Intel® Springware Architecture Products on Linux

Configuration Guide

November 2004

05-2399-001

Intel may make changes to specifications and product descriptions at any time, without notice.

Publication Date: November 2004

Document Number: 05-2399-001

Intel Converged Communications, Inc.
1515 Route 10
Parsippany, NJ 07054

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

For **Products and Services Information**, visit the Intel Telecom Products website at:
http://www.intel.com/design/network/products/telecom

For **Sales Offices** and other contact information, visit the Where to Buy Intel Telecom Products page at:
http://www.intel.com/buy/wtb/wtb1028.htm

Contents

Figures

Tables

Revision History

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

Document No.	Publication Date	Description of Revisions
05-2399-001	November 2004	Initial version of document.

About This Publication

The following topics provide information about this *Intel® Springware Architecture Products on Linux Configuration Guide*.

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

Purpose

This guide provides information about configuring Intel® Dialogic® Springware Architecture PCI boards in a Linux environment. Configuration procedures are included as well as descriptions of configuration files and configuration parameters.

Intended Audience

This information is for:

- Developers
  - System, application, and technology developers
  - Toolkit vendors
  - VARs/system integrators
- System Operators:
  - System and network administrators
  - Support personnel (crafts person)

How to Use This Publication

This information is organized as follows:

- Chapter 1, "Configuration Overview" describes the major configuration steps in the order in which they are performed, and provides a brief overview of each aspect of configuring a system containing Intel Dialogic on Springware architecture boards.
- Chapter 2, "Configuration Details" provides details about using the Dialogic.cfg file. Also provides details about SNMP agent configuration and the *Voice.prm* file.
- Chapter 3, "Configuration Procedures" contains detailed procedural information for configuring a system that uses Springware architecture boards.

- Chapter 4, “Dialogic.Cfg Parameter Reference” describes each parameter associated with the Dialogic.cfg file. Included are a description, list of values, and configuration guidelines.
- Chapter 5, “DNI Parameter Reference” describes each parameter associated with the digital network interface (DNI) parameter file (*Spandti.prm*). Included are a description, list of values, and configuration guidelines.
- Chapter 6, “Silence Compressed Record Parameter Reference” describes each parameter associated with Silence Compressed Record parameters contained in the *Voice.prm* file.

Related Information

For additional information related to configuring an Intel Dialogic product, see the following:

- For timely information that may affect configuration, see the Release Guide and Release Update. Be sure to check the online Release Update for the system release you are using for any updates or corrections to this publication.
- For information about installing the system software, see the system software installation guide supplied with your release.
- For information about administrative tasks related to this release, see the system administration guide supplied with your release.
- For information about administration functions relating to the SNMP agent software, see the *SNMP Agent Software for Linux Operating Systems Administration Guide*.
- *http://www.intel.com/design/network/products/telecom* for product information
- The Intel® Telecom Support Resources Web site at: *http://developer.intel.com/design/telecom/support/* provides wide ranging information in the form of technical notes, problem tracking, application notes, as well as other helpful documentation.

Configuration Overview 1

The configuration overview describes the major configuration steps in the order in which they are performed. The overview also provides a brief overview of each aspect of configuring a system containing Intel® Dialogic® boards based on the Springware Architecture.

1.1 Major Configuration Steps

The following major steps are used to configure a system containing Intel Dialogic Springware boards.

1. Modifying the Dialogic.cfg file parameters
2. Configuring Simple Network Management Protocol (SNMP) agent software (optional)
3. Assigning time slots when using a third-party board as the clock master (optional)
4. Configuring Digital Network Interface (DNI) parameters (optional)
5. Configuring Voice Parameters (optional)
6. Using Non-Facility Associated Signaling (NFAS)
7. Verifying Device names (optional)
8. Configuring Global Call CDP files (optional)
9. Intializing the system
10. Updating the PCI ID for D/42JCT-U and D/82JCT-U boards (optional)
11. Reconfiguring the system (optional)

1.2 The Configuration Process

When the `install.sh` installation procedure is completed, you can configure Intel Dialogic boards. You start the procedure by executing `config.sh` as explained in Section 3.3, “Starting the Config.sh Utility”, on page 22. For SpringWare boards, a utility called `mkcfg` is automatically invoked to prompt for configuration information. The procedure is described in Section 3.4, “Using the Mkcfg Utility”, on page 23. Reference information about the parameters, including configuration guidelines, can be found in *Chapter 4, “Dialogic.Cfg Parameter Reference”*.

Completing this procedure results in the creation of an ASCII text file named *dialogic.cfg* containing the SpringWare configuration information that you entered. The file is saved in */usr/dialogic/cfg* and is used by the downloader to initialize the system when the Intel Dialogic boards are started. In addition to modifying the *dialogic.cfg* file parameters, the configuration process includes the following:

Configuring SNMP agent software

SNMP agent software provides monitoring and administration of Intel Dialogic boards using the Simple Network Management Protocol (SNMP). The SNMP Configuration Tool (*dlgcsnmpconf*) is invoked as part of the configuration script *config.sh*. This tool provides both automatic and manual methods of configuration SNMP communities and SNMP v1 trap destinations. In addition, the SNMP Agent MIB files must be installed on the network management station after the main configuration script (*config.sh*) has completed.

Assigning time slots when using a third-party board as the clock master

Third-party boards and Intel Dialogic boards must not transmit data on the same telephony bus time slots. Also, the third-party technology (clock master) must execute before Intel Dialogic boards (slaves) in the startup sequence. To accommodate these requirements, adjustments must be made to the *.sctsbase* file and the *dlgsys.cfg* file.

Configuring the Digital Network Interface Parameters

The digital network interface (DNI) parameter file, *Spandti.prm*, is an ASCII text file used by the Intel Dialogic System Software to initialize the firmware configuration for the front end of digital network interface Springware boards. If the default settings in the *Spandti.prm* file aren't appropriate for your application, you can modify them.

Configuring Voice Parameters

This step involves adjusting parameters in the *Voice.prm* file for frequency shift keying (FSK), signal delay adjustments, and silence compressed record. The *Voice.prm* file is downloaded to all Springware voice boards during the installation and configuration process.

Configuring Non-Facility Associated Signaling (NFAS)

For T1 applications that require NFAS, you must edit a number of files to identify the trunk whose D channel will provide the signaling, as well as the trunk(s) that will share the NFAS D channel.

Verifying Device Names

This optional step consists of examining the *Voxcfg* file against the device name assignment rules.

Configuring CDP files (appropriate when using Global Call protocols)

When using Global Call protocols, the protocols and country dependent parameters (.cdp file) must be configured.

Initializing the System

During system initialization, all required firmware for an Intel Dialogic board is downloaded and configured using the identified files and parameter settings.

Updating the PCI ID for D/42JCT-U and D/82JCT-U Boards

After initial configuration and download, the Dialogic.cfg file shows PCI ID 0 for any D/42JCT-U and D/82JCT-U boards that are installed. This needs to be updated with the correct PCI IDs for these boards.

Reconfiguring the System

If hardware is added or configuration parameters need to be modified, the system must be reconfigured. Parameter changes can be made by editing the appropriate file or re-running the appropriate utility. The system is then re-initialized by using the *dlstop* and *dlstart* utilities.

Configuration Details 2

This chapter provides details about the following configuration information:

2.1 The Dialogic.cfg File

The *dialogic.cfg* file, located in */usr/dialogic/cfg*, is an ASCII file that contains board information required by the Dialogic board drivers and generic board downloader (Genload). The *dialogic.cfg* file is created when you run the `mkcfg` utility. You can update *dialogic.cfg* by rerunning the `mkcfg` utility or by editing the file manually if you prefer.

Although it is possible to use the *dialogic.cfg* file as created by `mkcfg` without modification, some configurations require manual editing of *dialogic.cfg* because the default parameter values are not appropriate. You also have to edit *dialogic.cfg* if you add or remove boards without performing another software installation.

When editing d*ialogic.cfg*, use the following conventions:

- The *dialogic.cfg* file contains a global parameter section and a board parameter section for each SpringWare board.
  - The global parameter section begins with:

    ```
    [Genload - All Boards]
    ```

    This section head indicates that the parameters below it apply to all the boards in the system and/or to the bus.
  - Each board parameter section begins with one of the following section heads:

    ```
    [Genload - ID <board-ID>]
    [Genload - PCI ID <board-ID>]
    [Genload - Address <board-shared-RAM-base-address>]
    ```

    A board section head indicates that the parameters below it apply to the board specified in the section head. Each board is identified by <*board-ID*>, that is, the identification number assigned to a board in the range 00H to 1FH.
- Within each section, the parameters can be in any order; Genload does not require any particular sequence of parameters. However, the **LogFile** parameter, if used, should be the first line in the [Genload - All Boards] section to ensure that all download information is captured in the log file.

- Some parameters can be used as either a global parameter or a board parameter. When a parameter is used as both a global and board parameter, the board parameter value overrides the global parameter value for the specified board. Many parameters apply only to certain boards or types of boards.
- Comments can be added to *dialogic.cfg*. If you use the pound sign (#) or semicolon (;) anywhere on a line, all text to the right of the character until the end of the line is treated as a comment (ignored). C code style comments are also allowed. If you use /* anywhere on a line, all text that follows is treated as a comment (ignored) until the */ character sequence is encountered.

2.2 SNMP Agent Configuration

Then SNMP agent software provides monitoring and administration of Intel Dialogic boards using the Simple Network Management Protocol (SNMP). Before using the SNMP agent, Net-SNMP must be installed. After the Intel software (including the SNMP agent software) has been installed, Net-SNMP must be configured to use the SNMP agent extension software.

When the `install.sh` installation procedure is completed, you can configure the SNMP agent software. You start the procedure by executing `config.sh` as explained in Section 3.3, "Starting the Config.sh Utility", on page 22. SpringWare and DM3 board configuration come first, and then a utility called `dlgcsnmpconf` is automatically invoked to start the SNMP Configuration Tool.

This configuration tool provides two methods of configuration: automatic and manual. Both methods configure SNMP communities and SNMP v1 trap destinations. The *admin* community is a mandatory configuration requirement that is needed by the SNMP agent extension to properly operate with the Net-SNMP master agent.

The automatic configuration method creates the *admin* community, giving it both read and write access on the local host. If the manual configuration method is used, then the user is responsible for creating the *admin* community with read and write privileges.

The automatic method configures the Net-SNMP agent by creating the read-write *admin* community and the *dialogic* community, which is set to read-only for all external managers.

The manual method allows the user to enter communities; it does not create any communities automatically. If the *admin* community already exists in the Net-SNMP configuration, the configuration tool indicates that the community is detected and does not require configuration.

Both the automatic and manual configuration methods provide an opportunity to configure trap destinations. Trap destinations are machines that are configured to receive SNMP v1 traps from managed nodes. Trap destinations are also called **trap sinks**. The configuration tool allows as many trap sinks as required by the user. If a trap sink is not reachable by the managed node, the configuration tool displays a warning message and allows the user to back out of the configuration.

Once the configuration tool has completed, it writes the configuration changes to */usr/share/snmp/snmpd.conf*. A backup of the original configuration file is created as */usr/share/snmp/snmpd.conf.backup*.

The SNMP Configuration Tool is normally used after installing the Intel software, as part of the `config.sh` configuration procedure. When your configuration is done and you reboot to start the System Service for the first time, the SNMP agent extension software is started.

If you need to use the configuration tool again at a later time (for example, to add an additional management station to receive trap notifications), you can invoke the tool by entering:

```
/usr/dialogic/lib/snmp/dlgcsnmpconf
```

After using the `dlgcsnmpconf` tool to modify the configuration, enter the following command to restart the SNMP agent extension software in order to apply the changes made:

```
/etc/init.d/dlgcsnmpd restart
```

To begin the SNMP agent configuration process, follow the instructions in Section 2.2, "SNMP Agent Configuration", on page 16.

2.3 Silence Compressed Record Feature

The silence compressed record (SCR) feature allows recording with silent pauses eliminated. This results in smaller size recorded files with no loss of intelligibility. The SCR feature is enabled in the *voice.prm* file, which is downloaded during initialization. You must edit this file and set appropriate values for the SCR parameters for your working environment before initializing the board(s). You cannot enable this feature through the Dialogic voice API.

The *voice.prm* file is in */usr/dialogic/data*. The section of the file dealing with SCR parameters is:

```
# ======================================================================
# SILENCE COMPRESSED RECORD Parameters
# To turn on SCR uncomment all of the lines in the block below.
# Recommended values are given.
# ======================================================================

# --- For Silence Compressed Record, uncomment the block below ---

#PARAM 134 : 100  # SCR_T = 1 second SCR trailing silence
#PARAM 135 : 100  # SCR_PC = 100 bytes of pre-compensation
#PARAM 136 : 43   # SCR_THRES = -43dB silence threshold
#PARAM 137 : 4    # SCR_DG = 40ms of non-silence deglitch
#PARAM 138 : 1    # SCR_ON = SCR is on
# ---- End of SCR block ----
```

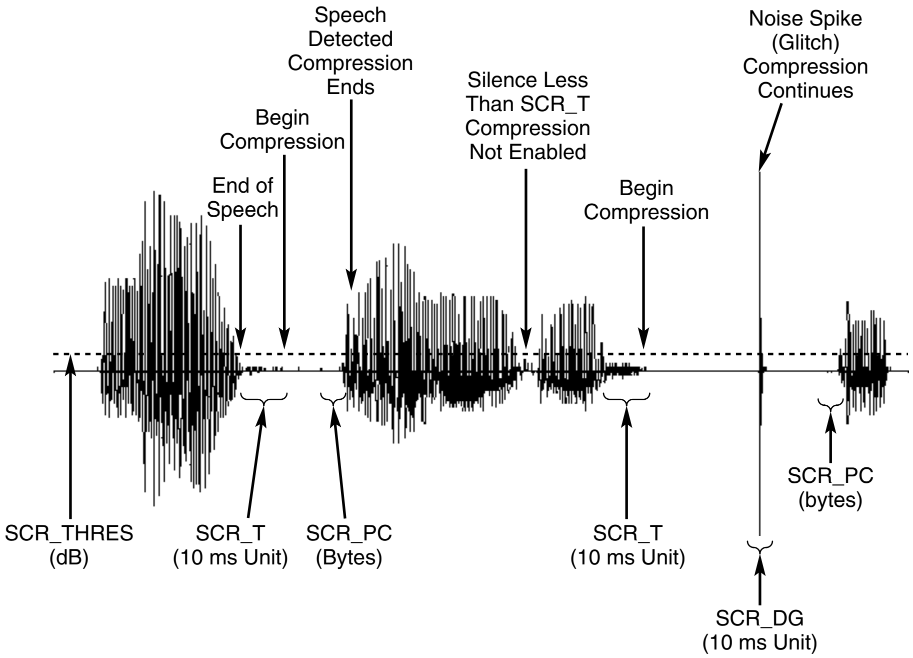
As distributed, the SCR parameters in the *voice.prm* file appear as comments (each line is preceded with #). To enable SCR, remove the # from the beginning of each line containing an SCR parameter and adjust the parameters if needed. The meaning of each SCR parameter is illustrated in Figure 1, "SCR Parameters Illustrated", on page 18 and described in Chapter 6, "Silence Compressed Record Parameter Reference".

After SCR is enabled in the *voice.prm* file, SCR is automatically activated through use of voice record functions such as **dx_rec( )**. When the audio level is at or falls below the silence threshold for a minimum duration of time, silence compressed record begins. When a short burst of noise (glitch) is detected, the compression does not end unless the glitch is longer than a specified period of time.

The *voice.prm* file is downloaded by default to all SpringWare voice boards. As such, SCR is available to all voice channels in the system. To enable SCR on only one board in a multi-board system configuration, see the procedure in Section 3.9.5, “Enabling Silence Compressed Record on Only One Board”, on page 35

Figure 1 illustrates the use of the SCR parameters.

Figure 1. SCR Parameters Illustrated



2.3.0.1 Encoding Algorithms

The following encoding algorithms and sampling rates are supported in silence compressed record:

- 6 kHz and 8 kHz OKI ADPCM
- 8 kHz and 11 kHz linear PCM
- 8 kHz and 11 kHz A-law PCM
- 8 kHz and 11 kHz Mu-law

2.4 Boards Supported

Table 1 lists the Intel Dialogic Boards supported by this Configuration Guide.

Table 1. Intel Dialogic Springware PCI Boards

Analog	D/4PCIU4S	4-port voice board with basic voice processing, CSP, and analog loop start
	D/4PCIUF	4-port voice board with basic voice processing, Fax, and analog loop start
	D/41JCT-LS	4-port voice board with call processing and analog loop start
	D/120JCT-LS	12 channels of analog line interface and voice processing
Single Span	D/240JCT-T1	24 channels of voice with play/record, tone, call progress analysis, and single T1 network interface
	D/300JCT-E1	30 channels of voice with play/record, tone, call progress analysis, and single E1 network interface
Dual Span	D/480JCT-1T1‡	24 channels of voice with CSP, play/record, tone, call progress analysis, softfax, and T1 network interface
	D/600JCT-1E1‡	30 channels of voice with CSP, play/record, tone, call progress analysis, softfax, and 75 or 120-Ohm E1 network interface
	D/480JCT-2T1	48 channels of voice processing with CSP and dual T1 network interface
	D/600JCT-2E1	60 channels of voice processing with CSP and dual E1 network interface
Voice/PBX Integration	D/42JCT-U	4-port voice-processing that emulates a number of phones and interfaces with supported PBXs
	D/82JCT-U	8-port voice-processing that emulates a number of phones and interfaces with supported PBXs
Fax	VFX/41JCT-LS	4 channels of analog voice and fax
‡ = These boards are displayed as D/480JCT-2T1 and D/600JCT-2E1 by the system.		

Configuration Procedures 3

This chapter provides detailed procedures for each major step in the configuration process for Intel Dialogic Springware PCI boards. Note that some of these configuration procedures may not apply to your specific system configuration. The following topics are discussed:

3.1 Assumptions and Prerequisites

The following assumptions and prerequisites exist regarding the configuration procedures:

- All required Intel® System Release software, including prerequisites, have been installed according to the procedures in the software installation guide supplied with the system release.
- The Intel® System Release was installed in the default directory */usr/dialogic*. Command instructions, directories paths and environment variable are shown relative to the default subdirectory.
- If SNMP agent software is installed, it is assumed all prerequisites have been met as outlined in the software installation guide and SNMP agent software administration guide supplied with your system release.
- If a third-party board is being used as the primary clock master, it is assumed that third-party technology can use a range of time slots starting at 0.
- The following GNU General Public License (GPL) tools have been installed: ld960, nm960, ldnew, ldarm, and nmarm. These files are required by PDKManager and the source files can be

downloaded from the GNU ftp site *ftp://ftp.gnu.org/gnu/binutils/binutils-2.13.tar.gz*. For more information about the GNU Binutils, see *http://www.gnu.org/directory/binutils.html*.

3.2 Order of Procedures

Procedures that are required when initially configuring any system are noted as such. The additional procedures may be required depending on your system. The configuration procedures should be performed in the order presented.

1. Starting the Board Configuration Utility (Config.sh) (**required**)
2. Using the Mkcfg Utility (**required**)
3. Using the SNMP Agents Configuration Tool
4. Assigning Time Slots When Using a Third-Party Board as Clock Master
5. Changing Digital Network Interface parameters
6. Configuring Voice Parameters
7. Using Non-Facility Associated Signaling (NFAS)
8. Verifying Device Names
9. Configuring Global Call CDP Files
10. Initializing the system (**required**)
11. Updating the PCI ID for D/42JCT-U and D/82JCT-U Boards
12. Reconfiguring the system

3.3 Starting the Config.sh Utility

If you want to keep a record of all configuration prompts and responses, use the Linux *script* utility prior to starting the board configuration procedure. You can then see the *script* output file for information such as configuration parameter selections. For information about using the *script* utility, see the Red Hat Linux documentation.

The following procedure explains the initial steps of board configuration. After these initial steps, further instructions depend on the system you are configuring.

1. Enter the following command to start the board configuration script:

   ```
   ./config.sh
   ```

   The system displays the following messages:

   ```
                         Intel(R)Dialogic(R)System Release 6.1 PCI for Linux
                                   CONFIGURATION

   This is the configuration tool for Intel(R) Dialogic(R) System Release 6.1 PCI
   for Linux.

   You will be asked to supply information for configuring the software.

   Would you like to configure SNMP on this system (y/n, default=n) ?
   ```

2. If you installed the SNMP agent software, type `y`; otherwise type `n`.

Note: Do not enter y to configure SNMP if you have not installed the SNMP agent software. If you do this, the configuration procedure is aborted and you will be prompted to run the installation script (install.sh) again so you can install the SNMP agent software.

If you enter y to configure SNMP, the SNMP Agents Configuration Tool is automatically invoked when board configuration is complete.

```
Copying driver files......

Drivers will now be loaded...

One or more Springware-family boards were detected.
Starting board-specific configuration...

Press ENTER to continue...
```

3. Press Enter.

   As indicated by the preceding messages, it may take several minutes before the drivers are loaded, and many lines of `makefile` output will be displayed.

At this time, configuration continues as follows:

For information about configuring Springware boards, see Section 3.4, “Using the Mkcfg Utility”, on page 23. Note that if you have both Springware boards and DM3 boards installed, you are prompted to configure the Springware boards first.

- If you have DM3 boards installed, the following message is displayed:

  ```
  Starting DM3-specific configuration...
  ```

  For information about configuring DM3 boards, refer to the *Intel® DM3 Architecture PCI Products on Linux Configuration Guide*.

- After Springware and DM3 board configuration, if you installed SNMP agent software, the following message is displayed:

  ```
  SNMP configuration...
  ```

  For information about configuring the SNMP agent software, refer to Section 3.5, “Using the SNMP Agents Configuration Tool (dlgcsnmpconf)”, on page 26.

3.4 Using the Mkcfg Utility

When you start the `config.sh` procedure, you are prompted to configure Springware boards first, before any DM3 boards that are installed. The Springware part of the configuration begins with the following message:

```
One or more Springware-family boards were detected.
Starting Springware specific configuration...

Press ENTER to continue...
```

1. Press Enter. The following messages are displayed:

   ```
   Intel(r) Dialogic(r) Configuration File Generator
   Version 5.00
   Copyright 1997-2003 Intel Corporation

   [Type ‘Q’ at any prompt to exit]
   [Type ‘?’ at any prompt for help]

   Press <return> to begin...
   ```

Note: You should respond to the entire series of prompts from the `mkcfg` utility even after you enter the information for your boards. Do not type Q to exit. Typing Q will abort the `mkcfg` utility without saving your configuration in the *dialogic.cfg* file. If this happens, you will have to run `mkcfg` again from the beginning.

2. Press Enter. The `mkcfg` utility starts prompting for the type(s) of Springware boards you have installed. For example:

```
CONFIG INFORMATION FOR D/XXJCT-U BOARDS
_______________________________________

(Includes D/42JCT-U, D/82JCT-U)

Hit RETURN to accept default values, which are listed in () at each prompt...

Enter the number of D/XXJCT-U boards in the system (0):
```

3. If you **don't** have this type of board installed, press Enter to accept the default value of 0. The `mkcfg` utility then prompts for another board type. Keep pressing Enter until `mkcfg` prompts for a board type that you **do** have installed.

   When you come to a prompt for a board type that you **do** have installed, respond with the number of boards that you have of that board type. Depending on the board type, the `mkcfg` utility then prompts you for:

   - Board ID—PCI boards can have any value from 0 to 15. If you set board specific parameters for PCI boards in the *dialogic.cfg* file, use unique board IDs for all PCI boards.
   - Encoding (ALAW or ULAW)—Normally, ALAW is used for E-1 and ULAW for T-1. See the PCMEncoding parameter in Chapter 4, "Dialogic.Cfg Parameter Reference" for further information.
   - PBX switch type—For PBX integration boards, select the PBX make and model that the board will interface to. See the PBXswitch parameter in Chapter 4, "Dialogic.Cfg Parameter Reference"for further information.

4. Respond to each prompt as requested. For help at any prompt, type ?.

   After you have completed the configuration prompts for all board types, a summary of your board information is displayed. For example:

```
                           +=====================+
+--------------------------|  Board Information  |--------------------------+
|                          +=====================+                          |
|                                                                           |
| D/4xxJCT-LS              ID  0:  Modtype: H100                            |
| T1/E1 PCI HD             ID  1:  Modtype: H100                            |
| T1/E1 PCI HD             ID  2:  Modtype: H100                            |
| ------------------------------------------------------------------------- |
|                                                                           |
|   NOTE:                                                                   |
| If the above information for Network board(s) is correct, you should     |
| record the information and modify the appropriate .prm file(s).          |

  Is this information correct (y/n)?
```

5. If the information is correct, enter y; otherwise enter n.

   If you enter n, the `mkcfg` utility takes you through the configuration prompts again. (You must reenter everything; none of your previous responses are saved.)

   Once the information is correct and you enter y, the configuration information is saved in the *dialogic.cfg* file (in */usr/dialogic/cfg*). If a previous version of *dialogic.cfg* exists (for example, if you entered updated configuration information), the previous version is saved as *dialogic.01* (or *dialogic.02,* etc.).

The `mkcfg` utility ends with the message:

```
Creating backup /usr/dialogic/cfg/dialogic.01
```

At this time, configuration continues as follows:

- If you have DM3 boards installed, the following message is displayed:

  ```
  Starting DM3-specific configuration...
  ```

  For information about configuring DM3 boards, see the *Intel® DM3 Architecture PCI Products on Linux Configuration Guide*.

- If you have the SNMP agent software installed, the following message is displayed:

  ```
      SNMP configuration...
  ```

  For information about configuring the SNMP agent software, see Section 3.5, "Using the SNMP Agents Configuration Tool (dlgcsnmpconf)", on page 26.

- When the configuration procedure is complete, the following messages are displayed:

  ```
  Configuration is complete.
  You must reboot the system to start the software for the first time.
  Thereafter, you may use the dlstop and dlstart scripts found in
  /usr/dialogic/bin
  ```

6. To verify the configuration, check the *dialogic.cfg* file that was generated (in */usr/dialogic/cfg*).

   The file contains a global parameter section and sections for each board. For example:

   ```
   [Genload - All Boards]
   Dialog/HD=YES
   BusType=SCBus
   SCBusClockMaster=AUTOMATIC
   SCBusClockMasterSource=AUTOMATIC
   PCMEncoding=AUTOMATIC

   [Genload - PCI ID 0]
   ParameterFile=spandti.prm
   .
   . (other board parameters go here)
   .
   [Genload - PCI ID 1]
   .
   .
   .
   ```

   Refer to Chapter 4, "Dialogic.Cfg Parameter Reference" for information about the *dialogic.cfg* file configuration parameters.

7. If necessary, edit the *dialogic.cfg* file. For example, you have to edit *dialogic.cfg* for the following features and configurations:

   - If you want to capture download information in a log file, add **LogFile = genload.log** as the first line in the global parameter section (after [Genload - All Boards]).
   - When H.100 Springware boards are installed, set **BusType = H100** and specify the clock master using the **PrimaryMaster** and **PrimaryMasterClockSource** parameters instead of the **SCbusClockMaster** and **SCbusClockMasterSource** parameters.
   - If you have DM3 boards installed and a DM3 board is the clock master, set the **SCBusClockMaster** and **SCBusClockMasterSource** parameters to **NONE**.
   - For country specific parameters, use the **Country** parameter to set the country code.

- For ISDN protocols, use the **ISDNProtocol** parameter to select the protocol.
- When Global Call Protocols and their corresponding Country Dependent Parameters (CDP) are installed, use the **ParameterFile** parameter to specify the correct parameter file for each board. See the *Global Call Country Dependent Parameters (CDP) Configuration Guide* for further information.
- To use Continuous Speech Processing (CSP) on Springware board(s), set the **CSPExtraTimeSlot**, **FirmwareFile**, and **FirmwareFile2** parameters to enable CSP.

8. Check to see if any other procedures are applicable to your configuration. (For example, additional configuration procedures are needed for using the SNMP agent software.) Perform the applicable procedures before you start the software.

   Note: If you have D/42JCT-U or D/82JCT-U boards installed, an additional procedure is needed **after** you start the software. See Section 3.14, "Updating the PCI ID for D/42JCT-U and D/82JCT-U Boards", on page 43.

Continue with any additional configuration procedures that are applicable to your system:

- If SNMP agent software is installed, once the Springware Configuration utility (*Mkcfg*) is complete, the SNMP Agents Configuration Tool (*dlgcsnmpconf*) is automatically invoked. Refer to Section 3.5, "Using the SNMP Agents Configuration Tool (dlgcsnmpconf)", on page 26.
- If you are using a third-party board as the clock master, see Section 3.6, "Assigning Time Slots When Using a Third-Party Board as Clock Master", on page 30.
- If you need to change any Digital Network Interface parameters, see Section 3.8, "Changing Digital Network Interface Parameters", on page 32.
- If you wish to modify any Voice parameters, refer to Section 3.9, "Configuring Voice Parameters", on page 33.
- If you will be using non-facility associated signaling, see Section 3.10, "Using Non-Facility Associated Signaling (NFAS)", on page 36.
- To verify Device Names, see Section 3.11, "Verifying Device Names", on page 37.
- If you have the Global Call Protocol Package installed, see Section 3.12, "Configuring Global Call CDP File", on page 41

When you are satisfied with all configuration information (including DM3 configuration if applicable), you must reboot to start the software for the first time. Rebooting the system initializes all the Intel Dialogic products in the system. For information about system startup, see Section 3.13, "Initializing the System", on page 41.

3.5 Using the SNMP Agents Configuration Tool (dlgcsnmpconf)

If you installed SNMP agent software, once the Springware Board Configuration utility (*Mkcfg*) is complete, the SNMP Agents Configuration Tool (*dlgcsnmpconf*) is automatically invoked. The following message is displayed:

```
SNMP configuration...
```

The procedure for configuring the SNMP agent software includes the following:

- Configuring the Community String
- Configuring Trap Destinations (Sinks)

3.5.1 Configuring the Community String

SNMP v1 uses community strings to provide simple access control for management information base (MIB) objects. If a management software tool uses an Intel® Dialogic SNMP MIB, it must use the identical community strings that the SNMP agent software is configured to use. Communities can be created in two ways:

- Configuring SNMP Communities Automatically
- Configuring SNMP Communities Manually

Both the automatic and manual configuration methods provide an opportunity to configure trap destinations.

Note: If the automatic configuration process is used, the *dialogic* community is created. This community grants external management stations **read-only** access to the Intel® Dialogic MIB. However, if the external management station requires **write** access to writable SNMP objects in the Intel® Dialogic MIB, then use the instructions in Section 3.5.1.2, "Configuring SNMP Communities Manually", on page 28 to create a community string that grants external managers read-write access.

3.5.1.1 Configuring SNMP Communities Automatically

The automatic configuration method creates the *admin* community, giving it both read and write access on the local host. The automatic method configures the Net-SNMP agent by creating the read-write *admin* community and the *dialogic* community, which is set to read-only for all external managers.

The SNMP agent software part of the configuration begins with the following messages:

```
SNMP configuration...
Dialogic SNMP Agents Configuration Tool
(C)2000-2001 Intel Corp.

You may choose to manually configure all communities and trap
sinks(destinations), or you may select an automatic configuration.  If the
automatic configuration is chosen, this tool will create the required
'admin' community and prompt you to enter trap sinks.  Selecting the manual
configuration allows you to easily create custom communities and configure
trap sinks.  If the 'admin' community does not exist yet, it may be created
the same way as other communities using the manual configuration.  Note, the
'admin' community MUST be assigned read-write priviledges or else abnormal
behavior will occur when the Intel Dialogic SNMP Agents are loaded.

Would you like to proceed with automatic configuration? (no will select
manual configuration) (y)es or (n)o?
```

Proceed as follows:

1. Type y for automatic configuration.

2. You are asked for confirmation; type the letter `y` again.

   The SNMP Agents Configuration Tool creates and configures the *admin* and *dialogic* communities.

   Configuration continues with the following prompt:

   ```
   Configure trap sink(destination) (y)es or (n)o?
   ```

Continue with the instructions in Section 3.5.2, “Configuring Trap Destinations (Sinks)”, on page 29.

3.5.1.2 Configuring SNMP Communities Manually

The manual method allows the user to enter communities; it does not create any communities automatically. If the manual configuration method is used, then the user is responsible for creating the *admin* community with read and write privileges. If the *admin* community already exists in the Net-SNMP configuration, the SNMP Agents Configuration Tool indicates that the community is detected and does not require configuration.

The SNMP agent software part of the configuration begins with the following messages:

```
SNMP configuration...
Dialogic SNMP Agents Configuration Tool
(C)2000-2001 Intel Corp.

You may choose to manually configure all communities and trap
sinks(destinations), or you may select an automatic configuration.  If the
automatic configuration is chosen, this tool will create the required
'admin' community and prompt you to enter trap sinks.  Selecting the manual
configuration allows you to easily create custom communities and configure
trap sinks.  If the 'admin' community does not exist yet, it may be created
the same way as other communities using the manual configuration.  Note, the
'admin' community MUST be assigned read-write priviledges or else abnormal
behavior will occur when the Intel Dialogic SNMP Agents are loaded.

Would you like to proceed with automatic configuration? (no will select
manual configuration) (y)es or (n)o?
```

Proceed as follows:

1. Type the letter `n` for manual configuration.

   The following prompt is displayed:

   ```
   Configure communities (access control)? (y)es or (n)o?
   ```

2. Type the letter `y` to create and configure communities. (Typing the letter `n` skips community configuration and proceeds to the trap destination configuration prompt shown in Step 7. of this procedure.)

   If you type the letter `y`, you are prompted for the community name:

   ```
   Enter community name (leave blank to cancel):
   ```

3. If the *admin* community has not been created yet, either manually or by the automatic configuration method, then type `admin` as the community name.

   You are prompted to enter the access privileges for the community:

   ```
   Make this community read-write? (y)es or (n)o?
   ```

4. Type the letter y if the community will allow write requests, or n if the community will allow only read requests.

   Note: For the *admin* community, access **must** be read-write.

   The next prompt asks if external managers will be allowed to use this community to access the Intel® Dialogic MIB:

   ```
   Allow external managers access with this community? (y)es or (n)o?
   ```

5. Type the letter y to grant access to the Intel® Dialogic MIB using this community, or the letter n to grant only the local host access to the MIB using this community.

   Note: For the *admin* community, local access only is recommended.

   The following prompt asks you to confirm the community configuration:

   ```
   Prepared to add "rwcommunity yourcommunityname localhost" to config file.
   Proceed (y)es or (n)o?
   ```

   where *yourcommmunityname* is the community name that you entered.

6. Type the letter y to write the community to the configuration file (*/usr/share/snmp/snmpd.conf*).

   The SNMP Agents Configuration Tool then allows you to configure additional communities:

   ```
   Add another community? (y)es or (n)o?
   ```

7. Type the letter y to add another community or n to continue with trap destination configuration.

   If you type the letter y, the prompt shown in Step 2. is repeated, allowing you to configure another community.

   If you type the letter n, the following prompt is displayed:

   ```
   Configure trap sink(destination) (y)es or (n)o?
   ```

Continue with the instructions in Section 3.5.2, “Configuring Trap Destinations (Sinks)”, on page 29.

3.5.2 Configuring Trap Destinations (Sinks)

Trap destinations are machines that are configured to receive SNMP v1 traps from managed nodes. Trap destinations are also called **trap sinks**. The SNMP Agents Configuration Tool allows as many trap sinks as required by the user. If a trap sink is not reachable by the managed node, the configuration tool displays a warning message and allows the user to back out of the configuration.

After starting the SNMP Agents Configuration Tool and using either the automatic or manual method to configure communities, configuration continues with the following prompt:

```
Configure trap sink(destination) (y)es or (n)o?
```

Proceed as follows:

1. Type the letter y to configure a trap destination or the letter n to exit the configuration tool.

   If you type the letter y, the following prompt is displayed:

```
Type host name to be trap sink:
```

2. Type the name of the management station that is configured to receive traps. The following prompt is displayed:

   ```
   Allow agent to send SNMPv1 traps to 'hostname' (y)es or (n)o?
   ```

   where *hostname* is the name of the management station that you entered.

3. Type the letter `y` to add the specified host as a trap destination.

   The prompts are repeated, allowing you to configure additional trap destination(s). When done, type n to exit the configuration tool and write the configuration changes to */usr/share/snmp/snmpd.conf.* A backup of the original configuration file is created as */usr/share/snmp/snmpd.conf.backup*.

Continue with any additional configuration procedures that are applicable to your system:

- If you are using a third-party board as the clock master, see Section 3.6, “Assigning Time Slots When Using a Third-Party Board as Clock Master”, on page 30.
- If you need to change any Digital Network Interface parameters, see Section 3.8, “Changing Digital Network Interface Parameters”, on page 32.
- If you wish to modify any Voice parameters, refer to Section 3.9, “Configuring Voice Parameters”, on page 33.
- If you will be using non-facility associated signaling, see Section 3.10, “Using Non-Facility Associated Signaling (NFAS)”, on page 36.
- To verify Device Names, see Section 3.11, “Verifying Device Names”, on page 37.
- If you have the Global Call Protocol Package installed, see Section 3.12, “Configuring Global Call CDP File”, on page 41

When you are satisfied with all configuration information (including DM3 configuration if applicable), you must reboot to start the software for the first time. Rebooting the system initializes all the Intel Dialogic products in the system. For information about system startup, see Section 3.13, “Initializing the System”, on page 41.

3.6 Assigning Time Slots When Using a Third-Party Board as Clock Master

Third-party boards and Intel Dialogic boards must not transmit data on the same telephony bus time slots. Also, the third-party technology (clock master) must execute before Intel Dialogic boards (slaves) in the startup sequence. Transmit time slots for Intel Dialogic boards are assigned during initialization as specified in the */usr/dialogic/cfg/.sctsbase* file. Use the following procedure to modify the time slot for Intel Dialogic boards to be a value greater than 0 (a value greater than the number of time slots required for the third-party board); then, the third-party board can use time slots in the beginning of the time slot range.

Also, you will need to set the clocking daemon to PASSIVE, thereby setting all Intel Dialogic boards to slaves and setting the clock daemon to not perform clock fallback. For information about setting the clocking daemon, see Section 3.7, “Setting the ClockDaemonMode”, on page 31.

Notes: ***1.*** It is assumed that the third-party technology can use a range of time slots starting at 0.

2. The third-party board must be configured as both the primary clock master and the reference master on the TDM bus.

Proceed as follows:

1. From the command prompt, go to the */usr/dialogic/cfg* directory and locate the *.sctsbase* file.

2. Using a text editor (for example, *vi*), open the *.sctsbase* file.

3. Add a line to */usr/dialogic/cfg/.sctsbase* that will cause the starting time slot for Intel Dialogic boards to be a value greater than 0. For example, if the third-party board uses time slots 0 through 1023 (1024 time slots), write the value “1024” to */usr/dialogic/cfg/.sctsbase* by adding this number as the first line in the file:

   ```
   1024
   ```

4. Save and close the */usr/dialogic/cfg/.sctsbase* file. Intel Dialogic boards will now use time slots above those specified in the *.sctsbase* file.

Continue with Section 3.7, “Setting the ClockDaemonMode”, on page 31

3.7 Setting the ClockDaemonMode

The ClockDaemonMode specifies whether or not the clock daemon supplied with the system software performs clock fallback. The ClockDaemonMode is modified by editing the dlgsys.cfg file.

1. From the command prompt, go to the `/usr/dialogic/cfg` directory and locate the *dlgsys.cfg* file.
2. Using a text editor (for example, vi), open the *dlgsys.cfg* file. Initially, the last line of the file reads:

   ```
   ClockDaemonMode          :ACTIVE
   ```

The ClockDaemonMode values supported are:

- ACTIVE [default]: Clocking daemon performs clock fallback. An Intel Dialogic board is selected as the clock master.
- PASSIVE: Clocking daemon does not perform clock fallback and sets all Intel NetStructure boards to slaves. Select PASSIVE when using a third-party board as a clock master.
- DISABLED: Clocking daemon does not control the TDM bus. Select DISABLED when using another application to control the TDM bus.

3. If you wish to change the ClockDaemonMode to a value other than the default value of ACTIVE, edit this line in the *dlgsys.cfg* file. For example, to change the mode to PASSIVE, you would edit the line to read:

   ```
   ClockDaemonMode          :PASSIVE
   ```

4. Save and close the *dlgsys.cfg* file.

   For further information about the clock daemon and clock fallback, see the *OA&M API for Linux Library Reference* and the *OA&M API for Linux Programming Guide*.

- If you need to change any Digital Network Interface parameters, see Section 3.8, “Changing Digital Network Interface Parameters”, on page 32.
- If you wish to modify any Voice parameters, refer to Section 3.9, “Configuring Voice Parameters”, on page 33.
- If you will be using non-facility associated signaling, see Section 3.10, “Using Non-Facility Associated Signaling (NFAS)”, on page 36.
- To verify Device Names, see Section 3.11, “Verifying Device Names”, on page 37.
- If you have the Global Call Protocol Package installed, see Section 3.12, “Configuring Global Call CDP File”, on page 41

When you are satisfied with all configuration information (including DM3 configuration if applicable), you must reboot to start the software for the first time. Rebooting the system initializes all the Intel Dialogic products in the system. For information about system startup, see Section 3.13, “Initializing the System”, on page 41.

3.8 Changing Digital Network Interface Parameters

The digital network interface parameter file, *spandti.prm*, is an unformatted ASCII file that the firmware downloader uses to initialize the basic firmware configuration for the digital network interface on Springware boards. This file contains a description of all possible values with comments, as well as examples of the parameters set to the default values.

To change a parameter from the default setting:

1. Preserve the original parameter file (*/usr/dialogic/data/spandti.prm*) by copying it to another file name such as *spandti.old* and do not modify the copy.

2. Modify *spandti.prm* to contain a list of the parameters and values that you want to change. It is not necessary to specify parameters that use the default values.

   To include comments in the file, place a semicolon (;) in the first column of a line used for comments. You should keep a record of the parameter settings that you change by using comments in the parameter file.

   Note: Do not change any settings unless you are sure of what you are doing. Settings must match those of your provider. If you are uncertain of the correct settings, ask your provider.

3. Set the **ParameterFile** parameter in the *dialogic.cfg* file to **ParameterFile = spandti.prm** for the boards that use the modified parameter settings. If a **ParameterFile** value is not specified, the default values are used.

Continue with any additional configuration procedures that are applicable to your system:

- If you wish to modify any Voice parameters, refer to Section 3.9, “Configuring Voice Parameters”, on page 33.
- If you will be using non-facility associated signaling, see Section 3.10, “Using Non-Facility Associated Signaling (NFAS)”, on page 36.
- To verify Device Names, see Section 3.11, “Verifying Device Names”, on page 37.
- If you have the Global Call Protocol Package installed, see Section 3.12, “Configuring Global Call CDP File”, on page 41

When you are satisfied with all configuration information (including DM3 configuration if applicable), you must reboot to start the software for the first time. Rebooting the system initializes all the Intel Dialogic products in the system. For information about system startup, see Section 3.13, “Initializing the System”, on page 41.

3.9 Configuring Voice Parameters

The procedures associated with configuring parameters in the *Voice.prm* file include:

- Adjusting FSK Receiver Carrier Detect Threshold
- Adjusting Two-Way FSK Transmit Framing Parameters
- Setting the Firmware Buffer Size
- Enabling and Modifying Silence Compressed Record Parameters
- Enabling Silence Compressed Record on Only One Board

3.9.1 Adjusting FSK Receiver Carrier Detect Threshold

Host applications can change the receiver carrier detect threshold from the default value (-44 dBm) to any value in the range of -22 dBm to -44 dBm. Values are adjusted in 2 dB intervals with a hysteresis of ±2 dB at each step. Odd numbers are valid values, but they will be rounded up to the next even number; for example, -27 becomes -26.

To edit the *Voice.prm* file to set the receiver carrier detect threshold, proceed as follows:

1. Open the file using any text editor.

2. Add parameter 255 and the appropriate value (26 in this example) as shown in bold text in the following example:

```
#beginning of voice.prm

AREA=VOICE
SIZE=WORD
BASE=DECIMAL
. . .
PARAM 255 : 26  # set receiver carrier detect threshold

#end of voice.prm
```

3. Save the file.

For the added parameter to take effect, you must specify the *Voice.rpm* file as the value for the **ParameterFile** parameter.

3.9.2 Adjusting Two-Way FSK Transmit Framing Parameters

The two-way frequency shift keying (FSK) transmit framing parameters and their respective default values are:

- TX channel seizure bits = 360
- TX onhook mark bits = 180
- TX offhook mark bits = 84
- TX endmark bits = 84
- TX to RX delay time in 10 millisecond units = 5

You may adjust any of these values to be different from their default setting by editing the *Voice.prm* file. To do this, perform the following:

1. Open the file using any text editor.
2. Add the following lines shown in bold to the *Voice.prm* file (example values are provided):

```
#beginning of voice.prm

AREA=VOICE
SIZE=WORD
BASE=DECIMAL
. . .

PARAM 257 : 80  # set number of 2-way FSK TX channel seizure bits
PARAM 258 : 40  # set number of 2-way FSK TX onhook mark bits (min > 0)
PARAM 259 : 40  # set number of 2-way FSK TX offhook mark bits (min > 0)
PARAM 260 : 10  # set number of 2-way FSK TX endmark bits (min > 0)
PARAM 261 : 5   # set in 10msec units of 2-way FSK TXRX TX to RX delay time

. . .

#end of voice.prm
```

3. Save the file.

For the added parameter to take effect, you must specify the *Voice.rpm* file as the value for the **ParameterFile** parameter.

3.9.3 Setting the Firmware Buffer Size

To edit the *Voice.prm* file to set the firmware play and record buffer sizes, proceed as follows:

1. Open the file using any text editor.
2. Add the appropriate play and record parameters as shown in bold text in the example below:

```
#beginning of voice.prm

AREA=VOICE
SIZE=WORD
BASE=DECIMAL
```

```
. . .
PARAM 246 : 256  # set firmware play buffer size to 256 bytes
PARAM 247 : 256  # set firmware record buffer size to 256 bytes

#end of voice.prm
```

3. Save the file.

For the added parameter to take effect, you must specify the *Voice.rpm* file as the value for the **ParameterFile** parameter.

3.9.4 Enabling and Modifying Silence Compressed Record Parameters

As distributed, the silence compressed record (SCR) parameters in the *Voice.prm* file appear as comments (each line is preceded with #). To enable the silence compressed record feature and edit this file, remove the # from the beginning of each line containing an SCR parameter. Recommended values for the SCR parameters are provided in the file.

```
# ===============================================================
# SILENCE COMPRESSED RECORD Parameters
# To turn on SCR uncomment all of the lines in the block below.
# Recommended values are given.
# ===============================================================

# --- For Silence Compressed Record, uncomment the block below ---

#PARAM 134 : 100  # SCR_T = 1 second SCR trailing silence
#PARAM 135 : 100  # SCR_PC = 100 bytes of pre-compensation
#PARAM 136 : 43   # SCR_THRES = -43dB silence threshold
#PARAM 137 : 4    # SCR_DG = 40ms of non-silence deglitch
#PARAM 138 : 1    # SCR_ON = SCR is on
# ---- End of SCR block ----
#
```

For additional information about the SCR parameters, refer to Chapter 6, “Silence Compressed Record Parameter Reference”.

3.9.5 Enabling Silence Compressed Record on Only One Board

The silence compressed record feature is enabled in the *Voice.prm* file. When this file is downloaded during initialization, SCR is enabled on all boards in your system.

To enable SCR on only one board in a multi-board system configuration, perform the following steps:

1. Disable the SCR parameters in the *Voice.prm* file.

2. Create a new parameter file that contains the SCR parameters, for example, by copying and renaming *Voice.prm* to *Voicescr.prm*, and then edit the SCR parameters in the new parameter file.

3. Download this new parameter file to the desired board by specifying it with the **ParameterFile** parameter in the *dialogic.cfg* file. For further information, see the ParameterFile parameter in Chapter 4, “Dialogic.Cfg Parameter Reference”.

For detailed information about SCR parameters, see Chapter 6, “Silence Compressed Record Parameter Reference”.

Continue with any additional configuration procedures that are applicable to your system:

- If you will be using non-facility associated signaling, see Section 3.10, “Using Non-Facility Associated Signaling (NFAS)”, on page 36.
- To verify Device Names, see Section 3.11, “Verifying Device Names”, on page 37.
- If you have the Global Call Protocol Package installed, see Section 3.12, “Configuring Global Call CDP File”, on page 41

When you are satisfied with all configuration information (including DM3 configuration if applicable), you must reboot to start the software for the first time. Rebooting the system initializes all the Intel Dialogic products in the system. For information about system startup, see Section 3.13, “Initializing the System”, on page 41.

3.10 Using Non-Facility Associated Signaling (NFAS)

For T1 applications that require Non-Facility Associated Signaling (NFAS), you must edit the following configuration files located in */usr/dialogic/cfg*:

- *nfas.cfg* file
- *(ISDN parameter) .prm* file
- *dialogic.cfg* file

No changes to an application are required. The application just needs to know that there is an additional bearer channel on the trunks that are not using a D channel.

To set up a system to use NFAS, perform the following steps:

1. Edit the *nfas.cfg* file to configure the NFAS group associations. This *nfas.cfg* file is used to inform the device driver which T1 trunks are associated with which ISDN D channels. Comments in the file explain how to perform the setup. The following is an example of an *nfas.cfg* file:

```
# NFAS group 1

# Board ID     Interface    ID D-Channel board ID   NFAS group ID

         1             1                        3               1

         2             2                        3               1

         3             0                        3               1
```

2. Edit the *(ISDN parameter).prm* file to disable the D channel on the trunks that will be sharing the NFAS D channel. Start with a properly configured D channel-equipped parameter file (for

example, *5ess.prm*). Make a copy of that file and name the copy so it is obvious that the two files relate, but are different (for example, name the copy *5ess_NoD.prm*). To disable the D channel in the new file, change parameter 0x0016 to the value 0x02 as shown in the following code segment:

```
;---
;--- ENABLE/DISABLE the D channel (Parameter type 16H)
;--- Use only when the protocol type (Parameter number 13H) is PRI ISDN
;--- for NFAS configuration
;--- Possible values for the data are as follows:
;--- 00H = Undefined.
;--- 01H = Enable the D channel.
;--- 02H = Disable the D channel.
0016 02
```

3. Edit the *dialogic.cfg* file to assign the appropriate parameter file to each trunk in the NFAS group. This is done by adding a ‘ParameterFile=’ line to each trunk in the group. The trunk that carries the actual NFAS D channel is assigned the base parameter file (for example, *5ess.prm*) and the trunks that are sharing the NFAS D channel are assigned the modified parameter file (for example, 5ess_NoD.prm) as indicated by the following sample segment from the dialogic.cfg file:

```
[Genload - ID 0]
ISDNProtocol=5ess
ParameterFile=5ess_NoD.prm

[Genload - ID 1]
ISDNProtocol=5ess
ParameterFile=5ess_NoD.prm

[Genload - ID 2]
ISDNProtocol=5ess
ParameterFile=5ess.prm
```

The NFAS specific changes will take effect the next time the system services are started.

Continue with any additional configuration procedures that are applicable to your system:

- To verify Device Names, see Section 3.11, “Verifying Device Names”, on page 37.
- If you have the Global Call Protocol Package installed, see Section 3.12, “Configuring Global Call CDP File”, on page 41

3.11 Verifying Device Names

This section describes how to verify the device names assigned to the boards in your system.

- Device Overview
- Device Types
- Sorting PCI Springware Boards

- Constructing Device Names

3.11.1 Device Overview

The following concepts are key to understanding Intel Dialogic devices:

device

A computer component controlled through a software device driver. An Intel Dialogic resource board, such as a voice resource, fax resource, and conferencing resource, and network interface board contain one or more logical board devices. Each channel or time slot on the board is also considered a device.

device channel

A data path that processes one incoming or outgoing call at a time (equivalent to the terminal equipment terminating a phone line). The first two numbers in the product naming scheme identify the number of device channels for a given product. For example, there are 24 voice device channels on a D/240JCT-T1 board, 30 on a D/300JCT-E1.

device name

A literal reference to a device, used to gain access to the device via an **xx_open( )** function, where “xx” is the prefix defining the device to be opened. The “xx” prefix is “dx” for voice device, “fx” for fax device, “ms” for modular station interface (MSI) device, and so on. For more information on device names, see Section 3.11.4, “Constructing Device Names”, on page 39.

physical and virtual boards

Intel Dialogic API functions distinguish between physical boards and virtual boards. The device driver views a single physical voice board with more than four channels as multiple emulated D/4x boards. These emulated boards are called virtual boards. For example, a D/120JCT-LS with 12 channels of voice processing contains three virtual boards. A D/480JCT-2T1 board with 48 channels of voice processing and two T1 trunk lines contains 12 virtual voice boards and two virtual network interface boards.

The Intel Dialogic System Software creates standard device and channel names for boards. These names are input as the **namep** parameter to, for example, the **dx_open( )** and **fx_open( )** functions, which return the device handles necessary for many essential API calls, such as **dx_play( )** and **dx_rec( )**.

When assigning device names, the Intel Dialogic System Software first groups the devices into device types and then sorts the devices within each group. Each group’s sort order depends on what kind of boards are installed in your system. Each device is then named according to its device type (group) sort number.

You can verify the Springware device names assigned to the boards in your system as follows:

1. Go to. This is the default location for configuration files. You may have specified a different location when installing the Intel Dialogic System Software.
2. Examine the *Voxcfg* file against the device naming rules described in Section 3.11.4, “Constructing Device Names”, on page 39. Do NOT modify this file.

3.11.2 Device Types

The Intel Dialogic System Software designates devices as the following types:

- **Voice and fax**. Device names for this type receive the prefix **dxxx**.
- **Digital network interface**. Device names for this type receive the prefix **dti**.
- **IP network interface**. Device names for this type are prefixed **ipt**.
- **IP media** (for example, DM3 IPLink boards). Device names for this type are prefixed **ipm**.

Voice boards with an integrated digital network interface are assigned both voice devices and one or two digital network interfaces.

3.11.3 Sorting PCI Springware Boards

The way in which PCI Springware boards are sorted depends on how the boards' rotary switches are set.

- Rotary switch settings are unique: The PCI boards are sorted in ascending order of rotary switch setting.
- Rotary switches are set to zero: The PCI boards are sorted in ascending order of bus and slot number.

Note: Both of these methods may be used in the same system.

Refer to Table 2 for an example.

Table 2. Device Sorting Example

Sort Order	Board	Address	Rotary Switch	Slot Number
1	VFX/41JCT-LS	N/A	0	2
2	D/41JCT-LS	N/A	0	3
3	D/240JCT-T1	N/A	1	1

3.11.4 Constructing Device Names

Once the Intel Dialogic System Software sorts the devices, it assigns names to both devices and channels within devices. The following topics discuss how to construct device names:

- Overview of Device Naming
- Board-Level Names
- Channel-Level Names

Overview of Device Naming

Although there is a great deal of consistency among different types of compatible Intel Dialogic hardware in how devices are numbered, device mapping (device naming or device numbering) is hardware dependent. If a programmer "hard-codes" an application to use device names based on specific Intel Dialogic boards, some of those device names may need to be changed if a different model board is used as a replacement.

A programmer can achieve the greatest degree of backward compatibility among Intel Dialogic boards by making the device mapping in the application program hardware independent. The method for achieving this, along with sample application code, is provided in the technical note entitled “Identifying the number and type of Intel Dialogic boards in a Windows NT system from within an application,” (*http://resource.intel.com/telecom/support/tnotes/tnbyos/winnt/tn193.htm*). This technical note also is available from the Intel Networking & Communications Telecom Support Resources web site *http://developer.intel.com/design/telecom/support/* by selecting Technical Notes, and then the operating system, Windows NT.

Board-Level Names

A device name is assigned to each device or each component in a board as follows:

- **dxxxBn**, where **n** is the device number assigned in sequential order down the list of sorted voice boards. A device corresponds to a grouping of two or four voice channels.

  For example, a D/240JCT-T1 board supports 24 voice channels; the Intel Dialogic System Software therefore divides the D/240JCT-T1 into six voice devices, each device consisting of four channels. Boards with an E1 interface, such as the D/300JCT-E1, support 30 voice channels; the Intel Dialogic System Software divides the D/300JCT-E1 into seven voice devices consisting of four channels each and one voice device consisting of two voice channels.
- **dcbBn**, where **n** is the device number assigned in sequential order down the list of sorted audio conferencing boards. A device corresponds to one DCB board.
- **iptBn**, where **n** is the logical board number that corresponds to a NIC or NIC address when using IP technology. These devices are used by the Global Call API.
- **ipmBn**, where **n** is the board device number assigned to a media board. These devices are used by the Global Call API and the IP Media Library API.
- **brdBn**, where **n** is a physical board name assigned to each board in the system. Given the opaque identifier (AUID) for a board, the **SRLGetPhysicalBoardName( )** function can be used to retrieve the physical board name.

Channel-Level Names

A board device name can be appended with a channel or component identifier. The following channel-level devices are used:

- **dxxxBnCy**: where **y** corresponds to one of the voice channels. Examples of channel device names for voice boards are dxxxB1C1 and dxxxB1C2.
- **dtiBnTy**: where **y** corresponds to one of the digital time slots. Examples of channel device names for digital network interface boards are dtiB1T1 and dtiB1T2.
- **msiBnCy**: where **y** corresponds to one of the conferencing channels.
- **iptBnTy**: where **y** corresponds to the logical channel number over which call signaling is transmitted when using IP technology. These devices are used by the Global Call API.

- **ipmBnTy**: where **y** corresponds to a media resource on a media board and is used to control media streaming and related functions when using IP technology. These devices are used by the Global Call API and the IP Media Library API.

For a given physical board, devices are enumerated sequentially. For example:

For a Springware D/600JCT board, devices are enumerated as follows:

- dxxxB1C1-dxxxB8C2 (span 1) then
- dxxxB9C1-dxxxB16C2 (span 2)

Continue with any additional configuration procedures that are applicable to your system:

- If you have the Global Call Protocol Package installed, see Section 3.12, "Configuring Global Call CDP File", on page 41.

When you are satisfied with all configuration information (including DM3 configuration if applicable), you must reboot to start the software for the first time. Rebooting the system initializes all the Intel Dialogic products in the system. For information about system startup, see Section 3.13, "Initializing the System", on page 41.

3.12 Configuring Global Call CDP File

If you are using the Global Call Protocol Package, the following configuration procedures are applicable:

- Configuring the country dependent parameters (CDP) file
- Downloading the protocol and CDP file

For detailed procedural information, see the *Global Call Country Dependent Parameters (CDP) Configuration Guide*.

3.13 Initializing the System

The new configuration settings will not take effect until the system is initialized. Before system initialization, make sure you perform all of the necessary configuration procedures. To initialize the system for the first time, proceed as follows:

3.13.1 Starting the System Service for the First Time

After you install the Intel Dialogic software with the `install.sh` command and configure the system with the `config.sh` command, the following messages are displayed when you exit from the configuration script:

```
Configuration is complete.

You must reboot the system to start the software for the first time.
Thereafter, you may use the dlstop and dlstart scripts found in
/usr/dialogic/bin
```

Before rebooting the system to start the software, make sure you perform all of the necessary configuration procedures.

Note: If you have D/42JCT-U or D/82JCT-U boards installed, an additional procedure is needed **after** you reboot the system. See Section 3.14, "Updating the PCI ID for D/42JCT-U and D/82JCT-U Boards", on page 43.

When you are satisfied with your configuration, shut down the system and restart it. Rebooting the system initializes all the Intel Dialogic products in the system.

Upon startup, check the screen or system log file for startup messages. The messages vary depending on the boards and software packages you installed. DM3 boards are downloaded before Springware boards (unless a Springware board has been configured as the primary clock master).

For DM3 boards, you should see:

```
Parsing SCD file /usr/dialogic/cfg/pyramid.scd succeeded
```

followed by messages for individual boards. For each board, look for a message that says:

```
Configuring and downloading board succeeded boardNumber=n
```

For Springware boards, you should see:

```
Using /usr/dialogic/cfg/dialogic.cfg to configure Dialogic Boards

System Download
...............................................................
```

followed by a list of boards that were detected and then:

```
n Dialogic Boards Successfully Installed
```

Finally, the system services are started. Once completed, the startup script will exit.

To ensure that the startup script has completed, enter the command:

```
ps -ef | grep S90dialogic | grep -v grep
```

When no output is seen, the startup script has completed.

To display information about boards that are present in the system and recognized by the device driver, enter the command:

```
detect
```

The `detect` command displays the board type (DM3 or Springware), PCI bus and slot number, logical ID, and other useful information for each board. For more information about the `detect` command, see Displaying Board Information in the System Release for Linux Springware Administration Guide.

After starting the system for the first time, you may want to use some of the tools provided by Intel to verify that your system is operating properly. Look in the */usr/dialogic/demos* directory for demo programs.

3.13.2 Starting the System Service after the Initial Startup

Startup should only be performed when the system is stopped, that is, after a `dlstop` command.

You only have to reboot the system for the **initial** startup. To restart the System Service at any time after the initial startup, enter the commands:

```
dlstop
dlstart
```

For information about startup messages, see Section 3.13.1, "Starting the System Service for the First Time", on page 41.

3.14 Updating the PCI ID for D/42JCT-U and D/82JCT-U Boards

Note: Perform this procedure **after** you reboot to start the system for the first time as described in Section 6.1, "Starting the Dialogic System Service for the First Time", on page 123.

After initial configuration and download, the *dialogic.cfg* file shows PCI ID 0 for any D/42JCT-U or D/82JCT-U boards that are installed. This has to be updated with the correct PCI IDs.

Use the following procedure to determine the PCI IDs for D/42JCT-U and D/82JCT-U boards, and to update the *dialogic.cfg* file. It is assumed that you have already installed the boards, installed the Intel Dialogic software, configured the boards using `mkcfg`, and rebooted the system.

1. To determine the PCI IDs of D/42JCT-U and D/82JCT-U boards, run the tool:

   ```
   /usr/dialogic/bin/DxxJCT_id
   ```

   The following shows sample output from this tool:

   ```
   Determining the D/xxJCT series board ID(s)
   Please standby...

   D/82JCT-U  (PCI ID 1) Download .. d82u Firmware Version 6.65 Build 0016
   D/82JCT-U Rev 2 (PCI ID 2) Download .. d82u Firmware Version 6.65 Build 0016

   Please edit /usr/dialogic/cfg/dialogic.cfg
   Set the respective IDs to those just displayed
   ```

   Note that the ID(s) shown will be a function of what boards are in the system and their location.

2. Edit the file */usr/dialogic/cfg/dialogic.cfg*. Look for the board section for each D/42JCT-U or D/82JCT-U board. The section head will show PCI ID 0, for example:

   ```
   [Genload - PCI ID 0]  /* D/XXJCT-U  */
   PBXswitch=Nortel_Norstar.fwl

   [Genload - PCI ID 0]  /* D/XXJCT-U  */
   PBXswitch=Nortel_Norstar.fwl
   ```

3. Change the zeros to the PCI IDs returned by the `DxxJCT_id` tool.

4. Run the `dlstart` command. For further information about `dlstart`, see Section 6.2, "Starting the Dialogic System Service after the Initial Startup", on page 125.

3.15 Reconfiguring the System

Once the system is initialized for the first time, the system must be stopped and restarted in order to make any additional configuration changes. You only have to reboot the system for the initial system startup. To restart (re-initialize) the system, you stop and then restart the system using the dlstop and dlstart utilities.

1. Before you stop the system, the application must be stopped and the application must ensure that all channels have been closed.

2. To stop the system, enter the command:

   ```
   /usr/dialogic/bin/dlstop
   ```

   The messages displayed depend on the boards installed and may include the following:

   ```
   [TEELOGGER]: Starting logging of admin script
   Shutting down Telephony Fault Detectors
   Stopping DM3 Boards
   REGVOX: Deleting DM3 Devices ...
   REGVOX: Delete DM3 Devices Done.
   Shutting down hot swap monitor
   Stopping timeslot doler
   Shutting down DeviceMapper Server
   [TEELOGGER]: Finishing logging of admin script
   Shutting down CORBA Name Server
   Shutting down CORBA Event Server
   ```

3. To restart the system, enter the command:

   ```
   /usr/dialogic/bin/dlstart
   ```

   Startup should only be performed when the system is stopped, that is, after a dlstop command.

   The messages displayed depend on the products installed and may include the following:

   ```
   Starting ORB Event Server:
   Starting Orbacus4 nameserv:
   Starting error logger
   [TEELOGGER]: Starting logging of admin script
   Starting DeviceMapper
   [TBPARMS]:Initializing System Variables for TDMBus
   [TBPARMS]:Base timeslot: 0
   [TBPARMS]:TDMBus Variables initialized
   Starting timeslot doler
   Starting DM3 Boards
   DM3 driver already loaded
   Creating device /dev/mercd
   Starting DM3 download phase
   downloader Version 2.47 Prod 0.02 Build: 00
   Copyright (c) Intel Corporation 1997-2002
     (Using: Host Library version: 6.10  Build: 1)
   Using data files from /usr/dialogic/data
   ```

   followed by screens of hardware configuration information and verification, and completing with the following:

   ```
   REGVOX: Adding DM3 Devices ...
   REGVOX: Adding DM3 Devices Done
   CHEETAHSTART: Removing temp files.
   CHEETAHSTART: Building shared memory.
   CHEETAHSTART: Done.
   Starting fault detection services
   Starting clocking daemon
   [TEELOGGER]: Finishing logging of admin script
   ```

For detailed procedures about other administrative tasks, see the *System Release for Linux Administration Guide* and the *SNMP Agent Software for Linux Administration Guide*.

Dialogic.Cfg Parameter Reference 4

This chapter lists and describes the parameters contained in the Dialogic.cfg file. Parameters are listed in the same order in which they appear in the file. The following parameters are included in the Dialogic.cfg file:

BLTAddress

Not applicable to this release.

BusType

Usage: Global or board parameter, required for boards that support multiple bus types (SCbus boards).

Description: Specifies the telephony bus type.

Guidelines: In systems with mixed bus types, the global parameter can be overridden on a board-by-board basis by including **BusType** as a board parameter.

For H.100, you must manually edit the *dialogic.cfg* file to enter **BusType = H100**. In addition, use the **PrimaryMaster** and **PrimaryMasterClockSource** parameters instead of the **SCbusClockMaster** and **SCbusClockMasterSource** parameters to specify clocking. For example:

```
#BusType=SCBus
#SCBusClockMaster=AUTOMATIC
#SCBusClockMasterSource=AUTOMATIC
BusType=H100
PrimaryMaster=0
PrimaryMasterClockSource=EXTERNAL1
```

Values: Valid values for **BusType** are:

- H100
- NONE
- SCBUS
- Standalone

Default value: SCBUS

Country

Usage: Global or board parameter, required outside North America.

Description: Specifies an international standard, two-letter country code, indicating that a parameter file containing country specific parameters is to be used when firmware is downloaded to the boards. For example, with **Country = FR**, parameter files in */usr/dialogic/data* beginning with FR (for France) are downloaded.

See the Features parameter in this section for information about selecting features from the country specific parameter file.

Guidelines: The Country Specific Parameters package (DLGCparms) is installed with the Springware Software. The install script installs the parameter files in the */usr/dialogic/data* directory, but you must manually configure *dialogic.cfg* with the correct country code.

If *dialogic.cfg* contains a **Country** parameter as well as a **ParameterFile** parameter, Genload selects the parameter file based on the following precedence: the **ParameterFile** board parameter takes precedence, followed by the **ParameterFile** global parameter, and then by the **Country** parameter.

- If the **Country** parameter is used, Genload uses any parameter files beginning with the specified country code, unless a **ParameterFile** parameter exists.
- If the **ParameterFile** parameter is used as a global parameter, Genload uses the specified parameter file name for all boards, except when a **ParameterFile** board parameter is used.
- If the **ParameterFile** parameter is used as a board parameter, Genload uses the specified parameter file name for that board only.

Values: Valid values for **Country** are:

Country	Country Parameter Value
Argentina	AR
Australia	AU
Australia/New Zealand	AN
Austria	AT
Belgium	BE
Brazil	BR
Chile	CL
China	CN
Columbia	CO
Denmark	DK
Euro (CTR-21)	EU
Finland	FI

Country	Country Parameter Value
France	FR
Germany	DE
Greece	GR
Hungary	HU
India	IN
Indonesia	ID
Ireland	IE
Israel	IL
Italy	IT
Japan	JP
Luxembourg	LU
Malaysia	MY
Mexico	MX
Morocco	MA
Netherlands	NL
New Zealand	NZ
Norway	NO
Poland	PL
Portugal	PT
Singapore	SG
South Africa	ZA
South Korea	KR
Spain	ES
Sweden	SE
Switzerland	CH
United Kingdom	UK
United States	US
Venezuela	VE

Default value: No country specific parameters.

CSPExtraTimeSlot

Usage: Board parameter, optional, applies to boards that use Continuous Speech Processing (CSP).

Description: In CSP, extra time slots must be reserved to send echo canceled data over a TDM bus such as the SCbus or CT Bus. When enabled, this parameter causes one extra time slot to be reserved on the TDM bus for each voice channel on a CSP-enabled span or board.

Guidelines: For boards that use CSP, you must set the **CSPExtraTimeSlot** parameter to ON.

Values: Valid values for **CSPExtraTimeSlot** are:

- ON: Extra time slots are reserved.
- OFF: Extra time slots are not reserved.

Default value: OFF

D41DAddress

Not applicable to this release.

D41E_Resource

Not applicable to this release.

DbFirmwareFile

Not applicable to this release.

Dialog/HD

Not applicable to this release.

Download4ChanDb

Not applicable to this release.

DownloadOnly

Usage: Global parameter, optional, applies to all boards.

Description: Specifies the boards to which you want firmware downloaded. If this parameter is not specified, the default is all boards. If this parameter is used, the firmware is downloaded **only** to the boards in this list.

When entering this parameter, the <board-list> can be a single board, a comma-separated list of boards, or a range of boards indicated by starting board, hyphen, and ending board (inclusive). A board is specified using one of the following methods:

- Board ID number: The unique Board Locator Technology (BLT) identification number assigned to a BLT board through hardware switch settings when the board was installed. The board ID number must be in the range 00 to 1F (hexadecimal).
- Board shared RAM address: The unique base memory address in shared RAM assigned to a hardware configurable board. The memory address must be in the range A0000 to DE000 (hexadecimal).

For example:

- **DownloadOnly = 00, 01** downloads firmware to activate the boards that are set to the board IDs 00 and 01.
- **DownloadOnly = 0-1F** downloads firmware to all boards with a board ID between 0 and 1FH. No other boards are downloaded.

For boards that have one or more spans, you must specify the ID of the board and its daughterboard(s) with the **DownloadOnly** parameter as follows:

- For a D/240JCT-T1, D/300JCT-E1, D/320JCT: **DownloadOnly =** ***n,2n***
- For a D/480JCT-2T1 or D/600JCT-2E1: **DownloadOnly =** ***n,2n,1n,3n***

For example, for a D/480JCT-2T1 board, you might specify **DownloadOnly = 2,22,12,32**, where 2 is the ID of the first span, 22 is the ID of the first daughterboard, 12 is the ID of the second span, and 32 is the ID of the second daughterboard. (The ID of the daughterboard is derived from the board ID plus 20.)

Guidelines: For SCbus installations, if you have downloaded Intel Dialogic boards using Genload and then change the *dialogic.cfg* board configuration through the **DownloadOnly** or **SkipBoards** parameter, the system must be rebooted before performing another download. Otherwise, the assigned SCbus time slots may be in conflict, and this can cause corrupt data or other adverse effects.

The **SkipBoards** parameter takes precedence over the **DownloadOnly** parameter.

Values: Valid values for **DownloadOnly** are:

- Board IDs from 00 to 1F (00 to 0F for DIALOG/HD boards).
- Board shared RAM addresses (as described for the **D41DAddress** parameter).

Default value: All boards are downloaded.

EC_Resource

Usage: Board parameter, optional, applies to DIALOG/HD boards.

Description: Enables the echo cancellation resource (ECR) feature on a supported board. The purpose of the echo canceller is to sufficiently reduce the magnitude of the echo component, such that it does not interfere with further processing or analysis of the echo canceled data stream. The echo cancellation capability becomes a system-wide resource that may be applied to any SCbus PCM stream.

Guidelines: To activate ECR after it is enabled, use the **dx_listenecr( )** or **dx_listenecrex( )** function in your application. When a channel is in ECR mode, the following voice operations are unavailable on that channel: play, dial, tone generation, R2MF, and transaction record. For record operations, only 8K PCM is supported.

Although DIALOG/HD Revision 1 boards and Revision 2 boards can coexist in a system, the ECR feature is not supported on DIALOG/HD Revision 1 boards. To identify a DIALOG/HD Revision 1 board, locate the serial number of the board. This number has the format *xxyyyyyy*, where *x* is a letter and *y* is a number. Serial numbers of DIALOG/HD Revision 1 boards begin with CV, CW, or CZ (for example, CZ005000).

Values: Valid values for **EC_Resource** are:

- ON: ECR feature is enabled for this board.
- OFF: ECR feature is not enabled for this board.

Default value: OFF

Features

Usage: Global or board parameter, optional, applies when country specific parameters are used.

Description: Specifies the features to use from the country specific parameter file.

When entering this parameter, the <feature-list> can be a single feature name or a comma-separated list of feature names.

Guidelines: The available features depend on the parameter file used. Parameter files are located in the */usr/dialogic/data* directory; available feature names are listed in each file. See the Country parameter in this section for information about specifying the country specific parameter file to download.

You can specify features on a board-by-board basis by including **Features** as a board parameter.

Values: Valid values for **Features** are:

CEPT1	The board uses the default DTMF output level, -11 dBm Lo-tone, -9 dBm Hi-tone.
CEPT2	Selects DTMF output level of -8 dBm Lo-tone, -6 dBm Hi-tone.
DPD_GENERIC	Activates dial pulse detection. Uses generic set of parameters.
DPD_NONE	Deactivates dial pulse detection. No DPD parameters specified. This is the default.

PPS_10 — The board uses the default 10 PPS for pulse dialing. This feature is used in Japan and Korea.

PPS_20 — The board uses 20 PPS for pulse dialing instead of the default 10 PPS. This feature is used in Japan and Korea.

PROT_BTCALLSTREAM — The board uses the default BT CallStream signaling. This feature is used only with digital front end boards and only in the United Kingdom.

PROT_MERCURYCAS — The board uses the Mercury Channel Associated Signaling instead of the default BT CallStream signaling. This feature is used only with digital front end boards and only in the United Kingdom.

RXGAIN_0 — The board uses the default value of 0 dB receive gain. This feature is available on SCbus analog interface boards in the United States and Japan only.

RXGAIN_N1 — The board has a negative receive gain of -1 dB instead of the default value of 0 dB. This feature is available on SCbus analog interface boards in the United States and Japan only.

RXGAIN_N2 — The board has a negative receive gain of -2 dB instead of the default value of 0 dB. This feature is available on SCbus analog interface boards in the United States and Japan only.

RXGAIN_N3 — The board has a negative receive gain of -3 dB instead of the default value of 0 dB. This feature is available on SCbus analog interface boards in the United States and Japan only.

RXGAIN_P1 — The board has a positive receive gain of +1 dB instead of the default value of 0 dB. This feature is available on SCbus analog interface boards in the United States and Japan only.

RXGAIN_P2 — The board has a positive receive gain of +2 dB instead of the default value of 0 dB. This feature is available on SCbus analog interface boards in the United States and Japan only.

RXGAIN_P3 — The board has a positive receive gain of +3 dB instead of the default value of 0 dB. This feature is available on SCbus analog interface boards in the United States and Japan only.

TS16_CLEAR — Selects Clear Channel Time Slot 16 (CCTS16) for E-1 interface boards, ignores E-1 signaling received from the network on time slot 16, and transmits FFH. Access to time slot 16 is not available. If CCTS16 is used, the corresponding network parameter must be set in the digital network interface parameter file (*spandti.prm*).

TS16_SIG — The E-1 interface board uses the default of E-1 signaling on time slot 16.

Default value: If the **Features** parameter is not specified, the default value depends on the country specific parameter file and the feature as listed above.

FirmwareFile

Usage: Board parameter, optional, applies to all baseboards.

Description: Specifies the name of a firmware load file for the system software to download to the board. This firmware file takes the place of the file that is normally downloaded.

For specifying the firmware load file of the second span on boards that have two spans, use the **FirmwareFile2** parameter.

Guidelines: When you execute Genload, the file that you specify here is located according to the following sequence:

- If a full pathname is specified (for example, **FirmwareFile = /usr/dialogic/data/spandti.fwl**), that file is used.
- If only a file name is specified (for example, **FirmwareFile = spandti.fwl**) and the file is in the directory from which Genload is executed, that file is used.
- Otherwise, the default firmware file location is */usr/dialogic/data*.

The default firmware file is the file specified using the **ISDNProtocol** parameter. If the **ISDNProtocol** parameter is set to NONE, the *spanplus.fwl* file is downloaded.

For Springware boards that support Continuous Speech Processing (CSP), a special firmware file is required. To enable CSP capability for Springware boards, you must explicitly specify the CSP firmware file. See the **FirmwareFile2** parameter for a list of standard (default) and CSP-specific firmware files.

To enable DSP-based fax on Span JCT-series boards, you must select the *spfax.fwl* firmware file.

Values: The firmware load files are installed in */usr/dialogic/data* and most have the extension *.fwl*.

Default value: Without this parameter, Genload automatically selects the correct firmware file to download.

FirmwareFile2

Usage: Board parameter, optional, applies to boards with two spans (for example, D/480JCT-2T1) and to enable Continuous Speech Processing (CSP) capability on Springware boards that support this feature.

Description: Specifies the name of a firmware load file for the system software to download to the second span of an applicable board. This firmware file takes the place of the file that is normally downloaded.

Specify the firmware load file for the first span using the **FirmwareFile** parameter.

Guidelines: For Springware boards that support CSP, a special firmware file is required. To enable CSP capability for Springware boards, you must explicitly specify the CSP firmware file.

For **D/480JCT-1T1** and **D/600JCT-1E1** boards, you can provide for ISDN support on one span and CSP support on the other by using two separate firmware files, one for each span.

- On the first span, you can specify an ISDN protocol and then the specific firmware file required for that ISDN protocol will be automatically downloaded to the board for that span. CSP capability is not available on this span.
- On the second span, you can enable CSP capability, without ISDN support, by specifying the CSP firmware file for that span and setting the ISDN protocol parameter value to **none**.

Note: For E-1 and T-1 boards that support CSP, specifying both an ISDN protocol (with **ISDNProtocol** or **ISDNProtocol2** parameter) and a CSP firmware file (with **FirmwareFile** or **FirmwareFile2** parameter) for the same span results in a download failure to that span. The Intel Dialogic System Service will not start.

Table 3 summarizes CSP and ISDN interoperability for D/480JCT-1T1 and D/600JCT-1E1 boards.

Table 3. CSP and ISDN Interoperability for D/480JCT-1T1 and D/600JCT-1E1 Boards

D/480JCT-1T1 or D/600JCT-1E1	ISDN Protocol Setting	Firmware File Setting	Result
First span	None	Standard firmware file	First span does not support ISDN.
	Specific ISDN protocol selected using the **ISDNProtocol** parameter	Firmware file specific to **ISDNProtocol** parameter automatically downloaded	First span supports ISDN.
Second span	None	CSP firmware file	Second span supports CSP.

For **D/480JCT-2T1** boards, you can provide for CSP support on one span and ISDN support on the other as follows:

- For CSP on the first span and ISDN on the second span:

```
[Genload - PCI ID xx]
FirmwareFile=spcsp.fwl /*for CSP on first span*/
ISDNProtocol2=DMS /*or other ISDN protocol for ISDN on second span*/
```

- For ISDN on the first span and CSP on the second span; note that the **ISDNProtocol2** parameter must explicitly be set to **none** in this case:

```
[Genload - PCI ID xx]
#FirmwareFile= (no FirmwareFile specified for first span)
ISDNProtocol=DMS /*or other ISDN protocol for ISDN on first span*/
FirmwareFile2=spcsp.fwl /*for CSP on second span*/
ISDNProtocol2=NONE /*must be set to none or else Genload will try to force
                   ISDNProtocol value to ISDNProtocol2 and this is not
                   supported with CSP*/
```

Values: Table 4 lists both the standard (default) firmware files and the CSP firmware files for Springware boards that support the CSP feature.

Table 4. Firmware Files for Default and CSP Configurations

	Standard (Default) Configuration		CSP Configuration	
Board Type	**Firmware File**	**Firmware File2**	**Firmware File**	**Firmware File2**
D/120JCT-LS	*spanplus.fwl*	not applicable	*spcsp.fwl*	not applicable
D/240JCT-T1	*spanplus.fwl*	not applicable	*spcsp.fwl*	not applicable
D/480JCT-2T1	*spanplus.fwl* or **ISDNProtocol** parameter value	*spanplus.fwl* or **ISDNProtocol2** parameter value	*spcsp.fwl*	*spcsp.fwl*
D/480JCT-1T1	*spanplus.fwl* or **ISDNProtocol** parameter value	*spanplus.fwl*	*spanplus.fwl* or **ISDNProtocol** parameter value	*spcsp.fwl*
D/600JCT-1E1	*spanplus.fwl* or **ISDNProtocol** parameter value	*spanplus.fwl*	*spanplus.fwl* or **ISDNProtocol** parameter value	*spe1csp.fwl*

Default value: See Table 4.

FrontEnd

Not applicable to this release.

IgnoreMissingBoards

Not applicable to this release.

ISABusWidth

Not applicable to this release.

ISDNProtocol

Usage: Global or board parameter, optional, applies to boards with a digital network interface.

Description: Specifies that the board's digital network interface should be configured for ISDN using the selected ISDN protocol.

For specifying the ISDN protocol of the second span on boards that have two spans, use the **ISDNProtocol2** parameter.

Guidelines: The ISDN PRI Protocols package (DLGCpri) is installed with the Springware Software.

If you use the **ISDNProtocol** parameter to download an ISDN protocol firmware file to a board, the **FirmwareFile** parameter must use its default value.

Note: For E-1 and T-1 boards that support Continuous Speech Processing (CSP), specifying an ISDN protocol and a CSP firmware file for the same span results in a download failure to that span. The Intel Dialogic System Service will not start.

For additional information about CSP interaction with ISDN operation, see the FirmwareFile2 parameter in this section.

Values: Valid values for **ISDNProtocol** are:

NONE	No ISDN protocol is used
1TR6	German National ISDN
4ESS	AT&T 4ESS custom switch TR41449/TR41459
5ESS	AT&T 5ESS custom switch 505-900-322
CTR4	EURO-ISDN ETSI300-102
DASS2	British National BTNR-190-1985
DMS	Northern Telecom custom switch A211-1 and A211-4
DPNSS (separately ordered)	British Private Branch Exchange DASS2 extension
ETN	EURO-ISDN ETSI300-102 for T-1
ETU	EURO-ISDN ETSI300-102 for T-1
NE1	EURO-ISDN ETSI300-102
NI2	National ISDN-2 Bellcore Special Report SR-NWT-002343
NT1	T-1 Network Emulation TR41449/TR41459
NTT	Japanese National ISDN INS-Net 1500
QNT	Q.SIG ISO 11572, ISO 11574
QTE	Q.SIG ISO 11572, ISO 11574
QTN	Q.SIG ECMA-142/143 for T-1
QTU	Q.SIG ECMA-142/143 for T-1
TPH	Australian National ISDN TS-0141 1990
TPHNT	Australian National ISDN TS-0141 1990
VN	French National ISDN VN3

VNNT	French National ISDN VN3 (Network Termination)

Default value: NONE (no ISDN protocol is used)

ISDNProtocol2

Usage: Board parameter, optional, applies to digital network interface boards with two spans (for example, D/480JCT-2T1).

Description: Specifies that the board's second digital network interface should be configured for ISDN using the selected ISDN protocol.

Specify the ISDN protocol for the first span using the **ISDNProtocol** parameter (which may be a global and/or board parameter).

Guidelines: The ISDN PRI Protocols package (DLGCpri) is installed with the Springware Software.

If you use the **ISDNProtocol2** parameter to download an ISDN protocol firmware file, the **FirmwareFile2** parameter must use its default value.

Note: For E-1 and T-1 boards that support Continuous Speech Processing (CSP), specifying an ISDN protocol and a CSP firmware file for the same span results in a download failure to that span. The Intel Dialogic System Service will not start.

For additional information about CSP interaction with ISDN operation, see the FirmwareFile2 parameter in this section.

Values: See the ISDNProtocol parameter in this section.

Default value: NONE (no ISDN protocol is used)

Katakana

Not applicable to this release. This parameter should be set to OFF.

LogFile

Usage: Global parameter, optional, applies to all boards.

Description: Specifies whether to copy screen information generated by Genload to a log file.

Guidelines: The **LogFile** parameter should be the first line in the [Genload - All Boards] section of *dialogic.cfg* to ensure that all download information is captured in the log file. For example:

```
[Genload - All Boards]  /* global parameters */
LogFile = genload.log
BLTAddress = D8000
.
.
.
[Genload - PCI ID 0]    /* board parameters */
.
.
.
```

Values: A file name or full pathname. If a full pathname is given, the directory must exist. If a file name with no path is given, the log file is stored by default in */usr/dialogic/log*.

Default value: No log file is generated.

Netref1Provider

Usage: Global parameter, optional, applies to systems with NETREF_1 as a clock source.

Description: Specifies the board ID for the board that serves as the network reference signal provider. See also the Netref1ProviderSource parameter in this section.

Guidelines: With H.100 CT Bus, the CT_NETREF signal (NETREF_1) carries a network clock signal that may be used by the primary clock master and secondary clock master as their reference. If the **PrimaryMasterClockSource** or **SecondaryMasterClockSource** parameter is set to NETREF1, use the **Netref1Provider** parameter to specify the board ID for the board providing this signal.

If a DM3 board rather than a Springware board is the network reference signal provider, do not use the **Netref1Provider** parameter; you use a parameter in the DM3 configuration file (*pyramid.scd*) instead. The **Netref1Provider** parameter is only applicable when a Springware board is the network reference signal provider.

Values: Valid values for **Netref1Provider** are:

- <board-id>: A board ID in hexadecimal. The specified board serves as the network reference signal provider.
- NONE: NETREF_1 is not used as a clock source.

Default value: NONE

Netref1ProviderSource

Usage: Global parameter, optional, applies to systems with NETREF_1 as a clock source.

Description: Specifies where the network reference signal provider (specified by the **Netref1Provider** parameter) derives its signal.

Guidelines: Clocking should be derived from a digital network trunk if available, not from a board's internal oscillator. The internal oscillator should be used as the clock source only for internal testing purposes.

If a DM3 board rather than a Springware board is the network reference signal provider, do not use the **Netref1ProviderSource** parameter; you use a parameter in the DM3 configuration file (*pyramid.scd*) instead. The **Netref1ProviderSource** parameter is only applicable when a Springware board is the network reference signal provider.

Values: Valid values for **Netref1ProviderSource** are:

- INTERNAL: Board uses its internal oscillator.
- EXTERNAL1: Board uses the clock signal from its front end.

  To specify the second front end on boards that have two spans, specify 0x1n as the **Netref1ProviderSource**, where 0x0n is the board ID of the baseboard.

Default value: EXTERNAL1

ParameterFile

Usage: Global or board parameter, optional, applies to boards with a digital network interface. Also, sometimes needed for voice boards.

Description: Specifies the name of a parameter file containing initialization data for customizing the network firmware for various communication parameters.

Guidelines: Specify a parameter file name only if you have changed the network parameters from the default values. The sample network parameter files are installed in */usr/dialogic/data*.

When you execute Genload, the file that you specify here is located according to the following sequence:

- If a full pathname is specified (for example, **ParameterFile = /usr/dialogic/data/spandti.prm**), that file is used.
- If only a file name is specified (for example, **ParameterFile = spandti.prm**) and the file is in the directory from which Genload is executed, that file is used.

If *dialogic.cfg* contains a **Country** parameter as well as a **ParameterFile** parameter, Genload selects the parameter file based on the following precedence: the **ParameterFile** board parameter takes precedence, followed by the **ParameterFile** global parameter, and then by the **Country** parameter.

Values: Sample parameter files are installed in */usr/dialogic/data* and include:

- *spandti.prm*: Sample parameter file for digital network interface boards.
- *Voice.prm*: Sample parameter file for voice boards.

User customized files may also be used.

Default value: No parameter file

ParameterFile2

Usage: Board parameter, optional, applies to digital network interface boards with two spans (for example, D/480JCT-2T1).

Description: Specifies the name of a parameter file containing initialization data for customizing the network firmware for various communication parameters for the second span of an applicable board.

Specify the parameter file for the first span using the **ParameterFile** parameter.

Guidelines: The parameter file used by the second span of a board is determined by the **ParameterFile** parameter unless you override it using the **ParameterFile2** parameter.

See the **ParameterFile** parameter for additional guidelines.

Values: See the **ParameterFile** parameter.

Default value: Value specified by the **ParameterFile** parameter (or no parameter file).

PBXswitch

Usage: Board parameter, optional, applies to PBX integration boards (for example, D/42JCT-U or D/82JCT-U type boards).

Description: Specifies the PBX (make and model) that the PBX integration board will interface to, so that the correct firmware is downloaded to the board.

Guidelines: Select the appropriate PBX make and model.

Values: Valid values for **PBXswitch** are:

- Lucent_2_wire.fwl
- Lucent_4_wire.fwl
- Mitel_DNIC_M420.fwl
- Mitel_DNIC_M430.fwl
- NEC_DTerm_III.fwl
- Nortel_Meridian_1.fwl
- Nortel_Norstar.fwl
- Siemens_Hicom.fwl
- Siemens_Rolm.fwl

Default value: Nortel_Norstar.fwl

PCMEncoding

Usage: Global or board parameter, required in mixed E-1/T-1 systems; otherwise optional, applies to boards that contain a network interface.

Description: Specifies the pulse code modulation (PCM) encoding method.

Guidelines: When **BusType = NONE** (by default or by explicit setting), the **PCMEncoding** parameter has no effect on D/xxE boards. A-law encoding is used.

If you have downloaded Intel Dialogic boards using Genload and then change the *dialogic.cfg* board configuration through the **PCMEncoding** parameter, the system must be rebooted before performing another download.

Values: Valid values for **PCMEncoding** are:

- ALAW: A-law encoding; normally used by CEPT administrations (E-1 areas).
- ULAW: Mu-law encoding; normally used in North America and Japan (T-1 areas).
- AUTOMATIC: The type of board and country specific support determine the method as follows:
  - A-law is used if a board with an E-1 interface is installed, or if country specific support has been installed for a country other than the United States or Japan.
  - Mu-law is used if a board with a T-1 interface is installed, or if country specific support has been installed for the United States or Japan, or if the board does not use T-1 or E-1.

Default value: AUTOMATIC

PrimaryMaster

Usage: Global parameter, required for systems with H.100 Springware boards.

Description: Specifies the board ID for the board that serves as the primary clock master. A clock master is one of the boards in a system that is designated to provide reference timing for all boards attached to the bus. This board must derive timing from a network reference which ultimately derives clock from a T-1 or E-1 line (for example, the H.100 CT_NETREF), or else must derive timing directly from a digital network interface or, as a last alternative, from its own internal oscillator. See also the PrimaryMasterClockSource parameter in this section.

The H.100 bus has two types of clock masters: primary clock master and secondary clock master. The secondary clock master becomes the clock master if the primary clock master fails or is removed from the system. See also the SecondaryMaster parameter in this section.

Guidelines: If your system contains H.100 Springware boards, use the **PrimaryMaster** and **PrimaryMasterClockSource** parameters instead of the **SCbusClockMaster** and **SCbusClockMasterSource** parameters. **PrimaryMaster** and **PrimaryMasterClockSource** must be added to the *dialogic.cfg* file manually. For example:

```
#BusType=SCBus
#SCBusClockMaster=AUTOMATIC
#SCBusClockMasterSource=AUTOMATIC
BusType=H100
PrimaryMaster=0
PrimaryMasterClockSource=EXTERNAL1
```

If your system contains both DM3 boards and Springware boards, and if a DM3 board is the clock master, do not use the **PrimaryMaster** and **PrimaryMasterClockSource** parameters, and set **SCbusClockMaster = NONE** and **SCbusClockMasterSource = NONE**.

Note: When both DM3 and Springware boards are installed in the same system, the the system automatically determines the technology (board type) that is to provide the clock master by locating the primary master in the configuration files.

Values: Valid values for **PrimaryMaster** are:

- AUTOMATIC: Genload selects a board for the primary clock master.
- NONE: Genload does not select a board as the primary clock master.
- <board-id>: A board ID in hexadecimal. The specified board serves as the primary clock master.

Default value: No default value. If an H.100 Springware board is to serve as the clock master, a value for **PrimaryMaster** must be specified. Otherwise, clocking defaults to **SCbusClockMaster = AUTOMATIC**.

PrimaryMasterClockSource

Usage: Global parameter, optional, applies to systems with H.100 Springware boards.

Description: Specifies where the primary clock master board (specified by the **PrimaryMaster** parameter) gets the clocking for the bus.

Guidelines: A clock master board must derive timing from a network reference which ultimately derives clock from a T-1 or E-1 line (for example, the H.100 CT_NETREF), or else must derive timing directly from a digital network interface or, as a last alternative, from its own internal oscillator. The internal oscillator should be used as the clock source only for internal testing purposes.

Values: Valid values for **PrimaryMasterClockSource** are:

- INTERNAL: Board uses its internal oscillator.
- EXTERNAL1: Board uses the clock signal from its front end.

  To specify the second front end on boards that have two spans, specify 0x1n as the **PrimaryMasterClockSource**, where 0x0n is the board ID of the baseboard.
- NETREF1: Board derives clocking from NETREF_1. See the Netref1Provider parameter in this section.

Default value: EXTERNAL1

SCbusClockMaster

Usage: Global parameter, optional, applies to all SCbus boards.

Description: Specifies the board ID for the board that serves as the master clock source for the SCbus. See also the SCbusClockMasterSource parameter in this section.

Guidelines: If your system contains H.100 Springware boards, use the **PrimaryMaster** and **PrimaryMasterClockSource** parameters instead of the **SCbusClockMaster** and **SCbusClockMasterSource** parameters.

If your system contains both DM3 boards and Springware boards, and if a DM3 board is the clock master, set **SCbusClockMaster = NONE** and **SCbusClockMasterSource = NONE**.

For SCbus installations, if you have downloaded Intel Dialogic boards using Genload and then change the *dialogic.cfg* board configuration through the **SCBusClockMaster** parameter, the system must be rebooted before performing another download.

Values: Valid values for **SCbusClockMaster** are:

- AUTOMATIC: Genload selects a board for the SCbus clock master.
- NONE: Genload does not select a board as the SCbus clock master.
- <board-id>: A board ID in hexadecimal from 00 to 1F (00 to 0F for DIALOG/HD boards). The specified board serves as the SCbus clock master.

Default value: AUTOMATIC

SCbusClockMasterSource

Usage: Global parameter, optional, applies to all SCbus boards.

Description: Specifies where the SCbus clock master board (specified by the **SCbusClockMaster** parameter) gets the clocking for the bus.

Guidelines: Clocking should be derived from a digital network trunk if available, not from a board's internal oscillator. The internal oscillator should be used as the clock source only for internal testing purposes.

If your system contains H.100 Springware boards, use the **PrimaryMaster** and **PrimaryMasterClockSource** parameters instead of **SCbusClockMaster** and **SCbusClockMasterSource**.

If your system contains both DM3 boards and Springware boards, and if a DM3 board is the clock master, set **SCbusClockMaster = NONE** and **SCbusClockMasterSource = NONE**.

Values: Valid values for **SCbusClockMasterSource** are:

- AUTOMATIC: Genload selects the clock source (LOOP for a board with a digital interface or INDEPENDENT for a board without a digital interface).

- INDEPENDENT: Board uses its internal oscillator.
- LOOP: Board uses the clock signal from its front end. This value is not available if the SCbus clock master board has an analog interface.
- NONE: An SCbus board is not the clock master board.

Default value: AUTOMATIC

SecondaryMaster

Usage: Global parameter, optional, applies to H.100 CT Bus applications only.

Description: Specifies the board ID for the board that serves as the secondary clock master. A clock master is one of the boards in a system that is designated to provide reference timing for all boards attached to the bus. This board must derive timing from a network reference which ultimately derives clock from a T-1 or E-1 line (for example, the H.100 CT_NETREF), or else must derive timing directly from a digital network interface or, as a last alternative, from its own internal oscillator. See the SecondaryMasterClockSource parameter in this section.

The H.100 bus has two types of clock masters: primary clock master and secondary clock master. The secondary clock master becomes the clock master if the primary clock master fails or is removed from the system. See the PrimaryMaster parameter in this section.

Guidelines: Use this parameter if you want to specify a secondary clock master for H.100 CT Bus applications. The **SecondaryMaster** parameter must be added to the *dialogic.cfg* file manually, because the default is no secondary clock master.

Values: Valid values for **SecondaryMaster** are:

- <board-id>: A board ID in hexadecimal. The specified board serves as the secondary clock master.
- NONE: No secondary clock master.

Default value: No secondary clock master.

SecondaryMasterClockSource

Usage: Global parameter, optional, applies to H.100 CT Bus applications only.

Description: Specifies where the secondary clock master board (specified by the **SecondaryMaster** parameter) gets the clocking for the bus.

Guidelines: A clock master board must derive timing from a network reference which ultimately derives clock from a T-1 or E-1 line (for example, the H.100 CT_NETREF), or else must derive timing directly from a digital network interface or, as a last alternative, from its own internal oscillator. The internal oscillator should be used as the clock source only for internal testing purposes.

Values: Valid values for **SecondaryMasterClockSource** are:

- INTERNAL: Board uses its internal oscillator.
- EXTERNAL1: Board uses the clock signal from its front end.

  To specify the second front end on boards that have two spans, specify 0x1n as the **SecondaryMasterClockSource**, where 0x0n is the board ID of the baseboard.
- NETREF1: Board derives clocking from NETREF_1. See the Netref1Provider parameter in this section.

Default value: EXTERNAL1

SkipBoards

Usage: Global parameter, optional, applies to all boards.

Description: Specifies the boards that you want Genload to skip when downloading firmware to the boards. Any board in this list does not get firmware downloaded.

When entering this parameter, the <board-list> can be a single board or a comma-separated list of boards. A board is specified using one of the following methods:

- Board ID number: The unique Board Locator Technology (BLT) identification number assigned to a BLT board through hardware switch settings when the board was installed. The board ID number must be in the range 00 to 1F (hexadecimal).
- Board shared RAM address: The unique base memory address in shared RAM assigned to a hardware configurable board. The memory address must be in the range A0000 to DE000 (hexadecimal).

For example, with **SkipBoards = 03** Genload does **not** download firmware to the board with board ID 03.

For boards that have one or more spans, you must specify the ID of the board and its daughterboard(s) with the **SkipBoards** parameter as follows:

- For a D/300JCT-E1 or D/320JCT: **SkipBoards = *n,2n***
- For a D/480JCT-2T1 or D/600JCT-2E1: **SkipBoards = *n,2n,1n,3n***

For example, for a D/480JCT-2T1 board, you might specify **SkipBoards = 2,22,12,32**, where 2 is the ID of the first span, 22 is the ID of the first daughterboard, 12 is the ID of the second span, and 32 is the ID of the second daughterboard. (The ID of the daughterboard is derived from the board ID plus 20.)

Guidelines: For SCbus installations, if you have downloaded Intel Dialogic boards using Genload and then change the *dialogic.cfg* board configuration through the **DownloadOnly** or **SkipBoards** parameter, the system must be rebooted before performing another download. Otherwise, the assigned SCbus time slots may be in conflict, and this can cause corrupt data or other adverse effects.

The **SkipBoards** parameter takes precedence over the **DownloadOnly** parameter.

Values: Valid values for **SkipBoards** are Board IDs from 00 to 1F.

Default value: All boards are downloaded.

DNI Parameter Reference 5

This section lists and describes the Digital Network Interface (DNI) parameters contained in the *Spandti.prm* file. The parameters are organized in the order in which they appear in the file. DNI Parameters include the following:

The DNI parameter file is an ASCII text file used by the Intel System Software to initialize the firmware configuration for the front end of Springware digital network interface boards. The DNI parameter file is named *Spandti.prm* and is installed in the *Data* subdirectory of the Intel Dialogic home directory (normally */usr/dialogic/data*).

If the default settings in *Spandti.prm* are not appropriate for your application, you can modify this file and create a new version of this file. In either case, you must set the **ParameterFile** parameter in the *dialogic.cfg* file to **ParameterFile = spandti.prm**.

All of the *Spandti.prm* parameter values are in hexadecimal.

Receive Wink Definition

Number: 0005

Description: Defines which bit will be examined to detect a wink received from the network and the polarity of the transition to be considered a wink.

Values:

- 01H: detect wink on A bit (lower nibble)
- 02H: detect wink on B bit (lower nibble)
- 04H: Detect wink on C bit (lower nibble) (E1 only)
- 08H: Detect wink on D bit (lower nibble) (E1 only)
- 10H: Positive polarity (off-on followed by on-off transition) (upper nibble)
- 00H: Negative polarity (on-off followed by off-on transition) (upper nibble)

Guidelines: Only one bit may be defined in each nibble. The default value for T1 is 11h (detect wink with positive polarity on A bit). The default value for E1 is 01h (detect wink with negative polarity on A bit).

Transmit National and International Bits

Number: 0006

Description: Defines the National and International signaling to be carried in time slot 0 of the odd frames in an E1 multiframe. Bit 1 is the International Bit and bits 4 through 8 are the National Bits.

Values: An 8-bit byte expressed in hexadecimal.

Guidelines: The default data value is 7Fh (all National bits and the International bit set to 1.

Transmit Extra Bits

Number: 0007

Description: Defines the value of the spare bits in time slot 16 of frame 0 in an E1 multiframe. This is the MultiFame Alignment Signal (MFAS).

Values: An 8-bit byte expressed in hexadecimal.

Guidelines: The default value is 07h (all spare bits set to 1).

Initial Signaling Insertion Pattern

Number: 0008

Description: Defines the default contents of the transmit signaling buffer for all channels. The transmit signaling on a channel will reflect the contents of the transmit signaling buffer for that channel when the channel is set to the signaling insert mode.

Values: The masks to set the corresponding signaling bits on are:

- 01H: A signaling bit
- 02H: B signaling bit
- 04H: C signaling bit (E1 only)
- 08H: D signaling bit (E1 only)

Guidelines: For T1, the default value is 00h (A and B bits are 0). For E1, the default value is 0Bh (A, B, and D bits are 1; C bit is 0) (blocking).

Signaling Mode

Number: 0009

Description: Defines the default signaling mode.

Values:

- 01H: Transparent mode (Default for T1)
- 00H: Insertion mode (Default for E1)

Guidelines: Transparent mode is used when the signaling from the TDM bus is transmitted to the T1 or E1 line. Insertion mode is used when the interface controls the signaling to the T1 or E1 line.

Idle Mode

Number: 000A

Description: Defines whether the T1 or E1 interface should transmit the idle pattern by default.

Values:

- 00H [default]: Do not transmit the idle pattern.
- 01H: Transmit the idle pattern.

Guidelines: If set to Do not transmit idle, data from the TDM bus is transmitted to the T1 or E1 line. If set to Transmit idle, the idle pattern is transmitted to the T1 or E1 line.

Transmit Idle Pattern

Number: 000B

Description: Defines the pattern to be used when the interface is transmitting the idle pattern to the T1 or E1 line.

Values:

- 00H[default]: An idle pattern of 7Fh will be transmitted if interface is T1; an idle pattern of 54h will be transmitted if interface is E1.
- 01H: An idle pattern of FFh will be transmitted if interface is T1; an idle pattern of D5h will be transmitted if interface is E1.

Guidelines: The default value for both T1 and E1 is 00H.

Transmit Wink Definition

Number: 000C

Description: Defines the state of the signaling bits used to transmit a wink. A wink starts by transmitting signaling state 0 for the pre-wink delay time. Then signaling state 1 is transmitted for the wink length time before returning to signaling state 0.

Values:

- 01H: State 0 on A signaling bit (lower nibble)
- 02H: State 0 on B signaling bit (lower nibble)
- 04H: State 0 on C signaling bit (lower nibble) (E1 only)
- 08H: State 0 on D signaling bit (lower nibble) (E1 only)
- 10H: State 1 on A signaling bit (upper nibble)
- 20H: State 1 on B signaling bit (upper nibble)
- 40H: State 1 on C signaling bit (upper nibble) (E1 only)
- 80H: State 1 on D signaling bit (upper nibble) (E1 only)

Guidelines: When a data bit is set, the corresponding signaling bit is ON (1) in the signaling state. For T1, the default value is 01h (A bit toggles from OFF to ON to OFF, B bit remains OFF). For E1, the default value is 89h (A bit toggles from ON to OFF to ON, B and C bits remain OFF, and D bit remains ON).

Transmit Pulse Digit Make/Break State Definition

Number: 000D

Description: Defines the signaling bit states used to transmit a pulse digit. A pulse digit consists of a series of pulses from a make signaling state to a break signaling state.

Values:

- 01H: Make state A signaling bit (lower nibble)
- 02H: Make state B signaling bit (lower nibble)
- 04H: Make state C signaling bit (lower nibble) (E1 only)
- 08H: Make state D signaling bit (lower nibble) (E1 only)
- 10H: Break state A signaling bit (upper nibble)
- 20H: Break state B signaling bit (upper nibble)
- 40H: Break state C signaling bit (upper nibble) (E1 only)
- 80H: Break state D signaling bit (upper nibble) (E1 only)

Guidelines: When a data bit is set, the corresponding signaling bit is ON (1) in the signaling state. For T1, the default value is 01h (A bit pulses from On to OFF to ON, B bit remains OFF). For E1, the default value is 98h (A bit pulses from OFF to ON to OFF, B and C bits remain OFF, and D bit remains ON).

Number of Pulses Per Digit

Number: 000E

Description: Defines the number of pulses in each digit dialed.

Values:

- 00H[default]: Digits 1 through 9 are represented by the corresponding number of pulses and digit 0 is represented by 10 pulses.
- 01H: Digits 0 through 9 are represented by the corresponding number of pulses +1 pulse.

Guidelines: The default value is 00H.

CRC Enable Switch

Number: 000F

Description: For E1 interfaces, turns the transmission of the CRC-4 pattern on (enabled) or off (disabled) and searches for such a pattern in the received signal.

Values:

- 00H[default]: Disable transmission of the CRC-4 pattern.
- 01H: Enable transmission of the CRC-4 pattern.

Guidelines: The default value is 00H.

Receive Pulse Digit Definition

Number: 0011

Description: Defines which bit will be examined to detect a pulse digit received from the network and the polarity of the transition to be considered a pulse.

Values:

- 01H: detect digit on A bit (lower nibble)
- 02H: detect digit on B bit (lower nibble)
- 04H: Detect digit on C bit (lower nibble) (E1 only)
- 08H: Detect digit on D bit (lower nibble) (E1 only)
- 00H: Positive polarity (off -on followed by on-off transition) (upper nibble)
- 10H: Negative polarity (on-off followed by off-on transition) (upper nibble)

Guidelines: Only one bit may be defined in each nibble. The default value for T1 is 11h (detect digit with negative polarity on A bit). The default value for E1 is 01h (detect digit with positive polarity on A bit).

Line Length

Number: 0012

Description: Defines output waveform template based on length of cable being driven for T1 interface.

Values:

- 00H [default]: 000 - 110 feet
- 01H: 110 - 220 feet
- 02H: 330 - 440 feet
- 03H: 220 - 330 feet
- 04H: Square template
- 05H: > 655 feet
- 06H: 440 - 550 feet
- 07H: 550 - 660 feet

Guidelines: The default value is 00H (000 - 110 feet)

CCTS16 (Clear Channel Time Slot 16)

Number: 0013

Description: Defines whether time slot 16 of an E1 multiframe will be used for signaling or for data (clear channel).

Values:

- 00H[default]: Time slot 16 is to be used for E1 signaling.
- 01H: Time slot 16 is defined as a clear channel and will be used to carry data.

Guidelines: The default is to use time slot 16 for signaling.

ESF Framing

Number: 0014

Description: Defines whether D4 Superframe or Extended Superframe (ESF) framing will be used in a T1 interface. D4 Superframe format uses 12 frames and ESF framing uses 24.

Values:

- 00 [default]: Superframe format will be used.
- 01: ESF framing will be used.

Guidelines: The default is D4 Superframe.

Zero Code Suppression

Number: 0020

Description: Specifies the type of zero code suppression to be used for a T1 interface.

Values:

- 00H[default]: No zero code suppression will be used.
- 01H: B8ZS - Binary eight zero code suppression will be used.
- 02H: Bit 7 jamming will be used for zero code suppression

Guidelines: The default is for no zero code suppression to be used.

Silence Compressed Record Parameter Reference 6

This section lists and describes the Silence Compressed Record (SCR) parameters contained in the *Voice.prm* file. The SCR parameters include:

As distributed, the SCR parameters in the *Voice.prm* file appear as comments (each line is preceded with #). To enable the SCR feature and edit this file, remove the # from the beginning of each line containing the SCR parameter. The recommended values for the SCR parameters are provided in the file.

SCR_T (Trailing Silence)

Description: The time, in 10-millisecond units, that silence can trail the end of speech before silence compression begins. This value impacts the amount of compression to be performed and, consequently, the final size. As you decrease the value, the amount of silence recorded between speech is decreased.

You can adjust this value to suit your environment. For example, increase this value if words or sentences run together, and decrease the value if the intervals of silence are too long.

Values: Time in 10-millisecond units (from 0 to 100 units)

Guidelines: The default value is 100 units (1 second).

SCR_PC (PreCompensation)

Description: The number of bytes of precompensation. Precompensation specifies the maximum length of silence that is recorded on the leading edge of speech. This prevents the beginning of speech that activates the recording from being dropped (clipped) after a period of silence.

When SCR is in use, two buffers of 512 bytes of shared RAM are allocated to store incoming audio. Data fills one buffer and is passed to the driver. If the audio is below the silence threshold (thus considered to be silence) for a specified period of time, data in the buffer is not passed to the driver, but is discarded.

Values: 0 to 512 bytes

Guidelines: The default is 512 bytes. For best performance, use the recommended value of 100 bytes. Otherwise, the recording may become garbled.

SCR_THRES (Silence Threshold)

Description: Defines the audio level in the phone line below which the signal is considered noise and above which is considered speech. When the audio level is at or below the value set by **SCR_THRES** for a minimum duration of time defined by the **SCR_T** parameter, silence compression begins.

The **SCR_THRES** numeric value is converted to a negative dB value by the firmware, where 20 represents -20 dB and 50 represents -50 dB.

Values: 20 to 50

Guidelines: The default is 43 (-43 dB). You can adjust this value to suit a particular environment. For example, the threshold might be higher in a noisy environment. If you specify an invalid value, the value is ignored and the default value is used.

SCR_DG (DeGlitch)

Description: Defines in 10-millisecond units the maximum non-silence period (glitch) that is ignored. A glitch may be a spike or short burst of noise on the line that is not speech. Silence compression continues if a glitch less than or equal in duration to **SCR_DG** occurs.

Values: Time in 10-millisecond units (from 0 to 20 units)

Guidelines: The default is 4 (40 milliseconds). You can increase this value if the recording includes too much noise, or decrease the value if you are losing speech.

SCR_ON

Description: Defines whether the SCR feature is enabled or disabled.

Values:

- 1 [default]: SCR is enabled.
- 0: SCR is disabled.

Index