifies that the SNMP entity, supporting a notification originator application, is reinitializing itself and that its configuration may have been altered.
authenticationFailure	Public	Signifies that the SNMP entity has received a protocol message that is not properly authenticated. While all implementations of SNMP entities <i>may</i> be capable of generating this trap, the snmpEnableAuthenTraps object indicates whether this trap will be generated.
ipmsSysEnvRunStatus	Private	Reports the run state of the overall IP Media Server system. The software that comprises the entire IP Media Server is treated as a system. Set to one of the following: ipmsSysEnvStatusUnknown ipmsSysEnvBootingUp ipmsSysEnvStatusInitialize ipmsSysEnvStatusActive ipmsSysEnvStatusInactive ipmsSysEnvStatusFault ipmsSysEnvShutdown Any time there is a state change, an SNMP notification (trap) is sent.
ipmsSysEnvSysReady	Private	Reports when the IPMS application operationally ready is achieved. Sent also as a notification whenever the state changes whether from Active or to Active state. The companion ipmsSysEnvRunStatus should be used by the SNMP client to retrieve a more detailed run status.
ipmsBridgeDeviceNotification	Private	Sent when operational status changes on a bridge device.
ipmsctBusClkNotification	Private	Sent when the TDM bus primary master value changes; for example, on CT_A/B line bad alarm, the previous primary master has stopped providing clock.

Table 9. Supported Traps (Continued)

Trap Name	Private/ Public	Description
ipmsctBusRefClkNotification	Private	Sent when the NetRef provider changes; for example, on LOSS_MASTER_SOURCE_INVALID alarm, the previous NetRef has stopped providing a valid clock source.
ipmsExtIfStatusChangedNotification	Private	Sends an SNMP trap when external interface link status changes (up or down). The SNMP notification message includes an identification of the specific external interface to which the state change applies using the ifIndex value to identify a particular of the ifTable entry.
ipmsTdmClkRefSwitchoverNotification	Private	Sent when clock source switchover situation occurs.

## 8.4 Stopping Media Services

One of the assumptions in this chapter is that all required software, including prerequisites, have been installed and Dialogic® services have been started according to the procedures in the software installation guide supplied with your release. As part of this process, media services (which include boards) are automatically started.

Media services must be stopped before configuring most components or subsystems. Therefore, it is recommended that this procedure always be performed.

Follow this procedure to stop media services. This procedure also stops all boards that are enabled.

1. View the system run status in the **ipmsSysEnv** MIB. Verify that the run status is active.
2. To stop media services, set the **ipmsSysEnvMediaProcCtl** object to **ipmsMediaProcStop** in the **ipmsSysEnv** MIB. If the SNMP client has registered itself to receive traps from the SNMP agent, then the SNMP client should receive the Run Status Trap, which notifies the user that media services have been stopped.

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 8.10, “Restarting Media Services”](#), on page 70.

## 8.5 Activating a License

Follow this procedure to activate a license you have obtained (the verification license supplied with the software is activated by default):

1. Verify that the media services and boards are stopped; see [Section 8.4, “Stopping Media Services”](#), on page 66.
2. In the **ipmsLicense** MIB, specify the path to the license file (that is, the directory where you downloaded the license) in the **ipmsLicenseFileDirectory** object. The default directory for license files is */usr/dialogic/data*.

3. In the **ipmsLicense** MIB, specify the license file name in the **ipmsLicenseFileSelected** object. The license will be effective on the next media services restart.

Alternatively, the license file also can be changed by setting **ipmsLicenseFileSelect** object to 1 in the **ipmsLicenseFileListEntry** table.

**Note:** As mentioned in [Section 8.1, “Assumptions and Prerequisites”](#), on page 63, when a private SNMP table is changed, for example the **ipmsLicenseFileList** table, it may take several seconds for the changes to be reflected in the table.

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 8.10, “Restarting Media Services”](#), on page 70.

## 8.6 Configuring PSTN Network Interfaces

To configure PSTN network interfaces for Dialogic® HMP Interface Boards, follow this procedure:

1. Verify that the media services and boards are stopped. See [Section 8.4, “Stopping Media Services”](#), on page 66.
2. In the **ipmsHwDM3Trunk** MIB, modify a current PSTN network interface setting for a specific device and network interface (trunk). For example, modify the protocol using **ipmsctHwDM3TrunkProtocol** object. The change takes effect on the next media services restart.

**Note:** Depending on the board, you may assign the same protocol or different protocols to each trunk on the board, with some limitations. For the Dialogic® DNI300TEPHMP board, you may assign an ISDN protocol or the CAS protocol to the trunk. For the Dialogic® DNI601TEPHMP board, you may assign an ISDN, CAS, or R2MF protocol to either trunk. For the Dialogic® DNI1200TEPHMP board, you may assign an ISDN protocol or the CAS protocol to any trunk.

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 8.10, “Restarting Media Services”](#), on page 70.

## 8.7 Configuring the TDM Bus

To modify default values for the TDM bus, follow this procedure:

1. Verify that the media services and boards are stopped. See [Section 8.4, “Stopping Media Services”](#), on page 66.
2. In the **ipmsctBus** MIB, modify TDM bus settings as needed. For example, modify primary line using the **ipmsctBusPrimaryLineBoard** object or the primary line clock source using the **ipmsctBusPrimaryLineClock** object. The changes take effect on the next media services restart.

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 8.10, “Restarting Media Services”](#), on page 70.

## 8.8 Configuring System-Level Bridging and Streaming

Dialogic® HMP Interface Boards have a bridge device that enables communication and media streaming between Dialogic® HMP Software and the boards on the CT Bus. You can set bridging and streaming characteristics at the system level or at the board (device) level. The media stream connections are managed by the bridge controller.

To modify default values for bridging and streaming at the system level, follow this procedure:

1. Verify that the media services and boards are stopped. See [Section 8.4, “Stopping Media Services”](#), on page 66.
2. In the **ipmsBridge** MIB, using the **ipmsBridgeSysAlawIdlePattern** and **ipmsBridgeSysMulawIdlePattern** objects, modify the system law (A-law or mu-law) and idle pattern that apply equally to all system TDM buses (CT bus and HMP soft CT bus). While these are not configured directly for bridge devices, this information is included as part of the configuration that is forwarded on to the bridge device during the startup process and must be provided to the bridge controller in the same manner. The change takes effect on the next media services restart.
3. In the **ipmsBridge** MIB, using the **ipmsBridgeSysHsiHoldBuffer** object, modify the number of hold buffers used by the host streaming interface. The greater the number of hold buffers specified, the greater the round trip media stream latency incurred between the host and the board. The change takes effect on the next media services restart.

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 8.10, “Restarting Media Services”](#), on page 70.

## 8.9 Configuring Device-Level Bridging and Streaming

Dialogic® HMP Interface Boards have a bridge device that enables communication and media streaming between Dialogic® HMP Software and the boards on the CT Bus. You can set bridging and streaming characteristics at the system level or at the board (device) level. The media stream connections are managed by the bridge controller.

To modify default values for bridging and streaming per device, follow this procedure:

1. Verify that the media services and boards are stopped. See [Section 8.4, “Stopping Media Services”](#), on page 66.
2. In the **ipmsBridge** MIB, using the **ipmsBridgeDevEnabled** object, modify the setting for bridge device enabled or disabled.

3. In the **ipmsBridge** MIB, using the **ipmsBridgeDevUsrDefFallback** object, modify the setting for HMP Clock Master Fallback number (user-defined) of the bridge device. The bridge device assigned the lowest number is the primary master. The bridge device assigned the next lowest number is the secondary master, and so on. The user defined (requested) fallback numbers are sent to the bridge controller. Since duplicates are allowed in the fallback number, the actual fallback number assigned (resolved) by the bridge controller may not be the same as the requested one. The assigned fallback number is called “Resolved”.

The setting for the HMP Clock Master Fallback number (resolved) of the bridge device is updated to reflect the resolved fallback number. The resolved setting cannot be modified directly.

4. In the **ipmsBridge** MIB, using the **ipmsBridgeBHDataStreams** object, modify the maximum board to host data streams. This parameter specifies the maximum number of data streams from the board to the host for the device. This parameter supports transparent mode.

Default value is board-dependent. For Dialogic® DNI boards, default value is 0.

**Note:** This value combined with the associated voice streams value (in [Step 5](#)) must not exceed the maximum streams for the device. For Dialogic® DNI boards, maximum streams is 253. Examples: bh-datastreams=0, bh-voicestreams must not exceed 253; bh-datastreams=126, bh-voicestreams must not exceed 127.

5. In the **ipmsBridge** MIB, using the **ipmsBridgeBHVoiceStreams** object, modify the maximum board to host voice streams. This parameter specifies the maximum number of voice streams from the board to the host for the device. This parameter supports transparent mode.

Default value is board-dependent. For Dialogic® DNI boards, default value is 253.

**Note:** See note in [Step 4](#) on maximum streams for the device.

6. In the **ipmsBridge** MIB, using the **ipmsBridgeHBDataStreams** object, modify the maximum host to board data streams. This parameter specifies the maximum number of data streams from the host to the board for the device. This parameter supports transparent mode.

Default value is board-dependent. For Dialogic® DNI boards, default value is 0.

**Note:** This value combined with the associated voice streams value (in [Step 7](#)) must not exceed the maximum streams for the device. For Dialogic® DNI boards, maximum streams is 253. Examples: hb-datastreams=0, hb-voicestreams must not exceed 253; hb-datastreams=126, hb-voicestreams must not exceed 127.

7. In the **ipmsBridge** MIB, using the **ipmsBridgeHBVoiceStreams** object, modify the maximum host to board voice streams. This parameter specifies the maximum number of voice streams from the host to the board for the device. This parameter supports transparent mode.

Default value is board-dependent. For Dialogic® DNI boards, default value is 253.

**Note:** See note in [Step 6](#) on maximum streams for the device.

For more information on transparent mode, see [Section 2.5, “Transparent Data Routing for 3G-324M”](#), on page 21.

## **Configuration Procedures Using SNMP**

Continue with any additional configuration procedures that are applicable to your system. When you are satisfied with all configuration information, proceed with [Section 8.10, “Restarting Media Services”](#), on page 70.

### **8.10 Restarting Media Services**

Media services, which include boards, must be stopped before performing any configuration procedures. Restart media services to activate new configuration settings.

Follow this procedure to restart media services:

1. View the system run status in the **ipmsSysEnv** MIB. Verify that the run status is inactive. Media services must be inactive before starting.
2. To start media services, set the **ipmsSysEnvMediaProcCtl** object to `ipmsMediaProcStart` in the **ipmsSysEnv** MIB. If the SNMP client has registered itself to receive traps from the SNMP agent, then the SNMP client and all registered stations should receive the Run Status Trap, which notifies the user that media services have been started.

This chapter provides guidelines and recommendations for tuning a system that has Dialogic® Host Media Processing Software Release 3.1LIN only or a system that also contains Dialogic® HMP Interface Boards:

- [Tuning a System with Dialogic® HMP Software](#) . . . . . 71
- [Tuning a System with Dialogic® HMP Interface Boards](#) . . . . . 72

## 9.1 Tuning a System with Dialogic® HMP Software

The following topics provide more information on tuning a system with Dialogic® Host Media Processing Software Release 3.1LIN (Dialogic® HMP Software 3.1LIN):

- [High Channel Density Recommendations](#)
- [Optimizing NIC Driver Settings](#)
- [Disabling High Precision Event Timer \(HPET\)](#)

### 9.1.1 High Channel Density Recommendations

The following recommendations are applicable in high channel density environments.

#### Increasing System Shared Memory Size

Systems with Dialogic® HMP Software 3.1LIN licenses that have more than 120 instances of a single resource type require an increase in the Linux operating system shared memory size to 64 MB or more.

To increase the size to 64 MB, use the command provided below. To make the command persistent across reboots, add the `kernel.shmmax` variable to the `/etc/sysctl.conf` file.

```
sysctl -w kernel.shmmax=65536000
sysctl -p /etc/sysctl.conf
```

To view the `shmmax` parameter, use the following command:

```
sysctl -a | grep shmmax
```

#### Memory Recommendations

When running Dialogic® HMP Software 3.1LIN with more than 120 channels of multimedia play and record, it is recommended that 4 GB of memory or more be available.

### 9.1.2 Optimizing NIC Driver Settings

When running Dialogic® HMP Software 3.1LIN, it is recommended to update network interface card (NIC) driver settings for improved system performance.

By default NIC driver settings are optimized for 100 Base-T NIC. For 1000 Base-T networks, make the following changes manually or set parameters in `/etc/sysctl.conf` file to be persistent across reboots:

```
echo 3000 > netdev_max_backlog -----> This is a recommended setting for
      1GB interfaces. Default is 300.)
echo 16777216 > rmem_default ----> related to size of TCP read window.
echo 16777216 > rmem_max
echo 16777216 > wmem_max -----> related to size of TCP write window.
echo 16777216 > wmem_default ethtool -G eth0 tx 4096 rx 4096 -----> set
      size of tx and rx ring buffers to 4096.
```

### 9.1.3 Disabling High Precision Event Timer (HPET)

When running Dialogic® HMP Software 3.1LIN, certain hardware configurations may cause problems with packet I/O.

At high densities, your hardware may require HPET to be disabled.

HPET is the High Precision Event timer tied to specific Intel architecture hardware. The Linux RTC driver uses the HPET when running in RTC compatibility mode. Using the HPET results in some performance issues in the Dialogic® HMP Software 3.1LIN real time system. Disabling the HPET allows the correct RTC logic to be used by the RTC driver.

To disable HPET, specify the following command in `/boot/grub/menu.lst` on the kernel line:

```
hpet=disable
```

Refer to your operating system manual for more information on how to set kernel boot parameters.

## 9.2 Tuning a System with Dialogic® HMP Interface Boards

The following topics provide more information on tuning a system with Dialogic® Host Media Processing Software Release 3.1LIN (Dialogic® HMP Software 3.1LIN) that also contains Dialogic® HMP Interface Boards:

- [Disabling Driver Status Messages](#)



## **9.2.1 Disabling Driver Status Messages**

In a system with Dialogic® HMP Software 3.1LIN and Dialogic® HMP Interface Boards, many driver status messages may be printed to your console screen during installation and during runtime, which may be distracting.

To prevent these driver status messages from being displayed on the console screen, run the following command at each login or add it to your login profile:

```
echo 0 >/proc/sys/kernel/printk
```

**Note:** Be aware that this command stops **all** driver and kernel messages from printing to the console screen, not just messages related to the Dialogic® HMP Interface Boards and Dialogic® HMP Software. All messages will still be printed to `/var/log/messages`.

## ***System Performance Tuning***

This chapter discusses echo cancellation support and configuration in the Dialogic® Host Media Processing (HMP) Software. The following topics are covered:

- [Echo Cancellation Background Information](#) . . . . . 75
- [Echo Cancellation Features](#) . . . . . 77
- [Using Echo Cancellation](#) . . . . . 78
- [Configuring Echo Cancellation](#) . . . . . 78
- [Echo Cancellation Sample Scenarios](#) . . . . . 82

## 10.1 Echo Cancellation Background Information

This section provides background information on echo cancellation:

- [Overview](#)
- [Acoustic Echo](#)
- [Electrical Echo](#)

### 10.1.1 Overview

The presence of echo on a media stream can negatively impact user satisfaction with voice quality. The degree to which echo is objectionable depends on both the loudness of the echo and the total round-trip delay. The impact of an echo increases as round-trip delays increase, affecting the user's perception of echo. If the delay is small (less than 20 ms), the user hears nothing or at most a reverberant sounding side-tone. Larger delays, however, lead to a subjective annoyance perceived as echo. The larger the delay, the less masking there is by the direct speech and the more annoying the echo becomes.

The presence of echo on a media stream can also negatively impact the reliability of automated media processing such as voice recognition or digit detection when voice prompts, for example, are played to a caller. In this case, a round-trip echo delay of less than 20 ms makes the returned echo unacceptable. For media processing algorithms, it is the loudness of the echo that is most problematic. Media processing algorithms typically function much better when echo is removed, or at a minimum suppressed to very low levels.

There are two basic types of echo to consider: acoustic echo and electrical echo.

### 10.1.2 Acoustic Echo

Acoustic echo occurs when sound waves emitted from the receiver of an analog or digital handset (telephone) are reflected back, or coupled back, into the microphone of that handset. If the echo is

## Echo Cancellation

not sufficiently cancelled or suppressed within the handset or elsewhere, the echo will return to the source of the sound waves, such as the remote caller, and can result in unacceptable voice quality.

Acoustic echo is typically cancelled at the handset. This is true for analog, digital, wireless, as well as IP handsets. Acoustic echo, however, can be an issue if low quality handsets are involved in a call or, in some cases, when a caller is using the hands-free mode of a handset that also does not do a proper job in canceling the acoustic echo.

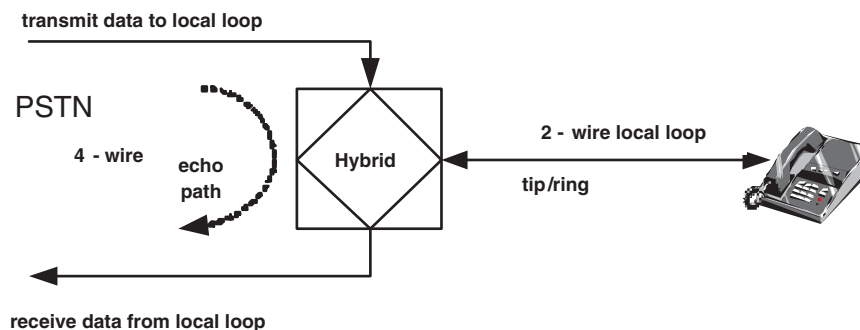
The public switched telephone network (PSTN) has no special provision to compensate for acoustic echo.

### 10.1.3 Electrical Echo

Electrical echo typically occurs when a call traverses between a 2-wire and 4-wire physical transport. On the 4-wire transport, the transmit signal and the receive signal travel on separate wire pairs. On the 2-wire transport, the transmit signal and the receive signal share the same wire pair. The interface between the 2-wire transport and the 4-wire transport is known as the **hybrid**. Since the 2-wire transport carries both the signal being sent to the 2-wire transport from the 4-wire transport (signal A) and the signal being sent from the 2-wire transport to the 4-wire transport (signal B), the hybrid must subtract signal A from the combined signal A+B on the 2-wire transport to obtain signal B to forward on to the 4-wire transport.

Let's examine the example where a 2-wire local analog loop from a subscriber's handset is connected via the hybrid to a 4-wire transport deeper within the PSTN, Figure 3. The hybrid attempts to subtract the signal that is being transmitted from the 4-wire transport to the 2-wire local loop from the combined signals on the 2-wire local loop to obtain the desired signal being sent toward the PSTN from the handset. Ideally, the signal being subtracted would match exactly the signal being sent to the 2-wire local loop from the 4-wire transport, so that the received data on the 4-wire side of the hybrid would contain only the signal sent to the PSTN by the handset. It would not contain any reflected signal or echo.

**Figure 3. Electrical Echo and the Hybrid**



However, in the real world, the ability of the hybrid to remove the transmitted signal is imperfect. The term hybrid balance refers to the ratio (expressed in dB) of the signal being transmitted from the 4-wire side to the apparent (reflected) signal received back at the 4-wire side. If the hybrid

balance is not perfect due to impedance mismatches, for instance, then a fraction of the signal being sent is reflected back to the sender. If a delay in the network occurs, the sender hears an echo, which can be objectionable.

The PSTN compensates for electrical echo created within the PSTN under certain connections by inserting echo suppressors or cancellation devices in the path of the call. This is done when a hybrid is included within the route of the call and when the round-trip echo delay exceeds a limit, typically less than 25 ms, above which the echo is deemed to be objectionable. This would be the case, for example, for some long distance calls as well as calls using a satellite link. For the majority of calls, including local calls and also many long distance calls, the round-trip delays are well below 25 ms and, therefore, echo compensation is not provided by the PSTN.

The criteria used to determine when or when not to insert echo compensation in the PSTN, however, assumes that terminal devices such as handsets and combinations of PBX's with handsets behind them add relatively small additional delays beyond the delays resulting from the PSTN connection. When echo exists on the received media stream and additional delays are introduced into the call connection beyond those budgeted by the PSTN, echo cancellation is required if these additional delays cause the total round-trip delay to exceed approximately 25 ms. If echo cancellation is not provided under these circumstances, callers may experience objectionable echo.

An example of when echo cancellation is not provided is for call connections that span PSTN and IP networks. While the round-trip delay may be small (a few milliseconds on the PSTN side) the delays introduced by the IP network portion of the connection may be in the order of 100 ms. IP networks can also introduce lost or replaced packets, as well as packet jitter, making for a non-linear environment in which it is very difficult to identify and cancel all echoes at the point where media leaves the packet network. Due to these significant delays and potential impairments, gateways which interface between the PSTN and IP network are required to remove any echo received from the PSTN side prior to transmitting audio onto the IP network. Likewise, IP phones must also remove echo before transmitting audio onto the IP network. Conferencing is another example where additional delays are added into the connection path. Therefore, conference bridges typically provide echo cancellation, removing the echo received from each conference party prior to summing the received audio.

## 10.2 Echo Cancellation Features

Echo cancellation support is available for circuit-switched connections that may contain electrical echo on the received media stream. This feature is applicable to Dialogic® HMP Interface Boards that provide public switched telephone network (PSTN) connectivity for Dialogic® HMP Software.

Echo cancellation support is in compliance with ITU-T G.168 recommendation and can be controlled using runtime API functions or CONFIG file parameters. It is recommended that you control echo cancellation on a channel by channel basis using runtime API functions when they are available. The boards support tail lengths of up to 64 ms.

Echo cancellation capability is provided on the following devices in Dialogic® Host Media Processing (HMP) Software:

- IP media device
- signal detector of the voice device

## Echo Cancellation

- conferencing device
- CSP/voice device

### 10.3 Using Echo Cancellation

Enabling the echo cancellation feature in certain devices (IP media, voice, conferencing, and continuous speech processing) in Dialogic® HMP Software is typically required when these devices receive media streams from the public switched telephone network (PSTN) via the DTI devices on the Dialogic® DNI300TEPHMP board or the Dialogic® DNI1200TEPHMP board.

**Note:** DTI devices refer to digital network interface devices, which is used interchangeably with digital trunk interface devices.

While T1/E1 interfaces are digital interfaces, analog links may be part of the end-to-end connections. When this is the case, electrical echo may be present unless the PSTN provides echo cancellation on the connection, which it does not normally do. The echo cancellation feature only addresses electrical echo. It does not address acoustic echo which is typically handled by the handsets (telephones) where acoustic echo could potentially be introduced into the network. See [Section 10.1, “Echo Cancellation Background Information”](#), on page 75 for an overview of echo cancellation and the types of echo.

Enabling the echo cancellation feature in the devices mentioned earlier in this section is **not required** for the following:

- Media streams received from PSTN connections via the Dialogic® DNI601TEPHMP board since this board provides on-board echo cancellation as part of the DTI resources.  
For information about echo cancellation settings on the Dialogic® DNI601TEPHMP board, see [Section 13.1, “\[0x2c\] Echo Cancellation Parameters”](#), on page 104.
- Media streams received from the IP packet network via IP media devices. Due to the delays associated with the packet network, echo must be removed prior to transmitting voice (audio) onto the packet network. This is the responsibility of IP endpoints such as gateways and IP phones. Therefore, it is assumed that the media streams received from the IP packet network are free of any objectionable echo.

Since the echo cancellation feature consumes CPU resources, echo cancellation should only be applied when needed. For this reason, runtime application control is available to both enable and disable echo cancellation on a per connection or per channel basis. See [Section 10.4, “Configuring Echo Cancellation”](#), on page 78 for more information.

### 10.4 Configuring Echo Cancellation

Echo cancellation can be configured on one or more devices as needed. The following topics provide more information about configuring echo cancellation:

- [Echo Cancellation on IP Media Devices](#)
- [Echo Cancellation on Voice Devices](#)
- [Echo Cancellation on Conferencing Devices](#)

- [Echo Cancellation on CSP/Voice Devices](#)

**Note:** The information in the following subsections does not apply to the Dialogic® DNI601TEPHMP board, because this board provides on-board echo cancellation as part of the DTI resources. For information about echo cancellation settings on the Dialogic® DNI601TEPHMP board, see [Section 13.1, “\[0x2c\] Echo Cancellation Parameters”](#), on page 104.

## 10.4.1 Echo Cancellation on IP Media Devices

The IP media device supports echo cancellation on tail lengths up to 64 ms. The default echo cancellation settings on IP media devices are as follows: echo cancellation is disabled, non-linear processing (NLP) is enabled, and the tail length is 16 ms.

Echo cancellation features of the IP media device can be managed using the Dialogic® IP Media Library API or the Dialogic® Global Call API library.

### Dialogic® IP Media Library API

Echo cancellation features of the IP media device can be controlled and monitored using Dialogic® IP Media Library API functions.

Echo cancellation can be turned on or off on a per channel (device) basis using **ipmSetParm()** in conjunction with PARMCH\_ECACTIVE in the IPM\_PARM\_INFO structure.

The echo cancellation tail length can be set using **ipmSetParm()** in conjunction with PARMCH\_ECHOTAIL in the IPM\_PARM\_INFO structure.

The echo cancellation NLP feature can be turned on or off using **ipmSetParm()** in conjunction with PARMCH\_EC\_NLP\_ACTIVE in the IPM\_PARM\_INFO structure.

Echo cancellation settings can be monitored using **ipm\_GetParm()** in conjunction with the following in the IPM\_PARM\_INFO structure: PARMCH\_ECACTIVE, PARMCH\_ECHOTAIL, and PARMCH\_EC\_NLP\_ACTIVE.

The following usage rules apply:

- NLP can only be disabled after echo cancellation has been enabled and the **ipm\_StartMedia()** function has been invoked.
- Disabling echo cancellation, invoking the **ipm\_Stop()** function, or invoking the **ipm\_ModifyMedia()** function will cause NLP to be reset to the enabled state. Therefore, if NLP needs to be disabled, you must invoke **ipm\_SetParm()** to disable NLP each time the **ipm\_StartMedia()** or **ipm\_ModifyMedia()** is invoked. This assumes that echo cancellation was previously enabled.
- If echo cancellation was also disabled, echo cancellation must be re-enabled either before or after invoking **ipm\_StartMedia()** or **ipm\_ModifyMedia()**. In either case, echo cancellation must also be enabled before **ipm\_SetParm()** is invoked to disable NLP.

For more information, see the Dialogic® IP Media API Library documentation.

### Dialogic® Global Call API Library

Alternatively, echo cancellation features of the IP media device can be controlled using the Dialogic® Global Call API Library.

Echo cancellation settings can be modified using the **gc\_SetUserInfo( )** function to set Dialogic® IP Media Library parameters via the GC\_PARM\_BLK structure. You can set any IP media parameter in the GC\_PARM\_BLK structure by using **gc\_util\_insert\_parm\_ref( )** with a set id of IPSET\_CONFIG and parameter IDs of IPPARM\_IPMPARM with a value as a pointer to the IPM\_PARM\_INFO structure.

The echo cancellation parameters that may be modified include PARMCH\_ECACTIVE, PARMCH\_ECHOTAIL, and PARMCH\_EC\_NLP\_ACTIVE.

The following usage rule applies:

- NLP can only be disabled after echo cancellation has been enabled and an active call has been established. Upon termination of a call, NLP will be reset to the enabled state. Therefore, if NLP needs to be disabled, the **gc\_SetUserInfo( )** function will need to be invoked to disable NLP each time a new call is established with echo cancellation enabled.

For more information, refer to the Dialogic® Global Call API Library documentation.

### 10.4.2 Echo Cancellation on Voice Devices

The signal detector of the voice resource supports echo cancellation on tail lengths up to 8 ms. By default, echo cancellation is disabled. The echo cancellation tail length cannot be modified.

#### Dialogic® Voice API Library

Echo cancellation features of the voice device can be controlled and monitored using the Dialogic® Voice API library.

Echo cancellation can be turned on or off on a per channel (device) basis using **dx\_setparm( )** in conjunction with DXCH\_EC\_ACTIVE.

Echo cancellation settings can be monitored using **dx\_getparm( )** in conjunction with DXCH\_EC\_ACTIVE.

For more information, refer to the Dialogic® Voice API library documentation.

### 10.4.3 Echo Cancellation on Conferencing Devices

The conferencing device supports echo cancellation on a per party basis for tail lengths up to 64 ms. By default, echo cancellation is disabled, and the default tail length is set to 16 ms.

Echo cancellation features of the conferencing device can be managed using the Dialogic® Conferencing (DCB) API library and the CONFIG file for Dialogic® HMP Software.



## Dialogic® Conferencing API Library

Echo cancellation features of the conferencing device can be controlled using Dialogic® Conferencing (DCB) API library functions.

Echo cancellation is disabled by default but can be enabled on a per party basis when establishing a conference using the **dcb\_estconf()** function or, when adding a party to a conference using the **dcb\_addtoconf()** function.

The **dcb\_estconf()** function uses an array of MS\_CDT data structures which define the attributes of the conference parties. The **dcb\_addtoconf()** function uses a single MS\_CDT structure to define the attributes of the party that is being added to the conference.

The `chan_attr` field in the MS\_CDT structure is a bitmask that specifies the party's properties within the conference. The bit that enables echo cancellation is `MSPA_ECHOXCLEN`.

The conferencing device echo cancellation tail length and NLP capability cannot be controlled via the conferencing API, but can be controlled via the CONFIG (`.config`) file. For more information, see [CONFIG File for Dialogic® HMP Software](#).

For more information on conferencing functions and data structures, refer to the Dialogic® Conferencing (DCB) API Library documentation.

## CONFIG File for Dialogic® HMP Software

You can modify the conferencing device echo cancellation tail length by editing the CONFIG (`.config`) file. The tail length setting, however, applies globally to all conferencing channels (devices). The setting also modifies the default value for continuous speech processing (CSP) devices, but the CSP values can be changed on a per channel basis using CSP APIs.

The following line in the CONFIG file allows the echo cancellation tail length to be modified:

```
[0x2c]
SetParm=0x2c03,128 ! Set EC tail length to 64(8ms), 80(10ms),
                  128(16ms),192(24ms), 256(32ms), 512(64ms). Default: 128(16ms)
```

You can enable or disable the conferencing device echo cancellation non-linear processing (NLP) feature by editing the CONFIG file. The NLP setting, however, applies globally to all conferencing channels (devices). The setting also modifies the default value for CSP, but the CSP values can be changed on a per channel basis using CSP APIs.

The following line in the CONFIG file allows the echo cancellation NLP feature to be enabled or disabled:

```
[0x2c]
SetParm=0x2c01, 0x1 ! Enable Echo Canceller NLP (0x1 (enable-default),
                  0x0 (disable))
```

## Echo Cancellation

After editing the CONFIG file, you must generate a new FCD file for the changes to take effect. For more information, see [Section 2.7, “Modifying the FCD File By Editing the CONFIG File”](#), on page 24.

### 10.4.4 Echo Cancellation on CSP/Voice Devices

The CSP/voice device supports echo cancellation on tail lengths up to 64 ms. By default, echo cancellation is disabled, and the default tail length is 16 ms.

**Note:** Modifying echo cancellation related conferencing parameters in the CONFIG file will change corresponding default CSP echo cancellation parameter values.

#### Dialogic® Continuous Speech Processing API Library

Echo cancellation features of the CSP/voice device can be controlled and monitored using Dialogic® CSP API library functions.

Echo cancellation can be turned on or off on a per channel (device) basis using `ec_setparm()` in conjunction with `ECCH_ECHOCANCELLER`.

The echo cancellation tail length can be set on a per channel (device) basis using `ec_setparm()` in conjunction with `DXCH_EC_TAP_LENGTH`.

The echo cancellation non-linear processing (NLP) feature can be turned on or off on a per channel (device) basis using `ec_setparm()` in conjunction with `ECCH_NLP`.

Echo cancellation settings can be monitored using `ec_getparm()` in conjunction with the following parameters: `ECCH_ECHOCANCELLER`, `DXCH_EC_TAP_LENGTH`, and `ECCH_NLP`.

For more information, refer to the Dialogic® Continuous Speech Processing API Library documentation.

## 10.5 Echo Cancellation Sample Scenarios

The following examples illustrate the digital network interface (DTI) devices connected to various Dialogic® HMP Software resources as well as IP only connections.

- [DTI and IP Gateway Example](#)
- [DTI and IP Conferencing Example](#)
- [DTI and CSP/Voice Example](#)
- [DTI and Voice Example](#)
- [IP and IP Connection Example](#)
- [IP and CSP/Voice Example](#)

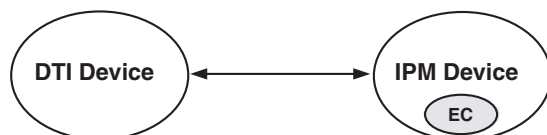
- IP and Voice Example

**Note:** These examples do not apply to the DTI devices on the Dialogic® DNI601TEPHMP board, because this board provides on-board echo cancellation as part of the DTI resources.

### 10.5.1 DTI and IP Gateway Example

Figure 4 shows an example of a T1/E1 and IP Gateway connection formed by connecting the DTI device to an IP media device, also called an IPM device. In this example, the echo cancellation feature (EC) of the IP media device is enabled, to handle echo on the media stream received from the PSTN via the DTI device.

Figure 4. T1/E1 Gateway Example

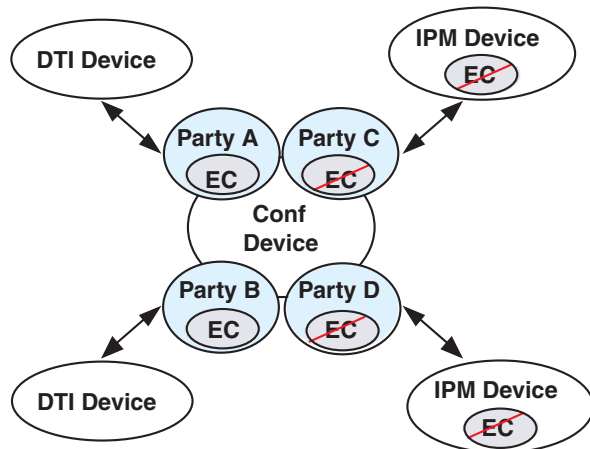


### 10.5.2 DTI and IP Conferencing Example

Figure 5 shows a four-party conferencing example where two DTI devices are connected to the conference as parties A and B, and two IP media devices are connected to the conference as parties C and D.

The echo cancellation feature (EC) of the conference device is enabled for parties A and B, to handle echo on the media stream received from the PSTN via the DTI devices. The EC feature is disabled for parties C and D, and also for both IP media devices.

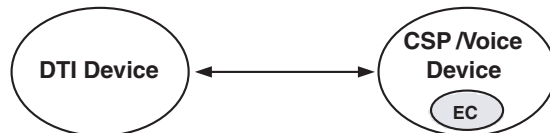
Figure 5. DTI and IP Conferencing Example



### 10.5.3 DTI and CSP/Voice Example

Figure 6 shows an example where the DTI device is connected to a CSP/voice device. The echo cancellation feature (EC) of the CSP/voice device is enabled, to handle echo on the media stream from the PSTN received via the DTI device.

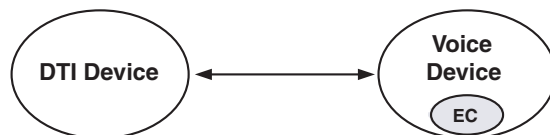
Figure 6. DTI and CSP/Voice Example



### 10.5.4 DTI and Voice Example

Figure 7 shows an example where the DTI device is connected to a voice device. The echo cancellation feature (EC) of the voice device is enabled, to handle echo on the media stream from the PSTN received via the DTI device.

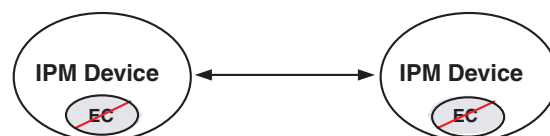
Figure 7. DTI and Voice Example



### 10.5.5 IP and IP Connection Example

Figure 8 shows an IP to IP connection example. The echo cancellation feature (EC) is disabled for both IP media devices. As discussed in Section 10.1.3, "Electrical Echo", on page 76, echo must be removed from the media stream prior to its transmission on the IP packet network.

Figure 8. IP and IP Connection Example

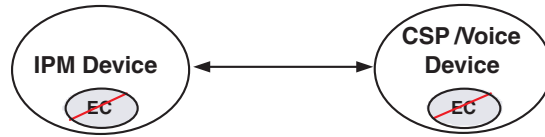


### 10.5.6 IP and CSP/Voice Example

Figure 9 shows an example where the IP media device is connected to a CSP/voice device. The echo cancellation feature (EC) of both the CSP/voice device and the IP media device is disabled.

As discussed in [Section 10.1.3, “Electrical Echo”](#), on page 76, echo must be removed from the media stream prior to its transmission on the IP packet network.

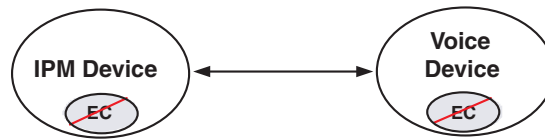
**Figure 9. IP and CSP/Voice Example**



## 10.5.7 IP and Voice Example

Figure 10 shows an example where the IP media device is connected to a voice device. The echo cancellation feature (EC) of both the voice device and the IP media device is disabled. As discussed in [Section 10.1.3, “Electrical Echo”](#), on page 76, echo must be removed from the media stream prior to its transmission on the IP packet network.

**Figure 10. IP and Voice Example**



## ***Echo Cancellation***

This chapter provides background information about CONFIG (.*config*) files including directory location and formatting conventions. This chapter also includes information to help you set the parameters contained in the CONFIG file.

- CONFIG File Formatting Conventions ..... 87
- CONFIG File Sections ..... 88
- [NFAS] Section ..... 88
- [CHP] Section ..... 89
- [TSC] Section ..... 90

## 11.1 CONFIG File Formatting Conventions

The CONFIG (.*config*) files, located in the *data* directory under INTEL\_DIALOGIC\_DIR (the environment variable for the directory in which the software is installed), are ASCII files that contain component configuration information required by Dialogic® boards. When manually editing the CONFIG file, use the following formatting conventions:

### Parameters

Many CONFIG file parameters use the SetParm command to assign values. The format is SetParm followed by an equal sign, followed by the hexadecimal parameter number, followed by a comma, followed by the parameter value:

```
SetParm=parameter-number, parameter-value
```

Additional commands used to set parameters include:

- AddNFASInterface (see “GroupID (Group Identifier)”, on page 115)
- Variant (see Section 11.4, “[CHP] Section”, on page 89)
- defineBSet (see Section 11.5, “[TSC] Section”, on page 90).

### Sections

Configuration parameters are grouped into sections. In general, each section begins with a section name enclosed in square brackets. (The section names are listed and described in Section 11.2, “CONFIG File Sections”, on page 88.) The parameters for the section immediately follow the section name.

```
[section-name]
```

Some sections group parameters that apply to a specific network interface (trunk) or channel (line). These section names are followed by a period (.) and the trunk number. For sections that group parameters like this, there is a separate section for each trunk.

```
[section-name.trunk-number]
```

## CONFIG File Details

### Comments

Comments can be added to the CONFIG file. If you use an exclamation point (!) anywhere on a line, all text to the right of the exclamation point until the end of the line is treated as a comment (ignored).

```
! comment
```

For a list of all CONFIG file parameters, see [Chapter 13, “CONFIG File Parameter Reference for Dialogic® HMP Interface Boards”](#).

## 11.2 CONFIG File Sections

CONFIG file parameters are grouped in sections based on the board components and subcomponents being configured. Modifiable CONFIG file sections include the following:

**Note:** CAS and R2MF protocols are configured using Protocol Development Kit (PDK) parameters. For more information, see the *Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide*.

### [0x2c]

Defines parameters used to set the tail length, or tap length, of the enhanced echo canceller.

### [lineAdmin.x]

Defines line device parameters applicable to each trunk on a board that has T1 or E1 trunks.

### [NFAS] and [NFAS.x]

Non-Facility Associated Signaling (NFAS). Defines the Primary D channel and NFAS trunk parameters. The [NFAS] section defines the number of NFAS groups on a board. The [NFAS.x] sections define the parameters specific to each group. For details about setting the NFAS parameters, see [Section 11.3, “\[NFAS\] Section”](#), on page 88.

### [CCS] and [CCS.x]

Common Channel Signaling (CCS). Defines common channel signaling parameters applicable to technologies such as ISDN. The [CCS] section defines board-based parameters and the [CCS.x] section defines the line-based parameters.

### [CHP]

Channel Protocol (CHP). Defines the telephony communication protocol that is used on each network interface using the Variant Define *n* command. For details about setting [CHP] parameters using the Variant Define *n* command, see [Section 11.4, “\[CHP\] Section”](#), on page 89.

### [TSC]

Telephony Service Component (TSC). Defines sets of B channels and associated characteristics using the defineBSet command. For details about setting [TSC] parameters using the defineBSet command, see [Section 11.5, “\[TSC\] Section”](#), on page 90.

## 11.3 [NFAS] Section

Non-Facility-Associated Signaling (NFAS) uses a single ISDN PRI D channel to provide signaling and control for up to 10 ISDN PRI lines. Normally, on an ISDN PRI line, one D channel is used for



signaling and 23 B channels (bearer channels) are used for transferring information. In an NFAS configuration, therefore, one D channel can support the signaling and control for up to 239 B channels. The trunk that provides the signaling is called the primary D channel. The trunks that use all 24 channels as B channels are called NFAS trunks.

- Notes:**
1. For a board containing multiple primary D channels, the maximum number of trunks supported by each NFAS group on that board is reduced. This is due to the additional message load on the board's CPU.
  2. NFAS is supported on ISDN NI-2, 4ESS, 5ESS, and DMS protocols.
  3. NFAS D channel backup (DCBU) is supported on ISDN 4ESS, 5ESS, NI-2 protocol.
  4. When NFAS is used, the **SignalingType** parameter in the [lineAdmin] section of the CONFIG file must be modified. For details about this parameter modification, see [Section 13.2, "\[lineAdmin.x\] Parameters \(Digital Voice\)"](#), on page 105.

The CONFIG file contains an [NFAS] section and multiple [NFAS.x] sections. The [NFAS] section defines the number of NFAS instances created, that is, defines the number of NFAS groups. For each NFAS group, there is an [NFAS.x] section in the CONFIG file. For example, if there are two NFAS groups defined in the [NFAS] section, there will be two [NFAS.x] sections, [NFAS.1] and [NFAS.2].

NFAS parameters are modified by editing the respective lines in the [NFAS] and [NFAS.x] sections of the CONFIG file. For example, to increase the number of NFAS groups per board from one to four, change the value of **NFAS\_INSTANCE\_MAP** (parameter = 0x3E02) from a value of 1 (one group per board) to a binary value of 1111 (four NFAS groups per board) represented by 0xF.

Following is an excerpt from the [NFAS] section of a CONFIG file that illustrates that part of the file before and after editing.

### Before editing:

```
[NFAS]
SetParm=0x3e02,0x1 !INSTANCE MAP, default = 1 (1 group/board)
```

### After editing:

```
[NFAS]
SetParm=0x3e02,0xf !INSTANCE MAP - 4 NFAS groups/board
```

## 11.4 [CHP] Section

The Channel Protocol (CHP) component implements the telephony communication protocol that is used on each network interface. There are different versions of this component for handling different signaling types as well as different protocol types on different B channels. There is one CHP instance created for each B channel in the system.

The [CHP] section of the CONFIG file is a subset of the [TSC] section. Protocol-specific parameters, primarily in the form of variants, are defined in the [CHP] section. The selection of

## CONFIG File Details

which of these protocol variants to use on which line (span) is determined in the [TSC] section. For more information on protocol variants selection, see [Section 11.5, “\[TSC\] Section”](#), on page 90.

A number of protocol variants are defined in the [CHP] section of the CONFIG file. Variants are defined by the Variant Define *n* command, where *n* is the variant identifier. The Variant Define *n* command defines variant “*n*” as all of the parameter definitions in the [CHP] section preceding the command.

**Note:** If a parameter is defined multiple times prior to the Variant Define *n* command, then only the last definition of the parameter is used for that variant.

**Note:** [CHP] T1 Protocol variants are configured using Protocol Development Kit (PDK) parameters. For more information, see the *Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide*.

Although protocol variants are defined in the [CHP] section, protocol variants are assigned in the [TSC] section of the CONFIG file. Selecting a particular Variant Define *n* is accomplished by changing the values of the **Inbound** and **Outbound** parameters for a particular line. The **Inbound** and **Outbound** parameters are the sixth and seventh parameters respectively in the defineBSet command in the [TSC] section of the CONFIG file.

For information about the defineBSet command and setting TSC parameters, see [Section 11.5, “\[TSC\] Section”](#), on page 90.

For information about each CHP parameter, see the following sections:

- [Section 13.6, “\[CHP\] Parameters”](#), on page 121
- [Section 13.7, “\[CHP\] ISDN Protocol Variant Definitions”](#), on page 122

## 11.5 [TSC] Section

The [TSC] section of the CONFIG file defines a set of B channels and associated characteristics using the defineBSet command. The syntax of the defineBSet command is:

```
defineBSet = SetId, LineId, StartChan, NumChans, BaseProtocol, Inbound, OutBound, DChanDesc, Admin, Width, BChanId, SlotId, Direction, Count, [BChanId, SlotId, Direction, Count,] 0
```

To change a [TSC] parameter, you change the value of the applicable defineBSet parameter in the CONFIG file. For example, to change the protocol variant from 2 to 4 for both inbound and outbound call processing on all 30 channels of line 2, you would change the value of the **Inbound** and **Outbound** parameters for line 2 (**SetId=20**) from 2 to 4. For information on defining protocol variants, see [Section 11.4, “\[CHP\] Section”](#), on page 89.

Following is an excerpt from the [TSC] section of a CONFIG file that illustrates that part of the file before and after editing.

### Before editing:

```
defineBSet=10,1,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0  
defineBSet=20,2,1,30, 0,2,2,1,20,1, 1,1,3,15, 16,17,3,15,0
```

**After editing:**

```
defineBSet=10,1,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0  
defineBSet=20,2,1,30, 0,4,4,1,20,1, 1,1,3,15, 16,17,3,15,0
```

For information about each TSC parameter, see [Section 13.9, “\[TSC\] defineBSet Parameters”](#), on page 130.

## ***CONFIG File Details***

# CONFIG File Parameter Reference 12 for Dialogic® HMP Software

---

This chapter lists and describes parameters contained in the CONFIG file for Dialogic® Host Media Processing (HMP) Software. Parameters are grouped according to the CONFIG file sections.

- [0x2c] Echo Cancellation Parameters ..... 93
- [0x3b] Conferencing Parameters ..... 95
- [IPVSC] IP Media Parameters ..... 96
- [decoder] Speed Control Parameter ..... 100
- [sigDet] Parameters ..... 100

**Note:** Not all parameters are included in each CONFIG file, as this depends on the license purchased. CONFIG file parameters that **should not be modified** by the user are omitted from this document.

**Note:** Modified CONFIG file parameters can be preserved and re-applied whenever a Dialogic® HMP Software license is changed or re-activated by using the user configuration file, *Hmp.Uconfig*. For more information, see [Section 2.6, “Preserving Data in User Configuration Files”](#), on page 23.

## 12.1 [0x2c] Echo Cancellation Parameters

The [0x2c] section of the CONFIG file includes the following parameters used to adjust echo cancellation settings for specific devices in Dialogic® HMP Software:

- EC Enablement
- EC Filter Length
- Echo Canceller NLP

### EC Enablement

**Number:** 0x2c00

**Description:** The **EC Enablement** parameter allows control of the echo canceller used in conferencing, continuous speech processing (CSP) and other applications requiring echo cancellation.

### Values:

- 0x0: EC disabled
- 0x1 [default]: EC enabled

**Guidelines:** This parameter is not included in the configuration file. To modify the parameter value, you must add this parameter manually in the configuration file in the [0x2c] section.

The **EC Enablement** parameter can be set in the CONFIG file for Dialogic® HMP Software or for the Dialogic® DNI601TEPHMP board. When set in the CONFIG file for the board, it affects EC on the DTI device of the board only. When set in the CONFIG file for the Dialogic® HMP Software, it affects EC for the conferencing and CSP devices. For more information on configuring echo cancellation features, see [Section 10.4, “Configuring Echo Cancellation”](#), on page 78. For more information about setting EC on Dialogic® DNI601TEPHMP boards, see [Chapter 13, “CONFIG File Parameter Reference for Dialogic® HMP Interface Boards”](#).

## EC Filter Length

**Number:** 0x2c03

**Description:** The **EC Filter Length** parameter specifies the tail length, or tap length, of the enhanced echo canceller used by a conferencing device and by a continuous speech processing (CSP) device.

**Note:** The CSP API library provides a parameter, through `ec_setparm()`, for setting the tap length. The audio conferencing (DCB) API library does not. When available, the API library method is preferable for setting parameters. Be aware that setting the tap length using this CONFIG file parameter affects both the conferencing device and the CSP device. For more information on configuring echo cancellation features, see [Section 10.4, “Configuring Echo Cancellation”](#), on page 78.

### Values:

- 64: 8 ms
- 80: 10 ms
- 128 [default]: 16 ms
- 192: 24 ms
- 256: 32 ms
- 512: 64 ms

**Guidelines:** When determining the tail length value, consider the length of the echo path delay your system will encounter as well as your overall system configuration. Longer tail lengths are provided to handle echo with longer path delays. To achieve better performance (that is, faster convergence and less noise), use the shortest tail length setting that is consistent with the expected echo path delay. The tail length setting should be at least as long as the expected echo path delay, if not longer.

## Echo Canceller NLP

**Number:** 0x2c01

**Description:** The **Echo Canceller NLP** parameter specifies whether non-linear processing (NLP) is enabled or not for all conferencing devices and for all continuous speech processing (CSP) devices.

**Note:** The CSP API library also provides a parameter, through **ec\_setparm()**, for setting NLP. The audio conferencing (DCB) API library does not. When available, the API library method is preferable for setting parameters. Be aware that setting the NLP using this CONFIG file parameter affects both the conferencing device and the CSP device. For more information on configuring echo cancellation features, see [Section 10.4, “Configuring Echo Cancellation”](#), on page 78.

### Values:

- 0x0: NLP disabled
- 0x1 [default]: NLP enabled

**Guidelines:** The **Echo Canceller NLP** parameter must be turned off for automatic speech recognition applications. This parameter is not included in the configuration file. To modify the parameter value, you must add this parameter manually in the configuration file in the [0x2c] section.

## 12.2 [0x3b] Conferencing Parameters

The [0x3b] section of the CONFIG file includes the following parameters associated with the conferencing device in Dialogic® HMP Software:

- [Comfort Noise Generation in Conferencing](#)
- [CSUMS\\_AGC\\_low\\_threshold](#) (Conferencing AGC Noise Level Threshold)

### Comfort Noise Generation in Conferencing

**Number:** 0x3b38

**Description:** Comfort noise generation can be generated on the outputs of a conference whenever the conference output drops below the no-talker level across the board. The comfort noise generation parameter is used to enable or disable comfort noise generation (CNG).

### Values:

- 0 [default]: comfort noise generation disabled
- 1: comfort noise generation enabled

**Guidelines:** It is recommended that you accept the default values for setting the silence threshold, noise amplitude, and hang time parameters. If it is necessary to modify these values,

## CONFIG File Parameter Reference for Dialogic® HMP Software

you must manually add the parameters to the CONFIG file in the [0x3b] section. The parameters are:

```
SetParm=0x3b39,100 ! Silence threshold, default is 100 (-43dbm), 1630  
(-18.5dbm) is maximum
```

```
SetParm=0x3b3a,200 ! Noise amplitude, default is 200 (-43dbm), 3270  
(-18.5dbm) is maximum
```

```
SetParm=0x3b3b,25 ! Hang time, default is 25 (25 gives 12msx25=300ms,  
but actually results in 620 ms due to exponential averaging)
```

### CSUMS\_AGC\_low\_threshold (Conferencing AGC Noise Level Threshold)

**Number:** 0x3b1f

**Description:** The **CSUMS\_AGC\_low\_threshold** parameter defines the upper threshold for noise level estimates. Any signal above this threshold is considered speech. Thus, this threshold should be set quite high in order to let the AGC algorithm determine when there are voiced and unvoiced periods. The parameter is given in terms of the average level.

**CSUMS\_AGC\_low\_threshold** is defined as:  $10^{(\text{outputlevel in dB})/20} * 2^{23}$ . Multiplying by  $2^{23}$  converts the value into a linear 24-bit value that accommodates the 24-bit DSPs used on the DM3 boards.

**Values:** 0x0020c5 to 0x0732ae (-60 dB<sub>avg</sub> to -25 dB<sub>avg</sub>).  
The default value is 0x0147ae (-40 dB<sub>avg</sub>).

**Guidelines:** The default value of -40 dB is the best setting for most applications. To modify the parameter value, you must add this parameter manually in the configuration file in the [0x3b] section.

The following example sets the AGC noise level to a value of -50 dB<sub>avg</sub>:

```
SetParm=0x3b1f,0x00679f
```

The calculation to get the hexadecimal value for an AGC noise threshold level of -50 dB<sub>avg</sub> is:

```
10(-50/20) * 223 = 0x00679f
```

It is recommended that the value be set in the range of -60 dB to -40 dB. Do not exceed the AGC high level threshold which is set to -34.6 dB in the current DM3 system.

## 12.3 [IPVSC] IP Media Parameters

The [IPVSC] section of the CONFIG file includes the following parameters associated with the IP media device in Dialogic® HMP Software:

- [UDP Port Base for Audio RTP](#)
- [UDP Port Base for Video RTP](#)
- [EC Enablement](#)
- [EC Tail Length](#)



- PrmPlr\_LatencyAlg\_Audio (Latency Mode)
- PrmRxLatencyFrameInit\_Audio (PLR Initial Latency)
- PrmRxLatencyFrameMax\_Audio (PLR Maximum Latency)

## UDP Port Base for Audio RTP

**Number:** 0x4005

**Description:** This parameter specifies the UDP port base for audio RTP. Two ports (1 RTP, 1 RTCP) are used for each instance of an IP media device. The default value for the base port is 49152. For example, in a configuration with 400 instances of IP media devices, the range of values would be 49152 to 49951. RTP and RTCP use paired ports; the even one is for RTP and the odd one is for RTCP. The first instance used would get 49152 for RTP and 49153 for RTCP. This parameter allows you to set the range used by Dialogic® HMP Software in order to avoid conflicts with other entities that use UDP ports.

**Values:** 0 to 0xffff. Default value for base port is 49152.

**Guidelines:** Although the range of values is 0 to 0xffff, the UDP standard may specify a more restricted range. Additionally, the range of values varies depending on the number of instances of IP media devices being used.

## UDP Port Base for Video RTP

**Number:** 0x4006

**Description:** This parameter specifies the UDP port base for video RTP. Two ports (1 RTP, 1 RTCP) are used for each instance of an IP media device. The default value for the base port is 57344. For example, in a configuration with 400 instances of IP media devices, the range of values would be 57344 to 58143. RTP and RTCP use paired ports; the even one is for RTP and the odd one is for RTCP. The first instance used would get 57344 for RTP and 57345 for RTCP. This parameter allows you to set the range used by Dialogic® HMP Software in order to avoid conflicts with other entities that use UDP ports.

**Values:** 0 to 0xffff. Default value for base port is 57344.

**Guidelines:** Although the range of values is 0 to 0xffff, the UDP standard may specify a more restricted range. Additionally, the range of values varies depending on the number of instances of IP media devices being used.

## EC Enablement

**Number:** 0x1b12

**Description:** The **EC Enablement** parameter specifies whether echo cancellation is enabled or not in Dialogic® HMP Software.

**Values:**

- 0 [default]: echo cancellation disabled
- 1: echo cancellation enabled

**Guidelines:** Use the **ipm\_SetParm()** function to turn on echo cancellation on a per-channel basis. See [Section 10.4.1, “Echo Cancellation on IP Media Devices”](#), on page 79 for more information.

## **EC Tail Length**

**Number:** 0x1b13

**Description:** The **EC Tail Length** parameter specifies the tail length of the echo canceller in Dialogic® HMP Software.

**Values:**

- 128 [default]: 16 ms
- 512: 64 ms

**Guidelines:** Use the **ipm\_SetParm()** function to modify the echo canceller tail length on a per-channel basis. See [Section 10.4.1, “Echo Cancellation on IP Media Devices”](#), on page 79 for more information.

## **PrmPlr\_LatencyAlg\_Audio (Latency Mode)**

**Number:** 0x1b95

**Description:** This parameter defines the mode of operation for the jitter buffer. By default, the jitter buffer uses the adaptive mode, which means latency can grow until the available number of frames (defined by the **PrmRxLatencyFrameMax\_Audio** parameter) is depleted. Using the fixed mode, the number of frames buffered can grow up to 1.5 times the starting value (defined by the **PrmRxLatencyFrameInit\_Audio** parameter).

**Values:**

- 0 [default]: adaptive mode enabled
- 1: fixed mode enabled

**Guidelines:** Use the adaptive mode to minimize audio loss due to abnormal packet reception conditions. Use the fixed mode to minimize latency growth by sacrificing audio quality. This parameter is not included in the configuration file. To modify the parameter value, you must add this parameter manually in the configuration file in the [IPVSC] section.

## PrmRxLatencyFrameInit\_Audio (PLR Initial Latency)

**Number:** 0x1b07

**Description:** This parameter specifies the amount of packet loss recovery (PLR) latency (delay) that can be introduced by defining the initial number of frames that can be buffered. This parameter defines the starting value for initial latency.

**Note:** When the fixed mode is enabled (see **PrmPlr\_LatencyAlg\_Audio** parameter), the number of frames buffered can grow up to 1.5 times the starting value. For example, if the parameter value is 6, the maximum number of frames that can be buffered is 9.

The packet loss recovery module attempts to restore packets arriving at the receive end as close as possible to their original time-stamped positions. Arriving packets are decomposed into individual frames, each with a unique time stamp.

Each new frame is then stored in a jitter buffer before sending it to the decoder. This is done to allow packets arriving out of order to be inserted in the queue in the correct order. The size of this jitter buffer is defined by the number of frames stored and is controlled by the **PrmRxLatencyFrameInit\_Audio**, **PrmRxLatencyFrameMax\_Audio**, and **PrmPlr\_LatencyAlg\_Audio** parameters.

**Values:** 1 to **PrmRxLatencyFrameMax\_Audio** value (frames). Default value is 6.

**Guidelines:** This parameter is not included in the configuration file. To modify the parameter value, you must add this parameter manually in the configuration file in the [IPVSC] section.

## PrmRxLatencyFrameMax\_Audio (PLR Maximum Latency)

**Number:** 0x1b08

**Description:** This parameter defines the maximum number of frames to be buffered in the Packet Loss Recovery (PLR) frame list. This parameter adds latency only when the buffer is already filled and additional frames arrive before there is space in the buffer. This provides for bursts of packets to arrive, which would have to be discarded otherwise.

The Packet Loss Recovery module attempts to restore packets arriving at the receive end as close as possible to their original time-stamped positions. Arriving packets are decomposed into individual frames, each with a unique time stamp.

Each new frame is then stored in a jitter buffer before sending it to the decoder. This is done to allow packets arriving out of order to be inserted in the queue in the correct order. The size of this jitter buffer is defined by the number of frames stored and is controlled by the **PrmRxLatencyFrameInit\_Audio**, **PrmRxLatencyFrameMax\_Audio**, and **PrmPlr\_LatencyAlg\_Audio** parameters.

**Values:** 30 to 200 (frames). Default value is 100.

**Note:** On the Windows® operating system, the upper limit of the value is dependent on channel density and Kernel memory availability.

**Guidelines:** While the number of frames to be buffered should be set as high as possible for best quality, too high a value will add unnecessary latency to the system. Generally, the number of frames buffered should be the same size or slightly larger than the number of frames per packet.

The coder used and its coder frame slice affects the frames per packet or PLR frames per packet. The coders and their coder frame slices are: G.711 (10 ms frame slice), G.723 (30 ms frame

slice), G.729 (10 ms frame slice), G.726 (20 ms frame slice). For example, the G.711 coder with 30 ms frame size uses 1 frame per packet but the PLR module stores this data as three 10 ms frames.

Not all coders are supported in all libraries. For more information on coder support in the Global Call API, see the *Global Call IP Technology Guide*. For more information on coder support in the IP media library (IPML) API, see the *IP Media Library API Library Reference*.

This parameter is not included in the configuration file. To modify the parameter value, you must add this parameter manually in the configuration file in the [IPVSC] section.

## 12.4 [decoder] Speed Control Parameter

The [decoder] section of the CONFIG file includes the following parameter associated with the voice device in Dialogic® HMP Software:

- [Speed Control Enablement](#)

### Speed Control Enablement

**Number:** 0x31e

**Description:** The **Speed Control Enablement** parameter specifies whether the speed control feature in the voice API library is enabled or not.

**Values:**

- 0: speed control disabled
- 1 [default]: speed control enabled

**Guidelines:** For more information on speed control, see the *Voice API Library Reference* and the *Voice API Programming Guide*.

## 12.5 [sigDet] Parameters

The signal detector section of the CONFIG file may include qualification templates for positive answering machine detection (PAMD) and positive voice detection (PVD) used in call progress analysis.

A tech note was written to provide instructions for modifying the PAMD and PVD qualification template parameters on Dialogic® DM3 boards to accomplish higher successful PAMD and PVD rates. For further information about these parameters, see the tech note at <http://www.dialogic.com/support/helpweb/dxall/tnotes/legacy/2000/tn030.htm>. Although the tech note was written for Dialogic® System Release 5.1.1 Feature Pack 1 for Windows®, the information applies to subsequent releases.

The modified parameters have now become the default in the firmware, so it is no longer necessary to tune the PAMD and PVD parameters as explained in the tech note. Operating with these new default values should result in improved accuracy of call progress analysis on Dialogic® DM3

boards. However, although these values are the most commonly used, they may not be suitable for every application environment. If needed, the PAMD and PVD templates are still tunable, as explained in the tech note, to achieve even better results based on the individual application environment.

The default PAMD qualification template ID is 106561 (0x1a041), but other valid PAMD qualification template IDs that can be defined in the CONFIG file are:

- 106564 (0x1a044)
- 106565 (0x1a045)
- 106566 (0x1a046)
- 106567 (0x1a047)

The default PVD qualification template ID is 128193 (0x1f4c1), but other valid PVD qualification template IDs that can be defined in the CONFIG file are:

- 128194 (0x1f4c2)
- 128195 (0x1f4c3)
- 128196 (0x1f4c4)
- 128197 (0x1f4c5)

**CONFIG File Parameter Reference for Dialogic® HMP Software**

# CONFIG File Parameter Reference 13

## for Dialogic® HMP Interface Boards

---

This chapter lists and describes the parameters contained in the CONFIG files used by Dialogic® HMP Interface Boards, which include Dialogic® DNI300TEPHMP, DNI601TEPHMP, and DNI1200TEPHMP boards. Parameters are grouped according to the CONFIG file sections. Within the CONFIG files, the parameters are grouped in the following sections.

- [0x2c] Echo Cancellation Parameters ..... 104
- [lineAdmin.x] Parameters (Digital Voice)..... 105
- [NFAS] Parameters ..... 113
- [NFAS.x] Parameters..... 114
- [CCS] Parameters ..... 116
- [CHP] Parameters ..... 121
- [CHP] ISDN Protocol Variant Definitions ..... 122
- [TSC] Parameters..... 130
- [TSC] defineBSet Parameters ..... 130
- [sigDet] Parameters ..... 137

**Note:** Not all parameters are included in each CONFIG file, as this depends on the board supported by that particular file. CONFIG file parameters that **should not be modified** by the user are omitted from this document. Exceptions are made for parameters that, although they should not be modified by the user, are needed in understanding a particular set of parameters (for example, the [TSC] `defineBSet Width` parameter). For these exceptions, the parameter description states that the value should not be modified by the user.

**Note:** Modified CONFIG file parameters can be preserved and re-applied whenever a Dialogic® HMP Software license is changed or re-activated by using the user configuration file, *Hmp.Uconfig*. For more information, see [Section 2.6, “Preserving Data in User Configuration Files”](#), on page 23.

**Note:** CAS and R2MF protocols are configured using Protocol Development Kit (PDK) parameters. For more information, see the *Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide*.

## 13.1 [0x2c] Echo Cancellation Parameters

The [0x2c] section of the CONFIG file defines parameters used to set the echo cancellation feature of the enhanced echo canceller for a digital network interface device (DTI device).

### EC Tail Length

**Number:** 0x2c03

**Description:** The **EC Tail Length** parameter specifies the tail length, or tap length, of the enhanced echo canceller for a digital network interface device (DTI device).

**Note:** The **EC Tail Length** parameter only applies to the Dialogic® DNI601TEPHMP board, which has an echo canceller on the board itself. The Dialogic® DNI300TEPHMP and DNI1200TEPHMP boards do not have an onboard echo canceller. For more information about using echo cancellation on these boards, see [Chapter 10, “Echo Cancellation”](#).

**Values:**

- 0x80: 16 ms
- 0xC0: 24 ms
- 0x100: 32 ms
- 0x200 [default]: 64 ms

**Guidelines:** When determining the tail length value, consider the length of the echo path delay your system will encounter as well as your overall system configuration. Longer tail lengths are provided to handle echo with longer path delays. To achieve better performance (that is, faster convergence and less noise), use the shortest tail length setting that is consistent with the expected echo path delay. The tail length setting should be at least as long as the expected echo path delay, if not longer.

### EC Enablement

**Number:** 0x2c00

**Description:** The **EC Enablement** parameter specifies whether echo cancellation is enabled or not on the board.

**Note:** The **EC Enablement** parameter only applies to the Dialogic® DNI601TEPHMP board, which has an echo canceller on the board itself. The Dialogic® DNI300TEPHMP and DNI1200TEPHMP boards do not have an onboard echo canceller. For more information about using echo cancellation on these boards, see [Chapter 10, “Echo Cancellation”](#).

**Values:**

- 0x0: Echo cancellation disabled
- 0x1 [default]: Echo cancellation enabled

**Guidelines:** None



## **EC NLP**

**Number:** 0x2c01

**Description:** The **EC NLP** parameter specifies whether non-linear processing (NLP) is enabled or not on the board.

**Note:** The **EC NLP** parameter only applies to the Dialogic® DNI601TEPHMP board, which has an echo canceller on the board itself. The Dialogic® DNI300TEPHMP and DNI1200TEPHMP boards do not have an onboard echo canceller. For more information about using echo cancellation on these boards, see [Chapter 10, “Echo Cancellation”](#).

### **Values:**

- 0x0: EC NLP disabled
- 0x1 [default]: EC NLP enabled

**Guidelines:** The EC NLP parameter must be turned off for automatic speech recognition applications.

## **13.2 [lineAdmin.x] Parameters (Digital Voice)**

For digital voice boards, the line administration parameters are associated with an individual T1 or E1 trunk. The parameters defined in the [lineAdmin.x] section are associated with line x. For example, parameters in the [lineAdmin.3] section of the CONFIG file are associated with line 3. Digital voice line administration parameters include:

- [LineType \(Line Type\)](#)
- [SignalingType \(Signaling Type\)](#)
- [Coding \(Coding\)](#)
- [ZeroCodeSuppression \(Zero Code Suppression\)](#)
- [FramingAlgorithm \(CRC Checking\)](#)
- [LOSDeclaredTime \(LOS Declared Time\)](#)
- [LOSClearedTime \(LOS Cleared Time\)](#)
- [REDCFADecay \(RED CFA Decay\)](#)
- [REDCFADeclareTime \(RED CFA Declare Time\)](#)
- [REDCFAClearedTime \(RED CFA Cleared Time\)](#)
- [YellowCFADeclareTime \(Yellow CFA Declare Time\)](#)
- [YellowCFAClearTime \(Yellow CFA Clear Time\)](#)
- [RAICRCCFADeclareTime \(RAI CRC CFA Declare Time\)](#)
- [RAICRCCFAClearTime \(RAI CRC CFA Clear Time\)](#)
- [Initial Alarm State](#)
- [BPVS Threshold Range](#)
- [OOF Threshold Range](#)
- [FERR Threshold Range](#)

- [ECS Threshold Range](#)
- [CECS Threshold Range](#)

## **LineType (Line Type)**

**Number:** 0x1601

**Description:** The **LineType** parameter defines the physical line type (T1 or E1) and the framing format (for example, D4 or ESF). Framing formats include:

### D4 framing (D4)

For T1 lines, in D4 framing, 12 frames of 193 bits each (2,316 bits total) constitute a superframe. This framing format supports AB signaling.

### Extended superframe (ESF)

For T1 lines, in ESF framing, 24 frames of 193 bits each (4,632 bits total) constitute an extended superframe. This framing format supports ABCD signaling.

### CEPT E1

For E1 lines, uses CEPT E1 framing.

### Cyclic redundancy check 4 (CRC-4) multi-frame

For E1 lines, this provides for CRC error detection. In this framing format, E1 lines have an extra framing that can coexist with the standard framing and the time slot 16 signaling framing. This extra framing is used to compute and check CRC-4 on incoming lines, to detect remote CRC-4 alarms, and to notify the remote line of CRC-4 errors. When CRC-4 framing is enabled, all CRC-related statistics will be collected and reported, and the RAI\_CRC\_CFA alarm will be detected and reported.

### **Values:**

- 0: T1 D4 (dsx1\_D4)
- 1: T1 ESF (dsx1\_ESF)
- 2: E1 CEPT E1 (dsx1\_E1)
- 3: E1 CRC 4 multi-frame (dsx1\_E1\_CRC)

## **SignalingType (Signaling Type)**

**Number:** 0x1602

**Description:** The **SignalingType** parameter defines the signaling type to be used by the T1 or E1 line. Signaling types include:

### Channel associated signaling (CAS)

In CAS, the signaling for each channel is directly associated with that channel. T1 robbed-bit signaling is an example of CAS.

### Common channel signaling (CCS)

In CCS, a common channel carries the signaling for all of the channels on that T1 or E1 line. ISDN is an example of CCS, where the D channel is used to carry the signaling for all of the B channels.

### Clear channel signaling (Clear)

In this type, none of the channels on the T1 or E1 line are used for signaling purposes. Clear channel signaling is the ability to access telephony channels in the system and configure them

to a user-defined call control protocol, or to simply leave the lines 'clear'. The resources should have access to the telephony bus for media routing purposes, as well as signal detection, signal generation, and tone generation capabilities, if desired.

**Note:** In a clear channel configuration, the CT Bus does not preserve frames, so any in-band signaling is lost. That is, T1 CAS robbed bit signaling cannot be performed on a line configured to use clear channel signaling.

**Values:**

- 4: CAS
- 5: CCS
- 6: Clear

**Guidelines:** When using Non-Facility-Associated Signaling (NFAS), Signaling Type is dependent on whether the T1 line is a primary, standby (DCBU), or NFAS ISDN trunk. The primary trunk must be set to CCS, and the standby and NFAS trunks must be set to Clear.

**Note:** NFAS is supported on ISDN NI-2, 4ESS, 5ESS and DMS protocols, and NFAS D channel backup (DCBU) is supported on ISDN 4ESS, 5ESS, NI-2 protocols.

For additional parameters that need to be modified for NFAS, see [Section 13.3, “\[NFAS\] Parameters”](#), on page 113

## **Coding (Coding)**

**Number:** 0x1603

**Description:** The **Coding** parameter defines the coding scheme to be used by a digital line type. Coding schemes include:

**Modified alternate mark inversion (B8ZS)**

This is a modified AMI code that only applies to T1 lines and is used to preserve one's density on the line. Whenever eight consecutive zeros occur on the line, they are replaced by an 8-bit string that violates the bipolar signaling. If the preceding pulse was positive, the polarity of the substituted eight bits is 000+-0-+. If the preceding pulse was negative, the polarity of the substituted eight bits is 000-+0+-.

**Alternate mark inversion (AMI)**

This is a bipolar signal conveying binary digits in which each successive 1 (mark) is of the opposite polarity. If the previous mark was a positive pulse, then the next mark will be a negative pulse. Spaces have an amplitude of zero (no pulse).

**High density bipolar three zero (HDB3)**

High density bipolar three zero is a modified AMI code that only applies to E1 and is used to preserve one's density on the line. Whenever four consecutive zeros appear, the four-zeros group is replaced with an HDB3 code. This could be either of two HDB3 codes, depending on whether there was an odd or even number of ones since the last bipolar violation. If an odd number of ones occurred, the substituted four bits are 000V, where V represents a bipolar

## CONFIG File Parameter Reference for Dialogic® HMP Interface Boards

violation. If an even number of ones occurred, the substituted four bits are P00V, where P represents a parity bit and V represents a bipolar violation.

### Values:

- 7: B8ZS
- 8: AMI
- 9: HDB3

## ZeroCodeSuppression (Zero Code Suppression)

**Number:** 0x1604

**Description:** The **ZeroCodeSuppression** parameter is an algorithm used by T1 lines that inserts a 1 bit into a stream to prevent the transmission of eight or more consecutive 0 bits, which could produce timing errors. Instead, this algorithm maintains a minimum one's density to reduce timing errors.

### Values:

- 10: Bell - Bell zero code suppression (Jam Bit 7)
- 11: GTE - GTE zero code suppression (Jam Bit 8, except in signaling frames when Jam Bit 7 is used if the signaling bit is 0)
- 12: DDS - Digital Data Service zero code suppression (data byte is replaced with 10011000)
- 13: None - No zero code suppression is used.

**Guidelines:** The **ZeroCodeSuppression** parameter is used when AMI line-coding is used, that is, when the **Coding** parameter is set to AMI. Since AMI does not perform zero code suppression, the **ZeroCodeSuppression** parameter ensures there are no long strings of consecutive zeros on the line.

If the **Coding** parameter is set to B8ZS or HDB3 (for E1), then zero code suppression is performed by the line-coding and the **ZeroCodeSuppression** parameter is ignored.

## FramingAlgorithm (CRC Checking)

**Number:** 0x1624

**Description:** A T1 front end can run two different framing algorithms when configured as extended superframe (ESF): a default algorithm and an alternate CRC-6 checking algorithm. The CRC-6 checking algorithm allows the circuit to confirm the CRC-6 bits in the received multiframe, as a guard against mimic framing patterns, before forcing a new frame alignment. The CRC Checking parameter allows you to enable the CRC-6 checking algorithm.

### Values:

- 0: Default algorithm
- 1: Alternate CRC-6 checking algorithm

**Guidelines:** This parameter only applies to T1 trunks whose Line Type parameter (0x1601) is set to 1 (dsx1\_ESF). For all other Line Types, this parameter is invalid.

To include this parameter and enable CRC checking, you must edit the applicable CONFIG file by adding the following line at the end of each [lineAdmin] section of the CONFIG file:

```
SetParm=0x1624,1! CRC checking OFF=0 (default), CRC checking ON=1
```

After editing the CONFIG file, you will need to generate a new FCD file. Refer to [Section 2.7](#), “Modifying the FCD File By Editing the CONFIG File”, on page 24 for more information.

### **LOSDeclaredTime (LOS Declared Time)**

**Number:** 0x160c

**Description:** The **LOSDeclaredTime** parameter defines the number of milliseconds for which no signal is detected at the input port before a loss of signal (LOS) or carrier-failure alarm (CFA) can be declared.

**Values:** 0 to 2500 (milliseconds)

### **LOSClearedTime (LOS Cleared Time)**

**Number:** 0x160d

**Description:** The **LOSClearedTime** parameter defines the number of milliseconds for which a signal must be detected at the input port before a declared LOS or CFA can be cleared.

**Values:** 0 to 2500 (milliseconds)

### **REDCFADecay (RED CFA Decay)**

**Number:** 0x1609

**Description:** The **REDCFADecay** parameter is the denominator of the fraction used to calculate the decay slope in the integration process when RED CFA condition has not been declared and LOS or LOF is intermittent.

**Values:** 4 to 15 (1/4 to 1/15)

### **REDCFADeclareTime (RED CFA Declare Time)**

**Number:** 0x160a

**Description:** The **REDCFADeclareTime** parameter defines the number of milliseconds that a red alarm condition must be received at the input port before a RED CFA condition can be declared.

**Values:** 0 to 2500 (milliseconds)

### **REDCFAClearedTime (RED CFA Cleared Time)**

**Number:** 0x160b

**Description:** The **REDCFAClearedTime** parameter defines the number of milliseconds that a normal signal must be received at the input port before a declared RED CFA condition can be cleared.

**Values:** 1000 to 15000 (milliseconds)

### **YellowCFADeclareTime (Yellow CFA Declare Time)**

**Number:** 0x160e

**Description:** The **YellowCFADeclareTime** parameter defines the number of milliseconds for which a Remote Alarm Indication (RAI) signal is detected at the input port before a yellow CFA condition can be declared.

**Values:** 0 to 2500 (milliseconds)

### **YellowCFAClearTime (Yellow CFA Clear Time)**

**Number:** 0x160f

**Description:** The **YellowCFAClearTime** parameter defines the number of milliseconds for which a RAI signal is not detected at the input port before a declared yellow CFA condition can be cleared.

**Values:** 0 to 2500 (milliseconds)

### **RAICRCCFADeclareTime (RAI CRC CFA Declare Time)**

**Number:** 0x1610

**Description:** The **RAICRCCFADeclareTime** parameter defines the number of seconds for which a RAI signal and CRC Error is detected at the input port before a RAI CRC CFA can be declared.

**Values:** 0 to 450 (milliseconds)

### **RAICRCCFAClearTime (RAI CRC CFA Clear Time)**

**Number:** 0x1611

**Description:** The **RAICRCCFAClearTime** parameter defines the number of seconds for which a RAI signal and Remote CRC Error is not detected at the input port before a declared RAI CRC CFA can be cleared.

**Values:** 0 to 450 (milliseconds)

## **InitialBitPattern (Initial CAS Signaling Bit Pattern)**

**Number:** 0x1625

**Description:** The **InitialBitPattern** parameter defines the values of the CAS ABCD signaling bits that are transmitted for all channels on the specified line at the time the firmware is downloaded and initialized.

**Values:** 0x0 to 0xf, where the hexadecimal value represents the binary ABCD bit values. For example, 0xd defines the ABCD bit pattern as 1101.

**Guidelines:** For a T1 line, the default is 0x0. For an E1 line, the default is 0xd.

## **Initial Alarm State**

**Number:** 0x1626

**Description:** Trunk preconditioning allows boards to be placed in an alarm state during board initialization.

While Dialogic® boards are starting up and are connected to network trunks, there is a period where the digital network interface begins transmitting frames and idle CAS signaling. This state can exist for a minute or more before the board and application program are prepared to handle calls. During this time, a service provider (CO) may begin alerting (ringing) for inbound calls, but the calls cannot be answered because the board or application has not finished initializing. This results in lost calls.

The Initial Alarm State parameter allows you to place trunks in an alarm state while the board is being initialized. This prevents the service provider from sending calls. The alarm clears and the trunks go inservice as soon as the first **gc\_OpenEx()** (or **gc\_Open()**) function for a trunk is executed in the application. (For T1 trunks, alarms clear after a 15-second delay to verify valid signaling.)

### **Values:**

- 0: No alarm is transmitted on the trunk; all trunk time slots signal Out of Service (Default)
- 1: TransmitAIS - An Alarm Indication Signal (AIS) alarm is transmitted on the trunk.
- 2: TransmitRAI - A Remote Alarm Indication (RAI) alarm is transmitted on the trunk.

**Note:** The default behavior also applies if the Initial Alarm State parameter is not used. The Initial Alarm State parameter setting applies only upon board initialization. After the initial alarm state is cleared (by **gc\_OpenEx()** or **gc\_Open()**), trunks do not return to the initial alarm state unless you restart the board. Stopping the board or unloading the application does not return a board to its initial alarm state.

**Note:** An RAI alarm could result from a response to a loss of sync from the network side. If the Initial Alarm State parameter is set to 2, but a loss of sync (or similar condition) persists even after the board is initialized and **gc\_OpenEx()** or **gc\_Open()** is invoked, the RAI will continue to be transmitted until the network condition is cleared.

**Note:** A board could transmit other alarms, as a response to a network condition, that are unrelated to this parameter. Those alarms will persist until the network condition is cleared.

## CONFIG File Parameter Reference for Dialogic® HMP Interface Boards

**Guidelines:** To use the Initial Alarm State parameter, it must be manually added to the CONFIG file that was selected for your board. The hexadecimal parameter number 0x1626 must be added in the [lineAdmin] section for each trunk on the board. For example:

```
[lineAdmin.1]
SetParm=0x1626,1      ! IntialAlarmState (None=0, AIS=1, RAI=2)

[lineAdmin.2]
SetParm=0x1626,1      ! IntialAlarmState (None=0, AIS=1, RAI=2)

[lineAdmin.3]
SetParm=0x1626,1      ! IntialAlarmState (None=0, AIS=1, RAI=2)

[lineAdmin.4]
SetParm=0x1626,1      ! IntialAlarmState (None=0, AIS=1, RAI=2)
```

### BPVS Threshold Range

**Number:** 0x1639

**Description:** To support the Dialogic® Global Call Alarm Management System (GCAMS) enhancements, this parameter allows you to change the default threshold value of the Bipolar Violation Count Saturation (BPVS) alarm (T1 or E1 alarm) by adding a parameter in the CONFIG file (.config) that corresponds to the PCD file in use on your board. The change is made per span. After threshold parameters are added, the FCD file is automatically updated when the new PCD file and modified CONFIG files are downloaded to the board.

**Values:** 0 to 255

**Guidelines:** To modify the default threshold for the BPVS alarm, add the following parameter (sample value of 100 shown) to the [lineAdmin.x] section of a CONFIG file:

```
SetParm=0x1639,100 ! BPVS threshold range 0 - 255, default 255
```

### OOF Threshold Range

**Number:** 0x163a

**Description:** To support the Dialogic® Global Call Alarm Management System (GCAMS) enhancements, this parameter allows you to change the default threshold value of the Out of Frame Error Count Saturation (OOF) alarm (T1 alarm) by adding a parameter in the CONFIG file (.config) that corresponds to the PCD file in use on your board. The change is made per span. After threshold parameters are added, the FCD file is automatically updated when the new PCD file and modified CONFIG files are downloaded to the board.

**Values:** 0 to 255

**Guidelines:** To modify the default threshold for the OOF alarm, add the following parameter (sample value of 100 shown) to the [lineAdmin.x] section of a CONFIG file:

```
SetParm=0x163a,100 ! OOF threshold range 0 - 255, default 0
```

### FERR Threshold Range

**Number:** 0x163b

**Description:** To support the Dialogic® Global Call Alarm Management System (GCAMS) enhancements, this parameter allows you to change the default threshold value of the Two out of



Four Consecutive Frame Bits (F bit) in Error (FERR) alarm (T1 alarm) by adding a parameter in the CONFIG file (.config) that corresponds to the PCD file in use on your board. The change is made per span. After threshold parameters are added, the FCD file is automatically updated when the new PCD file and modified CONFIG files are downloaded to the board.

**Values:** 0 to 255

**Guidelines:** To modify the default threshold for the FERR alarm, add the following parameter (sample value of 100 shown) to the [lineAdmin.x] section of a CONFIG file:

```
SetParm=0x163b,100 ! FERR threshold range 0 - 255, default 0
```

### ECS Threshold Range

**Number:** 0x163c

**Description:** To support the Dialogic® Global Call Alarm Management System (GCAMS) enhancements, this parameter allows you to change the default threshold value of the Frame Bit Error Count Saturation (ECS) alarm (T1 or E1 alarm) by adding a parameter in the CONFIG file (.config) that corresponds to the PCD file in use on your board. The change is made per span. After threshold parameters are added, the FCD file is automatically updated when the new PCD file and modified CONFIG files are downloaded to the board.

**Values:** 0 to 255

**Guidelines:** To modify the default threshold for the ECS alarm, add the following parameter (sample value of 100 shown) to the [lineAdmin.x] section of a CONFIG file:

```
SetParm=0x163c,100 ! ECS threshold range 0 - 255, default 0
```

### CECS Threshold Range

**Number:** 0x163d

**Description:** To support the Dialogic® Global Call Alarm Management System (GCAMS) enhancements, this parameter allows you to change the default threshold value of the CRC4 Error Count Saturation (CECS) alarm (E1 alarm) by adding a parameter in the CONFIG file (.config) that corresponds to the PCD file in use on your board. The change is made per span. After threshold parameters are added, the FCD file is automatically updated when the new PCD file and modified CONFIG files are downloaded to the board.

**Values:** 0 to 255

**Guidelines:** To modify the default threshold for the CECS alarm, add the following parameter (sample value of 100 shown) to the [lineAdmin.x] section of a CONFIG file:

```
SetParm=0x163d,100 ! CECS threshold range 0 - 255, default 255
```

## 13.3 [NFAS] Parameters

Non-Facility-Associated Signaling (NFAS) uses a single ISDN PRI D channel to provide signaling and control for multiple ISDN PRI lines. When using NFAS, modifications also need to be made to other sections of the CONFIG file. For details, see the following:

- “[SignalingType \(Signaling Type\)](#)”, on page 106.

- [Section 13.4, “\[NFAS.x\] Parameters”](#), on page 114.

There is only one NFAS component level parameter.

## **NFAS\_INSTANCE\_MAP (NFAS Instance Map)**

**Number:** 0x3E02

**Description:** The **NFAS\_INSTANCE\_MAP** parameter defines the number of NFAS groups or NFAS instances created on a particular board. One NFAS group is created for each primary D channel on the board.

**Values:**

- 0x0: 0 (0000)
- 0x1: 1 (0001)
- 0x3: 2 (0011)
- 0x7: 3 (0111)
- 0xF: 4 (1111)

**Guidelines:** The **NFAS\_INSTANCE\_MAP** parameter value is a hexadecimal bitmap that represents the number of NFAS groups that are needed. The bitmap’s least significant bit correlates to the first NFAS instance, the next least significant bit corresponds to the second NFAS instance, and so on. So, starting with the least significant bit and working towards the most significant bit, set each bit’s value to 1 for each NFAS instance needed. For example, to create three NFAS groups, set the value of the **NFAS\_INSTANCE\_MAP** parameter to 0x07 (0111).

## **13.4 [NFAS.x] Parameters**

Non-Facility-Associated Signaling (NFAS) uses a single ISDN PRI D channel to provide signaling and control for multiple ISDN PRI lines. For each group defined by the **NFAS\_INSTANCE\_MAP** parameter, there will be an [NFAS.x] section in the CONFIG file. For example, [NFAS.1] corresponds to the NFAS instance for the first group, [NFAS.2] corresponds to the NFAS instance for the second group, and so on.

When using NFAS, modifications also need to be made to other sections of the CONFIG file. For details, see the following parameters:

- [“NFAS\\_INSTANCE\\_MAP \(NFAS Instance Map\)”](#), on page 114.
- [“SignalingType \(Signaling Type\)”](#), on page 106

NFAS instance level parameters include:

- [GroupID \(Group Identifier\)](#)
- [NFAS\\_PrimaryIntID \(Primary Instance Identifier\)](#)
- [NFAS\\_Standby\\_IntID \(Standby Instance Identifier\)](#)

## **GroupID (Group Identifier)**

**Number:** 0x3E00

**Description:** The **GroupID** parameter is defined for each NFAS group created. This parameter defines the NFAS group including the trunks that are assigned to it.

**Values:** 1 to 4

**Guidelines:** When setting this parameter, the trunks assigned to the group must also be defined. For each group, multiple trunks are identified and added in recurring sets of triplets, using the following command:

```
AddNFASInterface(x)= a,b,c, a',b',c', ...
```

Where:

x = GroupID

NFAS group into which the interface needs to be added. For [NFAS.x], this would be “x”.

a = InterfaceID

Unique number for this interface assigned by the user. A maximum of 10 interfaces can be assigned to a single group.

b = BoardNumber

Logical board number (as defined by the **Logical ID** parameter) on which the trunk being assigned to the InterfaceID resides.

c = InstanceNumber

Instance number of the trunk that is being assigned to the InterfaceID. Trunks are numbered sequentially based on their physical location on the boards, from top to bottom.

For example, to add all four trunks on board 2 and the first two trunks on board 3 to the fourth NFAS group, enter the following to the [NFAS.4] section in the CONFIG file:

```
[NFAS.4]
AddNFASInterface(4)=0,2,1, 1,2,2, 2,2,3, 3,2,4, 4,3,1, 5,3,2
SetParm=0x3E04,0
```

## **NFAS\_PrimaryIntID (Primary Instance Identifier)**

**Number:** 0x3E04

**Description:** The **NFAS\_PrimaryIntID** parameter defines the primary D channel used by the NFAS group and is set for every [NFAS.x] group that is created.

**Values:** 0 to 9 (valid **InterfaceID** value)

**Guidelines:** The parameter is set to one of the [NFAS.x] InterfaceIDs defined by the **GroupID** parameter’s AddNFASInterface command. For details, see “**GroupID (Group Identifier)**”, on page 115.

For example, to define the primary D channel for NFAS group 4 to be the second trunk on board 3, enter the following to the [NFAS.4] section in the CONFIG file:

```
[NFAS.4]
AddNFASInterface(4)=0,2,1, 1,2,2, 2,2,3, 3,2,4, 4,3,1, 5,3,2
SetParm=0x3e04,5
```

## **NFAS\_Standby\_IntID (Standby Instance Identifier)**

**Number:** 0x3E05

**Description:** The **NFAS\_Standby\_IntID** parameter defines the standby, or backup, D channel used by the NFAS group. This parameter is set for every [NFAS.x] group that implements D channel backup (DCBU).

**Note:** DCBU is supported on boards using ISDN 4ESS, 5ESS, and NI-2 protocols.

**Values:** 0 to 9 (valid **InterfaceID** value)

**Guidelines:** The parameter is set to one of the [NFAS.x] InterfaceIDs defined by the **GroupID** parameter's **AddNFASInterface** command. For details about the **AddNFASInterface** command, see "**GroupID (Group Identifier)**", on page 115.

In the example:

```
[NFAS.4]
AddNFASInterface(4)=0,2,1, 1,2,2, 2,2,3, 3,2,4, 4,3,1, 5,3,2
SetParm=0x3e04,5
```

to define the first trunk on board 2 the standby D channel for the fourth NFAS group, add parameter 0x3e05 to the [NFAS.4] section of the CONFIG file and set it to a value of 0:

```
[NFAS.4]
AddNFASInterface(4)=0,2,1, 1,2,2, 2,2,3, 3,2,4, 4,3,1, 5,3,2
SetParm=0x3e04,5
SetParm=0x3e05,0
```

## **13.5 [CCS] Parameters**

Common Channel Signaling (CCS) supports ISDN PRI out-of-band signaling utilizing the Q.931 signaling protocol for messaging. The parameters in the [CCS] and [CCS.x] sections of the CONFIG file define the number of CCS component instances created and configure the parameters associated with each CCS instance.

The CCS parameters include:

- **INSTANCE\_MAP** (Instance Map)
- **CCS\_TMR\_302** (Q.931 Timer 302)
- **CCS\_TMR\_303** (Q.931 Timer 303)
- **CCS\_TMR\_304** (Q.931 Timer 304)
- **CCS\_TMR\_305** (Q.931 Timer 305)
- **CCS\_TMR\_308** (Q.931 Timer 308)
- **CCS\_TMR\_310** (Q.931 Timer 310)
- **CCS\_TMR\_313** (Q.931 Timer 313)
- **CCS\_TEI\_RETRY** (TEI Retry Timer)
- **CCS\_TEI\_STABILITY** (TEI Stability Timer)
- **SYMMETRICAL\_LINK** (Symmetrical Command Response Protocol)
- **CCS\_PROTOCOL\_MODE** (ISDN Protocol Mode)

- [CCS\\_SWITCH\\_TYPE](#) (Switch Type)
- [L2\\_TRACE](#) (Layer 2 Access Flag)

### **INSTANCE\_MAP (Instance Map)**

**Number:** 0x05

**Description:** The **INSTANCE\_MAP** parameter is a bitmap that defines the number of CCS instances created. A CCS instance is created for each network interface that supports common channel signaling. The bitmap's least significant bit corresponds to the CCS instance associated with the first network interface on the board. The next least significant bit corresponds to the CCS instance associated with the second network interface on the board, and so on. If the bit associated with a network interface has a value of 1, then a CCS instance is created for that network interface. For example, a value of 0x5 (0101) means that CCS instances 1 and 3 are created allowing for common channel signaling on network interfaces 1 and 3.

**Values:** 0 to 0xffff

### **CCS\_TMR\_302 (Q.931 Timer 302)**

**Number:** 0x14

**Description:** The **CCS\_TMR\_302** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

**Note:** This parameter only applies to E1 boards.

**Values:**

- 0: Use the default value for the switch (15000 ms)
- -1: Disable the timer (has the same effect as setting the timer value to 0)
- n > 1: Timer value (milliseconds)

### **CCS\_TMR\_303 (Q.931 Timer 303)**

**Number:** 0x0b

**Description:** The **CCS\_TMR\_303** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

**Values:**

- 0: Use the default value for the switch (4000 ms)
- -1: Disable the timer (has the same effect as setting the timer value to 0)
- n > 1: Timer value (milliseconds)

### **CCS\_TMR\_304 (Q.931 Timer 304)**

**Number:** 0x0c

**Description:** The **CCS\_TMR\_304** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

**Note:** This parameter only applies to E1 boards.

**Values:**

- 0: Use the default value for the switch (30000 ms)
- -1: Disable the timer (has the same effect as setting the timer value to 0)
- n > 1: Timer value (milliseconds)

### **CCS\_TMR\_305 (Q.931 Timer 305)**

**Number:** 0x0d

**Description:** The **CCS\_TMR\_305** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

**Values:**

- 0: Use the default value for the switch (4000 ms for T1, 30000 ms for E1)
- -1: Disable the timer (has the same effect as setting the timer value to 0)
- n > 1: Timer value (milliseconds)

### **CCS\_TMR\_308 (Q.931 Timer 308)**

**Number:** 0x0e

**Description:** The **CCS\_TMR\_308** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

**Values:**

- 0: Use the default value for the switch (4000 ms)
- -1: Disable the timer (has the same effect as setting the timer value to 0)
- n > 1: Timer value (milliseconds)

### **CCS\_TMR\_310 (Q.931 Timer 310)**

**Number:** 0x0f

**Description:** The **CCS\_TMR\_310** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

**Values:**

- 0: Use the default value for the switch (10000 ms)
- -1: Disable the timer (has the same effect as setting the timer value to 0)
- n > 1: Timer value (milliseconds)

### **CCS\_TMR\_313 (Q.931 Timer 313)**

**Number:** 0x10

**Description:** The **CCS\_TMR\_313** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

**Values:**

- 0: Use the default value for the switch (4000 ms)
- -1: Disable the timer (has the same effect as setting the timer value to 0)
- n > 1: Timer value (milliseconds)

### **CCS\_TEI\_RETRY (TEI Retry Timer)**

**Number:** 0x15

**Description:** The **CCS\_TEI\_RETRY** parameter defines the maximum amount of time that the data link remains in state 4 (TEI\_ASSIGNED) before transitioning to state 5 (TEI\_WAIT\_ESTABLISH).

**Values:** Time (milliseconds)

### **CCS\_TEI\_STABILITY (TEI Stability Timer)**

**Number:** 0x16

**Description:** The **CCS\_TEI\_STABILITY** parameter defines the minimum transition time between data link state 4 (TEI\_ASSIGNED) and data link state 5 (TEI\_WAIT\_ESTABLISH).

**Values:** 0 to 100,000 (milliseconds)

### **SYMMETRICAL\_LINK (Symmetrical Command Response Protocol)**

**Number:** 0x13

**Description:** The **SYMMETRICAL\_LINK** parameter enables or disables symmetrical data link operations.

**Values:**

- 0: Disable symmetrical data link operations
- 1: Enable symmetrical data link operations

### **CCS\_PROTOCOL\_MODE (ISDN Protocol Mode)**

**Number:** 0x17

**Description:** The **CCS\_PROTOCOL\_MODE** parameter sets the network user-side protocol. User-side protocol is also known as TE (terminal emulation) protocol and Network-side protocol

## CONFIG File Parameter Reference for Dialogic® HMP Interface Boards

is also known as NT (network termination) protocol. This parameter also can be used to configure QSIG Master/Slave.

**Note:** Master/Slave mode pertains to QSIG protocols only.

**Note:** With the exception of the QSIG protocol (where the User-side and Network-side protocols are symmetrical), using the CSS\_PROTOCOL\_MODE parameter to configure a Network-side protocol is supported for back-to-back testing purposes only. The Network-side firmware is not fully qualified for operation in a deployment environment.

### Values:

- 0: User or Slave Mode (QSIG)
- 1: Network or Master Mode (QSIG)

## CCS\_SWITCH\_TYPE (Switch Type)

**Number:** 0x07

**Description:** The CCS\_SWITCH\_TYPE parameter defines the network switch type.

### Values:

- 1: 4ESS
- 2: 5ESS
- 3: DMS
- 4: NTT
- 6: DASS2
- 7: NET5
- 10: QSIGE1
- 11: QSIGT1
- 12: NI2
- 13: DPNSS

## L2\_TRACE (Layer 2 Access Flag)

**Number:** 0x09

**Description:** The L2\_TRACE parameter is the ISDN Layer 2 access flag. When Layer 2 (Data Link layer) access is disabled, ISDN Link Access Protocol for the D channel (LAPD) functionality is obtained by accessing ISDN Call Control and Layer 3 (Network layer). When Layer 2 access is enabled, call control is no longer supported for the channels on this line and ISDN LAPD functionality is obtained by accessing Layer 2 directly.

### Values:

- 0: Disable Layer 2 access
- 1: Enable Layer 2 access



## 13.6 [CHP] Parameters

The Channel Protocol (CHP) component implements the telephony communication protocol that is used on the network interface. The CHP component parameters include:

- [R4Compatibility](#) (R4 Compatibility Flag)
- [InitialChanState](#) (Initial Channel State)
- [DisableBlock](#) (Disable Block)

### [R4Compatibility](#) (R4 Compatibility Flag)

**Number:** 0x1310

**Description:** The **R4Compatibility** parameter enables or disables R4 (Dialogic® Global Call API) compatibility features. This parameter also enables retrieval of DNIS and ANI information in the offered call state.

**Values:**

- 0 [default]: Disable R4 compatibility (same effect as the value 2)
- 1: Enable R4 compatibility
- 2: Disable R4 compatibility (same effect as the value 0)

### [InitialChanState](#) (Initial Channel State)

**Number:** 0x1311

**Description:** The **InitialChanState** parameter defines the initial B channel state (CHP channel state) at the end of system initialization. The initial state of the ISDN B channel is either InService or OutOfService. Once the board is initialized, this initial state will be set on all channels of the board until a user application is invoked and explicitly modifies the state of the channel.

**Values:**

- 1: InService
- 2: OutOfService

**Guidelines:** This parameter must be set to OutOfService for ISDN protocols.

### [DisableBlock](#) (Disable Block)

**Number:** 0x1312

**Description:** The **DisableBlock** parameter defines whether or not a blocking pattern (message) is sent on a channel when the channel is in the OutofService state. When **DisableBlock** is disabled, no pattern is sent (the switch will not present calls to the B channel).

When **DisableBlock** is enabled and a channel is in the InService state (**InitialChanState=1**), the protocol will send a non-blocking pattern on the channel (the switch will present calls to the B channel). When **DisableBlock** is enabled and a channel is in the OutofService state (**InitialChanState=2**), the protocol will send a blocking pattern on the channel (the switch will

present calls to the B channel but these calls will be abandoned by the switch since the application will not respond to the call).

**Values:**

- 0: Disable blocking
- 1: Enable blocking

## 13.7 [CHP] ISDN Protocol Variant Definitions

The CHP parameters define line configurations. Within the [CHP] section of the CONFIG file, ISDN protocol variants are defined using the `Variant Define n` command. For a detailed description of the `Variant Define n` command, see [Section 11.4, “\[CHP\] Section”](#), on page 89.

The ISDN protocol variant parameters include:

- `ProtocolType` (Protocol Type)
- `InterCallDelay` (Inter-call Delay)
- `DisconnectTimeout` (Disconnect Timeout)
- `Layer1Protocol` (Layer 1 Protocol)
- `InfoTransferRate` (Information Transfer Rate)
- `InfoTransferCap` (Information Transfer Cap)
- `CalledNumberType` (Called Number Type)
- `CalledNumberPlan` (Called Number Plan)
- `CalledNumberCount` (Called Number Count)
- `CallingNumberType` (Calling Number Type)
- `CallingNumberPlan` (Calling Number Plan)
- `CallingNumberPresentation` (Calling Number Presentation)
- `CallingNumberScreening` (Calling Number Screening)
- `CallingNumberCount` (Calling Number Count)
- `CallProgress` (Call Progress)
- `CaHdgLoHiGl` (Hello Edge/Low Glitch/High Glitch)
- `CaAnsdglPSV` (Answer Deglitcher/PAMD Speed Value)
- `CaHdgLoHiGl` (Hello Edge/Low Glitch/High Glitch)
- `CaBusySet` (Busy Signal)
- `CaSitSet` (SIT Signal)
- `CaFaxSet` (Fax Signal)
- `CaPvdId` (Voice Detection Signal)
- `CaPamdId` (Answering Machine Signal)
- `CaSignalTimeout` (Signal Timeout)
- `CaAnswerTimeout` (Answer Timeout)

- [CaPvdTimeout \(Voice Detection Timeout\)](#)

### **ProtocolType (Protocol Type)**

**Description:** The **ProtocolType** parameter defines the type of ISDN protocol used on a channel. The value of the parameter is dependent on the firmware being downloaded and the CONFIG files used. For example, when downloading the *ghmp1\_hmpdsb\_2\_4ess.config* file, **ProtocolType** should be set to a value of 1.

**Note:** The **ProtocolType** parameter is also used when defining T1 protocol variants.

**Values:**

- 1: 4ESS and NI-2
- 2: 5ESS
- 3: DMS100 and DMS250
- 4: NTT
- 7: NET5 and QSIG

### **InterCallDelay (Inter-call Delay)**

**Description:** The **InterCallDelay** parameter defines the minimum amount of time between outbound calls.

**Note:** The **InterCallDelay** parameter is also used when defining T1 protocol variants.

**Values:**  $n > 0$  (milliseconds)

### **DisconnectTimeout (Disconnect Timeout)**

**Description:** The **DisconnectTimeout** parameter defines the time delay between proceeding and alert/connect. The call will transition to idle after this time period (sooner if ClearConf is received).

**Note:** The **DisconnectTimeout** parameter is also used when defining T1 protocol variants.

**Values:**  $n > 0$  (milliseconds)

**Guidelines:** None.

### **Layer1Protocol (Layer 1 Protocol)**

**Description:** The **Layer1Protocol** parameter defines the User Layer 1 Protocol.

**Values:**

- 0x00: Protocol not present
- 0x01: CCITT
- 0x02: G.711 mu-law
- 0x03: G.711 A-law
- 0x04: G.721 ADPCM
- 0x05: G.721 kHz
- 0x06: 384 kHz Video
- 0x07: NS Rate Adaption
- 0x08: V120 Rate Adaption
- 0x09: X.31 HDLC

### **InfoTransferRate (Information Transfer Rate)**

**Description:** The **InfoTransferRate** parameter defines the information transfer rate.

**Values:**

- 0x00: Rate undefined
- 0x10: 64 kbps
- 0x11: 128 kbps
- 0x13: 384 kbps
- 0x15: 1536 kbps
- 0x17: 1920 kbps
- 0x18: Multi-rate

### **InfoTransferCap (Information Transfer Cap)**

**Description:** The **InfoTransferCap** parameter defines the information transfer capability.

**Values:**

- 0x00: Speech
- 0x08: Unrestricted digital
- 0x09: Restricted digital
- 0x10: 3 kHz
- 0x11: 7 kHz
- 0x18: Video

### **CalledNumberType (Called Number Type)**

**Description:** The **CalledNumberType** parameter defines the type of outbound calls (Called Party Numbers).

**Values:**

- 0x00: Unknown
- 0x01: International
- 0x02: National
- 0x03: Network specific
- 0x04: Network subscriber
- 0x06: Network abbreviated

### **CalledNumberPlan (Called Number Plan)**

**Description:** The **CalledNumberPlan** parameter defines the numbering plan to use for outbound calls (Called Party Numbers).

**Values:**

- 0x00: Unknown
- 0x01: ISDN
- 0x02: Telephony
- 0x03: Date X.121
- 0x04: Telex F.69
- 0x08: National standard
- 0x09: Private

### **CalledNumberCount (Called Number Count)**

**Description:** The **CalledNumberCount** parameter defines the number of digits to collect from an incoming call.

**Values:**

- 0: Collect all the digits provided
- n: Number of digits to collect

### **CallingNumberType (Calling Number Type)**

**Description:** The **CallingNumberType** parameter defines the type of outbound call (Calling Party Number).

**Values:**

- 0x00: Unknown
- 0x01: International
- 0x02: National
- 0x03: Network specific
- 0x04: Network subscriber
- 0x06: Network abbreviated

### **CallingNumberPlan (Calling Number Plan)**

**Description:** The **CallingNumberPlan** parameter defines the numbering plan to use for outbound calls (Calling Party Numbers).

**Values:**

- 0x00: Unknown
- 0x01: ISDN
- 0x02: Telephony
- 0x03: Date X.121
- 0x04: Telex F.69
- 0x08: National standard
- 0x09: Private

### **CallingNumberPresentation (Calling Number Presentation)**

**Description:** The **CallingNumberPresentation** parameter defines the presentation for calling number (outbound calls).

**Values:**

- 0x00: Allowed
- 0x01: Restricted
- 0x02: Not available

### **CallingNumberScreening (Calling Number Screening)**

**Description:** The **CallingNumberScreening** parameter defines the screening for calling number (outbound calls).

**Values:**

- 0x00: User provided
- 0x01: Verified and passed
- 0x02: Verified and failed
- 0x03: Network provided

## **CallingNumberCount (Calling Number Count)**

**Description:** The **CallingNumberCount** parameter defines the number of Calling Party Number digits to collect from incoming call.

**Values:**

- 0: Collect all the digits provided
- n: Number of digits to collect

## **CallProgress (Call Progress)**

**Description:** The **CallProgress** parameter enables or disables call progress detection for call setup.

**Note:** The **CallProgress** parameter is also used when defining T1 protocol variants.

**Values:**

- y: Enable call progress detection
- n: Disable call progress detection

## **CaHdgLoHiGl (Hello Edge/Low Glitch/High Glitch)**

The **CaHdgLoHiGl** parameter combines three parameters into one. They include the Hello Edge, Low Glitch, and High Glitch parameters. The values for all three parameters are contained in the **CaHdgLoHiGl** parameter value, 0xFF020F13, where 02 is the default hexadecimal value (2 decimal) for the Hello Edge parameter, 0F is the default hexadecimal value (15 decimal) for the Low Glitch parameter, and 13 is the hexadecimal value (19 decimal) for the High Glitch parameter.

**Description:** The **Hello Edge** parameter defines the point at which a connect will be returned to the application.

**Values:**

- 1: Rising edge (immediately when a connect is detected)
- 2: Falling edge (after the end of the salutation)

**Description:** The **Low Glitch** parameter defines, in intervals of 10 milliseconds, the maximum silence period to ignore. This maximum silence period helps to eliminate spurious silence intervals.

**Values:** The default value is 15 decimal (150 milliseconds).

**Description:** The **High Glitch** parameter defines, in intervals of 10 milliseconds, the maximum nonsilence period to ignore. This maximum nonsilence period helps to eliminate spurious nonsilence intervals.

**Values:** The default value is 19 decimal (190 milliseconds).

## **CaAnsdgIPSV (Answer Deglitcher/PAMD Speed Value)**

The **CaAnsdgIPSV** parameter combines two parameters into one. They include the Answer Deglitcher and PAMD Speed Value parameters. The values for both parameters are contained in

## CONFIG File Parameter Reference for Dialogic® HMP Interface Boards

the **CaAnsdglPSV** parameter value, 0xFFFFF01, where 01 is the default hexadecimal value (1 decimal) for the PAMD Speed Value parameter and FF is the default hexadecimal value (-1 decimal) for the Answer Deglitcher parameter, which corresponds to disabling it. This parameter should only be enabled if you are concerned with measuring the length of the salutation.

**Description:** The **Answer Deglitcher** parameter defines the maximum silence period, in 10 millisecond intervals, allowed between words in a salutation.

**Values:** The default value is -1 (FFFF), for disabled.

**Description:** The **PAMD Speed Value** parameter defines the PAMD algorithm: PAMD\_ACCU, PAMD\_FULL, and PAMD\_QUICK. PAMD\_QUICK provides the fastest results based on the connect circumstances, but is the least accurate. PAMD\_FULL performs hiss noise analysis to determine if this is an answer machine response, and then performs a full evaluation of the voice response if the hiss information is not sufficient to make the decision. PAMD\_ACCU will not perform hiss noise analysis, since this is not required with today's digital answering systems, but will perform a full answer size voice response to achieve the most accurate result.

**Values:**

- 1 [default]: PAMD\_ACCU
- 2: PAMD\_FULL
- 3: PAMD\_QUICK

### CaRingingSet (Ringing Signal)

**Description:** The **CaRingingSet** parameter defines the signal set used to detect ringing for call progress analysis.

**Note:** The **CaRingingSet** parameter is also used when defining T1 protocol variants.

**Values:** 0x024940

### CaBusySet (Busy Signal)

**Description:** The **CaBusySet** parameter defines the signal set used to detect busy for call progress analysis.

**Note:** The **CaBusySet** parameter is also used when defining T1 protocol variants.

**Values:** 0x004DE0

### CaSitSet (SIT Signal)

**Description:** The **CaSiteSet** parameter defines the signal set used to detect Special Information Tones (SIT) tones for call progress analysis.

**Note:** The **CaSiteSet** parameter is also used when defining T1 protocol variants.

**Values:** 0x02F240



### **CaFaxSet (Fax Signal)**

**Description:** The **CaFaxSet** parameter defines the signal set used to detect fax tones for call progress analysis.

**Note:** The **CaFaxSet** parameter is also used when defining T1 protocol variants.

**Values:** 0x014B80

### **CaPvdId (Voice Detection Signal)**

**Description:** The **CaPvdId** parameter defines the signal to use for positive voice detection in call progress analysis.

**Note:** The **CaPvdId** parameter is also used when defining T1 protocol variants.

**Values:** 0x01F4C1

### **CaPamdId (Answering Machine Signal)**

**Description:** The **CaPamdId** parameter defines the signal to use for positive answering machine detection in call progress analysis.

**Note:** The **CaPamdId** parameter is also used when defining T1 protocol variants.

**Values:** 0x01A041

### **CaSignalTimeout (Signal Timeout)**

**Description:** The **CaSignalTimeout** parameter defines the maximum amount of time to wait to detect a call progress tone from one of the call analysis signal sets. For T1 loop start and ground start protocols, if this time is exceeded, then the outbound call will fail with the reason being NoAnswer.

**Note:** The **CaSignalTimeout** parameter is also used when defining T1 protocol variants.

**Values:**  $n > 0$  (the value must be a multiple of 10 ms)

### **CaAnswerTimeout (Answer Timeout)**

**Description:** The **CaAnswerTimeout** parameter defines the maximum amount of time that call analysis will wait for ringback to stop (equivalent to the number of rings). If this time is exceeded, then the outbound call will fail with the reason being NoAnswer.

**Note:** The **CaAnswerTimeout** parameter is also used when defining T1 protocol variants.

**Values:**  $n > 0$  (the value must be a multiple of 10 ms)

### **CaPvdTimeout (Voice Detection Timeout)**

**Description:** The **CaPvdTimeout** parameter defines the maximum amount of time that call analysis will wait to detect positive answering machine detection (PAMD) or positive voice detection (PVD) once ringback has ceased. If this time is exceeded, then the call state will

transition to “Connected” with the reason being Normal. If PAMD or PVD is detected within this time period, then the “Connected” reason will be PAMD or PVD respectively.

**Note:** The **CaPvdTimeout** parameter is also used when defining T1 protocol variants.

**Values:**  $n > 0$  (the value is expressed in multiples of 10 milliseconds. For example, a value of 200 equals 2000 milliseconds, or 2 seconds)

- For digital boards: default = 400

## 13.8 [TSC] Parameters

The parameter in the [TSC] section of the CONFIG file is associated with the B channel sets.

### Encoding (Encoding Method)

**Number:** 0x1209

**Description:** The **Encoding** parameter defines the encoding method used on a line.

**Values:**

- 1: A-law
- 2: mu-law

## 13.9 [TSC] defineBSet Parameters

The parameters defined by the `defineBSet` command in the [TSC] section of the CONFIG file are associated with the B channel sets. The syntax of the `defineBSet` command is:

```
defineBSet = SetId, LineId, StartChan, NumChans, BaseProtocol, Inbound, OutBound, DChanDesc, Admin, Width, BChanId, SlotId, Direction, Count, [BChanId, SlotId, Direction, Count,] 0
```

**Note:** The [TSC] `defineBSet` parameters do not have parameter numbers explicitly defined within the CONFIG file.

The `defineBSet` parameters include:

- **SetId** (Set Identifier)
- **LineId** (Line Identifier)
- **StartChan** (Start Channel)
- **NumChans** (Number of B Channels)
- **BaseProtocol** (Base Protocol)
- **Inbound** (Inbound Variant)
- **Outbound** (Outbound Variant)
- **DChanDesc** (D Channel Identifier)
- **Admin** (Admin)
- **Width** (Width)

- [BChanId \(B Channel Identifier\)](#)
- [SlotId \(Slot Identifier\)](#)
- [Direction \(Direction\)](#)
- [Count \(Count\)](#)

## SetId (Set Identifier)

**Description:** The **SetId** parameter is an arbitrary identifier set by the user that identifies the B channel set in which the B channels are a member.

**Values:** Number

**Guidelines:** Each B channel set must have a unique identifier.

For example, for each line on a board, **SetId** can be set sequentially to a value that is a multiple of 10 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23, 0,1,1,2,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23, 0,1,1,3,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23, 0,1,1,4,20,1, 1,1,3,23,0
```

## LineId (Line Identifier)

**Description:** The **LineId** parameter defines the T1 or E1 line that carries all of the B channels in the set.

**Values:** 1 to 16

**Guidelines:** For example, on a board with four network interfaces, the value of **LineId** is set to 1 for line 1, 2 for line 2, and so on for each line as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

## StartChan (Start Channel)

**Description:** The **StartChan** parameter defines the first B channel in the set. This parameter is used in combination with the **NumChans** parameter to define a contiguous set of B channels.

**Values:** The value range depends on the technology, because the number of available B channels varies.

- 1 to 24: T1
- 1 to 30: E1
- 1 to 31: E1 clear channel

**Guidelines:** For example, on a T1 line where 23 of the 24 channels are used as B channels, the value of **StartChan** is set to 1 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

## **NumChans (Number of B Channels)**

**Description:** The **NumChans** parameter defines the total number of B channels in the set. This parameter is used in combination with the **StartChan** parameter to define a contiguous set of B channels.

**Values:** The range of values varies with technology because the number of time slots varies.

- 1 to 24: T1
- 1 to 30: E1
- 1 to 31: E1 clear channel

**Guidelines:** For example, on a T1 line, a value of 1 for **StartChan** and a value of 23 for **NumChans** defines 23 B channels numbered from 1 to 23:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23, 0,1,1,2,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23, 0,1,1,3,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23, 0,1,1,4,20,1, 1,1,3,23,0
```

## **BaseProtocol (Base Protocol)**

**Description:** The **BaseProtocol** parameter defines the base protocol on which the B channel set will run.

**Values:**

- 0: T1 CAS, ISDN or Dialogic® Global Call protocols (where the default protocol is defined by the firmware) or clear channel

**Guidelines:** For T1 CAS, ISDN, and Dialogic® Global Call protocols, each firmware load supports only one base protocol, so this parameter will be set to 0 for these protocols. This parameter is also set to 0 for clear channel. Clear channel is the ability to access telephony channels in the system and configure them to a user-defined call control protocol, or to simply leave the lines “clear”. The resources should have access to the telephony bus for media routing purposes, as well as signal detection, signal generation, and tone generation capabilities, if desired.

For example, on T1 ISDN lines, **BaseProtocol** is set to a value of 0 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

## **Inbound (Inbound Variant)**

**Description:** The **Inbound** parameter selects one of the protocol type variant parameter sets defined in the [CHP] section of the CONFIG file to use for inbound calls. The protocol variant defines the type of protocol running on the set of B channels.

**Values:**

- 0: Clear channel (disable inbound calls)
- n: Variant identifier as defined in the [CHP] section of the CONFIG file

**Guidelines:** This parameter is set to 0 for clear channel. Clear channel is the ability to access telephony channels in the system and configure them to a user-defined call control protocol, or to simply leave the lines “clear”. The resources should have access to the telephony bus for media routing purposes, as well as signal detection, signal generation, and tone generation capabilities, if desired.

For example, on T1 ISDN lines, **Inbound** is set to a value of 1 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

## Outbound (Outbound Variant)

**Description:** The **Outbound** parameter selects one of the protocol type variant parameter sets defined in the [CHP] section of the CONFIG file to use for outbound calls. The protocol variant defines the type of protocol running on the set of B channels.

**Values:**

- 0: Clear channels (disable outbound calls)
- n: Variant identifier as defined in the [CHP] section of the CONFIG file

**Guidelines:** This parameter is set to 0 for clear channel (disable outbound calls). Clear channel is the ability to access telephony channels in the system and configure them to a user-defined call control protocol, or to simply leave the lines “clear”. The resources should have access to the telephony bus for media routing purposes, as well as signal detection, signal generation, and tone generation capabilities, if desired.

For example, on T1 ISDN lines, **Outbound** is set to a value of 1 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

## DChanDesc (D Channel Identifier)

**Description:** The **DChanDesc** parameter is an ISDN parameter that identifies which trunk the D-channel resides for this B-set. This parameter is ignored for T1 CAS, clear channel, and Dialogic® Global Call protocols.

**Values:** 1 to 16

**Guidelines:** For example, on a board with four T1 ISDN lines, **DChanDesc** is set as follows:

```
defineBSet=10,1,1,24, 0,1,1,1,20,1, 1,1,3,24,0  
defineBSet=20,2,1,24, 0,1,1,2,20,1, 1,1,3,24,0  
defineBSet=30,3,1,24, 0,1,1,3,20,1, 1,1,3,24,0  
defineBSet=40,4,1,24, 0,1,1,4,20,1, 1,1,3,24,0
```

## Admin (Admin)

**Description:** The **Admin** parameter is an arbitrary 32-bit value set by the user that is exported to the TSC\_AttrAdminGroup attribute of the TSC cluster for each B channel in the set. This attribute can be used to find and/or allocate TSC clusters.

**Values:** 0 to 0xFFFFFFFF

**Guidelines:** For example, on a T1 line, **Admin** is set to a value of 20 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

## Width (Width)

**Description:** The **Width** parameter specifies the number of time slots used by each B channel. Currently, only one time slot per channel is used.

**Note:** This **Width** should not be modified by the user.

**Values:** 1

**Guidelines:** For example, on a T1 line, **Width** is set to a value of 1 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

## BChanId (B Channel Identifier)

**Description:** The **BChanId** parameter defines the initial B channel in the set to which the TSC instance is associated. It is also the channel to which the initial time slot, defined by **SlotId**, will be mapped. B channels are then sequentially mapped to time slots for a count of **Count**.

**Values:** The range of values varies with technology because the number of time slots varies.

- 1 to 24: T1
- 1 to 31: E1

**Guidelines:** For example, on a T1 board where the D channel is mapped to time slot 24 on all four lines, **BChanId** and **SlotId** are set to a value of 1 and **NumChans** is set to a value of 23. This defines 23 B channels numbered 1 to 23 mapped to time slots 1 to 23.

```
defineBSet=10,1,1,23,0,1,1,1,20,1, 1,1,3,23,0  
defineBSet=20,2,1,23,0,1,1,2,20,1, 1,1,3,23,0  
defineBSet=30,3,1,23,0,1,1,3,20,1, 1,1,3,23,0  
defineBSet=40,4,1,23,0,1,1,4,20,1, 1,1,3,23,0
```

For E1 ISDN lines that usually contain a D channel mapped to time slot 16, the mapping of channels to time slots occurs in two sets of **BChanId**, **SlotId**, **Direction** and **Count** definitions. The first set of definitions maps time slots before the D channel and the second set maps time slots after the D channel.

For example, on an E1 ISDN board with four network interfaces, where time slot 16 is used for signaling on all four lines, **BChanId** would be defined on each line as follows:

## CONFIG File Parameter Reference for Dialogic® HMP Interface Boards

```
defineBSet=10,1,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=20,2,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=30,3,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=40,4,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
```

In this example, channels 1 to 15 are mapped to time slots 1 to 15 and channels 16 to 30 are mapped to time slots 17 to 31.

For E1 clear channel lines where the time slot 16 is not used for signaling, additional `defineBSet` commands are added to clear channel 31. Both **StartChan** and **BChanId** are set to a value of 31, **NumChans** and **Count** are set to a value of 1, and **SlotId** is set to 16 as follows:

```
defineBSet=50,1,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=60,2,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=70,3,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=80,4,31,1, 0,0,0,1,21,1, 31,16,3,1,0
```

### SlotId (Slot Identifier)

**Description:** The **SlotId** parameter defines the logical time slot the initial B channel, defined by **BChanId**, is using. B channels are then sequentially mapped to time slots for a count of **Count**.

**Values:** The range of values varies with technology because the number of time slots varies.

- 1 to 24: T1
- 1 to 31: E1 ISDN
- 1 to 31: E1 clear channel

**Guidelines:** For E1 ISDN, the mapping of channels to time slots occurs in two sets of **BChanId**, **SlotId**, **Direction** and **Count** definitions. The first set of definitions maps the time slots before the D channel, and the second set maps the slots after the D channel.

For example, on an E1 ISDN board with four network interfaces, where time slot 16 is used for signaling on all four lines, **SlotId** for all four lines would be as follows

```
defineBSet=10,1,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=20,2,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=30,3,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=40,4,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
```

For all lines in this example, channels 1 to 15 are sequentially mapped to time slots 1 to 15 and channels 16 to 30 are mapped to time slots 17 to 31.

For E1 clear channel lines where time slot 16 is not used for signaling, additional `defineBSet` commands are added to clear channel 31 and to map time slot 16. Both **StartChan** and **BChanId** are set to a value of 31, **NumChans** and **Count** are set to a value of 1, and **SlotId** is set to 16 as follows:

```
defineBSet=50,1,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=60,2,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=70,3,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=80,4,31,1, 0,0,0,1,21,1, 31,16,3,1,0
```

## Direction (Direction)

**Description:** The **Direction** parameter defines the direction in which the data can be sent: inbound, outbound, or both.

**Values:**

- 1: Inbound
- 2: Outbound
- 3: Both

**Guidelines:** For example, on an T1 line where data is transferred both inbound and outbound, **Direction** is set to a value of 3 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,2,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,3,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,4,20,1, 1,1,3,23,0
```

## Count (Count)

**Description:** The **Count** parameter defines the number of time slots that are being mapped to B channels. This value is limited to the value of **NumChans** since only the number of channels that exist on a line can be mapped to a time slots.

**Values:** 1 to **NumChans**

**Guidelines:** For example, on a T1 line containing two network interfaces, where time slot 24 is used as a D channel on both lines, the **Count** for both lines would be as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

For an E1 line, **Count** is set to a value of 30 for lines that contain only B channels. For lines that contain a single D channel, the mapping of channels to time slots occurs in two sets of **BChanId**, **SlotId**, **Direction** and **Count** definitions. The first set of definitions maps the time slots before the D channel, and the second set maps the slots after the D channel. For example, on an E1 board with four network interfaces, where time slot 16 is used for signaling on all four lines, the **Count** for all four lines would be as follows:

```
defineBSet=10,1,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=20,2,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=30,3,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=40,4,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
```

For all lines in this example, channels 1 to 15 are mapped to time slots 1 to 15 and channels 16 to 30 are mapped to time slots 17 to 31.

For E1 clear channel lines where the time slot 16 is not used for signaling, additional **defineBSet** commands are added to clear channel 31 and to map time slot 16. **Count** is set to a value of 1 (also the value of **NumChans**) as follows:

```
defineBSet=50,1,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=60,2,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=70,3,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=80,4,31,1, 0,0,0,1,21,1, 31,16,3,1,0
```



## **13.10 [sigDet] Parameters**

**Note:** The information in this section only applies to the Dialogic® DNI601TEPHMP board.

The signal detector section of the CONFIG file may include qualification templates for positive answering machine detection (PAMD) and positive voice detection (PVD) used in call progress analysis.

A tech note was written to provide instructions for modifying the PAMD and PVD qualification template parameters on Dialogic® DM3 boards to accomplish higher successful PAMD and PVD rates. For further information about these parameters, see the tech note at <http://www.dialogic.com/support/helpweb/dxall/tnotes/legacy/2000/tn030.htm>. Although the tech note was written for Dialogic® System Release 5.1.1 Feature Pack 1 for Windows, the information applies to subsequent releases.

The modified parameters have now become the default in the firmware, so it is no longer necessary to tune the PAMD and PVD parameters as explained in the tech note. Operating with these new default values should result in improved accuracy of call progress analysis on Dialogic® DM3 boards. However, although these values are the most commonly used, they may not be suitable for every application environment. If needed, the PAMD and PVD templates are still tunable, as explained in the tech note, to achieve even better results based on the individual application environment.

The default PAMD qualification template ID is 106561 (0x1a041), but other valid PAMD qualification template IDs that can be defined in the CONFIG file are:

- 106564 (0x1a044)
- 106565 (0x1a045)
- 106566 (0x1a046)
- 106567 (0x1a047)

The default PVD qualification template ID is 128193 (0x1f4c1), but other valid PVD qualification template IDs that can be defined in the CONFIG file are:

- 128194 (0x1f4c2)
- 128195 (0x1f4c3)
- 128196 (0x1f4c4)
- 128197 (0x1f4c5)

**CONFIG File Parameter Reference for Dialogic® HMP Interface Boards**