DM3 Configuration File Reference

05-1272-005

Publication Date: January 2003

Document Number: 05-1272-005

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 Communications Systems Products website at:
http://www.intel.com/network/csp/

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

Table of Contents

List of Figures

List of Tables

About This Information

The following topics provide information about this guide:

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

Purpose

This document provides reference information about DM3 configuration files. It also provides instructions for editing configuration files and using the FCDGEN and PDKManager utilities.

Intended Audience

This information is intended for:

- Distributors
- System Integrators
- Toolkit Developers
- Value Added Resellers (VARs)
- Original Equipment Manufacturers (OEMs)

How to Use This Information

This information is organized as follows:

- Chapter 1, “Configuration Overview”provides an overview of the configuration process and describes the files associated with this process as well as the initialization process.
- Chapter 2, “Configuration Procedures” describes how to modify the configuration parameters by editing the CONFIG and CDP files, and running the applicable configuration utility.

- Chapter 3, "Boards and Applicable CONFIG File Sections" lists the various DM3 boards by product category and lists the CONFIG file used by each board as well as the sections in that CONFIG file that apply to the board.
- Chapter 4, "Parameter Reference" lists each CONFIG file parameter and provides detailed information about that parameter, including the parameter name, a description of the parameter, and the values that can be selected for that parameter.

This manual covers both Windows and Linux operating systems. Where these operating systems are similar, that is, the same except for syntax, Windows instructions are shown. Where these operating systems differ, special Windows Notes or Linux Notes are provided.

NOTE: Command instructions, directories paths and environment variable settings are shown relative to the Dialogic subdirectory. By default, the System Release software is installed in the following directory:

- Windows - C:\Program Files\Dialogic
- Linux - /usr/dialogic

When this manual specifies the Dialogic\Data directory it is referring to C:\Program File\Dialogic\Data in Windows and /usr/dialogic/data in Linux.

Related Information

For information on Dialogic products, visit our website at *http://www.dialogic.com*.

See the following for more information:

- *Compatibility Guide for the Dialogic R4 API on DM3 Products*
- *DM3 GlobalCall Resource Analog Protocol User's Guide*
- *DM3 GlobalCall Resource ISDN Protocol User's Guide*
- *DM3 GlobalCall Resource T-1 Protocol User's Guide*
- *DM3 GlobalCall Resource User's Guide*
- *DM3 IPLink User's Guide*
- *System Release Installation and Configuration Guide for Windows*
- *System Release for Linux Installation Guide*

- DCM Online Help

The following documentation supports GlobalCall protocols which are provided in a separately orderable protocol package:

- *GlobalCall Country Dependent Parameter (CDP) Reference*

For information on the GlobalCall Protocol Package, see your Dialogic Sales representative.

1. Configuration Overview

This section provides information on the following:

- Configuration File Sets
- Media Loads and Routing Configurations
- Configuration (CONFIG) File
- Feature Configuration Description (FCD) File
- Product Configuration Description (PCD) File
- System-level and Additional Board-level Parameters
- System Initialization
- Country Dependent Parameters (CDP) File

A DM3 board is configured by downloading firmware. Configuration files are used to identify which software components are downloaded and how these individual components are configured. The major steps used to configure a DM3 board include:

1. Editing the CONFIG file.
2. Generating an FCD file.
3. Configuring system-level and additional board-level parameters.
4. Initializing the system.
5. Modifying and Downloading CDP file (applicable when using PDK Protocols.

The files and utilities associated with the DM3 board configuration process are shown in Figure 1, “DM3 Board Configuration Process”, on page 2.

Detailed information about the DM3 board configuration procedures is provided in Chapter 2, “Configuration Procedures”.

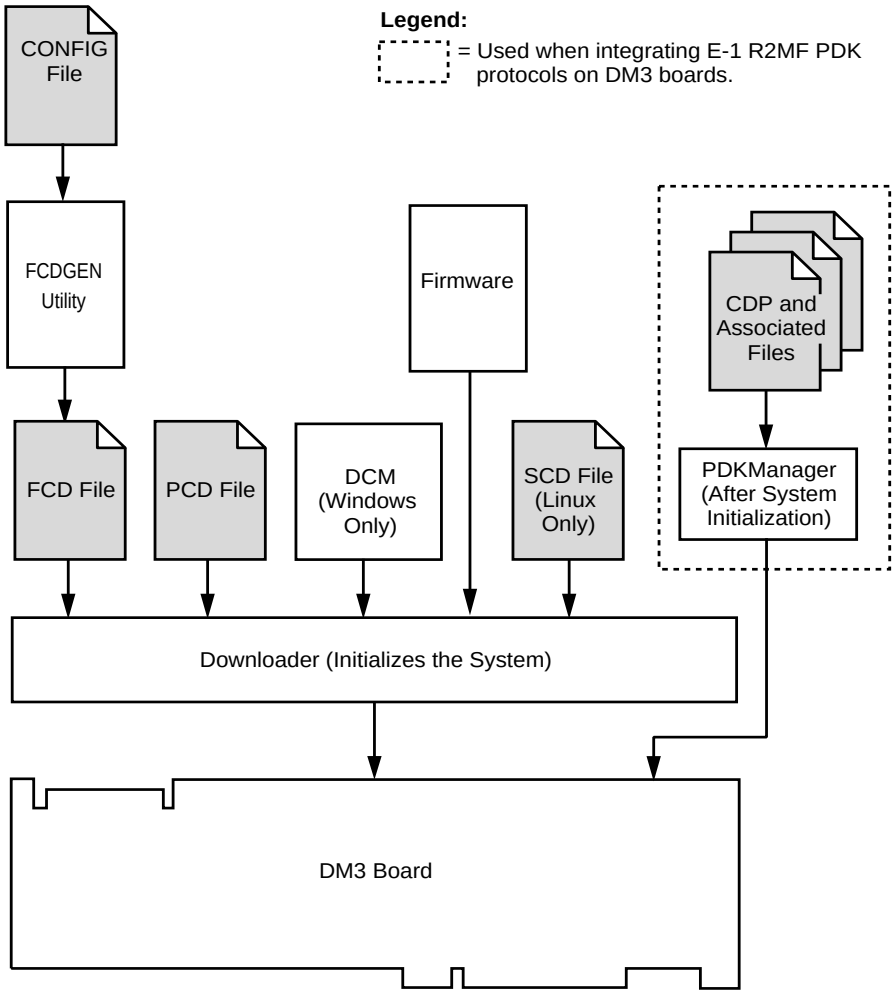


Figure 1. DM3 Board Configuration Process

1.1. Configuration File Sets

Except for the SCD (Linux only) and CDP files, the configuration files associated with a specific configuration have the same name; only the extensions (*.pcd*, *.fcd* and *.config*) differ. A set of these files with the same name are used for a specific DM3 board type. The board type can include a single board or a group of similar boards. Depending on the board type and the protocol that the board will use, a specific FCD and PCD file are downloaded to that board. In addition, if the FCD file needs to be modified, the CONFIG file in that same set is updated and used to generate a new FCD file. Configuration file sets are located in the \Dialogic\Data directory.

For example, if you were using the robbed bit protocol on a DM/V960A-4T1-PCI board, then you might use the configuration file set that included:

- *ml2_qsa_t1.config*
- *ml2_qsa_t1.fcd*
- *ml2_qsa_t1.pcd*

For a detailed list of boards and supported configuration file sets, see Chapter 3, "Boards and Applicable CONFIG File Sections".

1.2. Media Loads and Routing Configurations

Media loads are pre-defined sets of features supported by certain DM3 boards. A media load is downloaded to a board by choosing a configuration file set that supports that particular media load.

Most feature sets are numbered (for example, 1, 2, 9b) for identification purposes and each media load supports a different combination of features. Examples of media features include speed control, volume control, transaction recording, silence compressed record (SCR) and conferencing. For a complete list of features supported by each media load, consult the *Release Guide* accompanying your Dialogic System Software.

R4 on DM3 supports two types of routing configurations, fixed and flexible routing. With fixed routing, the resource devices (voice/fax) and network interface devices are permanently coupled together in a fixed configuration. Only the

network interface time slot device has access to the CT Bus. With flexible routing, the resource devices (voice/fax) and network interface devices are independent, which allows exporting and sharing of the resources. All resources have access to the CT Bus. Except on some Intel® Dialogic® Integrated (DI) Series boards, media loads support flexible routing. For a complete discussion on fixed and flexible routing, see the *Compatibility Guide for the Dialogic R4 API on DM3 Products*.

For a list of CONFIG files supported by each board, see the CONFIG file tables listed in Chapter 3, "Boards and Applicable CONFIG File Sections".

MediaSpan Media Loads

QuadSpan, DualSpan, and Voice Resource media load configuration files are identified by having an ml*x* prefix, where *x* represents the specific media load. Configuration files that do not begin with the ml*x* prefix do not support media loads and do not support flexible routing. For example, on a DM/V960A-4T1-cPCI board using the T-1 ISDN 4ESS protocol:

- *qs2_isdn_4ess.config* supports a fixed routing configuration.
- *ml2_qs2_4ess.config* supports media load 2 in a flexible routing configuration.

IPLink Media Loads

IPLink media loads and routing configurations are identified as follows:

- ml*x* included in the filename indicates media load *x* and flexible routing is supported
- evr (exportable voice resources) excluded from the filename indicates media load 1 and fixed routing is supported
- evr included in the filename indicates flexible routing and the following media load support:
  - media load 1 support (applicable to boards containing 303 or 307 DSPs)
  - media load 2 support (applicable to all other boards)

For example, on a DM/IP301-1E1-PCI-100BT board using the E-1 ISDN NET5 protocol:

- *ipt_isdn_net5_311.configs* supports media load 1 and a fixed routing configuration
- *ipt_evr_isdn_net5_311.config* supports media load 2 in a flexible routing configuration
- *ipvs_evr_isdn_net5_ml11_311.config* supports media load 11 in a flexible routing configuration

Fax and VFN Media Loads

Fax and VFN media load configuration files are identified by having an fn3 or a vfn3 prefix, respectively. Configuration files that do not begin either of these prefixes do not support media loads and do not support flexible routing. For example, on a DM/VF240-1T1-PCI board using the T-1 ISDN DMS protocol:

- *vfn_isdn_dms.config* supports a fixed routing configuration.
- *vfn3_isdn_dms.config* supports media loads in a flexible routing configuration.

Dialogic Integrated Series Media Loads

Dialogic Integrated (DI) Series media load 1 configuration files do not contain any media load identifier in the file name. For media loads greater than 1 (for example, media load 4), the media load configuration files are identified by having an ml*x* suffix, where *x* represents the specific media load. For example, on a DI/0408-LS-A-R2 board:

- *DI0408LSA_REV2.config* supports media load 1.
- *DI0408LSA_REV2_ML4.config* supports media load 4.

Configuration files that do not contain an ml*x* prefix may or may not support media load 1. For example:

- *DI0SI16.config* does not support media load 1 on the DI/SI-16 board.
- *DI0408LSA.config* supports media load 1 on the DI/0408-LS-A board.

For a complete list of media loads supported on each DI board, consult the *Release Guide* accompanying your Dialogic System Software.

1.3. Configuration (CONFIG) File

The CONFIG file (*.config*) contains the modifiable parameter settings used to configure DM3 components. Parameters are modified in the CONFIG file and then the CONFIG file is used to generate a modified FCD file.

The CONFIG file is divided into a number of sections that correspond to the DM3 components and subcomponents being configured. These include:

- [encoder] Section
- [recorder] Section
- [0x39] Section
- [0x3b] Section
- [0x3b.x] Section
- [lineAdmin.x] Section
- [NFAS] Section
- [CAS] Section
- [CCS] Section
- [CHP] Section
- [TSC] Section
- [0x1b] Section
- [NetTSC] Section
- [0x1d] Section
- [0x1c] Section

1.3.1. [encoder] Section

The encoder component resides on the signal processor (SP) and can be instantiated to perform an encoding process on a media stream. The most common type of encoder is a component that is part of a Recorder resource. Resources such as the Recorder may include several different encoder components to support different coding algorithms or provide varying capabilities.

Encoder parameters are modified by editing the respective lines in the [encoder] section of the CONFIG file. For example, to change the **PrmAGCk** parameter

(parameter number =0x401) from 0x2061 (-30.0 dB) to 0xd5f1 (-13.6 dB), you would change the value in the [encoder] section from 0x2061 to 0xd5f1.

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

[encoder] before editing:

```
[encoder]
SetParm=0x401,0x2061              ! PrmAGCk
SetParm=0x403,0x028f5c            ! PrmAGCMEM_sil_reset
```

[encoder] after editing:

```
[encoder]
SetParm=0x401,0xd5f1              ! PrmAGCk
SetParm=0x403,0x028f5c            ! PrmAGCMEM_sil_reset
```

For information about each encoder parameter, see Section 4.1, “[encoder] Parameters”, on page 110.

1.3.2. [recorder] Section

The recorder component and its instances reside on the control processor (CP) and handle the resource’s message exchanges with the host, as well as media stream management and encoder component control functions. Each Recorder component instance has an associated instance of an encoder component, which resides on an SP and which actually processes the data to be recorded.

Recorder parameters are modified by editing the respective lines in the [recorder] section of the CONFIG file. For example, to change the **AGCOnOff** parameter (parameter number=0x205) from a value of Enable to Disable, you would change the value in the [recorder] section from 1 to 0.

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

[recorder] before editing:

```
[recorder]
SetParm=0x205,1              ! Enable AGC for recording (1=Enable, 0=Disable)
!SetParm=0x203,0x71835e01    ! BEEP_DGSD, Digital Sound Beep
```

[recorder] after editing:

```
[recorder]
SetParm=0x205,0              ! Enable AGC for recording (1=Enable, 0=Disable)
!SetParm=0x203,0x71835e01    ! BEEP_DGSD, Digital Sound Beep
```

For information about each recorder parameter, see Section 4.2, "[recorder] Parameters", on page 118.

1.3.3. [0x39] Section

The [0x39] section of the CONFIG file defines conferencing parameters applicable to all conferencing lines. Parameters are modified by editing (or adding) the respective lines in the [0x39] section of the CONFIG file.

NOTE: Only modifications to the **ToneClamping** parameter are supported in this CONFIG file section. The **ToneClamping** parameter applies only to Dialogic Integrated Series boards.

Tone clamping is enabled by default. To disable tone clamping, the **ToneClaming** parameter (parameter number=0x3925) must be added to the CONFIG file and set to a value of 0.

NOTE: For Dialogic Integrated Series boards, disabling tone clamping also requires modifications to the [0x3b.*x*] sections of the CONFIG file. For details, see Section 1.3.5, "[0x3b.x] Section", on page 10.

Following is an excerpt from the [0x39] section of a CONFIG file that illustrate that part of the file before and after editing.

[0x39] before editing:

```
[0x39]
SetParm=0x3926,16 ! Total # of parties (Conf_MaxTotalParties)
                  ! per conferencing board.
```

[0x39] after editing:

```
[0x39]
SetParm=0x3925,0  ! Disable Tone Clamping
SetParm=0x3926,16 ! Total # of parties (Conf_MaxTotalParties)
                  ! per conferencing board.
```

To enable tone clamping, delete the **ToneClamping** parameter from the [0x39] section of the CONFIG file.

For information about each [0x39] parameter, see Section 4.3, "[0x39] Parameters", on page 119.

1.3.4. [0x3b] Section

The [0x3b] section of the CONFIG file defines conferencing parameters. These parameters are modified by editing the respective lines in the [0x3b] section of the CONFIG file. For example, to change the **ActiveTalkerNotifyInterval** parameter (parameter number=0x3b02) from a value of 100 (1 second) to 150 (1.5 seconds), you would change the value in the [0x3b] section from 100 to 150.

Following is an excerpt from the [0x3b] section of a CONFIG file that illustrate that part of the file before and after editing.

[0x3b] before editing:

```
[0x3b]
SetParm=0x3b02,100 !ActiveTalkerNotifyInterval
                   !100 frame interval; 10ms * 100 = 1sec)
```

[0x3b] after editing:

```
[0x3b]
SetParm=0x3b02,150 !ActiveTalkerNotifyInterval
                   !(150 frame interval; 10ms * 150 = 1.5sec)
```

For information about each [0x3b] parameter, see Section 4.4, "[0x3b] Parameters", on page 120.

1.3.5. [0x3b.x] Section

The [0x3b.*x*] section of the CONFIG file defines conferencing parameters specific to each conferencing line. There is an [0x3b.*x*] section for each line. For example, [0x3b.1] applies to conferencing line 1, [0x3b.2] applies to conferencing line 2, and so on.

Parameters are modified by editing (or adding) the respective lines in each [0x3b.*x*] section of the CONFIG file.

NOTE: Only modifications to the **NotificationTone** parameter are supported in this CONFIG file section. The **NotificationTone** parameter applies to QuadSpan Voice, DualSpan Voice, and Dialogic Integrated Series boards.

On DM3 boards, the conference notification tone is enabled by default. To disable the conference notification tone, the **NotificationTone** parameter (parameter number=0x3b06) must be added each line and set to a value of 0.

NOTE: For Dialogic Integrated Series boards, disabling tone clamping also requires modifications to the [0x39] section of the CONFIG file. For details, see Section 1.3.3, “[0x39] Section”, on page 8.

Following is an excerpt from the [0x3b.**x**] sections of a DI/0408-LS-A CONFIG file (three conferencing lines) that illustrate that part of the file before and after editing.

[0x3b.x] before editing:

```
[0x3b.1]
SetParm=0x3b05,0  ! Tariff Tone OFF

[0x3b.2]
SetParm=0x3b05,0  ! Tariff Tone OFF

[0x3b.3]
SetParm=0x3b05,0  ! Tariff Tone OFF
```

[0x3b.x] after editing:

```
[0x3b.1]
SetParm=0x3b05,0  ! Tariff Tone OFF
SetParm=0x3b06,0  !Notification Tone OFF

[0x3b.2]
SetParm=0x3b05,0  ! Tariff Tone OFF
SetParm=0x3b06,0  !Notification Tone OFF

[0x3b.3]
SetParm=0x3b05,0  ! Tariff Tone OFF
SetParm=0x3b06,0  !Notification Tone OFF
```

To enable the conference notification tone, delete the **NotificationTone** parameter from each conferencing line, that is, each [0x3b.*x*] section of the CONFIG file.

For information about each [0x3b.x] parameter, see Section 4.5, “[0x3b.x] Parameters”, on page 121.

1.3.6. [lineAdmin.x] Section

The Line Administration (LCON) component resides on the control processor and manages line devices. There is one instance of the LCON component for each line (span).

There is a [lineAdmin.x] section in the CONFIG file for each line on the board. The parameters defined in a [lineAdmin.x] section apply to the line specified, for example, parameters in the [lineAdmin.3] section apply to line 3.

Line Administrator parameters are modified by editing the respective lines in the [lineAdmin.x] section of the CONFIG file. For example, to change the **SignalingType** parameter (parameter number=0x1602) from a value of CAS to CCS for the second T-1 span, you would change the value in the [lineAdmin.2] section of the CONFIG file from 4 to 5.

Following is an excerpt from the [lineAdmin.2] section of a CONFIG file that illustrates that part of the file before and after editing.

Before editing:

```
SetParm=0x1601,0    !LineType (dsx1_D4=0, dsx1_ESF=1)
SetParm=0x1602,4    !SignalingType (CAS=4, CCS=5, Clear=6)
SetParm=0x1603,7    !Coding (B8ZS=7, AMI=8)
```

After editing:

```
[lineAdmin.2]
SetParm=0x1601,0    !LineType (dsx1_D4=0, dsx1_ESF=1)
SetParm=0x1602,5    !SignalingType (CAS=4, CCS=5, Clear=6)
SetParm=0x1603,7    !Coding (B8ZS=7, AMI=8)
```

For information about each Line Administrator parameter, see Section 4.6, "[lineAdmin.x] Parameters", on page 122.

1.3.7. [NFAS] Section

Non-Facility-Associated Signaling (NFAS) uses a single ISDN PRI D channel to provide signaling and control for up to 10 ISDN PRI lines. Normally, on an ISDN PRI line, one D channel is used for signaling and 23 B channels (bearer channels) are used for transferring information. In an NFAS configuration, therefore, one D channel can support the signaling and control for up to 239 B channels. The trunk that provides the signaling is called the primary D channel. The trunks that use all 24 channels as B channels are called NFAS trunks.

NOTES:
1. For a board containing multiple primary D channels, the maximum number of trunks supported by each NFAS group on that board is reduced. This is due to the additional message load on the board's CPU.
2. NFAS is only supported by the NI2, 4ESS, 5ESS and DMS protocols.

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

NFAS parameters are modified by editing the respective lines in the [NFAS] and [NFAS.x] sections of the CONFIG file. For example, to increase the number of NFAS groups per board from one to four, change the value of **NFAS_INSTANCE_MAP** (parameter = 0x3e02) from a value of 1 (one group per board) to a binary value of 1111 (four NFAS groups per board) represented by 0xf.

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

Before editing:

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

After editing:

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

Also, when NFAS is used, the **SignalingType** parameter in the [lineAdmin] section of the CONFIG file must be modified. For details about this parameter modification, see Section 4.6.2, “SignalingType (Signaling Type)”, on page 123.

1.3.8. [CAS] Section

The Channel Associated Signaling (CAS) component resides on the signal processor. It is responsible for managing the generation and detection of digital line signaling functions required to manage voice channels. Each CAS instance corresponds to the CHP instance of the same voice channel.

The [CAS] section of the CONFIG file is a subcomponent of the [TSC] section. Commands in the [CAS] section define the signaling types used by a CAS protocol and the [TSC] section assigns these signaling type to voice channels. For example, many CAS protocols use off-hook and wink signals, which can be defined in this section. For an explanation of the [TSC] section of the CONFIG file, see Section 1.3.11, “[TSC] Section”, on page 29.

NOTE: The CAS signaling parameters should only be modified by experienced users if the default settings do not match what the line carrier or PBX is sending or expecting for the line protocol configuration running on the card.

For information about specific CAS parameter, see the following sections:

- Section 4.9, “[CAS] Parameters for T-1 E&M Signals”, on page 131.
- Section 4.10, “[CAS] Parameters for T-1 Loop Start Signals”, on page 134.
- Section 4.11, “[CAS] Parameters for T-1 Ground Start Signals”, on page 141.
-

CAS parameters are defined using the following signal definition types:

- Transition Signal - the current signaling state changes to a new signaling state.
- Pulse Signal - the current signaling state changes to a new signaling state, and then reverts to the original signaling state.
- Train Signal - the signaling state alternates between two predefined signaling states in a regular defined pattern (series of pulses).
- Sequence Signal - the signaling state is defined by a set of train signals.

Transition Signal

The **transition** command defines an ABCD-bit transition from one state to another. It is used to define the CAS transition signals required by a protocol. The **transition** command uses the following syntax:

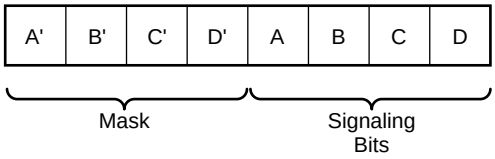
```
transition = SigId, Preval, PostVal, PreTM, PostTM
```

where,

- **SigId** is a unique identifier (parameter number) of the transition signal. The Channel Protocol (CHP) uses the **SigId** to recognize the transition when it is received, and also to generate the transition when needed.

  NOTE: **SigId** should not be modified by the user.
- **PreVal** defines the ABCD bit states on the line before the transition occurs. The four least significant bits represent the ABCD signaling bits (0 or 1). The four most significant bits represent a mask (A’B’C’D’) that specifies if each corresponding signaling bit value counts. If a mask bit is set to 1, the

corresponding signaling bit is counted. If a mask bit is set to 0, the corresponding signaling bit is ignored.



- **PostVal** defines the ABCD bit states on the line after the transition occurs. The format of this field is the same as the PreVal field.
- **PreTm** defines the minimum amount of time, in milliseconds, for the duration of the pre-transition interval.
- **PostTm** defines the minimum amount of time, in milliseconds, for the duration of the post-transition interval.

Transition Example

The following is an example of a transition command that defines a transition signal:

```
transition = 0xC15CA001, 0xF0,0xFF, 100, 300
```

In the example shown, the transition signal is defined as having the following values:

- SigId = 0xC15CA001 - Defines the CAS T-1 E&M transition signal off-hook.
- PreVal = 0xF0 (11110000) - Defines the mask as having a hexadecimal value of F (1111) and the signaling bits as having a hexadecimal value of 0 (0000). Since all of the mask bits are 1, all of the signaling bits are significant. Thus, the A, B, C, and D bits all have a value of 0 before the transition.
- PostVal = 0xFF (11111111) - Defines both the mask and the signaling bits as having a hexadecimal value of F (1111). Since all of the mask bits are 1, all of the signaling bits are significant. Thus, the A, B, C, and D bits all have a value of 1 after the transition.
- PreTm = 100 ms - Specifies that the PreVal signaling bits must be present for at least 100 milliseconds before they transition to the OnVal signaling values.
- PostTm = 300 ms - Specifies that the PostVal signaling bits must be present for at least 300 milliseconds before the signal is reported to the protocol (or if the

signal is being sent, then the CAS subcomponent ensures that the OffVal signaling value is generated for at least 300 milliseconds).

Figure 2 is a graphical representation of this signal definition.

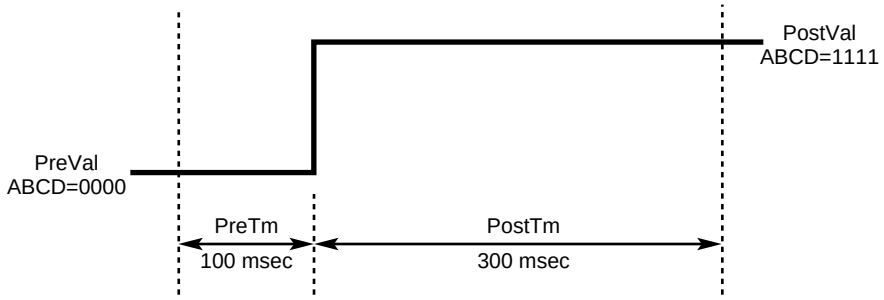


Figure 2. Example of Off-hook Transition Signal (0xC15CA001)

Pulse Signal

The **pulse** command defines an ABCD-bit transition from one state to another, and then back to the original state. It is used to define the CAS pulse signals required by a protocol. The **pulse** command uses the following syntax:

```
pulse = SigId, OffVal, OnVal, PreTm, MinTm, NomTm, MaxTm, PostTm
```

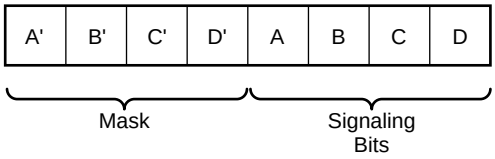
where,

- **SigId** is a unique identifier (parameter number) of the pulse signal. The Channel Protocol (CHP) uses the **SigId** to recognize the pulse when it is received, and also to generate the pulse when needed.

  NOTE: SigId should not be modified by the user.

- **OffVal** defines the ABCD bit states on the line before the transition occurs. The four least significant bits represent the ABCD signaling bits (0 or 1). The four most significant bits represent a mask (A'B'C'D') that specifies if each corresponding signaling bit value counts. If a mask bit is set to 1, the

corresponding signaling bit is counted. If a mask bit is set to 0, the corresponding signaling bit is ignored.



- **OnVal** defines the ABCD bit states on the line during the pulse. The format of this field is the same as the OffVal field.
- **PreTm** defines the minimum time, in milliseconds, for the duration of the pre-pulse interval.
- **MinTm** defines the minimum time, in milliseconds, for the duration of the pulse interval.
- **NomTm** defines the nominal time, in milliseconds, for the duration of the pulse interval.
- **MaxTm** defines the maximum time, in milliseconds, for the duration of the pulse interval.
- **PostTm** defines the minimum time, in milliseconds, for the duration of the end-of-pulse interval.

Pulse Example

The following is an example of a pulse command that defines a pulse signal:

```
pulse = 0xC15CA011, 0xF0, 0xFF, 100, 220, 250, 280, 100
```

In the example shown, the pulse signal is defined as having the following values:

- SigId = 0xC15CA011 - Defines the CAS T-1 E&M pulse signal Wink.
- OffVal = 0xF0 - Defines the mask as having a hexadecimal value of F (1111) and the signaling bits as having a hexadecimal value of 0 (0000). Since all of the mask bits are 1, all of the signaling bits are significant. Thus, the A, B, C, and D bits all have a value of 0 before the transition from the OffVal to the OnVal, and after the transition from the OnVal to the OffVal.
- OnVal = 0xFF - Defines both the mask and the signaling bits as having a hexadecimal value of F (1111). Since all of the mask bits are 1, all of the

signaling bits are significant. Thus, the A, B, C, and D bits all have a value of 1 after the transition to the OnVal.

- PreTm = 100 ms - Specifies that the OffVal signaling bits must be present for at least 100 milliseconds before they transition to the OnVal signaling values.
- MinTm = 220 ms - Specifies that the OnVal signaling bits must be present for at least 220 milliseconds before they transition to the OffVal signaling values.
- NomTm = 250 ms - Specifies that the OnVal signaling bits are generated for 250 milliseconds before transitioning to the OffVal signaling values.
- MaxTm = 280 ms - Specifies that the OnVal signaling bits must be present for no longer than 280 milliseconds before they transition to the OffVal signaling values.
- PostTm = 100 ms - Specifies that the OffVal signaling bits must be present for at least 100 milliseconds before the signal is reported to the protocol (or if the signal is being sent, then the CAS component ensures that the OffVal signaling value is generated for at least 100 milliseconds).

Figure 3 is a graphical representation of this signal definition.

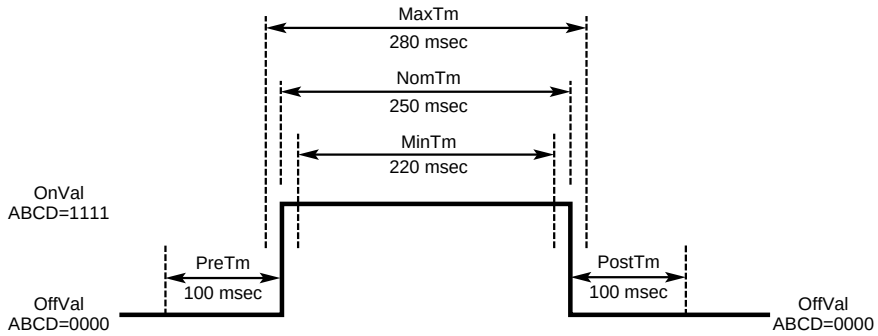


Figure 3. Example of Wink Pulse Signal (0xC15CA011)

Train Signal

The **train** command defines a set of transitions from one signaling state to another in a predefined pattern (set of pulses). It is used to define CAS signals required by a protocol. The **train** command uses the following syntax:

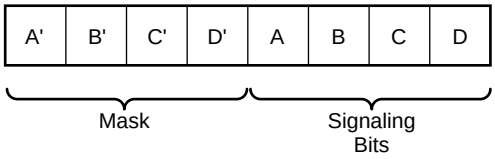
```
train = SigId, OffVal, OnVal, PulseTmlMin, PulseTmMax, PulseTmNom, preTm,
interTmMin, interTmMax, interTmNom, postTm, digitCount, pulseCount, label,
pulseCount, label, ...
```

where,

- **SigId** is a unique identifier (parameter number) of the train signal. The Channel Protocol (CHP) uses the **SigId** to recognize the train when it is received, and also to generate the train when needed.

  NOTE: SigId should not be modified by the user.

- **OffVal** defines the ABCD bit states on the line before the transition occurs. The four least significant bits represent the ABCD signaling bits (0 or 1). The four most significant bits represent a mask (A'B'C'D') that specifies if each corresponding bit value count. If a mask bit is set to 1, the corresponding signaling bit is counted. If a mask bit is set to 0, the corresponding signaling bit is ignored.



- **OnVal** defines the ABCD bit states on the line during one pulse of the train. The format of this field is the same as the OffVal field.
- **pulseTmMin** defines the minimum time, in milliseconds, for the duration of the pulse interval.
- **pulseTmMax** defines the maximum time, in milliseconds, for the duration of the pulse interval.
- **pulseTmNom** defines the nominal time, in milliseconds, for the duration of the pulse interval.
- **preTm** defines the minimum time, in milliseconds, for the duration of the pre-train interval.

- **interTmMin** defines the minimum time, in milliseconds, for the duration of the inter-pulse interval.
- **interTmMax** defines the maximum time, in milliseconds, for the duration of the inter-pulse interval.
- **interTmNom** defines the nominal time, in milliseconds, for the duration of the inter-pulse interval.
- **postTm** defines the maximum time, in milliseconds, for the duration of the post-train interval.
- **digitCount** defines the number of digit definitions in the train. The pulse count for each digit (ASCII character) is defined by the label pairs following digitCount.
- **pulseCount** defines the number of train pulses that define the digit (ASCII character) identified by the label parameter.
- **label** defines the digit (ASCII character) associated with the corresponding pulseCount value.

Train Example

The following is an example of a train command that defines a train signal:

```
train = 0xC15CA032, 0xCC, 0xC4, 31, 33, 32, 600, 62, 66, 64, 20, 12, 10, 0, 1, 1,
2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 11, #, 12, *
```

where,

- SigId = 0xC15CA032 - Defines the CAS T-1 loop start train signal parameter.
- OffVal = 0xCC - Defines both the mask and the signaling bits as having a hexadecimal value of C (1100). Since only mask bits A and B have a value of 1, only signaling bits A and B are significant. Thus, the A and B bits both have a value of 1 before the transition from the OffVal to the OnVal.
- OnVal = 0xC4 - Defines the mask as having a hexadecimal value of C (1100) and the signaling bits as having a hexadecimal value of 4 (0100). Since mask bits A and B have a value of 1, signaling bits A and B are significant. Thus, the A bit has a value of 0 and the B bit has a value of 1 after the transition to the OnVal.
- pulseTmMin = 31 - Specifies that the OnVal signaling bits must be present for at least 31 milliseconds before they transition to the OffVal signaling values.

- pulseTmMax = 33 - Specifies that the OnVal signaling bits must be present for no longer than 33 milliseconds before they transition to the OffVal signaling values.
- pulseTmNom = 32 - Specifies that the OnVal signaling bits must be present for 32 milliseconds before they transition to the OffVal signaling values.
- preTm = 600 - Specifies that the OffVal signaling bits must be present for 600 milliseconds before the train signal begins.
- interTmMin = 62 - Specifies that the OffVal signaling bits must be present for at least 62 milliseconds before they transition to the OnVal signaling values.
- interTmMax = 66 - Specifies that the OffVal signaling bits must be present for no longer than 66 milliseconds before they transition to the OnVal signaling values.
- interTmNom = 64 - Specifies that the OffVal signaling bits must be present for 64 milliseconds before they transition to the OnVal signaling values.
- postTm = 20 - Specifies that the OffVal signaling bits must be present for 20 milliseconds before the signal is reported to the protocol (or if the signal is being sent, then the CAS component ensures that the OffVal value is generated for at least 20 milliseconds).
- digitCount = 12 - Specifies that 12 digits/characters are defined by this train signal.
- pulseCount, label pairs = 10,0 1,1 2,2 3,3 4,4 5,5 6,6 7,7 8,8 9,9 11,# 12,* - The first pair indicates that 10 pulses correspond to the digit “0”, the next pair indicates that 1 pulse corresponds to the digit “1’, and the last pair indicates that 12 pulses correspond to the ASCII character “*”.

Figure 4 is a graphical representation of this signal definition.

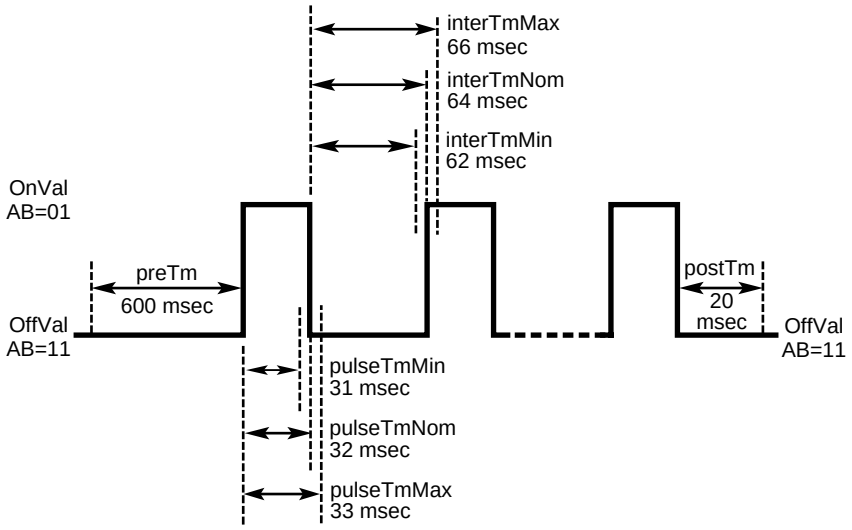


Figure 4. Example of T-1 Loop Start Train Signal (0xC15CA032)

Sequence Signal

The **sequence** command defines a set of train signals. It is used to define CAS signals required by a protocol. The **sequence** command uses the following syntax:

```
sequence = SigId, TrainSigId, preTm, interTmMin, interTmMax, interTmNom, postTm
```

where,

- **SigId** is a unique identifier (parameter number) of the sequence signal. The Channel Protocol (CHP) uses the **SigId** to recognize the sequence when it is received, and also to generate the sequence when needed.

  NOTE: SigId should not be modified by the user.
- **TrainSigId** defines the train signal that the sequence signal uses.
- **preTm** defines the minimum time, in milliseconds, for the duration of the pre-sequence interval.

- **interTmMin** defines the minimum time, in milliseconds, for the duration of the inter-train interval.
- **interTmMax** defines the maximum time, in milliseconds, for the duration of the inter-train interval.
- **interTmNom** defines the nominal time, in milliseconds, for the duration of the inter-train interval.
- **postTm** defines the minimal time, in milliseconds, for the duration of the post-sequence interval.

Sequence Example

The following is an example of a sequence command that defines a sequence signal:

```
sequence = 0xC15CA033, 0xC15CA032, 720, 640, 680, 660, 1600
```

where,

- SigId = 0xC15CA033 - Specifies the CAS T-1 loop start sequence signal parameter.
- TrainSigId = 0xC15CA032 - Specifies the train signal definition that the sequence signal uses.
- preTm = 720 - Specifies that the OffVal signaling bits (as defined in the train definition) must be present for 720 milliseconds before the sequence signal begins (that is, before the first train signal begins).
- interTmMin = 640 - Specifies that the OffVal signaling bits (as defined in the train definition) must be present for at least 640 milliseconds between train signals.
- interTmMax = 680 - Specifies that the OffVal signaling bits (as defined in the train definition) must be present for no longer than 680 milliseconds between train signals.
- interTmNom = 660 - Specifies that the OffVal signaling bits (as defined in the train definition) must be present for 660 milliseconds between train signals.
- postTm = 1600 - Specifies that the OffVal signaling bits (as defined in the train definition) must be present for 1600 milliseconds before the signal is reported to the protocol (or if the signal is being sent, then the CAS component ensures that the OffVal value is generated for at least 1600 milliseconds).

Figure 5 is a graphical representation of this signal definition.

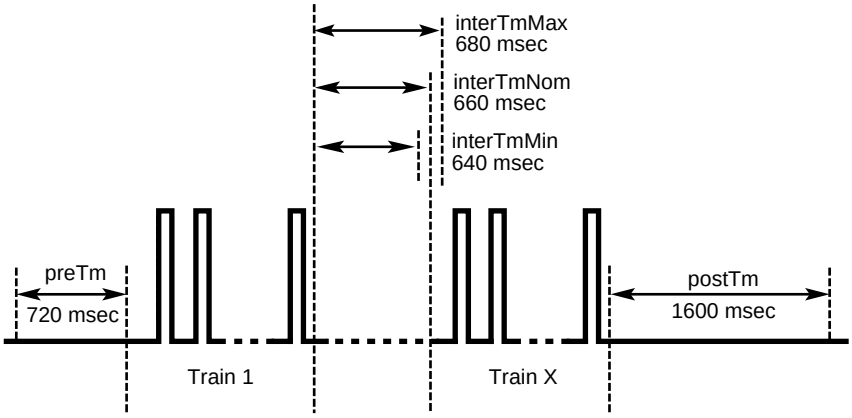


Figure 5. Example of T-1 Loop Start Sequence Signal (0xC15CA033)

1.3.9. [CCS] Section

The Common Channel Signaling (CCS) component resides on the control processor. The CCS component applies to technologies such as ISDN that use common channel signaling.

The [CCS] section of the CONFIG file is a subcomponent of the [TSC] section and contains common channel signaling parameters. Under the [CCS] section, there is typically one [CCS.x] section for each line on the board. The [CCS] section of the CONFIG file defines board based parameters and the [CCS.x] section defines the line based parameters. Each [CCS.x] section defines the CCS parameters that apply to line x, for example, parameters in the [CCS.2] section apply to line 2.

CCS parameters are modified by editing the respective lines in the [CCS] or [CCS.x] sections of the CONFIG file. For example, to change the **CCS_TMR_308** parameter (parameter number=0x0e) from a value of 4000 milliseconds to 3000 milliseconds for the fourth T-1 span, you would change the value in the [CSS.4] section from 4000 to 3000.

Following is an excerpt from the [CCS.4] section of the CONFIG file that illustrates that part of the file before and after editing:

Before editing:

```
Setparm=0x0d,4000     !Q.931 timer 305. Default=4000 msec.
Setparm=0x0e,4000     !Q.931 timer 308. Default=4000 msec.
Setparm=0x0f,10000    !Q.931 timer 310. Default=10000 msec.
```

After editing:

```
Setparm=0x0d,4000     !Q.931 timer 305. Default=4000 msec.
Setparm=0x0e,3000     !Q.931 timer 308. Default=4000 msec.
Setparm=0x0f,10000    !Q.931 timer 310. Default=10000 msec.
```

For information about each CCS parameter, see Section 4.14, “[CCS] Parameters”, on page 151.

1.3.10. [CHP] Section

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

The [CHP] section of the CONFIG file is a subset of the [TSC] section. Protocol specific parameters, primarily in the form of variants, are defined in the [CHP] section. The selection of which of these protocol variants to use on which line (span) is determined in the [TSC] section. For more information on protocol variants selection, see Section 1.3.11, “[TSC] Section”, on page 29.

A number of protocol variants are defined in the CONFIG file. Variants are defined by the `Variant Define n` command, where n is the variant identifier. The

`Variant Define n` command defines variant “n” as all of the parameter definitions in the [CHP] section preceding the command.

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

Example:

```
! T1 Protocol variant definitions
Variant VariantFormat         1     !T1 CAS format
Variant ProtocolType          1     !E&M=1,LS-FXS=2,GS-FXS=3
Variant Wink                  y
Variant Dial                  y
Variant DialFormat            1     !DTMF=1, MF, DP
Variant ANI                   0     !No=0, Pre, Post
Variant ANIFormat             1     !DTMF=1, MF, DP
Variant ANICount              0
Variant DNIS                  y
Variant DNISFormat            1     !DTMF=1, MF, DP
Variant DNISCount             0
Variant CallProgress          y
  .
  .
  .
! Define Protocol Variant 2 as T1 E&M Wink Start, DTMF Dial &
! DNIS + callProgress
Variant Define 2

! Note: Previous variant parms are kept, and the following
! commands replace the specified parameters

! Define Protocol Variant 5 as T1 Loop Start FXS, DTMF Dial
! & DNIS, callProgress, and DialTone detection.
Variant ProtocolType          2     !E&M=1,LS-FXS=2,GS-FXS=3
Variant Wink                  n
Variant Define 5

! Define Protocol Variant 4 as T1 E&M Wink Start, no tone
```

```
! and no callProgress
Variant ProtocolType          1     !E&M=1,LS-FXS=2,GS-FXS=3
Variant Wink                  y
Variant Dial                  n
Variant DNIS                  n
Variant CallProgress          n
Variant Define 4

! Define Protocol Variant 6 as Ground Start FXS
Variant ProtocolType          3     !E&M=1,LS-FXS=2,GS-FXS=3
Variant Wink                  n
Variant Dial                  y
Variant DNIS                  y
Variant CallProgress          y
Variant Define 6
```

From the [CHP] example, selecting protocol variant 2 would include all the parameter definitions from the beginning of the [CHP] section to the `Variant Define 2` line. This would define the protocol as T1 E&M Wink start, DTMF dialing, DNIS digits, with call progress (ProtocolType = 1, Wink = y, Dial = y, DialFormat = 1, DNIS = y, CallProgress = y).

If, however, you were to select protocol variant 5, this would include all of the parameter definitions from the beginning of the [CHP] section to the `Variant Define 5` line. In this case, the protocol type would change to LS-FSX (Loop start) and Wink Start would be disabled, but DTMF dialing, DNIS digits, and call progress would still be used (ProtocolType = 2, Wink = n).

If protocol variant 4 were selected, all of the parameter definitions from the beginning of the [CHP] section to the `Variant Define 4` line would be included. Now, the protocol type would change back to E&M Wink start with no DTMF dialing, no DNIS digits, and no call progress (ProtocolType = 1, Wink = y, Dial = n, DNIS = n, CallProgress = n).

You may also create your own variant if none of the existing defined protocol variants match your need. For example, to create a new protocol variant in which you want to use E&M Immediate start (instead of Wink start) with no ANI/DNIS digits provided, you may add another `Variant Define n` after the `Variant Define 2` statement. In this example, we can use n = 1 because this number has

not yet been defined in the CONFIG file. That part of the [CHP] section would then become:

```
! Define Protocol Variant 2 as T1 E&M Wink Start, DTMF Dial &
DNIS + callProgress
Variant Define 2
! Define Protocol Variant 1 as T1 E&M Immediate Start, DTMF
Dial + callProgress
Variant Wink                    n
Variant DNIS                    n
Variant Define 1
```

By disabling DNIS in protocol variant 1, which follows protocol variant 2, we have also caused DNIS to be disabled in protocol variant 5. DNIS was originally enabled in protocol variant 5 because protocol variant 5 followed protocol variant 2 which defined it as enabled. We will now need to re-enable DNIS in protocol variant 5 as shown in the following example:

```
! Define Protocol Variant 5 as T1 Loop Start FXS, DTMF Dial
! & DNIS, callProgress, and DialTone detection.
! Add DNIS back to this protocol variant
Variant DNIS                    y
Variant ProtocolType 2  ! E&M=1, LS-FXS=2, GS-FXS=3
Variant Wink n
Variant Define 5
```

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

For information about the `defineBSet` command and setting TSC parameters, see Section 1.3.11, “[TSC] Section”, on page 29.

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

- Section 4.15, “[CHP] Parameters”, on page 157.

- Section 4.16, “[CHP] T-1 Protocol Variant Definitions”, on page 159.
- Section 4.17, “[CHP] ISDN Protocol Variant Definitions”, on page 182

1.3.11. [TSC] Section

The Telephony Service Component (TSC) resides on the control processor and manages the B channel sets. B channel sets define the number of TSC-instance clusters created, as well as the line and B channels that each cluster will manage. These sets use the same inbound and outbound protocols.

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

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

To change a [TSC] parameter, you change the value of the applicable `defineBSet` parameter in the CONFIG file. For example, to change the protocol variant from 2 to 4 for both inbound and outbound call processing on all 30 channels of line 2, you would change the value of the **Inbound** and **Outbound** parameters for line 2 (**SetId**=20) from 2 to 4. For information on defining protocol variants, see Section 1.3.10, “[CHP] Section”, on page 25

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

Before editing:

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

After editing:

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

For information about each TSC parameter, see Section 4.18, “[TSC] Parameters”, on page 191.

1.3.12. [0x1b] Section

The [0x1b] section is used to set parameters relating to Echo Cancellation, Packet Loss Recovery, Volume controls to network output, and Gain controls applied to data received from the network.

NOTE: This section only applies to IPLink technologies.

[0x1b] parameters are modified by editing the respective lines in the [0x1b] section of the CONFIG file. For example, to change the **prmECNLPActive** parameter (parameter number=0x1b1b) from a value of Enable to Disable, you would change the value from 2 to 0.

Following is an excerpt from the [0x1b] section of a CONFIG file that illustrates that part of the file before and after editing.

Before editing:

```
SetParm=0x1b13,128      !prmECOrder     (48 - 128)
SetParm=0x1b16,0x20C    !prmECMu        (0x20C - 0x28F5C)
SetParm=0x1b1b,2        !prmECNLPActive (2=Enable, 0=Disable)
```

After editing:

```
SetParm=0x1b13,128      !prmECOrder     (48 - 128)
SetParm=0x1b16,0x20C    !prmECMu        (0x20C - 0x28F5C)
SetParm=0x1b1b,0        !prmECNLPActive (2=Enable, 0=Disable)
```

For information about each 0x1b parameter, see Section 4.20, “[0x1b] Parameters”, on page 202.

1.3.13. [NetTSC] Section

The Network Telephony Service Component (NetTSC) resides on the control processor and manages print level and debug level parameters.

The [NetTSC] section is used to set parameters relating to the H.323 print level and the NetTSC debug level.

NOTE: This section only applies to IPLink technologies.

[NetTSC] parameters are modified by editing the respective lines in the [NetTSC] section of the CONFIG file. For example, to change the **prmDebugLevelStream** parameter (parameter number=0x1e10) value from ERROR to Warning, you would change the value from 2 to 3.

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

Before editing:

```
SetParm=0x1e0e,2     !prmDebugLevelStack (0=OFF, 1=FATAL,
                      2=ERROR, 3=Warning, 4=Info, 5=Expand)
SetParm=0x1e0f,2     !prmDebugLevelMsg (0=OFF, 1=FATAL,
                      2=ERROR, 3=Warning, 4=Info, 5=Expand)
SetParm=0x1e10,2     !prmDebugLevelStream (0=OFF, 1=FATAL,
                      2=ERROR, 3=Warning, 4=Info, 5=Expand)
```

After editing:

```
SetParm=0x1e0e,2     !prmDebugLevelStack (0=OFF, 1=FATAL,
                      2=ERROR, 3=Warning, 4=Info, 5=Expand)
SetParm=0x1e0f,2     !prmDebugLevelMsg (0=OFF, 1=FATAL,
                      2=ERROR, 3=Warning, 4=Info, 5=Expand)
SetParm=0x1e10,3     !prmDebugLevelStream (0=OFF, 1=FATAL,
                      2=ERROR, 3=Warning, 4=Info, 5=Expand)
```

For information about each NetTSC parameter, see Section 4.21, “[NetTSC] Parameters”, on page 222.

1.3.14. [0x1d] Section

The [0x1d] section is used to set the Type of Service (TOS) byte in the IP header of transmitted datagrams in order to improve the mobility of the UDP/TCP packets

NOTE: This section only applies to IPLink technologies.

[0x1d] parameter is modified by editing the respective line in the [0x1d] section of the CONFIG file. For example, to change the **prmTOS** parameter (parameter number=0x1d01) value from 0 to 16, you would change the value from 0 to 16.

Following is an excerpt from the [0x1d] section of a CONFIG file that illustrates that part of the file before and after editing.

Before editing:

```
SetParm=0x1d01, 0   ! PrmTOS (LOWDELAY 0x10 THROUGHPUT 0x08
                    ! RELIABILITY 0x04 MINCOST 0x02
                    ! No Priority 0)
```

After editing:

```
SetParm=0x1d01, 16 ! PrmTOS (LOWDELAY 0x10 THROUGHPUT 0x08
                    ! RELIABILITY 0x04 MINCOST 0x02
                    ! No Priority 0)
```

For information about the [0x1d] parameter, see Section 4.22, “[0x1d] Parameters”, on page 241.

1.3.15. [0x1c] Section

The [0x1c] section is used to set the type of call signaling messages used on the internet when transferring call configuration to the destination end.

NOTE: This section only applies to IPLink technologies.

[0x1c] parameters are modified by editing the respective lines in the [0x1c] section of the CONFIG file. For example, to change the **prmForceSlowStart** parameter (parameter number=0x1c05) value from FastStart to SlowStart, you would change the value from 0 to1.

Following is an excerpt from the [0x1c] section of a CONFIG file that illustrates that part of the file before and after editing.

Before editing:

```
SetParm=0x1c05, 0    !prmForceSlowStart (0=FastStart-default,
                      1=SlowStart)
```

After editing:

```
SetParm=0x1c05, 1    !prmForceSlowStart (0=FastStart-default,
                      1=SlowStart)
```

The default method used for sending call signaling messages over the internet is FastStart. For IPLink CONFIG files that do not contain an [0x1c] section, you can force a SlowStart by adding the [0x1c] section to the end of the CONFIG file and setting the **prmForceSlowStart** parameter to a value of 1.

```
[0x1c]
SetParm=0x1c05, 1    !prmForceSlowStart
                     !(0=FastStart-default, 1=SlowStart)
```

For information about each [0x1c] parameter, see Section 4.23, “[0x1c] Parameters”, on page 243.

1.4. Feature Configuration Description (FCD) File

An FCD file (*.fcd*) must be downloaded to each DM3 board in the system. The purpose of the FCD file is to adjust the settings of the components that make up each product. For example, the FCD file may contain instructions to set certain country codes, or may send messages that configure the Telephony Service Provider (TSP) component to operate with a particular network protocol.

The FCD file defines a simple message form that the downloader parses and sends to a specific component. These parameters are sent to a component within a DM3 message and can be thought of as configurable *features* of a component.

To modify the FCD file parameters, the associated CONFIG file and FCDGEN utility is used. FCDGEN is a command line utility that uses the modified CONFIG file to generate an updated FCD file. After the FCD file is updated, it is downloaded

to the board. For a detailed procedure, refer to Section 2.1, “Modifying the FCD File Parameters Using FCDGEN”, on page 37.

CAUTION

The FCD file should not be edited directly. If parameters require modification, the changes are made by editing the associated CONFIG file. After editing the CONFIG file, the FCDGEN utility is used to generate a modified FCD file.

1.5. Product Configuration Description (PCD) File

A PCD file (*.pcd*) must be downloaded to each DM3 board in the system. The purpose of the PCD file is to determine the software components your system will use. It defines the product by mapping download object files to specific processors, configuring the kernel for each processor and setting the number of component instances to run on each processor.

CAUTION

The PCD file should not be modified by the user.

1.6. System-level and Additional Board-level Parameters

In addition to the parameters set by the FCD file, there are system-level and additional board-level parameters used during the configuration process. These parameters are defined in the following:

- Dialogic Configuration Manager (DCM) (Windows only)
- System Configuration Description (SCD) File (Linux only)

1.6.1. Dialogic Configuration Manager (DCM)

In a Windows environment, the DCM is a utility that can be used to define additional system and component parameters. It also is used for assigning a configuration file to a DM3 board, starting and stopping the system, and detecting the boards. For more information about DCM parameters, refer to the DCM Online Help supplied with your Windows System Release.

1.6.2. System Configuration Description (SCD) File

In a Linux environment, the SCD file (*.scd*) defines the physical parameters of a platform and includes references to the PCD and the FCD files. The parameters defined in an SCD file are product and site specific. For more information about SCD files, refer to the *System Release Installation and Configuration Guide* supplied with your Linux System Release.

1.7. System Initialization

During initialization, all required firmware for DM3 products is downloaded and configured using the identified configuration files. To initialize the system:

- In a Windows environment, start the Intel® Dialogic® System using DCM. For details about DCM, refer to the DCM Online Help supplied with the system release.
- In a Linux environment, stop and start the system.

1.8. Country Dependent Parameters (CDP) File

When using PDK protocols on DM3 boards, the protocols and CDP (*.cdp*) parameter information are downloaded to the board separately after system initialization. The PDKManager provides this functionality and up to four CDP files (one per span) can be used. For detailed downloading procedures, refer to Section 2.3, “Downloading CDP File Parameters Using the PDKManager”, on page 40.

NOTE: PDK Protocols are provided in a separately orderable GlobalCall Protocol Package. For information about the GlobalCall Protocol Package, see your Dialogic Sales representative. For information about country dependent parameters, see the *GlobalCall Country Dependent Parameter (CDP) Reference* supplied with the GlobalCall Protocol Package.

2. Configuration Procedures

This manual covers both Windows and Linux operating systems. Where these operating systems are similar, that is, the same except for syntax, Windows instructions are shown. Where these operating systems differ, special Windows Notes or Linux Notes are provided.

NOTE: Command instructions, directories paths and environment variable settings are shown relative to the Dialogic subdirectory. By default, the System Release software is installed in the following directory:

- Windows - C:\Program Files\Dialogic
- Linux - /usr/dialogic

When this manual specifies the Dialogic\Data directory it is referring to C:\Program File\Dialogic\Data in Windows and /usr/dialogic/data in Linux.

For both Windows and Linux, a number of parameters are modified by changing the Feature Configuration Description (FCD) file associated with a specific board type. In addition, when using PDK protocols, parameters are modified by changing the Country Dependent Parameter (CDP) file. The process for modifying FCD and CDP parameters includes the following procedures:

1. Modifying the FCD File Parameters Using FCDGEN
2. Initializing the System
3. Downloading CDP File Parameters Using the PDKManager (if applicable)

2.1. Modifying the FCD File Parameters Using FCDGEN

If the default settings in the FCD files are not appropriate for your configuration, you can modify the FCD file parameters using FCDGEN. For each FCD file to be modified, the procedure includes:

1. Editing the CONFIG File

2. Generating the FCD File

2.1.1. Editing the CONFIG File

In order to generate an FCD file, the corresponding CONFIG file must be updated.

NOTE: If you want to preserve the default parameter values contained in the CONFIG file, make a backup copy of the file prior to editing it.

To edit the CONFIG file:

1. From the command prompt, go to the Dialogic\Data directory and locate the CONFIG file. For a list of boards and applicable CONFIG files, see Chapter 3, "Boards and Applicable CONFIG File Sections".

2. Using a text editor (for example, WordPad in Windows or vi in Linux), open the CONFIG file that corresponds to the FCD file you want to modify. By default, the CONFIG file will have the same file name as the FCD file, but with a *.config* extension.

3. Edit the CONFIG file as necessary.

4. Save and close the CONFIG file.

2.1.2. Generating the FCD File

FCDGEN converts the CONFIG file into a format that can be read directly by the DM3 downloader. By default, the output FCD file will have the same filename as

the user-modified input configuration file, but with a *.fcd* extension (that is, if the -o option is omitted from the command).

1. From the command prompt, go the Dialogic\Data directory.

2. Execute FCDGEN as follows:

   ```
   ..\bin\fcdgen <input file>.config
   ```

   For example:

   ```
   ..\bin\fcdgen qs_t1.config
   ```

   The resulting FCD file, *qs_t1.fcd*, will reside in the Dialogic\Data directory.

2.2. Initializing the System

The newly generated FCD file will not take effect until the system is initialized.

2.2.1. Windows Environment

In a Windows environment, FCD parameters are downloaded to the board using DCM. For details about DCM, including starting and stopping the Intel® Dialogic ® System, refer to the DCM Online Help supplied with the system release.

To download parameters in a Windows environment:

1. Stop the Intel ®Dialogic® System by selecting **Stop Service** from the DCM Service pull-down menu.

2. Double-click the board model name to display the configuration data property sheets pertaining to the board.

3. Click the **Misc** property sheet.

4. Highlight the **FCDFileName** parameter.

5. Set the value to the new FCD file name if you renamed the file. Otherwise, proceed with step 6.

6. Click **Apply** to save the new value.

7. Start the Intel ®Dialogic® System by selecting **Start Service** from the DCM Service pull-down menu. The FCD parameters are downloaded once the system is started.

2.2.2. Linux Environment

In a Linux environment, the FCD parameters are downloaded to the board during the board configuration procedure. For detailed information about the board configuration procedure, refer to the *System Release for Linux Installation Guide*.

If you need to modify and re-download FCD parameters after the boards are configured, you can force a re-read of the FCD file by performing the following:

1. Stop the system by typing:
   ```
   /usr/dialogic/bin/dlstop
   ```

2. Re-start the system by typing:
   ```
   /usr/dialogic/bin/dlstart
   ```

2.3. Downloading CDP File Parameters Using the PDKManager

If you are using PDK Protocols on DM3 boards, the protocols and their parameters are downloaded to the boards after system initialization. The PDKManager utility is used to download the "hotloadable" protocol (*.hot*) file and configure the protocol as defined by the CDP (*.cdp*) file. Configuration is performed on a per board, line, or channel basis depending on the PDKManager options used.

The "hotloadable" protocol (HOT) file is the PDK protocol module that is downloaded to the board. The HOT file is the compiled object that contains the protocol logic for a particular country.

NOTE: PDK Protocols are provided in a separately orderable GlobalCall Protocol Package. For information about the GlobalCall Protocol Package, see your Dialogic Sales representative. For information about country dependent parameters, see the *GlobalCall Country Dependent Parameter (CDP) Reference* supplied with the GlobalCall Protocol Package.

PDKManager is invoked after the FCD file is downloaded to the board and the system is initialized. The procedure includes:

1. Editing the CDP File

2. Downloading the CDP and HOT Files by one of the following methods:

   2.a. Running PDKManager Manually

   2.b. Running PDKManager Automatically

2.3.1. Editing the CDP File

The CDP file can be updated from the command line using a text editor.

NOTE: If you want to preserve the default parameter values contained in the CDP file, make a backup copy of the file prior to editing it.

To edit the CDP file:

1. From the command prompt, go to the Dialogic\Cfg directory to locate the CDP file.

2. Using a text editor (for example, WordPad in Windows or vi in Linux), open the CDP file you want to modify.

3. Edit the CDP file as necessary. For more information about the CDP files, see the *GlobalCall Country Dependent Parameter (CDP) Reference* supplied with the GlobalCall Protocol Package.

4. Save and close the CDP file.

2.3.2. Downloading the CDP and HOT Files

PDKManager is used to download the PDK protocol firmware (HOT file) and the protocol parameters (CDP file) to a DM3 board.

There are two ways to run PDKManager:

- Running PDKManager Manually (from the command line)

- Running PDKManager Automatically (from DCM in Windows or dlstart in Linux)

Running PDKManager Manually

To manually download and configure the protocol using PDKManager from the command line, perform the following:

1. From the command prompt, go to the Dialogic\Cfg directory.

2. In Windows, run PDKManager by typing:

   ```
   PDKManager -board <n> [-options]
   ```

   In Linux, run PDKManager by typing:

   ```
   PDKManager -board <n> [-options]
   ```

   where <n> is the DM3 logical board identifier and is defined as follows:

   - Windows - the value of the **LogicalID** parameter as set in DCM. For details, refer to the DCM Online Help.
   - Linux - **board ID** as defined in the *pyramid.scd* file. For details, refer to the *System Release Installation and Configuration Guide* for your operating system.

   The protocol and parameter settings are downloaded to the board(s) as specified by the command. For a list of the PDKManager command line options, refer to Table 1.

Table 1. PDKManager Command Line Options

Option	Description
-board <n>	**Required**. Specifies the board(s) to which the command applies. For multiple boards, n = {n1 n2 ... nx} For example, to download and assign the *pdk_ar_r2_io* protocol to all lines on boards 1 and 3, type: `PDKManager -board {1 3} -variant pdk_ar_R2_io.cdp`

Table 1. PDKManager Command Line Options (Continued)

Option	Description
-line <n>	Specifies the T-1 or E-1 line(s) to which the command applies. If this parameter is not specified, then all lines defined by the FCD file are used. For multiple lines, n = {n1 n2 ... nx} For example, to download and assign the *pdk_ar_r2_io* protocol to lines 1 and 2 on board 1, type: `PDKManager -board 1 -line {1 2} -variant pdk_ar_R2_io.cdp`
-chan <n>	Specifies the channel(s) to which the command applies. If this parameter is not specified, then all channels defined by the FCD file are used. For multiple channels, n = {n1 n2 ... nx} For example, to download and assign the *pdk_ar_r2_io* protocol to channels 1 and 2 on line 1 on board 2, type: `PDKManager -board 2 -line 1 -channel {1 2} -variant pdk_ar_R2_io.cpd`
-pcdfile <file>	**Required** if -mlm file option is not used. Specifies the *.mlm* file by parsing the PCD file.
-fcdfile <file>	**Required** if -line and -channel options are not used. Determines line and channel configurations by parsing the FCD file.
-variant <file>	**Required** if the -load option is not used. Specifies the CDP file used. Downloads and configures the protocol on the board(s) specified, and then assigns the variant to the lines and channels.
-load <file>	**Required** if the -variant option is not used. Specifies the CDP file used. Downloads and configures the protocol to the board(s) specified, but does not assign the variant to any lines.

Table 1. PDKManager Command Line Options (Continued)

Option	Description
-mlmfile <file>	Overrides the firmware file (*.mlm*) specified in the PCD file.
-fastset	Not supported.
-help	Invokes the PDKManager help menu which displays the command options.

Running PDKManager Automatically

To automatically invoke PDKManager from DCM (Windows) or dlstart (Linux), the user must create a file called pdk.cfg. This file specifies the protocol and the parameter settings downloaded to each board.

1. From the command prompt, go to the Dialogic\Cfg directory.

2. Using a text editor (for example, WordPad in Windows or vi in Linux), create a file called pdk.cfg.

3. For each board or line to be configured, add the following entry to the file:

   ```
   board <n> [options] variant <file>
   ```

   where <n> is the DM3 logical board identifier and is defined as follows:

   - Windows - the value of the **LogicalID** parameter as set in DCM. For details, refer to the DCM Online Help.
   - Linux - **board ID** as defined in the *pyramid.scd* file. For details, refer to the *System Release for Linux Installation Guide*.

   and <file> is the CDP file to be downloaded.

4. In a Windows environment, to automatically run PDKManager whenever DCM is invoked, type the following at the command line:

   ```
   pdkmanagerregsetup add
   ```

   To stop PDKManager from automatically running when DCM is invoked, type the following at the command prompt:

   ```
   pdkmanagerregsetup remove
   ```

NOTE: In a Linux environment, dlstart will automatically invoke PDKManager if the pdk.cfg file is present in the dialogic/cfg directory. To stop PDKManager from automatically running, remove the pdk.cfg file from the dialogic/cfg directory.

For a list of PDKManager options used in the pdk.cfg file, refer to Table 2.

Table 2. PDKManager pdk.cfg File Options

Option	Description
board <n>	**Required**. Specifies the board(s) to which the command applies. For multiple boards, n = {n1 n2 ... nx} For example, to download and assign the *pdk_ar_r2_io* protocol to all lines on boards 1 and 3, type: `board {1 3} variant pdk_ar_R2_io.cdp`
line <n>	Specifies the T-1 or E-1 line(s) to which the command applies. If this parameter is not specified, then all lines defined by the FCD file are used. For multiple lines, n = {n1 n2 ... nx} For example, to download and assign the *pdk_ar_r2_io* protocol to lines 1 and 2 on board 1, type: `board 1 line {1 2} variant pdk_ar_R2_io.cdp`
chan <n>	Specifies the channel(s) to which the command applies. If this parameter is not specified, then all channels defined by the FCD file are used. For multiple channels, n = {n1 n2 ... nx} For example, to download and assign the *pdk_ar_r2_io* protocol to channels 1 and 2 on line 1 on board 2, type: `board 2 line 1 channel {1 2} variant pdk_ar_R2_io.cpd`
pcdfile <file>	**Required** if the default PCD file, qs_r2mf.pcd, is not used. Specifies the *.mlm* file by parsing the PCD file.

Table 2. PDKManager pdk.cfg File Options

Option	Description
fcdfile <file>	**Required** if the default FCD file, qs_r2mf.fcd, is not used. Determines line and channel configurations by parsing the FCD file.
variant <file>	**Required**. Specifies the CDP file used. Downloads and configures the protocol on the board(s) specified, and then assigns the variant to the lines and channels.
mlmfile <file>	Overrides the firmware file (*.mlm*) specified in the PCD file.

3. Boards and Applicable CONFIG File Sections

The purpose of this section is to help you determine which configuration (CONFIG) files and CONFIG file sections are applicable to particular DM3 board type. The DM3 boards are listed by product category and, within each product category, the boards are further listed by groups based on the CONFIG files they share. For each group of boards, a table is provided that shows which sections within the CONFIG file apply to that DM3 board, depending on the protocol selected.

DM3 Boards include the following product categories:

- IPLink Boards
- IPLink Resource Board
- Intel® Dialogic® DMV160LP
- QuadSpan Voice Boards
- DualSpan Voice Boards
- QuadSpan DTI Boards
- Voice Resource Boards
- DM3 Fax Boards
- DM3 Fax Resource Boards
- DM3 VFN Boards
- CP Fax (GDK) Boards
- CP Fax (GDK) Resource Boards
- High Density Station Interface (HDSI) Series Boards
- Dialogic Integrated Series Boards

3.1. IPLink Boards

The IPLink boards are organized according to boards that share a common set of CONFIG files and include the following groups of boards:

- DM/IP241-1T1-PCI
- DM/IP241-1T1-cPCI, DM/IP2431A-1T1-PCI
- DM/IP301-1E1-PCI
- DM/IP301-1E1-cPCI, DM/IP3031A-1E1-PCI
- DM/IP0821A-T1
- DM/IP0821A-E1-120
- DM/IP241-1T1-PCI-100BT
- DM/IP301-1E1-PCI-100BT
- DM/IP481-2T1-PCI-100BT
- DM/IP481-2T1-cPCI-100BT
- DM/IP601-2E1-PCI-100BT
- DM/IP601-2E1-cPCI-100BT

The IPLink configuration files' naming convention is as follows:

xxxx _yyy_r_#zzzzK_ml#_tttc	
xxxx	ipt = embedded call control (NetTSC) ipvs = split call control (IPVS)
yyy	evr = exportable voice resources (flexible routing)
r	If present, an "r" indicates a resource only load (no dti support)
#	If present, indicates the number of PSTN interfaces. If missing, then assume that it supports 1 PSTN interface.
zzzz	PSTN protocol
K	If present, a "k" indicates that the PSTN interface is disabled
ml#	If present, indicates a specific media load supported.
ttt	If present, indicateds the type of DSP: 307 or 311. If missing, then assume 303.

xxxx _yyy_r_#zzzzK_ml#_tttc	
c	If present, a “c” indicates that this is for a cPCI product If missing, indicates a PCI product.

3.1.1. DM/IP241-1T1-PCI

Table 3 and Table 4 list each CONFIG file for the DM/IP241-1T1-PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 3. DM/IP241-1T1-PCI CONFIG File Sections, Part 1

Applicable CONFIG File Sections	**CONFIG Files (by Protocol)**			
	ipt_evr_cas_307.config ipvs_evr_cas_307.config ipt_cas_307.config ipvs_cas_307.config	**ipt_evr_isdn_4ess_307.config ipvs_evr_isdn_4ess_307.config ipt_isdn_4ess_307.config ipvs_isdn_4ess_307.config**	**ipt_evr_isdn_5ess_307.config ipvs_evr_isdn_5ess_307.config ipt_isdn_5ess_307.config ipvs_isdn_5ess_307.config**	**ipt_evr_isdn_dms_307.config ipvs_evr_isdn_dms_307.config ipt_isdn_dms_307.config ipvs_isdn_dms_307.config**
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
NFAS		✓	✓	✓
CAS	✓			
CCS		✓	✓	✓
CHP	✓	✓	✓	✓

NOTE: See Table 4 for NI2, NTT and QSIGT1 protocols, and resource only CONFIG files.

Table 3. DM/IP241-1T1-PCI CONFIG File Sections, Part 1 (Continued)

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ipt_evr_cas_307.config ipvs_evr_cas_307.config ipt_cas_307.config ipvs_cas_307.config	**ipt_evr_isdn_4ess_307.config ipvs_evr_isdn_4ess_307.config ipt_isdn_4ess_307.config ipvs_isdn_4ess_307.config**	**ipt_evr_isdn_5ess_307.config ipvs_evr_isdn_5ess_307.config ipt_isdn_5ess_307.config ipvs_isdn_5ess_307.config**	**ipt_evr_isdn_dms_307.config ipvs_evr_isdn_dms_307.config ipt_isdn_dms_307.config ipvs_isdn_dms_307.config**
TSC	✓	✓	✓	✓
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓
0x1c	✓	✓	✓	✓

NOTE: See Table 4 for NI2, NTT and QSIGT1 protocols, and resource only CONFIG files.

Table 4. DM/IP241-1T1-PCI CONFIG File Sections, Part 2

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ipt_evr_ni2_307.config ipvs_evr_ni2_307.config ipt_ni2_307.config ipvs_ni2_307.config	**ipt_evr_isdn_ntt_307.config ipvs_evr_isdn_ntt_307.config ipt_isdn_ntt_307.config ipvs_isdn_ntt_307.config**	**ipt_evr_isdn_qsigt1_307.config ipvs_evr_isdn_qsigt1_307.config ipt_isdn_qsigt1_307.config ipvs_isdn_qsigt1_307.config**	**iptk_t1_307.config ①**
recorder	✓	✓	✓	
lineAdmin	✓	✓	✓	
CAS				
NFAS	✓	✓	✓	
CCS	✓	✓	✓	
CHP	✓	✓	✓	
TSC	✓	✓	✓	
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓
0x1c	✓	✓	✓	✓

① These files are for resource only configurations and are not associated with a protocol.
NOTE: See Table 3 for CAS, 4ESS, 5ESS, and DMS protocol CONFIG files.

3.1.2. DM/IP241-1T1-cPCI, DM/IP2431A-1T1-PCI

Table 5 and Table 6 list each CONFIG file for the IPLink boards in this group. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 5. DM/IP241-1T1-cPCI, DM/IP2431A-1T1-PCI CONFIG File Sections, Part 1

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ipt_evr_cas.config ipvs_evr_cas.config ipt_cas.config ipvs_cas.config	**ipt_evr_isdn_4ess.config ipvs_evr_isdn_4ess.config ipt_isdn_4ess.config ipvs_isdn_4ess.config**	**ipt_evr_isdn_5ess.config ipvs_evr_isdn_5ess.config ipt_isdn_5ess.config ipvs_isdn_5ess.config**	**ipt_evr_isdn_dms.config ipvs_evr_isdn_dms.config ipt_isdn_dms.config ipvs_isdn_dms.config**
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
NFAS		✓	✓	✓
CAS	✓			
CCS		✓	✓	✓
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓
0x1c	✓			

NOTE: See Table 6 for NI2, NTT, and QSIGT1 protocols, and resource only CONFIG files.

Table 6. DM/IP241-1T1-cPCI, DM/IP2431A-1T1-PCI CONFIG File Sections, Part 2

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	iipt_evr_isdn_ni2.config ipvs_evr_isdn_ni2.config ipt_isdn_ni2.config ipvs_isdn_ni2.config	**ipt_evr_isdn_ntt.config ipvs_evr_isdn_ntt.config ipt_isdn_ntt.config ipvs_isdn_ntt.config**	**ipt_evr_isdn_qsigt1.config ipvs_evr_isdn_qsigt1.config ipt_isdn_qsigt1.config ipvs_isdn_qsigt1.config**	**iptk_t1.confi g ① ipvsk_t1.config ①**
recorder	✓	✓	✓	
lineAdmin	✓	✓	✓	
CCS	✓	✓	✓	
CHP	✓	✓	✓	
TSC	✓	✓	✓	
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓
0x1c				✓

① These files only apply to the DM/IP2431A-1T1-PCI board. They are for resource only configurations and are not associated with a protocol.
NOTE: See Table 5 for CAS, 4ESS 5ESS, and DMS protocol CONFIG files.

3.1.3. DM/IP301-1E1-PCI

Table 7 lists each CONFIG file for the DM/IP301-1E1-PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 7. DM/IP301-1E1-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ipt_evr_isdn_net5_307.config ipvs_evr_isdn_net5_307.config ipt_isdn_net5_307.config ipvs_isdn_net5_307.config	**ipt_evr_isdn_qsige1_307.config ipvs_evr_isdn_qsige1_307.config ipt_isdn_qsige1_307.config ipvs_isdn_qsige1_307.config**	**ipt_evr_r2mf_307.config ipvs_evr_r2mf_307.config ipt_r2mf_307.config ipvs_r2mf_307.config**	**iptk_e1_307.config ① ipvsk_e1_307.config ①**
recorder	✓	✓	✓	
lineAdmin	✓	✓	✓	
CCS	✓	✓	✓	
CHP	✓	✓	✓	
TSC	✓	✓	✓	
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓
0x1c	✓	✓		✓

① These files are for resource only configurations and are not associated with a protocol.

3.1.4. DM/IP301-1E1-cPCI, DM/IP3031A-1E1-PCI

Table 8 lists each CONFIG file for the IPLink boards in this group. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 8. DM/IP301-1E1-cPCI, DM/IP3031A-1E1-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ipt_evr_isdn_net5.config ipvs_evr_isdn_net5.config ipt_isdn_net5.config ipvs_isdn_net5.config	**ipt_evr_isdn_qsige1.config ipvs_evr_isdn_qsige1.config ipt_isdn_qsige1.config ipvs_isdn_qsige1.config**	**ipt_evr_r2mf.config ipvs_evr_r2mf.config ipt_r2mf.config ipvs_r2mf.config**	**iptk_e1.config ① ipvsk_e1.config ①**
encoder			✓	
recorder	✓	✓	✓	
lineAdmin	✓	✓	✓	
CCS	✓	✓		
CHP	✓	✓	✓	
TSC	✓	✓	✓	
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓
0x1c				✓

① These files only apply to the DM/IP3031A-1E1-PCI board. They are for resource only configurations and are not associated with a protocol.

3.1.5. DM/IP0821A-T1

Table 9 lists each CONFIG file for the DM/IP0821A-T1 board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 9. DM/IP0821A-T1 CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)						
	ipt_cas1d.confg	**ipt_isdn_4ess1d.config**	**ipt_isdn_5ess1d.config**	**ipt_isdn_dms1d.config**	**ipt_isdn_ni21d.config**	**ipt_isdn_ntt1d.config**	**ipt_isdn_qsigt11d.config**
recorder	✓	✓	✓	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓	✓	✓	✓
NFAS		✓	✓	✓	✓	✓	✓
CAS	✓						
CCS		✓	✓	✓	✓	✓	✓
CHP	✓	✓	✓	✓	✓	✓	✓
TSC	✓	✓	✓	✓	✓	✓	✓
0x1b	✓	✓	✓	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓	✓	✓	✓
0x1c	✓	✓	✓	✓	✓	✓	✓

3.1.6. DM/IP0821A-E1-120

Table 10 lists each CONFIG file for the DM/IP0821A-E1-120 board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 10. DM/IP0821A-E1-120 CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	ipt_isdn_net51d.confg	ipt_isdn_qsige11d.confg	ipt_r2mf1d.confg
encoder			✓
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
CCS	✓	✓	
CHP	✓	✓	✓
TSC	✓	✓	✓
0x1b	✓	✓	✓
NetTSC	✓	✓	✓
0x1c	✓	✓	✓

3.1.7. DM/IP241-1T1-PCI-100BT

Table 11 and Table 12 list each CONFIG file for the DM/IP241-1T1-PCI-100BT board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 11. DM/IP241-1T1-PCI-100BT CONFIG File Sections, Part 1

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ipt_cas_311.config **ipt_evr_cas_311.config** **ipvs_cas_311.config** **ipvs_evr_cas_311.config** **ipvs_evr_cas_ml11_311.config**	**ipt_isdn_4ess_311.config** **ipt_evr_isdn_4ess_311.config** **ipvs_isdn_4ess_311.config** **ipvs_evr_isdn_4ess_311.config** **ipvs_evr_isdn_4ess_ml11_311.config**	**ipt_isdn_5ess_311.config** **ipt_evr_isdn_5ess_311.config** **ipvs_isdn_5ess_311.config** **ipvs_evr_isdn_5ess_311.config** **ipvs_evr_isdn_5ess_ml11_311.config**	**ipt_isdn_dms_311.config** **ipt_evr_isdn_dms_311.config** **ipvs_isdn_dms_311.config** **ipvs_evr_isdn_dms_311.config** **ipvs_evr_isdn_dms_ml11_311.config**
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
NFAS		✓	✓	✓
CAS	✓			
CCS		✓	✓	✓
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓
0x1c	✓			

NOTE: See Table 12 for NI2, NTT, and QSIGT1 protocol, and resource only CONFIG files.

Table 12. DM/IP241-1T1-PCI-100BT CONFIG File Sections, Part 2

Applicable CONFIG File Sections	CONFIG Files (by Protocol)				
	ipt_isdn_ni2_311.config ipt_evr_isdn_ni2_311.config ipvs_isdn_ni2_311.config ipvs_evr_isdn_ni2_311.config ipvs_evr_isdn_ni2_ml11_311.config	**ipt_isdn_ntt_311.config ipt_evr_isdn_ntt_311.config ipvs_isdn_ntt_311.config ipvs_evr_isdn_ntt_311.config ipvs_evr_isdn_ntt_ml11_311.config**	**ipt_isdn_qsigt1_311.config ipt_evr_isdn_qsigt1_311.config ipvs_isdn_qsigt1_311.config ipvs_evr_isdn_qsigt1_311.config ipvs_evr_isdn_qsigt1_ml11_311.config**	**iptk_t1_311.config ①**	**ipvsk_t1_311.config ①**
recorder	✓	✓	✓		✓
lineAdmin	✓	✓	✓		✓
NFAS	✓				
CAS					✓
CCS	✓	✓	✓		
CHP	✓	✓	✓		✓
TSC	✓	✓	✓		✓
0x1b	✓	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓	
0x1c				✓	✓

① = These files are for resource only configurations and are not associated with a protocol.

NOTE: See Table 11 for CAS, 4ESS 5ESS, and DMS protocol CONFIG files.

3.1.8. DM/IP301-1E1-PCI-100BT

Table 13 lists each CONFIG file for the DM/IP301-1E1-PCI-100BT board. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 13. DM/IP301-1E1-PCI-100BT CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ipt_isdn_net5_311.config ipt_evr_isdn_net5_311.config ipvs_isdn_net5_311.config ipvs_evr_isdn_net5_311.config	**ipt_isdn_qsige1_311.config ipt_evr_isdn_qsige1_311.config ipvs_isdn_qsige1_311.config ipvs_evr_isdn_qsige1_311.config**	**ipt_r2mf_311.config ipt_evr_r2mf_311.config ipvs_r2mf_311.config ipvs_evr_r2mf_311.config**	**iptk_e1_311.config ① ipvsk_e1_311.config ①**
encoder			✓	
recorder	✓	✓	✓	
lineAdmin	✓	✓	✓	
CCS	✓	✓		
CHP	✓	✓	✓	
TSC	✓	✓	✓	
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓
0x1c				✓

① These files are for resource only configurations and are not associated with a protocol.

3.1.9. DM/IP481-2T1-PCI-100BT

Table 14 and Table 15 list each CONFIG file for the DM/IP481-2T1-PCI-100BT board. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 14. DM/IP481-2T1-PCI-100BT CONFIG File Sections, Part 1

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ipt_2cas_311.config ipt_evr_2cas_311.config ipvs_2cas_311.config ipvs_evr_2cas_311.config	**ipt_2isdn_4ess_311.config ipt_evr_2isdn_4ess_311.config ipvs_2isdn_4ess_311.config ipvs_evr_2isdn_4ess_311.config**	**ipt_2isdn_5ess_311.config ipt_evr_2isdn_5ess_311.config ipvs_2isdn_5ess_311.config ipvs_evr_2isdn_5ess_311.config**	**ipt_2isdn_dms_311.config ipt_evr_2isdn_dms_311.config ipvs_2isdn_dms_311.config ipvs_evr_2isdn_dms_311.config**
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
NFAS		✓	✓	✓
CAS	✓			
CCS		✓	✓	✓
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓

NOTE: See Table 15 for NI2, NTT, and QSIGT1 protocol CONFIG files.

Table 15. DM/IP481-2T1-PCI-100BT CONFIG File Sections, Part 2

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	ipt_2isdn_ni2_311.config ipt_evr_2isdn_ni2_311.config ipvs_2isdn_ni2_311.config ipvs_evr_2isdn_ni2_311.config	**ipt_2isdn_ntt_311.config ipt_evr_2isdn_ntt_311.config ipvs_2isdn_ntt_311.config ipvs_evr_2isdn_ntt_311.config**	**ipt_2isdn_qsigt1_311.config ipt_evr_2isdn_qsigt1_311.config ipvs_2isdn_qsigt1_311.config ipvs_evr_2isdn_qsigt1_311.config**
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
NFAS	✓		
CCS	✓	✓	✓
CHP	✓	✓	✓
TSC	✓	✓	✓
0x1b	✓	✓	✓
NetTSC	✓	✓	✓
NOTE: See Table 14 for CAS, 4ESS 5ESS, and DMS protocol CONFIG files.			

3.1.10. DM/IP481-2T1-cPCI-100BT

Table 16 and Table 17 list each CONFIG file for the DM/IP481-2T1-cPCI-100BT board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 16. DM/IP481-2T1-cPCI-100BT CONFIG File Sections, Part 1

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ipt_2cas_311c.config ipt_evr_2cas_311c.config ipvs_2cas_311c.config ipvs_evr_2cas_311c.config	**ipt_2isdn_4ess_311c.config ipt_evr_2isdn_4ess_311c.config ipvs_2isdn_4ess_311c.config ipvs_evr_2isdn_4ess_311c.config**	**ipt_2isdn_5ess_311c.config ipt_evr_2isdn_5ess_311c.config ipvs_2isdn_5ess_311c.config ipvs_evr_2isdn_5ess_311c.config**	**ipt_2isdn_dms_311c.config ipt_evr_2isdn_dms_311c.config ipvs_2isdn_dms_311c.config ipvs_evr_2isdn_dms_311c.config**
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
NFAS		✓	✓	✓
CAS	✓			
CCS		✓	✓	✓
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓
0x1b	✓	✓	✓	✓
NetTSC	✓	✓	✓	✓

NOTE: See Table 17 for NI2, NTT, and QSIGT1 protocol CONFIG files.

Table 17. DM/IP481-2T1-cPCI-100BT CONFIG File Sections, Part 2

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	ipt_2isdn_ni2_311c.config ipt_evr_2isdn_ni2_311c.config ipvs_2isdn_ni2_311c.config ipvs_evr_2isdn_ni2_311c.config	**ipt_2isdn_ntt_311c.config ipt_evr_2isdn_ntt_311c.config ipvs_2isdn_ntt_311c.config ipvs_evr_2isdn_ntt_311c.config**	**ipt_2isdn_qsigt1_311c.config ipt_evr_2isdn_qsigt1_311c.config ipvs_2isdn_qsigt1_311c.config ipvs_evr_2isdn_qsigt1_311c.config**
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
NFAS	✓		
CCS	✓	✓	✓
CHP	✓	✓	✓
TSC	✓	✓	✓
0x1b	✓	✓	✓
NetTSC	✓	✓	✓
NOTE: See Table 16 for CAS, 4ESS, 5ESS and DMS protocol CONFIG files.			

3.1.11. DM/IP601-2E1-PCI-100BT

Table 18 lists each CONFIG file for the DM/IP601-2E1-PCI-100BT board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 18. DM/IP601-2E1-PCI-100BT CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)				
	ipt_2isdn_net5_311.config **ipt_evr_2isdn_net5_311.config** **ipvs_2isdn_net5_311.config** **ipvs_evr_2isdn_net5_311.config** **ipvs_evr_2isdn_net5_ml11_311.config**	**ipt_2isdn_qsige1_311.config** **ipt_evr_2isdn_qsige1_311.config** **ipvs_2isdn_qsige1_311.config** **ipvs_evr_2isdn_qsige1_311.config** **ipvs_evr_2isdn_qsige1_ml11_311.config**	**ipt_2r2mf_311.config** **ipt_evr_2r2mf_311.config** **ipvs_2r2mf_311.config** **ipvs_evr_2r2mf_311.config** **ipvs_evr_2r2mf_ml11_311.config**	**ipt_evr_r_311.config** ①	**ipvs_evr_r_311.config** ① **ipvs_evr_r_311_ml10.config** ① **ipvs_evr_r_ml11_311c.config** ①
encoder			✓		
recorder	✓	✓	✓	✓	✓
lineAdmin	✓	✓	✓		
CCS	✓	✓			
CHP	✓	✓	✓		
TSC	✓	✓	✓		
0x1b	✓	✓	✓	✓	✓
NetTSC	✓	✓	✓		✓
0x1c				✓	✓

① These files are for resource only configurations and are not associated with a protocol.

3.1.12. DM/IP601-2E1-cPCI-100BT

Table 19 lists each CONFIG file for the DM/IP601-2E1-cPCI-100BT board. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 19. DM/IP601-2E1-cPCI-100BT CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	ipt_2isdn_net5_311c.config **ipt_evr_2isdn_net5_311c.config** **ipvs_2isdn_net5_311c.config** **ipvs_evr_2isdn_net5_311c.config** **ipvs_evr_2isdn_net5_ml11_311c.config**	**ipt_2isdn_qsige1_311c.config** **ipt_evr_2isdn_qsige1_311c.config** **ipvs_2isdn_qsige1_311c.config** **ipvs_evr_2isdn_qsige1_311c.config** **ipvs_evr_2isdn_qsige1_ml11_311c.config**	**ipt_2r2mf_311c.config** **ipt_evr_2r2mf_311c.config** **ipvs_2r2mf_311c.config** **ipvs_evr_2r2mf_311c.config**
encoder			✓
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
CCS	✓	✓	
CHP	✓	✓	✓
TSC	✓	✓	✓
0x1b	✓	✓	✓
NetTSC	✓	✓	✓

3.2. IPLink Resource Board

The IPLink Resource board includes only the DM/IP601-cPCI-100BT board. For a explaination of the configuration file naming convention, refer to Section 3.1, “IPLink Boards”, on page 48.

3.2.1. DM/IP601-cPCI-100BT

Table 20 lists each CONFIG file for the DM/IP601-cPCI-100BT board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular resource board configuration.

Table 20. DM/IP601-cPCI-100BT CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)	
	ipt_evr_r_311c.config ipvs_ evr_r_311c.config	ipvs_ evr_r_ml11_311c.config
0x39		✓
0x3b		✓
recorder	✓	
0x1b	✓	✓
NetTSC	✓	✓
0x1c	✓	✓

3.3. Intel® Dialogic® DMV160LP

The Intel® Dialogic® DMV160LP is an analog voice board based on the DM3 architecture.

3.3.1. DMV160LP

Table 21 lists each CONFIG file for the DMV160LP board. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 21. DMV160LP CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)
	dmv160lp.config **dmv160lp_JP.config**
encoder	✓
recorder	✓
lineAdmin	✓
CAS	✓
CHP	✓
TSC	✓

3.4. QuadSpan Voice Boards

The QuadSpan voice boards are organized according to boards that share a common set of CONFIG files and include the following groups of boards:

- DM/V480-4T1-PCI, DM/V480-4T1-cPCI
- DM/V600-4E1-PCI, DM/V600-4E1-cPCI
- DM/V960-4T1-PCI, DM/V960-4T1-cPCI
- DM/V960A-4T1-PCI
- DM/V960A-4T1-cPCI
- DM/V1200-4E1-PCI, DM/V1200-4E1-cPCI

- DM/V1200A-4E1-PCI
- DM/V1200A-4E1-cPCI

3.4.1. DM/V480-4T1-PCI, DM/V480-4T1-cPCI

Table 22 lists each CONFIG file for the QuadSpan voice boards in this group. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files fall into the following categories:

- Files that provide media loads and support flexible routing (ml*x* preface)
- Files that do not provide media loads and do not support flexible routing (4xt prefix)

For more information on media loads and flexible routing, see Section 1.2, "Media Loads and Routing Configurations", on page 3.

Table 22. DM/V480-4T1-PCI, DM/V480-4T1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)						
	4x2_cas.config ① 4x2_t1.config ml1_4x2_cas.config ① ml1_4x2_t1.config	4x2_isdn_4ess.config ml1_4x2_4ess.config	4x2_isdn_5ess.config ml1_4x2_5ess.config	4x2_isdn_dms.config ml1_4x2_dms.config	4x2_isdn_ni2.config ml1_4x2_ni2.config	4x2_isdn_ntt.config ml1_4x2_ntt.config	4x2_isdn_qsigt1.config ml1_4x2_qsigt1.config
encoder	✓	✓	✓	✓	✓	✓	✓
recorder	✓	✓	✓	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓	✓	✓	✓
NFAS		✓	✓	✓	✓		

① The CAS section does not apply to this file.

Table 22. DM/V480-4T1-PCI, DM/V480-4T1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)						
	4x2_cas.config ① 4x2_t1.config ml1_4x2_cas.config ① ml1_4x2_t1.config	4x2_isdn_4ess.config ml1_4x2_4ess.config	4x2_isdn_5ess.config ml1_4x2_5ess.config	4x2_isdn_dms.config ml1_4x2_dms.config	4x2_isdn_ni2.config ml1_4x2_ni2.config	4x2_isdn_ntt.config ml1_4x2_ntt.config	4x2_isdn_qsigt1.config ml1_4x2_qsigt1.config
CAS	✓						
CCS		✓	✓	✓	✓	✓	✓
CHP	✓	✓	✓	✓	✓	✓	✓
TSC	✓	✓	✓	✓	✓	✓	✓

① The CAS section does not apply to this file.

3.4.2. DM/V600-4E1-PCI, DM/V600-4E1-cPCI

Table 23 lists each CONFIG file for the QuadSpan voice boards in this group. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files fall into the following categories:

- Files that provide media loads and support flexible routing (ml*x* preface)
- Files that do not provide media loads and do not support flexible routing (4x2 prefix)

For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 23. DM/V600-4E1-PCI, DM/V600-4E1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	4x2_isdn_net5.config ml1_4x2_net5.config	4x2_isdn_qsige1.config ml1_4x2_qsige1.config	4x2_r2mf.config ml1_4x2_r2mf.config
encoder	✓	✓	✓
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
CCS	✓	✓	
CHP	✓	✓	
TSC	✓	✓	✓

3.4.3. DM/V960-4T1-PCI, DM/V960-4T1-cPCI

Table 24 lists each CONFIG file for the QuadSpan voice boards in this group. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files fall into the following categories:

- Files that provide media loads and support flexible routing (ml*x* preface)
- Files that do not provide media loads and do not support flexible routing (qs prefix)

For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 24. DM/V960-4T1-PCI, DM/V960-4T1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)						
	qs_cas.config ① qs_t1.config ml1_qs_cas.config ① ml1_qs_t1.config	qs_isdn_4ess.config ml1_qs_4ess.config	qs_isdn_5ess.config ml1_qs_5ess.config	qs_isdn_dms.config ml1_qs_dms.config	qs_isdn_ni2.config ml1_qs_ni2.config	qs_isdn_ntt.config ml1_qs_ntt.config	qs_isdn_qsigt1.config ml1_qs_qsigt1.config
encoder	✓	✓	✓	✓	✓	✓	✓
recorder	✓	✓	✓	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓	✓	✓	✓
NFAS		✓	✓	✓	✓		
CAS	✓						
CCS		✓	✓	✓	✓	✓	✓
CHP	✓	✓	✓	✓	✓	✓	✓
TSC	✓	✓	✓	✓	✓	✓	✓

① The CAS section does not apply to this file.

3.4.4. DM/V960A-4T1-PCI

Table 25 and Table 26 list each CONFIG file for the DM/V960A-4T1-PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 25. DM/V960A-4T1-PCI CONFIG File Sections, Part 1

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	ml1b_qsa_cas.config ① ml1b_qsa_t1.config ml2_qsa_cas.config ① ml2_qsa_t1.config ml5_qsa_cas.config ① ml6_qsa_cas.config ① ml6_qsa_t1.config ml9b_qsa_t1.config	**ml1b_qsa_4ess.config ml2_qsa_4ess.config ml5_qsa_4ess.config ml6_qsa_4ess.config**	**ml1b_qsa_5ess.config ml2_qsa_5ess.config ml5_qsa_5ess.config ml6_qsa_5ess.config**
encoder	✓	✓	✓
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
NFAS		✓	✓
CAS	✓		
CCS		✓	✓
CHP	✓	✓	✓
TSC	✓	✓	✓

① The CAS section does not apply to this file.
NOTE: See Table 26 for NI2, NTT, and QSIGT1 protocol CONFIG files.

Table 26. DM/V960A-4T1-PCI CONFIG File Sections, Part 2

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ml1b_qsa_dms.config ml2_qsa_dms.config ml5_qsa_dms.config ml6_qsa_dms.config	**ml1b_qsa_ni2.config ml2_qsa_ni2.config ml5_qsa_ni2.config ml6_qsa_ni2.config**	**ml1b_qsa_ntt.config ml2_qsa_ntt.config ml5_qsa_ntt.config ml6_qsa_ntt.config**	**ml1b_qsa_qsigt1.config ml2_qsa_qsigt1.config ml5_qsa_qsigt1.config ml6_qsa_qsigt1.config**
encoder	✓	✓	✓	✓
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
NFAS	✓	✓		
CAS				
CCS	✓	✓	✓	✓
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓

NOTE: See Table 25 for CAS, 4ESS, 5ESS, and DMS protocol CONFIG files.

3.4.5. DM/V960A-4T1-cPCI

Table 27 and Table 28 list each CONFIG file for the DM/V960A-4T1-cPCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files fall into the following categories:

- Files that provide media loads and support flexible routing (ml*x* preface)

- Files that do not provide media loads and do not support flexible routing (qs2 prefix)

For more information on media loads and flexible routing, see Section 1.2, "Media Loads and Routing Configurations", on page 3.

Table 27. DM/V960A-4T1-cPCI CONFIG File Sections, Part 1

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	qs2_cas_.config ① qs2_t1.config ml1b_qs2_cas.cobfig ① ml1b_qs2_t1.config ml2_qs2_cas.config ① ml2_qs2_t1.config ml9b_qs2_t1.config	qs2_isdn_4ess.config ml1b_qs2_4ess.config ml2_qs2_4ess.config	qs2_isdn_5ess.config ml1b_qs2_5ess.config ml2_qs2_5ess.config	qs2_isdn_dms.config ml1b_qs2_dms.config ml2_qs2_dms.config
encoder	✓	✓	✓	✓
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
NFAS		✓	✓	✓
CAS	✓			
CCS		✓	✓	✓
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓

① The CAS section does not apply to this file.

NOTE: See Table 28 for NI2, NTT, and QSIGT1 protocol CONFIG files.

Table 28. DM/V960A-4T1-cPCI CONFIG File Sections, Part 2

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	qs2_isdn_ni2.config ml1b_qs2_ni2.config ml2_qs2_ni2.config	**qs2_isdn_ntt.config ml1b_qs2_ntt.config ml2_qs2_ntt.config**	**qs2_isdn_qsigt1.config ml1b_qs2_qsigt1.config ml2_qs2_qsigt1.config**
encoder	✓	✓	✓
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
NFAS	✓		
CAS			
CCS	✓	✓	✓
CHP	✓	✓	✓
TSC	✓	✓	✓

NOTE: See Table 27 for CAS, 4ESS, 5ESS, and DMS protocol CONFIG files.

3.4.6. DM/V1200-4E1-PCI, DM/V1200-4E1-cPCI

Table 29 lists each CONFIG file for the QuadSpan voice boards in this group. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files fall into the following categories:

- Files that provide media loads and support flexible routing (ml*x* preface)
- Files that do not provide media loads and do not support flexible routing (qs prefix)

For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 29. DM/V1200-4E1-PCI, DM/V1200-4T1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	qs_isdn_net5.config ml1_qs_net5.config	qs_isdn_qsige1.config ml1_qs_qsige1.config	qs_r2mf.config ml1_qs_r2mf.config	ml1_qs_ts16.config
encoder	✓	✓	✓	✓
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
CCS	✓	✓		
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓

3.4.7. DM/V1200A-4E1-PCI

Table 30 lists each CONFIG file for the DM/V1200A-4E1-PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 30. DM/V1200A-4E1-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ml1b_qsa_net5.config ml2_qsa_net5.config ml6_qsa_net5.config ul1_qsa_net5.config	**ml1b_qsa_qsige1.config ml2_qsa_qsige1.config ml6_qsa_qsige1.config ul1_qsa_qsige1.config**	**ml1b_qsa_r2mf.config ml2_qsa_r2mf.config ml6_qsa_r2mf.config**	**ml2_qsa_ts16.config ml9b_qsa_e1.config**
encoder	✓	✓	✓	✓
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
CCS	✓	✓		
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓

3.4.8. DM/V1200A-4E1-cPCI

Table 31 lists each CONFIG file for the DM/V1200A-4E1-cPCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, "Media Loads and Routing Configurations", on page 3.

Table 31. DM/V1200A-4E1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	qs2_isdn_net5.config ml1b_isdn_net5.config ml2_qs2_net5.config	qs2_isdn_qsige1.config ml1b_isdn_qsige1.config ml2_qs2_qsige1.config	qs2_r2mf.config ml1b_isdn_r2mf.config ml2_qs2_r2mf.config	ml2_qs2_ts16.config ml9b_qs2_e1.config
encoder	✓	✓	✓	✓
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
CAS			✓	
CCS	✓	✓		
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓

3.5. DualSpan Voice Boards

The DualSpan boards include the following groups:

- DM/V480A-2T1-PCI
- DM/V480A-2T1-cPCI
- DM/V600A-2E1-PCI
- DM/V600A-2E1-cPCI

- DM/V480-2T1-PCI-HiZ
- DM/V600-2E1-PCI-HiZ

3.5.1. DM/V480A-2T1-PCI

Table 32 and Table 33 list each CONFIG file for the DM/V480A-2T1-PCI board. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, "Media Loads and Routing Configurations", on page 3.

Table 32. DM/V480A-2T1-PCI CONFIG File Sections, Part 1

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	ml1b_dsa_cas.config ① **ml2_dsa_cas.config** ① **ml2_dsa_t1.config** **ml6_dsa_cas.config** ① **ml6_dsa_t1.config** **ml10_dsa_cas.config** ① **ml10_dsa_t1.config**	**ml1b_dsa_4ess.config** **ml2_dsa_4ess.config** **ml6_dsa_4ess.config** **ml10_dsa_4ess.config**	**ml1b_dsa_5ess.config** **ml2_dsa_5ess.config** **ml6_dsa_5ess.config** **ml10_dsa_5ess.config**
0x3b ②	✓	✓	✓
encoder	✓	✓	✓
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
NFAS		✓	✓

① The CAS section does not apply to this file.

② This section only applies to the ml10 CONFIG files.

NOTE: See Table 28 for DMS, NI2, NTT, and QSIGT1 protocol CONFIG files.

Table 32. DM/V480A-2T1-PCI CONFIG File Sections, Part 1

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	ml1b_dsa_cas.config ① ml2_dsa_cas.config ① ml2_dsa_t1.config ml6_dsa_cas.config ① ml6_dsa_t1.config ml10_dsa_cas.config ① ml10_dsa_t1.config	**ml1b_dsa_4ess.config ml2_dsa_4ess.config ml6_dsa_4ess.config ml10_dsa_4ess.config**	**ml1b_dsa_5ess.config ml2_dsa_5ess.config ml6_dsa_5ess.config ml10_dsa_5ess.config**
CAS	✓		
CCS		✓	✓
CHP	✓	✓	✓
TSC	✓	✓	✓

① The CAS section does not apply to this file.

② This section only applies to the ml10 CONFIG files.

NOTE: See Table 28 for DMS, NI2, NTT, and QSIGT1 protocol CONFIG files.

Table 33. DM/V480A-2T1-PCI CONFIG File Sections, Part 2

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ml1b_dsa_dms.config ml2_dsa_dms.config ml6_dsa_dms.config ml10_dsa_dms.config	**ml1b_dsa_ni2.config ml2_dsa_ni2.config ml6_dsa_ni2.config ml10_dsa_ni2.config**	**ml1b_dsa_ntt.config ml2_dsa_ntt.config ml6_dsa_ntt.config ml10_dsa_ntt.config**	**ml1b_dsa_qsigt1.config ml2_dsa_qsigt1.config ml6_dsa_qsigt1.config ml10_dsa_qsigt1.config**
0x3b ①	✓	✓	✓	✓
encoder	✓	✓	✓	✓
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
NFAS	✓	✓		
CAS				
CCS	✓	✓	✓	✓
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓

① This section only applies to the ml10 CONFIG files.

3.5.2. DM/V480A-2T1-cPCI

Table 34 and Table 35 list each CONFIG file for the DM/V480A-2T1-cPCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, "Media Loads and Routing Configurations", on page 3.

Table 34. DM/V480A-2T1-cPCI CONFIG File Sections, Part 1

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ml1b_ds2_cas.config ① ml2_ds2_cas.config ① ml2_ds_t1.config ml10_ds_t1.config	**ml1b_ds_4ess.config ml2_ds_4ess.config ml10_ds_4ess.config**	**ml1b_ds_5ess.config ml2_ds_5ess.config ml10_ds_5ess.config**	**ml1b_ds_dms.config ml2_ds_dms.config ml10_ds_dms.config**
encoder	✓	✓	✓	✓
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
NFAS		✓	✓	✓
CAS	✓			
CCS		✓	✓	✓
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓

① The CAS section does not apply to this file.
NOTE: See Table 35 for NI2, NTT, and QSIGT1 protocol CONFIG files.

Table 35. DM/V480A-2T1-cPCI CONFIG File Sections, Part 2

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	ml1b_ds_ni2.config ml2_ds_ni2.config ml10_ds_ni2.config	**ml1b_ds_ntt.config ml2_ds_ntt.config ml10_ds_ntt.config**	**ml1b_ds_qsigt1.config ml2_ds_qsigt1.config ml10_ds_qsigt1.config**
encoder	✓	✓	✓
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
NFAS	✓		
CAS			
CCS	✓	✓	✓
CHP	✓	✓	✓
TSC	✓	✓	✓
NOTE: See Table 34 for CAS, 4ESS, 5ESS and DMS protocol CONFIG files.			

3.5.3. DM/V600A-2E1-PCI

Table 36 lists each CONFIG file for the DM/V600A-2E1-PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 36. DM/V600A-2E1-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ml1b_dsa_net5.config ml2_dsa_net5.config ml6_dsa_net5.config ml10_dsa_net5.config	**ml1b_dsa_qsige1.config ml2_dsa_qsige1.config ml6_dsa_qsige1.config ml10_dsa_qsige1.config**	**ml1b_dsa_r2mf.config ml2_dsa_r2mf.config ml6_dsa_r2mf.config ml10_dsa_r2mf.confi**	**ml2_dsa_ts16.config**
0x3b ①	✓	✓	✓	✓
encoder	✓	✓	✓	✓
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
CAS				✓
CCS	✓	✓		
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓

① This section only applies to the ml2 CONFIG files.

3.5.4. DM/V600A-2E1-cPCI

Table 37 lists each CONFIG file for the DM/V600A-2E1-cPCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 37. DM/V600A-2E1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)			
	ml1b_ds2_net5.config ml2_ds2_net5.config ml10_ds2_net5.config	**ml1b_ds2_qsige1.config ml2_ds2_qsige1.config ml10_ds2_qsige1.config**	**ml1b_ds2_r2mf.config ml2_ds2_r2mf.config ml10_ds2_r2mf.confi**	**ml2_ds2_ts16.config**
0x3b ①	✓	✓	✓	✓
encoder	✓	✓	✓	✓
recorder	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓
CAS				✓
CCS	✓	✓		
CHP	✓	✓	✓	✓
TSC	✓	✓	✓	✓

① This section only applies to the ml2 CONFIG files.

3.5.5. DM/V480-2T1-PCI-HiZ

Table 38 lists each CONFIG file for the DM/V480-2T1-PCI-HiZ board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 38. DM/V480-2T1-PCI-HiZ CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)
	ml2_hiz_isdnt1.config ml2_hiz_t1lite.config
encoder	✓
recorder	✓
lineAdmin	✓
CCS	✓
CHP	✓
TSC	✓

3.5.6. DM/V600-2E1-PCI-HiZ

Table 39 lists each CONFIG file for the DM/V600-2E1-PCI-HiZ board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

This CONFIG file provides media load and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 39. DM/V600-2E1-PCI-HiZ CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)
	ml2_hiz_isdne1.config
encoder	✓
recorder	✓
lineAdmin	✓

Table 39. DM/V600-2E1-PCI-HiZ CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)
	ml2_hiz_isdne1.config
CCS	✓
CHP	✓
TSC	✓

3.6. QuadSpan DTI Boards

The QuadSpan DTI boards include the following groups:

- DM/T960-4T1-PCI, DM/T960-4T1-cPCI
- DM/T1200-4E1-PCI, DM/T1200-4E1-cPCI
- DM/N960-4T1-PCI, DM/N960-4T1-cPCI
- DM/N1200-4E1-PCI

3.6.1. DM/T960-4T1-PCI, DM/T960-4T1-cPCI

Table 40 lists each CONFIG file for the QuadSpan DTI boards in this group. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 40. DM/T960-4T1-PCI, DM/T960-4T1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)						
	4xt_cas.config ① 4xt_t1.config mn_4xt_cas.config ① mn_4xt_t1.config	4xt_isdn_4ess.config mn_4xt_4ess.config	4xt_isdn_5ess.config mn_4xt_5ess.config	4xt_isdn_dms.config mn_4xt_dms.config	4xt_isdn_ni2.config mn_4xt_ni2.config	4xt_isdn_ntt.config mn_4xt_ntt.config	4xt_isdn_qsigt1.config mn_4xt_qsigt1.config
encoder	✓	✓	✓	✓	✓	✓	✓
recorder	✓	✓	✓	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓	✓	✓	✓
NFAS		✓	✓	✓	✓		
CAS	✓						
CCS		✓	✓	✓	✓	✓	✓
CHP	✓	✓	✓	✓	✓	✓	✓
TSC	✓	✓	✓	✓	✓	✓	✓

① The CAS section does not apply to this file.

3.6.2. DM/T1200-4E1-PCI, DM/T1200-4E1-cPCI

Table 41 lists each CONFIG file for the QuadSpan DTI boards in this group. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 41. DM/T1200-4E1-PCI, DM/T1200-4E1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	4xt_isdn_net5.config mn_4xt_net5.config	**4xt_isdn_qsige1.config mn_4xt_qsige1.config**	**4xt_r2mf.config mn_4xt_r2mf.config**
encoder	✓	✓	✓
lineAdmin	✓	✓	✓
CCS	✓	✓	
CHP	✓	✓	✓
TSC	✓	✓	✓

3.6.3. DM/N960-4T1-PCI, DM/N960-4T1-cPCI

Table 42 lists each CONFIG file for the QuadSpan DTI boards in this group. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 42. DM/N960-4T1-PCI, DM/N960-4T1-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)					
	4x0_isdn_4ess.config	4x0_isdn_5ess.config	4x0_isdn_dms.config	4x0_isdn_ni2.config	4x0_isdn_ntt.config	4x0_isdn_qsigt1.config
encoder	✓	✓	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓	✓	✓
NFAS	✓	✓	✓	✓		
CCS	✓	✓	✓	✓	✓	✓
CHP	✓	✓	✓	✓	✓	✓
TSC	✓	✓	✓	✓	✓	✓

3.6.4. DM/N1200-4E1-PCI

Table 43 lists each CONFIG file for the DM/N1200-4E1-PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 43. DM/N1200-4E1-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)	
	4xt_isdn_net5.config	4xt_isdn_qsige1.config
encoder	✓	✓
lineAdmin	✓	✓
CCS	✓	✓
CHP	✓	✓
TSC	✓	✓

3.7. Voice Resource Boards

The DM3 Voice Resource boards include the following group:

- DM/V2400A-PCI
- DM/V2400A-cPCI

3.7.1. DM/V2400A-PCI

Table 44 lists each CONFIG file for the DM/V2400A-PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, “Media Loads and Routing Configurations”, on page 3.

Table 44. DM/V2400A-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol) ml1_pcires.config ml9b_pcires.config ml10_pcires.config
0x3b ①	✓
recorder ②	✓

① This section does not apply to the ml1 CONFIG file.
② This section does not apply to the ml9b CONFIG files.

3.7.2. DM/V2400A-cPCI

Table 45 lists each CONFIG file for the DM/V2400A-cPCI board. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

These CONFIG files provide media loads and support flexible routing (ml*x* preface) only. For more information on media loads and flexible routing, see Section 1.2, "Media Loads and Routing Configurations", on page 3.

Table 45. DM/V2400A-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol) ml1_cpcires.config ml9b_cpcires.config ml10_cpcires.config
0x3b ①	✓
recorder ②	✓

① This section does not apply to the ml1 CONFIG file.
② This section does not apply to the ml9b CONFIG files.

3.8. DM3 Fax Boards

The DM3 Fax boards include the following groups:

- DM/F240-1T1-PCI
- DM/F300-1E1-PCI

3.8.1. DM/F240-1T1-PCI

Table 46 lists each CONFIG file for the DM/F240-1T1-PCI board. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 46. DM/F240-1T1-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)						
	fn_t1.config fn3_t1.config	**fn_isdn_4ess.config fn3_isdn_4ess.config**	**fn_isdn_5ess.config fn3_isdn_5ess.config**	**fn_isdn_dms.config fn3_isdn_dms.config**	**fn_isdn_ni2.config fn3_isdn_ni2.config**	**fn_isdn_ntt.config fn3_isdn_ntt.config**	**fn_isdn_qsigt1.config fn3_isdn_qsigt1.config**
encoder	✓	✓	✓	✓	✓	✓	✓
recorder	✓	✓	✓	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓	✓	✓	✓
CAS	✓						
CCS		✓	✓	✓	✓	✓	✓
CHP	✓	✓	✓	✓	✓	✓	✓
TSC	✓	✓	✓	✓	✓	✓	✓

3.8.2. DM/F300-1E1-PCI

Table 47 lists each CONFIG file for the DM/F300-1E1-PCI board. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 47. DM/F300-1E1-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	fn_isdn_net5.config fn3_isdn_net5.config	**fn_isdn_qsige1.config fn3_isdn_qsige1.config**	**fn_isdn_r2mf.config fn3_isdn_r2mf.config**
encoder	✓	✓	✓
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
CCS	✓	✓	✓
CHP	✓	✓	✓
TSC	✓	✓	✓

3.9. DM3 Fax Resource Boards

The DM3 Fax Resource boards include the following groups:

- DM/F240-PCI and DM/F240-cPCI
- DM/F300-PCI and DM/F300-cPCI

3.9.1. DM/F240-PCI and DM/F240-cPCI

There are no CONFIG files associated with the DM3 Fax Resource boards in this group.

NOTE: The PCD and FCD files associated with the DM3 Fax Resource boards in this group are *fax24.pcd* and *fax24.fcd*.

3.9.2. DM/F300-PCI and DM/F300-cPCI

There are no CONFIG files associated with the DM3 Fax Resource boards in this group.

NOTE: The PCD and FCD files associated with the DM3 Fax Resource boards in this group are *fax30.pcd* and *fax30.fcd*.

3.10. DM3 VFN Boards

The DM3 VFN boards include the following groups:

- DM/VF240-1T1-PCI
- DM/VF300-1E1-PCI

3.10.1. DM/VF240-1T1-PCI

Table 48 lists each CONFIG file for the DM/F240-1T1-PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 48. DM/VF240-1T1-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)						
	vfn_t1.config vfn3_t1.config	**vfn_isdn_4ess.config vfn3_isdn_4ess.config**	**vfn_isdn_5ess.config vfn3_isdn_5ess.config**	**vfn_isdn_dms.config vfn3_isdn_dms.config**	**vfn_isdn_ni2.config vfn3_isdn_ni2.config**	**vfn_isdn_ntt.config vfn3_isdn_ntt.config**	**vfn_isdn_qsigt1.config vfn3_isdn_qsigt1.config**
encoder	✓	✓	✓	✓	✓	✓	✓
recorder	✓	✓	✓	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓	✓	✓	✓
CAS	✓						
CCS		✓	✓	✓	✓	✓	✓
CHP	✓	✓	✓	✓	✓	✓	✓
TSC	✓	✓	✓	✓	✓	✓	✓

3.10.2. DM/VF300-1E1-PCI

Table 49 lists each CONFIG file for the DM/F300-1E1-PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 49. DM/VF300-1E1-PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)		
	vfn_isdn_net5.config vfn3_isdn_net5.config	**fvn_isdn_qsige1.config vfn3_isdn_qsige1.config**	**vfn_isdn_r2mf.config vfn3_isdn_r2mf.config**
encoder	✓	✓	✓
recorder	✓	✓	✓
lineAdmin	✓	✓	✓
CCS	✓	✓	✓
CHP	✓	✓	✓
TSC	✓	✓	✓

3.11. CP Fax (GDK) Boards

The CP Fax (GDK) boards include the following groups:

- CPi/2400CT-T1
- CPi/3000CT-E1
- CPi/400 BRI PCI

3.11.1. CPi/2400CT-T1

Table 50 lists each CONFIG file for the CPi/2400CT-T1 (GDK) board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 50. CPi/2400CT-T1 CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)				
	gdk_t1_em.config gdk_t1_gs.config gdk_t1_ls.config	gdk_isdn_4ess.config	gdk_isdn_5ess.config	gdk_isdn_dms.config	gdk_isdn_ntt.config
encoder	✓	✓	✓	✓	✓
recorder	✓	✓	✓	✓	✓
lineAdmin	✓	✓	✓	✓	✓
CAS	✓				
CCS		✓	✓	✓	✓
CHP	✓	✓	✓	✓	✓
TSC	✓	✓	✓	✓	✓

3.11.2. CPi/3000CT-E1

Table 51 lists each CONFIG file for the CPi/3000CT-E1 (GDK) board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 51. CPi/3000CT-E1 CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)
	gdk_isdn_net5.config
encoder	✓

Table 51. CPi/3000CT-E1 CONFIG File Sections (Continued)

Applicable CONFIG File Sections	CONFIG Files (by Protocol)
	gdk_isdn_net5.config
recorder	✓
lineAdmin	✓
CCS	✓
CHP	✓
TSC	✓

3.11.3. CPi/400 BRI PCI

Table 51 lists each CONFIG file for the CPi/400 BRI PCI board. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 52. CPi/400 BRI PCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol)
	cpi400bripcipmp.config **cpi400bripcipp.config**
TSC	✓

3.12. CP Fax (GDK) Resource Boards

The CP Fax (GDK) Resource boards include the following groups:

- CPi/2400CT
- CPi/3000CT

3.12.1. CPi/2400CT

There are no CONFIG files associated with the CPi/2400CT (GDK) board.

NOTE: The PCD and FCD files associated with the CPi/2400CT (GDK) board are *fax24.pcd* and *fax24.fcd*.

3.12.2. CPi/3000CT

There are no CONFIG files associated with the CPi/3000CT (GDK) board.

NOTE: The PCD and FCD files associated with the CPi/3000CT (GDK) board are *fax30.pcd* and *fax30.fcd*.

3.13. High Density Station Interface (HDSI) Series Boards

The High Density Station Interface (HDSI) Series board includes the following boards:

- HDSI/480-PCI and HDSI/480-cPCI
- HDSI/720-PCI and HDSI/720-cPCI
- HDSI/960-PCI and HDSI/960-cPCI

High Density Station Interface board CONFIG files are prefaced with a country code. This code represents the country-specific protocol that is supported. Refer to Table 53 for a list of country codes for each supported country.

Table 53. Country Codes

Code	Country
at	Austria
au	Australia
be	Belgium
ch	Switzerland
de	Germany

Table 53. Country Codes (Continued)

Code	Country
dk	Denmark
es	Spain
fr	France
gb	United Kingdom
hk	Hong Kong
ie	Ireland
it	Italy
jp	Japan
lu	Luxembourg
mx	Mexico
my	Malaysia
nl	Netherlands
no	Norway
nz	New Zealand
pt	Portugal
sg	Singapore
us	United States
za	South Africa

3.13.1. HDSI/480-PCI and HDSI/480-cPCI

Table 54 lists each CONFIG file for the HDSI/480-PCI and HDSI/480-cPCI boards. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 54. HDSI/480-PCI and HDSI/480-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol) ①
	***_hdsi.config** ***_hdsi_48_play_rec.config**
CAS	✓
CHP	✓
TSC	✓

① HDSI config files are prefaced with a country code. For example, the HDSI CONFIG file for Australia E-1 (code = at) is at_hdsi_48_play_rec.config. For a list of country codes see Table 53.

3.13.2. HDSI/720-PCI and HDSI/720-cPCI

Table 54 lists each CONFIG file for the HDSI/720-PCI and HDSI/720-cPCI boards. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 55. HDSI/720-PCI and HDSI/720-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol) ①
	***_hdsi.config** ***_hdsi_48_play_rec.config** ***_hdsi_72_play_rec.config**
CAS	✓
CHP	✓
TSC	✓

Table 55. HDSI/720-PCI and HDSI/720-cPCI CONFIG File Sections (Continued)

Applicable CONFIG File Sections	CONFIG Files (by Protocol) ①
	***_hdsi.config** ***_hdsi_48_play_rec.config** ***_hdsi_72_play_rec.config**

① HDSI config files are prefaced with a country code. For example, the HDSI CONFIG files for Australia E-1 (code = at) are and *at_hdsi_48_play_rec.config* and *at_hdsi_72_play_rec.config*. For a list of country codes see Table 53.

3.13.3. HDSI/960-PCI and HDSI/960-cPCI

Table 54 lists each CONFIG file for the HDSI/960-PCI and HDSI/960-cPCI boards. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular protocol.

Table 56. HDSI/960-PCI and HDSI/960-cPCI CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Protocol) ①
	***_hdsi.config** ***_hdsi_48_play_rec.config** ***_hdsi_72_play_rec.config** ***_hdsi_96_play_rec.config**
CAS	✓
CHP	✓
TSC	✓

① HDSI config files are prefaced with a country code. For example, the HDSI CONFIG file for Australia E-1 (code = at) are *at_hdsi_48_play_rec.config*, *at_hdsi_72_play_rec.config* and *at_hdsi_96_play_rec.config*. For a list of country codes see Table 53.

3.14. Dialogic Integrated Series Boards

The Dialogic Integrated Series boards include the following groups:

- DI/0408-LS-A and DI/0408-LS-A
- D/SI-32
- DI/SI-16-R2, DI/SI-24-R2, and DI/SI-32-R2

3.14.1. DI/0408-LS-A and DI/0408-LS-A

Table 57 lists each CONFIG file for the DI/0408-LS-A and DI/0408-LS-A boards. The rows associated with the “Applicable CONFIG File Sections” of the table denote by check mark which parameter sections in a CONFIG file apply to a particular board.

Table 57. DI/0408-LS-A and DI/0408-LS-A-R2 CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Board)	
	DI0408LSA.config ① **DI0408LSA_ML2.config**	**DI0408LSA_REV2.config ①** **DI0408LSA_REV2_EU.config ①** **DI0408LSA_REV2_EU_ML2.config** **DI0408LSA_REV2_ML2.config** **DI0408LSA_REV2_EU_ML4.config** **DI0408LSA_REV2_ML4.config**
recorder	✓	✓
encoder	✓	✓
0x39	✓	✓
0x3b	✓	✓
0x3b.x	✓	✓
CAS	✓	✓
CHP	✓	✓
TSC	✓	✓

① These files support media load 1 as defined by the applicable board.

NOTE: The CONFIG file names that include the country code EU apply to all European Union Member Countries, that is, Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, the Netherlands, Portugal, Spain, Sweden, and the United Kingdom.

3.14.2. D/SI-32

Table 58 lists each CONFIG file for the DI/SI-32 board. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to the board.

Table 58. DI/SI-32 CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Board)
	DISI16.config
recorder	✓
0x39	✓
0x3b.x	✓
CAS	✓
CHP	✓
TSC	✓

3.14.3. DI/SI-16-R2, DI/SI-24-R2, and DI/SI-32-R2

Table 59 lists each CONFIG file for the Dialogic Integrated Series boards in this group. The rows associated with the "Applicable CONFIG File Sections" of the table denote by check mark which parameter sections in a CONFIG file apply to a particular board.

Table 59. DI/SI-16-R2, DI/SI-24-R2, and DI/SI-32-R2 CONFIG File Sections

Applicable CONFIG File Sections	CONFIG Files (by Board)		
	DISI16_R2.config DISI16_R2_DE.config DISI16_R2_FR.config DISI16_R2_IT.config DISI16_R2_JP.config DISI16_R2_UK.config	DISI24_R2.config DISI24_R2_DE.config DISI24_R2_FR.config DISI24_R2_IT.config DISI24_R2_JP.config DISI24_R2_UK.config	DISI32_R2.config DISI32_R2_DE.config DISI32_R2_FR.config DISI32_R2_IT.config DISI32_R2_JP.config DISI32_R2_UK.config
recorder	✓	✓	✓
0x39	✓	✓	✓
0x3b.x	✓	✓	✓
CAS	✓	✓	✓
CHP	✓	✓	✓
TSC	✓	✓	✓

① All files support media load 1 as defined by the applicable board.

NOTE: For CONFIG file names that contain a country code, refer to Table 53 for definitions. DI country codes include DE, FR, IT, JP, and UK.

4. Parameter Reference

This section lists the parameters contained in the CONFIG files. Not all parameters are included in each CONFIG file, as this depends on the board supported by that particular file. For a list of supported boards and associated CONFIG files, see Chapter 3, “Boards and Applicable CONFIG File Sections”.

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

Parameters are listed in the same order as they appear in the CONFIG files and they are grouped according to the CONFIG file sections. Within the CONFIG files, the parameters are grouped in the following sections:

- [encoder] Parameters
- [recorder] Parameters
- [0x39] Parameters
- [0x3b] Parameters
- [0x3b.x] Parameters
- [lineAdmin.x] Parameters
- [NFAS] Parameters
- [NFAS.x] Parameters
- [CAS] Parameters for T-1 E&M Signals
- [CAS] Parameters for T-1 Loop Start Signals
- [CAS] Parameters for T-1 Ground Start Signals
- [CAS] User-defined CAS and Tone Signal Parameters
- [CAS] User-defined Signals for Selectable Rings Parameters
- [CCS] Parameters
- [CHP] Parameters
- [CHP] T-1 Protocol Variant Definitions

- [CHP] ISDN Protocol Variant Definitions
- [TSC] Parameters
- [TSC] defineBSet Parameters
- [0x1b] Parameters
- [NetTSC] Parameters
- [0x1d] Parameters
- [0x1c] Parameters

4.1. [encoder] Parameters

The encoder parameters are used to perform an encoding process on a media stream. Automatic Gain Control (AGC) and Silence Compressed Record (SCR) are two algorithms used as part of this encoding process.

The AGC is an algorithm for normalizing an input signal to a target record level. The target record level should be chosen to be the optimum level for an encoder and, at the same time, produce a suitable playback level for a listener.

The AGC algorithm is controlled by three parameters: **PrmAGCk**, **PrmAGCmax_gain,** and **PrmAGClow_threshold**. **PrmAGCK**, is a target output level. **PrmAGCmax_gain** is the a limit on the possible maximum gain. The ratio, **PrmAGCk**/**PrmAGCmax_gain** gives the AGC High Threshold value. This is the threshold for which inputs above it produce output level at the **PrmAGCk** level and inputs with a level below it produce outputs which linearly decrease with the input level. The **PrmAGClow_threshold**, on the other hand, is an upper limit for a noise level estimate. That is, a signal with a level above the **PrmAGClow_threshold** is declared speech, independently of whether it is or not. Below the threshold, the AGC algorithm itself tries to discriminate between voiced and unvoiced signals.

Figure 6 is a graphical representation of the AGC gain relative to input average.

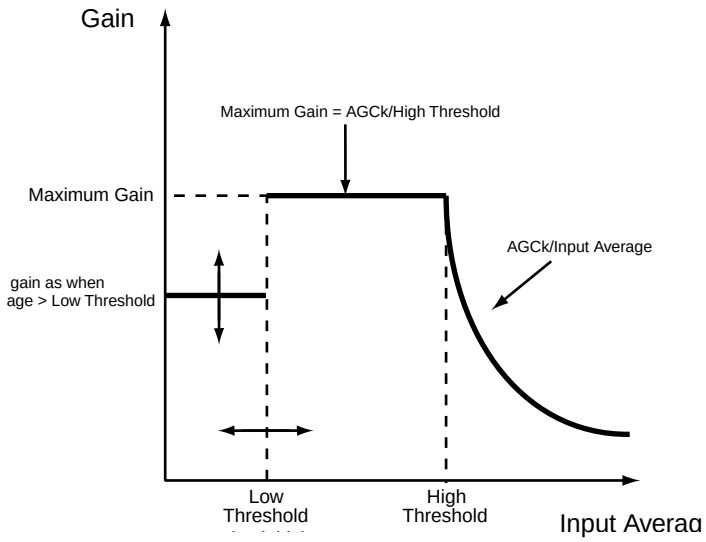


Figure 6. AGC Gain vs. Input Average

The SCR algorithm operates on 1 millisecond blocks of speech and uses a twofold approach to determine whether a sample is speech or silence. Two Probability of Speech values are calculated using a Zero Crossing algorithm and an Energy Detection algorithm. These values are combined to calculate a Combined Probability of Speech.

The Zero Crossing algorithm counts the number of times a sample block crosses a zero line, thus establishing a rough “average frequency” for the sample. If the count for the sample falls within a predetermined range, the sample is considered speech.

The Energy Detection algorithm allows user input at the component level (via the **SCR_LO_THR** and **SCR_HI_THR** parameters) of a background noise threshold range. Signals above the high threshold are declared speech and signals below the low threshold are declared silence.

SCR declares speech or silence for the current 1 millisecond sample based on the following:

- previous 1 millisecond sample declaration (speech or silence)
- Combined Probability of Speech in relation to the Speech Probability Threshold (**SCR_PR_SP**)
- Combined Probability of Speech in relation to the Silence Probability Threshold (**SCR_PR_SIL**)
- Trailing Silence (**SCR_T**) relative to Silence Duration

The logic is as follows:

Previous sample = Silence

```
If Combined Probability of Speech > Speech Probability Threshold
     then Declare Speech
     else Declare Silence
```

Previous sample = Speech

```
If Combined Probability of Speech > Silence Probability Threshold
     then Declare Speech
     else If Silence Duration < Trailing Silence
          then Declare Speech
          else Declare Silence
```

The encoder section of the CONFIG file includes the following parameters:

- PrmAGCk (AGC K Constant)
- PrmAGClow_threshold (AGC Noise Level Lower Threshold)
- PrmAGCmax_gain (AGC Maximum Gain)
- SCR_T (SCR Trailing Silence)
- SCR_PR_SP (SCR Speech Probability Threshold)
- SCR_PR_SIL (SCR Silence Probability Threshold)
- SCR_LO_THR (SCR Low Background Noise Threshold)
- SCR_HI_THR (SCR High Background Noise Threshold)

4.1.1. PrmAGCk (AGC K Constant)

Parameter Number: 0x401

Description: The **PrmAGCk** parameter is the target output level to the TDM bus divided by 32 (to limit K to the range 0 to -1). Note that K is the average level for the output.

PrmAGCk is defined as:

$K * 2^{23}$

where K is defined as follows:

$(10^{(\text{output level in dB})/20})/32$

NOTE: Multiplying by 2^{23} converts the value into a linear 24-bit value that accommodates the 24-bit DSPs used on the DM3 boards.

Therefore, K = 0.006529 corresponds to -13.6 dB average since

$0.006529 = (10^{(-13.6/20)})/32$

Note that -13.6 dB average would result in -6.6 dBm level of the analog output signal.

Here is a sample calculation to obtain a hexadecimal value of **PrmAGCk** for an output level of -19.6 dB:

$(10^{(-19.6/20)})/32 * 2^{23} = 0x006b39$

It is recommended that the value be set in the range of -30.0 dB to -13.6 dB. Higher values may result in strong peak to average compression if it is enabled or just severe clipping if peak to average compression is disabled.

Values: 0x2061 to 0xd5f1 (-30.0 dB to -13.6 dB output levels)

4.1.2. PrmAGClow_threshold (AGC Noise Level Lower Threshold)

Parameter Number: 0x405

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

AGClow_threshold is defined as:

$10^{(\text{output level in dB})/20} * 2^{23}$

NOTES: **1.** Multiplying by 2^{23} converts the value into a linear 24-bit value that accommodates the 24-bit DSPs used on the DM3 boards.

2. The AGC high threshold is determined by the ratio of the **PrmAGCk** value over the **AGCmax_gain** value.

Here is a sample calculation to get a hexadecimal value of **AGClow_threshold** for a noise threshold level of -50 dB_{avg}:

$10^{(-50/20)} * 2^{23} = 0x679f$

It is recommended that the value be set in the range of -60 dB to -40 dB.

Values: 0x20c5 to 0x3000 (-60 dB to -25 dB)

4.1.3. PrmAGCmax_gain (AGC Maximum Gain)

Parameter Number: 0x408

Description: The **PrmAGCmax_gain** Gain parameter defines the maximum gain divided by 32. This parameter controls the maximum possible gain applied by the AGC algorithm. It also implies the High Threshold Level above which all the inputs produce the target output levels and below which produce the levels linearly decreasing with their input level.

AGCmax_gain is defined as:

$((10^{((\text{maximum gain in dB})/20)})/32) * 2^{23}$

NOTE: Multiplying by 2^{23} converts the value into a linear 24-bit value that accommodates the 24-bit DSPs used on the DM3 boards.

Here is a sample calculation to obtain a hexadecimal value of **AGCmax_gain** for a maximum gain of 21 dB:

$((10^{(21/20)})/32) * 2^{23}$

It is recommended that the value be set in the range of 0 dB to 30 dB.

Values: 0x040000 to 0x7e7db9 (0 dB to 30 dB)

4.1.4. SCR_T (SCR Trailing Silence)

Parameter Number: 0x415

Description: The **SCR_T** parameter is used for Silence Compressed Recording and defines the duration of silence allowed following speech before SCR begins (trailing silence). If it is set too low, words and sentences will run together; if it is set too high, SCR efficiency will be reduced, resulting in larger files.

Values: 100 to 1000 (milliseconds)

4.1.5. SCR_PR_SP (SCR Speech Probability Threshold)

Parameter Number: 0x417

Description: The **SCR_PR_SP** parameter is used for Silence Compressed Recording and sets the threshold that the calculated probability of speech value is compared to in declaring speech. If the probability of speech is greater than this parameter, speech is declared.

The **SCR_PR_SP** parameter primarily affects the sensitivity of detecting speech at the leading time of a speech period verses false speech detection.

SCR_PR_SP is defined as:

(Probability Value) * 2^{23}

NOTE: Multiplying by 2^{23} converts the value into a linear 24-bit value that accommodates the 24-bit DSPs used on the DM3 boards.

For example, for a speech probability threshold of 0.58, **SCR_PR_SP** would have a value of:

$0.58 * 2^{23} = 4865393$

Values: 4194304 to 2^{23} (0.50 to 1.00 probability)

4.1.6. SCR_PR_SIL (SCR Silence Probability Threshold)

Parameter Number: 0x418

Description: The **SCR_PR_SIL** parameter is used for Silence Compressed Recording and sets the threshold that the calculated probability of speech value is compared to in declaring silence. If the probability of speech is less than this parameter, silence is declared.

The **SCR_PR_SIL** parameter primarily affects the sensitivity of detecting silence at the trailing time of a speech period verses false silence detection.

SCR_PR_SIL is defined as:

(Probability Value) * 2^{23}

NOTE: Multiplying by 2^{23} converts the value into a linear 24-bit value that accommodates the 24-bit DSPs used on the DM3 boards.

For example, a silence probability threshold of 0.39, **SCR_PR_SIL** would have a value of:

$0.39 * 2^{23} = 3271557$

Values: 0 to 4194304 (0.0 to 0.50 probability)

4.1.7. SCR_LO_THR (SCR Low Background Noise Threshold)

Parameter Number: 0x419

Description: The **SCR_LO_THR** parameter is used for Silence Compressed Recording and controls the low threshold for background noise estimation and, along with **SCR_HI_THR** parameter, forms a range of loudness. Any signal below

this threshold is declared silence. Increasing this threshold increases the probability of losing speech and decreases the probability of recording noise.

SCR_LO_THR is defined as:

$10^{(\text{dB value}/20)} * 2^{23}$

NOTE: Multiplying by 2^{23} converts the value into a linear 24-bit value that accommodates the 24-bit DSPs used on the DM3 boards.

For example, for a low noise threshold of -78 dB, **SCR_LO_THR** would have a value of:

$10^{(-78/20)} * 2^{23} = 1057$

Values: 839 to 83887 (-80 dB to -40 dB)

4.1.8. SCR_HI_THR (SCR High Background Noise Threshold)

Parameter Number: 0x41a

Description: The **SCR_HI_THR** parameter is used for Silence Compressed Recording and controls the high threshold for background noise estimation and, along with **SCR_LO_THR** parameter, forms a range of loudness. Any signal above this threshold is declared speech. Reducing this threshold increases the probability of recording noise and decreases the probability of losing speech.

SCR_HI_THR is defined as:

$10^{(\text{dB value}/20)} * 2^{23}$

NOTE: Multiplying by 2^{23} converts the value into a linear 24-bit value that accommodates the 24-bit DSPs used on the DM3 boards.

For example, for a high noise threshold of -20 dB, **SCR_HI_THR** would have a value of:

$10^{(-20/20)} * 2^{23} = 838861$

Values: 83887 to 2^{23}+ (-40 dB to 0 dB)

4.2. [recorder] Parameters

The recorder section of the CONFIG file includes the following parameters:

- Duration (Record Duration)
- BufferTruncate (Buffer Truncate)
- BeepSignalID (Pre-Recording Beep)
- AGCOnOff (AGC Flag)
- SCR (SCR Flag)

4.2.1. Duration (Record Duration)

Parameter Number: 0x200

Description: The **Duration** parameter specifies the maximum duration for which to record. The maximum duration time is 72 hours (259,200,000 milliseconds). The time specified must be divisible by 4.

Values: 0 to 259,200,000 (in milliseconds and the time specified must be divisible by 4)

4.2.2. BufferTruncate (Buffer Truncate)

Parameter Number: 0x202

Description: The **BufferTruncate** parameter specifies the amount of data (in milliseconds) to truncate from the record buffer at the end of a recording. The suggested range is about 50-150 milliseconds (varies with coder).

Values: 0 to 4000 (milliseconds)

4.2.3. BeepSignalID (Pre-Recording Beep)

Parameter Number: 0x203

Description: The **BeepSignalID** parameter is the signal identifier of the beep tone preceding the recording.

Values:

- 0x21 = 444 Hz tone for 400 milliseconds
- 0x22 = 1000 Hz tone for 400 milliseconds

4.2.4. AGCOnOff (AGC Flag)

Parameter Number: 0x205

Description: The **AGCOnOff** parameter enables or disables Automatic Gain Control (AGC) on a per board basis. These settings can be changed for individual channels using API calls.

Values:

- 0 = Disable AGC
- 1 = Enable AGC

4.2.5. SCR (SCR Flag)

Parameter Number: 0x209

Description: The **SCR** parameter enables or disables Silence Compressed Record (SCR) on a per board basis. These settings can be changed for individual channels using API calls.

Values:

- 9 = Enable SCR
- 10 = Disable SCR

4.3. [0x39] Parameters

The [0x39] section of the CONFIG file includes the following parameters:

- ToneClamping (Tone Clamping)

4.3.1. ToneClamping (Tone Clamping)

Parameter Number: 0x3925

Description: The **ToneClamping** parameter is the used to disable conference tone clamping on Dialogic Integrated Series boards. Tone clamping reduces the amount of DTMF tones heard in a conference. The tone is enabled by default on DM3 boards, that is, the **ToneClamping** parameter does not appear in the CONFIG file. To disable tone clamping, the **ToneClamping** parameter must be added to the [0x39] section of the CONFIG file and set to a value of 0.

NOTE: To disable the conference notification tone on Dialogic Integrated Series, as well as QuadSpan Voice and DualSpan Voice boards, modifications to the **NotificationTone** parameter are also required. For details, see Section 4.5.1, "NotificationTone (Conferencing Notification Tone)", on page 121.

Once disabled, to re-enable the conferencing notification tone, the **ToneClamping** parameter must be removed from the [0x39] section of the CONFIG file.

Values: 0

4.4. [0x3b] Parameters

The [0x3b] section of the CONFIG file includes the following parameters:

- ActiveTalkerNotifyInterval (Active Talker Notification Interval)

4.4.1. ActiveTalkerNotifyInterval (Active Talker Notification Interval)

Parameter Number: 0x3b02

Description: The **ActiveTalkerNotificationInterval** parameter is the periodic duration at which a message is sent to the host listing the active talkers in a conference. A low value can affect system performance due to excessive usage. A high value can affect the quality of the reporting. For example setting the value to 20 will result in good quality active talker notification whenever any party says a "yes" or "no". Setting the parameter to 100 will not result in such events being reported.

Values are in 10 millisecond units. For example, to send a notification message once per second, the **ActiveTalkerNotifyInterval** parameter should be set to a value of 100 (100 * 10 = 1000 milliseconds = 1 second).

Values: Time in 10 millisecond intervals

4.5. [0x3b.x] Parameters

The [0x3b.x] section of the CONFIG file includes the following parameters:

- NotificationTone (Conferencing Notification Tone)

4.5.1. NotificationTone (Conferencing Notification Tone)

Parameter Number: 0x3b06

Description: The **NotificationTone** parameter is the used to disable the conferencing notification tone on QuadSpan Voice, DualSpan Voice, and Dialogic Integrated Series boards. The conferencing notification tone is generated to alert conferees when a party enters or exits a conference. This tone is enabled by default on DM3 boards and the **NotificationTone** parameter is added to the CONFIG file to disable the tone.

NOTE: To disable the tone clamping on Dialogic Integrated Series boards, modifications to the **ToneClamping** parameter are also required. For details, see Section 4.3.1, “ToneClamping (Tone Clamping)”, on page 120.

To disable the conferencing notification tone, the **NotificationTone** parameter must be added to the [0x3b.*x*] sections of the CONFIG file and set to a value of 0. Each [0x3b.*x*] section applies to a specific conferencing line. That is, [0x3b.1] applies to conferencing line 1, [0x3b.2] applies to conferencing line 2, and so on.

For QuadSpan Voice and DualSpan Voice there are 30 conferencing lines, and therefore, 30 [0x3b.*x*] sections in the CONFIG file: [0x3b.1] through [0x3b.30].

For the DI/0408-LS-A Dialogic Integrated Series board, there are three conferencing lines, and therefore, three [0x3b.*x*] sections in the CONFIG file: [0x3b.1] through [0x3b.3].

For the DI/SI-16, DI/SI-24. and DI/SI-32 Dialogic Integrated Series boards, there are five conferencing lines, and therefore, five [0x3b.*x*] sections in the CONFIG file: [0x3b.1] through [0x3b.5].

To enable the conferencing notification tone, the **NotificationTone** parameter must be removed from the [0x3b.*x*] sections of the CONFIG file.

Values: 0

4.6. [lineAdmin.x] Parameters

Line administration parameters are associated with an individual T-1 or E-1 trunk. The parameters defined in the [lineAdmin.x] section are associated with line x, for example, parameters in the [lineAdmin.3] section of the CONFIG file are associated with line 3. Line administration parameters include:

- LineType (Line Type)
- SignalingType (Signaling Type)
- Coding (Coding)
- ZeroCodeSuppression (Zero Code Suppression)
- LOSDeclaredTime (LOS Declared Time)
- LOSClearedTime (LOS Cleared Time)
- REDCFADecay (RED CFA Decay)
- REDCFADeclareTime (RED CFA Declare Time)
- REDCFAClearedTime (RED CFA Cleared Time)
- YellowCFADeclareTime (Yellow CFA Declare Time)
- YellowCFAClearTime (Yellow CFA Clear Time)
- RAICRCCFADeclareTime (RAI CRC CFA Declare Time)
- RAICRCCFAClearTime (RAI CRC CFA Clear Time)

4.6.1. LineType (Line Type)

Parameter Number: 0x1601

Description: The **LineType** parameter defines the physical line type (T-1 or E-1) and the framing format (for example, D4 or ESF). Framing formats include:

- D4 framing - For T-1 lines, in D4 framing, 12 frames of 193 bits each (2,316 bits total) constitute a superframe. This framing format supports AB signaling
- Extended superframe (ESF) - For T-1 lines, in ESF framing, 24 frames of 193 bits each (4,632 bits total) constitute an extended superframe. This framing format supports ABCD signaling.
- CEPT E-1 - For E-1 lines, uses CEPT E-1 framing.
- Cyclic redundancy check 4 (CRC-4) multiframe - For E-1 lines, this provides for CRC error detection. In this framing format, E-1 lines have an extra framing that can coexist with the standard framing and the time slot 16 signaling framing. This extra framing is used to compute and check CRC-4 on incoming lines, to detect remote CRC-4 alarms, and to notify the remote line of CRC-4 errors. When CRC-4 framing is enabled, all CRC-related statistics will be collected and reported, and the RAI_CRC_CFA alarm will be detected and reported.
- Analog - Analog is the framing used for Dialogic Integrated Series boards.

Values:

- **0 = dsx1_D4** - Defines the line type as T-1 using D4 framing format.
- **1 = dsx1_ESF** - Defines the line type as T-1 using extended superframe (ESF) framing format.
- **2 = dsx1_E1** - Defines the line type as E-1 using CEPT E-1 framing.
- **3 = dsx1_E1_CRC** - Defines the line type as E-1 using CRC-4 multiframe.
- **4 = analog** - Defines the line type as analog for use with Dialogic Integrated Series boards.

4.6.2. SignalingType (Signaling Type)

Parameter Number: 0x1602

Description: The **SignalingType** parameter defines the signaling type to be used by the T-1 or E-1 line. Signaling types include:

- Channel associated signaling (CAS) - In CAS, the signaling for each channel is directly associated with that channel. T-1 robbed-bit signaling is an example of CAS.
- Common channel signaling (CCS) - In CCS, a common channel carries the signaling for all of the channels on that T-1 or E-1 line. ISDN is an example of CCS, where the D channel is used to carry the signaling for all of the B channels.

  When using non-facility associated signaling (NFAS) in ISDN applications, the SignalingType must be set to CCS for the primary ISDN trunk.
- Clear channel signaling (Clear) - In this type, none of the channels on the T-1 or E-1 line are used for signaling purposes.

  Clear channel signaling is the ability to access telephony channels in the system and configure them to a user defined call control protocol, or to simply leave the lines ‘clear’. The resources should have access to the telephony bus for media routing purposes, as well as signal detection, signal generation, and tone generation capabilities, if desired.

When using Non-Facility-Associated Signaling (NFAS), Signaling Type is dependent on whether the T-1 or E-1 line is a primary ISDN trunk or NFAS trunk. When using NFAS, SignalingType must be set to Clear for all of the NFAS ISDN trunks. For additional parameters that need to be modified for NFAS, see Section 4.7, “[NFAS] Parameters”, on page 128

Values:

- **4 = CAS** - Defines the signaling type as channel associated signaling (CAS).
- **5 = CCS** - Defines the signaling type as common channel signaling (CCS).
- **6 = Clear** - Defines the signaling type as clear channel.

4.6.3. Coding (Coding)

Parameter Number: 0x1603

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

Values:

- **7 = B8ZS** - This is a modified AMI code that only applies to T-1 lines and is used to preserve one's density on the line. Whenever eight consecutive zeros occur on the line, they are replaced by an 8-bit string that violates the bipolar signaling. If the preceding pulse was positive, the polarity of the substituted eight bits is 000+-0-+. If the preceding pulse was negative, the polarity of the substituted eight bits is 000-+0+-.
- **8 = AMI** - Alternate mark inversion is a bipolar signal conveying binary digits in which each successive 1 (mark) is of the opposite polarity. If the previous mark was a positive pulse, then the next mark will be a negative pulse. Spaces have an amplitude of zero (no pulse).
- **9 = HDB3** - High density bipolar three zero is a modified AMI code that only applies to E-1 and is used to preserve one's density on the line. Whenever four consecutive zeros appear, the four-zeros group is replaced with an HDB3 code. This could be either of two HDB3 codes, depending on whether there was an odd or even number of ones since the last bipolar violation. If an odd number of ones occurred, the substituted four bits are 000V, where V represents a bipolar violation. If an even number of ones occurred, the substituted four bits are P00V, where P represents a parity bit and V represents a bipolar violation.

4.6.4. ZeroCodeSuppression (Zero Code Suppression)

Parameter Number: 0x1604

Description: The **ZeroCodeSuppression** parameter is an algorithm used by T-1 lines that inserts a 1 bit into a stream to prevent the transmission of eight or more consecutive 0 bits, which could produce timing errors. Instead, this algorithm maintains a minimum one's density to reduce timing errors.

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

If the **Coding** parameter is set to B8ZS or HDB3 (for E-1), then zero code suppression is performed by the line-coding and the **ZeroCodeSuppression** parameter is ignored.

Values:

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

4.6.5. LOSDeclaredTime (LOS Declared Time)

Parameter Number: 0x160c

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

Values: 0 to 2500 (0 to 2.5 seconds)

4.6.6. LOSClearedTime (LOS Cleared Time)

Parameter Number: 0x160d

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

Values: 0 to 2500 (0 to 2.5 seconds)

4.6.7. REDCFADecay (RED CFA Decay)

Parameter Number: 0x1609

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

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

4.6.8. REDCFADeclareTime (RED CFA Declare Time)

Parameter Number: 0x160a

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

Values: 0 to 2500 (0 to 2.5 seconds)

4.6.9. REDCFAClearedTime (RED CFA Cleared Time)

Parameter Number: 0x160b

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

Values: 1000 to 15000 (1 to 15 seconds)

4.6.10. YellowCFADeclareTime (Yellow CFA Declare Time)

Parameter Number: 0x160e

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

Values: 0 to 2500 (0 to 2.5 seconds)

4.6.11. YellowCFAClearTime (Yellow CFA Clear Time)

Parameter Number: 0x160f

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

Values: 0 to 2500 (0 to 2.5 seconds)

4.6.12. RAICRCCFADeclareTime (RAI CRC CFA Declare Time)

Parameter Number: 0x1610

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

Values: 0 to 450 (milliseconds)

4.6.13. RAICRCCFAClearTime (RAI CRC CFA Clear Time)

Parameter Number: 0x1611

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

Values: 0 to 450 (milliseconds)

4.7. [NFAS] Parameters

Non-Facility-Associated Signaling (NFAS) uses a single ISDN PRI D channel to provide signaling and control for multiple ISDN PRI lines. When using NFAS, modifications also need to be made to other sections of the CONFIG file. For details, see the following sections:

- Section 4.6.2, “SignalingType (Signaling Type)”, on page 123
- Section 4.8, “[NFAS.x] Parameters”, on page 129

NFAS component level parameters include:

- NFAS_INSTANCE_MAP (NFAS Instance Map)

4.7.1. NFAS_INSTANCE_MAP (NFAS Instance Map)

Parameter Number: 0x3e02

Description: The **NFAS_INSTANCE_MAP** parameter defines the number of NFAS instances or NFAS groups created on a particular board. One NFAS group is created for each primary D channel on the board. The bit map's least significant bit correlates to the first NFAS instance, the next least significant bit corresponds to the second NFAS instance, and so on. So, starting with the least significant bit and working towards the most significant bit, set each bit's value to 1 for each NFAS instance needed. For example, to create three NFAS groups, set the value of the **NFAS_INSTANCE_MAP** parameter to 0x07 (0111).

Values:

- 0x0 = 0 (0000)
- 0x1 = 1 (0001)
- 0x3 = 2 (0011)
- 0x7 = 3 (0111)
- 0xf = 4 (1111)

4.8. [NFAS.x] Parameters

Non-Facility-Associated Signaling (NFAS) uses a single ISDN PRI D channel to provide signaling and control for multiple ISDN PRI lines. For each group defined by the **NFAS_INSTANCE_MAP** parameter, there will be an [NFAS.x] section in the CONFIG file. For example, [NFAS.1] corresponds to the NFAS instance for the first group, [NFAS.2] corresponds to the NFAS instance for the second group, and so on.

When using NFAS, modifications also need to be made to other sections of the CONFIG file. For details, see Section 4.7, "[NFAS] Parameters", on page 128.

NFAS instance level parameters include:

- GroupID (Group Identifier)
- NFAS_PrimaryIntID (Primary Instance Identifier)

4.8.1. GroupID (Group Identifier)

Parameter Number: 0x3e00

Description: The **GroupID** parameter defines the trunks that are assigned to a particular NFAS group. For each group, multiple trunks are identified and added in recurring sets of triplets, using the following command:

```
AddNFASInterface(x)= a,b,c,  a',b',c', ...
```

where,

- x = GroupID - NFAS group into which the interface needs to be added. For [NFAS.x], this would be "x".
- a = InterfaceID - Unique number for this interface assigned by the user. A maximum of 10 interfaces can be assigned to a single group.
- b = BoardNumber - Logical board number on which the trunk being assigned to the InterfaceID resides.
- c = InstanceNumber - Instance number of the trunk that is being assigned to the InterfaceID.

For example, to add all four trunks on board 2 and the first two trunks on board 3 to the fourth NFAS group, enter the following to the [NFAS.4] section in the CONFIG file:

```
[NFAS.4]
AddNFASInterface(4)=0,2,1, 1,2,2, 2,2,3, 3,2,4, 4,3,1, 5,3,2
SetParm=0x3e04,0
```

Values: 1 to 4

4.8.2. NFAS_PrimaryIntID (Primary Instance Identifier)

Parameter Number: 0x3e04

Description: The **NFAS_PrimaryIntID** parameter defines the primary D channel used by the NFAS group and is set for every [NFAS.x] group that is created. The parameter is set to one of the [NFAS.x] InterfaceIDs defined by the **GroupID** parameter's `AddNFASInterface` command. For details, see Section 4.8.1, "GroupID (Group Identifier)", on page 130.

For example, to define the primary D channel for NFAS group 4 to be the second trunk on board 3, enter the following to the [NFAS.4] section in the CONFIG file:

```
[NFAS.4]
AddNFASInterface(4)=0,2,1, 1,2,2, 2,2,3, 3,2,4, 4,3,1, 5,3,2
SetParm=0x3e04,5
```

Values: 0 to 9

4.9. [CAS] Parameters for T-1 E&M Signals

The basis for the T-1 E&M wink protocol is Channel Associated Signaling (CAS). The CAS component is responsible for the generation and detection of CAS signals on the phone network interface. The CAS T-1 E&M wink signals are defined in this section of the CONFIG file and assigned as variants in the [CHP] T-1 Protocol Variant Definition section of the CONFIG file. For details, see Section 4.16, "[CHP] T-1 Protocol Variant Definitions", on page 159.

NOTE: The CAS signaling parameters should only be modified by experienced users if the default settings do not match what the line carrier or PBX is sending or expecting for the line protocol configuration running on the card.

CAS T-1 E&M wink parameters include:

- Offhook (E&M Off-hook Signal)
- Onhook (E&M On-hook Signal)
- FlashOnhook (E&M Flash On-hook Signal)

- Wink (E&M Wink Signal)
- Flash (E&M Flash Signal)

4.9.1. Offhook (E&M Off-hook Signal)

Parameter Number: 0xC15CA001

Description: The **Offhook** parameter defines the transition signal from an on-hook state to an off-hook state. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15 of the Configuration Overview.

Values:

- PreVal = 0xF0 (11110000)
- PostVal = 0xFF (11111111)
- PreTm = 100 ms
- PostTm = 300 ms

4.9.2. Onhook (E&M On-hook Signal)

Parameter Number: 0xC15CA002

Description: The **Onhook** parameter defines the transition signal from an off-hook state to an on-hook state. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xFF (11111111)
- PostVal = 0xF0 (11110000)
- PreTm = 300 ms
- PostTm = 100 ms

4.9.3. FlashOnhook (E&M Flash On-hook Signal)

Parameter Number: 0xC15CA003

Description: The **FlashOnhook** parameter defines the transition signal from an off-hook state to a flash on-hook state during a blind transfer. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xFF (11111111)
- PostVal = 0xF0 (11110000)
- PreTm = 300 ms
- PostTm = 1000 ms

4.9.4. Wink (E&M Wink Signal)

Parameter Number: 0xC15CA011

Description: The **Wink** parameter defines a pulse signal for the purposes of protocol hand-shaking and is typically used as an acknowledgment signal to the line carrier or PBX. It is most often used to acknowledge signaling bit changes detected from the carrier or to signal the start or end of digit collection. The signal transitions from OffVal to OnVal and back to OffVal. For detailed information about pulse signals and their associated values, see “Pulse Signal” on page 16 and “Pulse Example” on page 17.

Values:

- OffVal = 0xF0 (11110000)
- OnVal = 0xFF (11111111)
- PreTm = 100 ms
- MinTm = 210 ms
- NomTm = 250 ms
- MaxTm = 280 ms
- PostTm = 100 ms

4.9.5. Flash (E&M Flash Signal)

Parameter Number: 0xC15CA012

Description: The **Flash** parameter defines a pulse signal for the purposes of requesting special processing. This signal is typically sent to transfer a call to another phone or channel while the call is connected and in progress. The signal goes from OffVal to OnVal and back to OffVal. For detailed information about pulse signals and their associated values, see “Pulse Signal” on page 16 and “Pulse Example” on page 17.

Values:

- OffVal = 0xFF (11111111)
- OnVal = 0xF0 (11110000)
- PreTm = 100 ms
- MinTm = 210 ms
- NomTm = 250 ms
- MaxTm = 280 ms
- PostTm = 100 ms

4.10. [CAS] Parameters for T-1 Loop Start Signals

The basis for the T-1 loop start protocol is Channel Associated Signaling (CAS). The CAS component is responsible for the generation and detection of CAS signals on the phone network interface. The CAS T-1 loop start signals are defined in this section of the CONFIG file and assigned as variants in the [CHP] T-1 Protocol Variant Definition section of the CONFIG file. For details, see Section 4.16, “[CHP] T-1 Protocol Variant Definitions”, on page 159.

NOTE: The CAS signaling parameters should only be modified by experienced users if the default settings do not match what the line carrier or PBX is sending or expecting for the line protocol configuration running on the card.

CAS T-1 loop start parameters include:

- PBX_Open (Loop Start PBX Open Signal)

- PBX_Close (Loop Start PBX Close Signal)
- Net_Answer (Loop Start Net Answer Signal)
- Net_Drop (Loop Start Net Drop Signal)
- Net_Abandon (Loop Start Net Abandon Signal)
- Net_RingOn (Loop Start Net Ring On Signal)
- Net_RingOff (Loop Start Net Ring Off Signal)
- PBX_FlashOpen (Loop Start PBX Flash Open Signal)
- Net_FlashDrop (Loop Start Net Flash Drop Signal)
- PBX_Flash (Loop Start PBX Flash Signal)
- Loop Start Train Definition
- Loop Start Sequence Definition

4.10.1. PBX_Open (Loop Start PBX Open Signal)

Parameter Number: 0xC15CA021

Description: The **PBX_Open** parameter defines the transition signal sent to drop a call. In an analog environment, the station goes from off-hook to on-hook. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xFF (11111111)
- PostVal = 0xF5 (11110101)
- PreTm = 100 ms
- PostTm = 100 ms

4.10.2. PBX_Close (Loop Start PBX Close Signal)

Parameter Number: 0xC15CA022

Description: The **PBX_Close** parameter defines the transition signal sent to make an outbound call, or to answer an incoming call. In an analog environment, the station goes from on-hook to off-hook. For detailed information about transition

signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xF5 (11110101)
- PostVal = 0xFF (11111111)
- PreTm = 100 ms
- PostTm = 100 ms

4.10.3. Net_Answer (Loop Start Net Answer Signal)

Parameter Number: 0xC15CA023

Description: The **Net_Answer** parameter defines the transition signal that, when received, indicates that the network has answered an outbound call. In an analog environment, when the station goes off-hook, the network answers with loop current. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xF5 (11110101)
- PostVal = 0xF0 (11110000)
- PreTm = 100 ms
- PostTm = 100 ms

4.10.4. Net_Drop (Loop Start Net Drop Signal)

Parameter Number: 0xC15CA024

Description: The **Net_Drop** parameter defines the transition signal that, when received, indicates that the network has dropped the call. In an analog environment, with the station off-hook, the network hangs up by dropping loop current. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xA0 (10100000)
- PostVal = 0xAA (10101010)
- PreTm = 100 ms
- PostTm = 100 ms

4.10.5. Net_Abandon (Loop Start Net Abandon Signal)

Parameter Number: 0xC15CA025

Description: The **Net_Abandon** parameter defines the transition signal that, when received, indicates the network has dropped an offered call, that is, the network stops ringing the line. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xF0 (11110000)
- PostVal = 0xF5 (11110101)
- PreTm = 1300 ms
- PostTm = 4500 ms

4.10.6. Net_RingOn (Loop Start Net Ring On Signal)

Parameter Number: 0xC15CA026

Description: The **Net_RingOn** parameter defines the transition signal that, when received, indicates that the network is ringing the line, and an inbound call is offered. In an analog environment, the station is on-hook and the network rings the station. This is the leading edge of the ring. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xF5 (11110101)

- PostVal = 0xF0 (11110000)
- PreTm = 3900 ms
- PostTm = 50 ms

4.10.7. Net_RingOff (Loop Start Net Ring Off Signal)

Parameter Number: 0xC15CA027

Description: The **Net_RingOff** parameter defines the transition signal that, when received, indicates that the network is still offering an inbound call, but has stopped ringing the line. In an analog environment, the station is on-hook and the network pauses between rings. This is the trailing edge of the ring. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xF0 (11110000)
- PostVal = 0xF5 (11110101)
- PreTm = 1300 ms
- PostTm = 0 ms

4.10.8. PBX_FlashOpen (Loop Start PBX Flash Open Signal)

Parameter Number: 0xC15CA029

Description: When flash hook transfer is enabled, the **PBX_FlashOpen** parameter defines the transition signal that is sent to drop a call. In an analog environment, the station drops the call with a flash hook. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xFF (11111111)
- PostVal = 0xF5 (11110101)
- PreTm = 100 ms

- PostTm = 1000 ms

4.10.9. Net_FlashDrop (Loop Start Net Flash Drop Signal)

Parameter Number: 0xC15CA02A

Description: When flash hook transfer is enabled, the **Net_FlashDrop** parameter defines the transition signal that, when received, indicates that the network has dropped the call. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xA0 (10100000)
- PostVal = 0xAA (10101010)
- PreTm = 100 ms
- PostTm = 1000 ms

4.10.10. PBX_Flash (Loop Start PBX Flash Signal)

Parameter Number: 0xC15CA031

Description: The **PBX_Flash** parameter defines the pulse signal used to initiate a blind transfer while the call is connected. For detailed information about pulse signals and their associated values, see “Pulse Signal” on page 16 and “Pulse Example” on page 17.

Values:

- OffVal = 0xFF (11111111)
- OnVal = 0xF5 (11110101)
- PreTm = 100 ms
- MinTm = 210 ms
- NomTm = 250 ms
- MaxTm = 280 ms
- PostTm = 100 ms

4.10.11. Loop Start Train Definition

Parameter Number: 0xC15CA032

Description: The Loop Start Train Definition defines a set of transitions from one signaling state to another in a predefined pattern (set of pulses). This parameter is used to define CAS signals required by a protocol. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- OffVal =0xCC (11001100)
- OnVal = 0xC4 (11000100)
- pulseTmMin = 32
- pulseTmMax = 32
- pulseTmNom = 32
- preTm = 600
- interTmMin = 64
- interTmMax = 64
- interTmNom = 64
- postTm = 20
- digitCount = 12
- pulseCount = 10,0
- label = 1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,11,#,12,*

4.10.12. Loop Start Sequence Definition

Parameter Number: 0xC15CA033

Description: The Loop Start Sequence Definition defines a set of trains for use with HDSI and Dialogic Integrated Series boards. This parameter is used to define CAS signals required by a protocol. For detailed information about sequence signals and their associated values, see “Sequence Signal” on page 22 and “Sequence Example” on page 23.

Values:

- TrainSigId = 0xC15CA03
- preTm = 2
- interTmMin = 720
- interTmMax = 660
- interTmNom = 660
- postTm = 1600

4.11. [CAS] Parameters for T-1 Ground Start Signals

The basis for the T-1 ground start protocol is Channel Associated Signaling (CAS). The CAS component is responsible for the generation and detection of CAS signals on the phone network interface. The CAS T-1 ground start signals are defined in this section of the CONFIG file and assigned as variants in the [CHP] T-1 Protocol Variant Definition section of the CONFIG file. For details, see Section 4.16, "[CHP] T-1 Protocol Variant Definitions", on page 159.

NOTE: The CAS signaling parameters should only be modified by experienced users if the default settings do not match what the line carrier or PBX is sending or expecting for the line protocol configuration running on the card.

CAS T-1 loop start parameters include:

- PBX_Ground (Ground Start PBX Ground Signal)
- PBX_Answer (Ground Start PBX Answer Signal)
- PBX_Release (Ground Start PBX Release Signal)
- PBX_Drop (Ground Start PBX Drop Signal)
- Net_Ground (Ground Start Net Ground Signal)
- Net_Drop (Ground Start Net Drop Signal)
- Ring_On (Ground Start Ring On Signal)
- Ring_Off (Ground Start Ring Off Signal)
- PBX_FlashDrop (Ground Start PBX Flash Drop Signal)
- Net_FlashDrop (Ground Start Net Flash Drop Signal)
- Net_Answer (Ground Start Net Answer Signal)

- PBX_Flash (Ground Start PBX Flash Signal)

4.11.1. PBX_Ground (Ground Start PBX Ground Signal)

Parameter Number: 0xC15CA041

Description: The **PBX_Ground** parameter defines the transition signal sent by the station to make an outbound call, or to answer an incoming call (off-hook). From the station side, it is the GS-FXS transmitting a generic seize, and from the corresponding office or network side, it is the GS-FXO receiving a generic seize. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xF5 (11110101)
- PostVal = 0xF0 (11110000)
- PreTm = 100 ms
- PostTm = 100 ms

4.11.2. PBX_Answer (Ground Start PBX Answer Signal)

Parameter Number: 0xC15CA042

Description: The **PBX_Answer** parameter defines the transition signal sent by the station (GS-FXS) when an inbound call is answered. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xA5 (10100101)
- PostVal = 0xFF (11111111)
- PreTm = 100 ms
- PostTm = 100 ms

4.11.3. PBX_Release (Ground Start PBX Release Signal)

Parameter Number: 0xC15CA043

Description: The **PBX_Release** parameter defines the transition signal sent by the station (GS-FXS) to release an outbound call. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xF0 (11110000)
- PostVal = 0xF5 (11110101)
- PreTm = 50 ms
- PostTm = 50 ms

4.11.4. PBX_Drop (Ground Start PBX Drop Signal)

Parameter Number: 0xC15CA044

Description: The **PBX_Drop** parameter defines the transition signal used by the station (GS-FXS) to know that the network (GS-FXO) has dropped the call. The network generates a GS_Net_Drop signal. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xFF (11111111)
- PostVal = 0xF5 (11110101)
- PreTm = 50 ms
- PostTm = 100 ms

4.11.5. Net_Ground (Ground Start Net Ground Signal)

Parameter Number: 0xC15CA045

Description: The **Net_Ground** parameter defines the transition signal sent by the network (GS-FXO) to make an outbound call, or to answer an incoming call. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xAA (10101010)
- PostVal = 0xA0 (10100000)
- PreTm = 100 ms
- PostTm = 50 ms

4.11.6. Net_Drop (Ground Start Net Drop Signal)

Parameter Number: 0xC15CA046

Description: The **Net_Drop** parameter defines the transition signal that, when received by the station (GS-FXS), indicates that the network has dropped the call. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xA0 (10100000)
- PostVal = 0xAA (10101010)
- PreTm = 100 ms
- PostTm = 50 ms

4.11.7. Ring_On (Ground Start Ring On Signal)

Parameter Number: 0xC15CA047

Description: The **Ring_On** parameter defines the transition signal generated by the network (GS-FXO) to ring the line, and indicate an inbound call is offered to the station (GS-FXS). This is the leading edge of the ring. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xF5 (11110101)
- PostVal = 0xF0 (11110000)
- PreTm = 3900 ms
- PostTm = 50 ms

4.11.8. Ring_Off (Ground Start Ring Off Signal)

Parameter Number: 0xC15CA048

Description: The **Ring_Off** parameter defines the transition signal generated by the network (GS-FXO) to stop the "ring" on a line when a call is offered to the station (GS_FSO). This is the trailing edge of the ring. For detailed information about transition signals and their associated values, see "Transition Signal" on page 14 and "Transition Example" on page 15.

signal indicates that the network (GS-FXO) is "ringing" the line, and an inbound call is offered to the station (GS-FXS). This is the trailing edge of the ring.

Values:

- PreVal = 0xF0 (11110000)
- PostVal = 0xF5 (11110101)
- PreTm = 1300 ms
- PostTm = 0 ms

4.11.9. PBX_FlashDrop (Ground Start PBX Flash Drop Signal)

Parameter Number: 0xC15CA049

Description: When flash hook transfer is enabled, the **PBX_FlashDrop** parameter defines the transition signal that, when received, indicates that the PBX has dropped the call. For detailed information about transition signals and their associated values, see "Transition Signal" on page 14 and "Transition Example" on page 15.

Values:

- PreVal = 0xFF (11111111)
- PostVal = 0xF5 (11110101)
- PreTm = 50 ms
- PostTm = 1000 ms

4.11.10. Net_FlashDrop (Ground Start Net Flash Drop Signal)

Parameter Number: 0xC15CA04A

Description: When flash hook transfer is enabled, the **Net_FlashDrop** parameter defines the transition signal that, when received, indicates that the network has dropped the call. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xA0 (10100000)
- PostVal = 0xAA (10101010)
- PreTm = 100 ms
- PostTm = 1000 ms

4.11.11. Net_Answer (Ground Start Net Answer Signal)

Parameter Number: 0xC15CA04B

Description: The **Net_Answer** parameter defines the transition signal that, when received, indicates that the network has answered an outbound call. For detailed information about transition signals and their associated values, see “Transition Signal” on page 14 and “Transition Example” on page 15.

Values:

- PreVal = 0xAA (10101010)
- PostVal = 0xA0 (10100000)
- PreTm = 100 ms

- PostTm = 50 ms

4.11.12. PBX_Flash (Ground Start PBX Flash Signal)

Parameter Number: 0xC15CA051

Description: The **PBX_Flash** parameter defines the pulse signal used by the station to initiate a blind transfer while the call is connected. For detailed information about pulse signals and their associated values, see “Pulse Signal” on page 16 and “Pulse Example” on page 17.

Values:

- OffVal = 0xFF (11111111)
- OnVal = 0xF5 (11110101)
- PreTm = 100 ms
- MinTm = 210 ms
- NomTm = 250 ms
- MaxTm = 280 ms
- PostTm = 100 ms

4.12. [CAS] User-defined CAS and Tone Signal Parameters

The CAS component is responsible for the generation and detection of CAS signals on the phone network interface. The CAS user-defined and tone signals are defined in this section of the CONFIG file and assigned as variants in the [CHP] T-1 Protocol Variant Definition section of the CONFIG file. For details, see Section 4.16, “[CHP] T-1 Protocol Variant Definitions”, on page 159.

NOTE: The CAS signaling parameters should only be modified by experienced users if the default settings do not match what the line carrier or PBX is sending or expecting for the line protocol configuration running on the card.

- HookFlash (Hook Flash)

4.12.1. HookFlash (Hook Flash)

Parameter Number: 0x9201

Description: The **HookFlash (Hook Flash)** parameter defines the pulse signal used to define flash times for HDSI and Dialogic Integrated (DI) Series boards. For detailed information about pulse signals and their associated values, see “Pulse Signal” on page 16 and “Pulse Example” on page 17.

NOTE: Only modifications to the MinTm and MaxTm values are supported.

Values:

- HDSI Boards:
  - OffVal = 0xFF (11111111)
  - OnVal = 0xF5 (11110101)
  - PreTm = 100 ms
  - MinTm = Time (in milliseconds)
  - NomTm = 650 ms
  - MaxTm = Time (in milliseconds)
  - PostTm = 100 ms
- Dialogic Integrated Series Boards:
  - OffVal = 0xCC (11001100)
  - OnVal = 0xC4 (11110100)
  - PreTm = 100 ms
  - MinTm = Time (in milliseconds)
  - NomTm = 250 ms
  - MaxTm = Time (in milliseconds)
  - PostTm = 100 ms

4.13. [CAS] User-defined Signals for Selectable Rings Parameters

The CAS component is responsible for the generation and detection of CAS signals on the phone network interface. The CAS user-defined signals for selectable rings

are defined in this section of the CONFIG file and assigned as variants in the [CHP] T-1 Protocol Variant Definition section of the CONFIG file. For details, see Section 4.16, "[CHP] T-1 Protocol Variant Definitions", on page 159.

NOTE: The CAS signaling parameters should only be modified by experienced users if the default settings do not match what the line carrier or PBX is sending or expecting for the line protocol configuration running on the card.

- Net_RingOn (Ring Cadence On-time)
- Net_RingOff (Ring Cadence Off-time)

4.13.1. Net_RingOn (Ring Cadence On-time)

Parameter Number: 0x9110

Description: The **Net_RingOn** parameter is one of two pulse signal used to define the ring cadence for HDSI and Dialogic Integrated Series boards. **Net_RingOn** defines the on-time signal and **Net_RingOff** defines the off-time signal. For detailed information about pulse signals and their associated values, see "Pulse Signal" on page 16 and "Pulse Example" on page 17.

The value used for the MinTm, NomTm, and MaxTm (the same time must be used for all three values) is the total on-time duration of the pulse. If MinTm, NomTm, and MaxTm are set to 2000, then the total on-time duration of the pulse is 2000 milliseconds. The PostTm value is not included in the total on-time duration since this value defines part of the off-time duration.

NOTE: Only modifications to the MinTm, NomTm, and MaxTm values are supported. When modifying these values, the same time must be used for all three values.

Values:

- OffVal = 0xA4 (10101010)
- OnVal = 0xAA (10100100)
- PreTm = 0 ms
- MinTm = 2000 ms
- NomTm = 2000 ms

- MaxTm = 2000 ms
- PostTm = 50 ms

4.13.2. Net_RingOff (Ring Cadence Off-time)

Parameter Number: 0x9111

Description: The **Net_RingOff** parameter is one of two pulse signal used to define the ring cadence for HDSI and Dialogic Integrated Series boards. **Net_RingOn** defines the on-time signal and **Net_RingOff** defines the off-time signal. For detailed information about pulse signals and their associated values, see "Pulse Signal" on page 16 and "Pulse Example" on page 17.

The total off-time duration of the pulse includes the **Net_RingOn** PostTm duration, the **NetRingOff** PrePulse duration, and the **NetRingOff** on-time duration (defined by MinTm, NomTm, or MaxTm since all three times must be set to the same value). If the **Net_RingOn** PostTm is set to 50 milliseconds, the **Net_RingOff** PreTm duration is set to 50 milliseconds, and the **Net_RingOff** on-time duration is set to 3900 milliseconds, the total off-time duration is 4000 milliseconds.

NOTE: Only modifications to the MinTm, NomTm, and MaxTm values are supported. When modifying these values, the same time must be used for all three values.

Values:

- OffVal = 0xA4 (10100100)
- OnVal = 0xA4 (10100100)
- PreTm = 50 ms
- MinTm = 3900 ms
- NomTm = 3900 ms
- MaxTm = 3900 ms
- PostTm = 0 ms

4.14. [CCS] Parameters

Common Channel Signaling (CCS) supports ISDN PRI out-of-band signaling utilizing the Q.931 signaling protocol for messaging. The parameters in the [CCS] and [CSS.x] sections of the CONFIG file define the number of CCS component instances created and configure the parameters associated with each CCS instance. The CCS parameters include:

- INSTANCE_MAP (Instance Map)
- CCS_TMR_302 (Q.931 Timer 302)
- CCS_TMR_303 (Q.931 Timer 303)
- CCS_TMR_304 (Q.931 Timer 304)
- CCS_TMR_305 (Q.931 Timer 305)
- CCS_TMR_308 (Q.931 Timer 308)
- CCS_TMR_310 (Q.931 Timer 310)
- CCS_TMR_313 (Q.931 Timer 313)
- CCS_TEI_RETRY (TEI Retry Timer)
- CCS_TEI_STABILITY (TEI Stability Timer)
- SYMMETRICAL_LINK (Symmetrical Command Response Protocol)
- CCS_PROTOCOL_MODE (ISDN Protocol Mode)
- CCS_SWITCH_TYPE (Switch Type)
- L2_TRACE (Layer 2 Access Flag)

4.14.1. INSTANCE_MAP (Instance Map)

Parameter Number = 5

Description: The **INSTANCE_MAP** parameter is a bit map that defines the number of CCS instances created. A CCS instance is created for each network interface that supports common channel signaling. The bit map's least significant bit corresponds to the CCS instance associated with the first network interface on the board. The next least significant bit corresponds to the CCS instance associated with the second network interface on the board, and so on. If the bit associated with a network interface has a value of 1, then a CCS instance is created for that network

interface. For example, a value of 0x5 (0101) means that CCS instances 1 and 3 are created allowing for common channel signaling on network interfaces 1 and 3.

Values: 0 to 0x0f

4.14.2. CCS_TMR_302 (Q.931 Timer 302)

Parameter Number: 0x14

Description: The **CCS_TMR_302** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

Values:

- 0 = Use the default value for the switch
  - T-1 = not applicable
  - E-1= 15000 milliseconds
- -1 = Disable the timer (has the same effect as setting the timer value to 0)
- 0 < n < -1 = Timer value (in milliseconds)

4.14.3. CCS_TMR_303 (Q.931 Timer 303)

Parameter Number: 0x0b

Description: The **CCS_TMR_303** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

Values:

- 0 = Use the default value for the switch
  - T-1 = 4000 milliseconds
  - E-1= 4000 milliseconds
- -1 = Disable the timer (has the same effect as setting the timer value to 0)
- 0 < n < -1 = Timer value (in milliseconds)

4.14.4. CCS_TMR_304 (Q.931 Timer 304)

Parameter Number: 0x0c

Description: The **CCS_TMR_304** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

Values:

- 0 = Use the default value for the switch
  - T-1 = not applicable
  - E-1 = 30000 milliseconds
- -1 = Disable the timer (has the same effect as setting the timer value to 0)
- 0 < n < -1 = Timer value (in milliseconds)

4.14.5. CCS_TMR_305 (Q.931 Timer 305)

Parameter Number: 0x0d

Description: The **CCS_TMR_305** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

Values:

- 0 = Use the default value for the switch
  - T-1 = 4000 milliseconds
  - E-1 = 30000 milliseconds
- -1 = Disable the timer (has the same effect as setting the timer value to 0)
- 0 < n < -1 = Timer value (in milliseconds)

4.14.6. CCS_TMR_308 (Q.931 Timer 308)

Parameter Number: 0x0e

Description: The **CCS_TMR_308** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

Values:

- 0 = Use the default value for the switch
  - T-1 = 4000 milliseconds
  - E-1 = 4000 milliseconds
- -1 = Disable the timer (has the same effect as setting the timer value to 0)
- 0 < n < -1 = Timer value (in milliseconds)

4.14.7. CCS_TMR_310 (Q.931 Timer 310)

Parameter Number: 0x0f

Description: The **CCS_TMR_310** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

Values:

- 0 = Use the default value for the switch
  - T-1 = 10000 milliseconds
  - E-1 = 10000 milliseconds
- -1 = Disable the timer (has the same effect as setting the timer value to 0)
- 0 < n < -1 = Timer value (in milliseconds)

4.14.8. CCS_TMR_313 (Q.931 Timer 313)

Parameter Number: 0x10

Description: The **CCS_TMR_313** parameter is an ISDN Layer 3 timer. For exact timer definitions, refer to the Q.931 specification and the switch specifications.

Values:

- 0 = Use the default value for the switch
  - T-1 = 4000 milliseconds
  - E-1 = 4000 milliseconds
- -1 = Disable the timer (has the same effect as setting the timer value to 0)
- 0 < n < -1 = Timer value (in milliseconds)

4.14.9. CCS_TEI_RETRY (TEI Retry Timer)

Parameter Number: 0x15

Description: The **CCS_TEI_RETRY** parameter defines the maximum amount of time that the data link remains in state 4 (TEI_ASSIGNED) before transitioning to state 5 (**TEI_WAIT_ESTABLISH**).

Values: Time (in milliseconds)

4.14.10. CCS_TEI_STABILITY (TEI Stability Timer)

Parameter Number: 0x16

Description: The **CCS_TEI_STABILITY** parameter defines the minimum transition time between data link state 4 (TEI_ASSIGNED) and data link state 5 (TEI_WAIT_ESTABLISH).

Values: 0 to 100,000 (milliseconds)

4.14.11. SYMMETRICAL_LINK (Symmetrical Command Response Protocol)

Parameter Number: 0x13

Description: The **SYMMETRICAL_LINK** parameter enables or disables symmetrical data link operations.

Values:

- 0 = Disable symmetrical data link operations
- 1 = Enable symmetrical data link operations (for debugging purposes only)

  NOTE: When symmetrical data link operations is enabled, the board's front end cannot be used (i.e., cannot make or receive calls).

4.14.12. CCS_PROTOCOL_MODE (ISDN Protocol Mode)

Parameter Number: 0x17

Description: The **CCS_PROTOCOL_MODE** parameter sets the network user-side protocol. User-side protocol is also known as TE (terminal emulation) protocol and Network-side protocol is also known as NT (network termination) protocol. This parameter also can be use to configure QSIG Master/Slave.

NOTE: Master/Slave mode pertains to QSIG protocols only.

Values:

- 0 = User or Slave Mode (QSIG)
- 1 = Network or Master Mode (QSIG)

4.14.13. CCS_SWITCH_TYPE (Switch Type)

Parameter Number: 0x07

Description: The **CCS_SWITCH_TYPE** parameter defines the network switch type. It is used to distinguish between QSIG T-1 and QSIG E-1 only and ignored for any other protocols.

Values:

- 10 = QSIGE1
- 11 = QSIGT1

4.14.14. L2_TRACE (Layer 2 Access Flag)

Parameter Number: 0x09

Description: The **L2_TRACE** parameter is the ISDN Layer 2 access flag. When Layer 2 (Data Link layer) access is disabled, ISDN Link Access Protocol for the D channel (LAPD) functionality is obtained by accessing ISDN Call Control and Layer 3 (Network layer). When Layer 2 access is enabled, call control is no longer supported for the channels on this line and ISDN LAPD functionality is obtained by accessing Layer 2 directly.

Values:

- 0 = Disable Layer 2 access
- 1 = Enable Layer 2 access

4.15. [CHP] Parameters

The Channel Protocol (CHP) component implements the telephony communication protocol that is used on the network interface. The CHP component parameters include:

- R4Compatibility (R4 Compatibility Flag)
- InitialChanState (Initial Channel State)
- DisableBlock (Disable Block)

4.15.1. R4Compatibility (R4 Compatibility Flag)

Parameter Number: 0x1310

Description: The **R4Compatibility** parameter enables or disables R4 (GlobalCall) compatibility features. This parameter also enables retrieval of DNIS and ANI information in the offered call state.

Values:

- 0 = Default (The default value disables R4 compatibility.)
- 1 = Enable R4 compatibility

- 2 = Disable R4 compatibility

4.15.2. InitialChanState (Initial Channel State)

Parameter Number: 0x1311

Description: The **InitialChanState** parameter defines the initial B channel state (CHP channel state) at the end of system initialization. The initial state of the ISDN B channel is either InService or OutOfService. This parameter must be set to OutofService for ISDN protocols. Once the board is initialized, this initial state will be set on all channels of the board until a user application is invoked and explicitly modifies the state of the channel.

Values:

- 1 = InService
- 2 = OutOfService

4.15.3. DisableBlock (Disable Block)

Parameter Number: 0x1312

Description: The **DisableBlock** parameter defines whether or not a blocking pattern (message) is sent on a channel when the channel is in the OutofService state. When **DisableBlock** is disabled, no pattern is sent (the switch will not present calls to the B channel).

When **DisableBlock** is enabled and a channel is in the InService state (**InitialChanState**=1), the protocol will send a non-blocking pattern on the channel (the switch will present calls to the B channel). When **DisableBlock** is enabled and a channel is in the OutofService state (**InitialChanState**=2), the protocol will send a blocking pattern on the channel (the switch will present calls to the B channel but these calls will be abandoned by the switch since the application will not respond to the call).

Values:

- 0 = Disable blocking
- 1 = Enable blocking

4.16. [CHP] T-1 Protocol Variant Definitions

The CHP parameters define the line configurations that will be used by each network interface on the DM3 board. Within the [CHP] section of the CONFIG file, different possible T-1 protocol variants (protocol configuration settings) can be implemented and are defined using the `Variant Define n` command. The Variant Define n is later set for each network channel in the [TSC] defineBSet section of the CONFIG file.

For a detailed description of the `Variant Define n` command, see Section 1.3.10, “[CHP] Section”, on page 25. For a detailed description of the `defineBSet` command, also see Section 1.3.11, “[TSC] Section”, on page 29.

Although T-1 signals (E&M wink, loop start and ground start) are assigned as part of the T-1 protocol variant definitions, these signals are defined in the [CAS] section of the CONFIG file. For detailed information about specific T-1 signal definitions, see the following:

- Section 4.9, “[CAS] Parameters for T-1 E&M Signals”, on page 131
- Section 4.10, “[CAS] Parameters for T-1 Loop Start Signals”, on page 134
- Section 4.11, “[CAS] Parameters for T-1 Ground Start Signals”, on page 141

Unless otherwise noted, T-1 Protocol Variant parameters apply to all signal types: E&M wink, loop start, and ground start. Parameters that only apply to a specific signal type, are prefaced with the applicable signal type. For example, the parameter **EM_offhook** only applies to E&M wink signaling and the parameter **ProtocolType** applies to E&M wink, loop start, and ground start signaling.

The T-1 Protocol Variant parameters include:

- ProtocolType (Protocol Type)
- Wink (Wink Flag)
- StartTimeout (Start Timeout)
- FarEndAnswer (Far End Answer)
- AnswerTimeout (Answer Timeout)
- ReconnectTimeout (Reconnect Timeout)
- DisconnectTimeout (Disconnect Timeout)

- InterCallDelay (Inter-call Delay)
- Dial (Outbound Dialing Flag)
- DialFormat (Dial Digits Format)
- ANI (ANI Flag)
- ANIFormat (ANI Digit Format)
- ANICount (ANI Digit Count)
- DNIS (DNIS Flag)
- DNISFormat (DNIS Digit Format)
- DNISCount (DNIS Digit Count)
- PreDigitTimeout (Pre-digit Timeout)
- InterDigitTimeout (Inter-digit Timeout)
- CallProgress (Call Progress Flag)
- CaRingingSet (Ringing Signal)
- CaBusySet (Busy Signal)
- CaSitSet (SIT Signal)
- CaFaxSet (Fax Signal)
- CaPvdId (Voice Detection Signal)
- CaPamdId (Answering Machine Signal)
- CaSignalTimeout (Signal Timeout)
- CaAnswerTimeout (Answer Timeout)
- CaPvdTimeout (Voice Detection Timeout)
- DialToneId (Dial Tone Signal)
- CaDialTimeout (Dial Timeout)
- BlindTransfer (Blind Transfer)
- BtDialToneId (DTD Signal)
- BtStartTimeout (DTD Timeout)
- BtAddressDef (Address Definition)
- BtOrigFormat (Originator Address Digits)
- BtDestFormat (Destination Address Digits)
- BtCancelDigitsFormat (Cancel Digits Format)
- BtCancelFlashCount (Cancel Flash Count)

- BtCancelInterFlashDuration (Cancel Inter-flash Duration)
- BtCancelDigits (Cancel Digits)
- BtDestSuffix (Destination Suffix Digits)
- BtDestPrefix (Destination Prefix Digits)
- BtOrigPrefix (Originator Prefix Digits)
- BtOrigSuffix (Originator Suffix Digits)
- PolarityDetection (Polarity Flag)
- EM_Offhook (E&M Off-hook Signal)
- EM_Onhook (E&M On-hook Signal)
- EM_FlashOnhook (E&M Flash On-hook Signal)
- EM_Wink (E&M Wink Signal)
- EM_Flash (E&M Flash Signal)
- LS_PBX_Open (Loop Start PBX Open)
- LS_PBX_FlashOpen (Loop Start PBX Flash Open Signal)
- LS_PBX_Close (Loop Start PBX Close Signal)
- LS_PBX_Flash (Loop Start PBX Flash Signal)
- LS_Net_Answer (Loop Start Net Answer Signal)
- LS_Net_Drop (Loop Start Net Drop Signal)
- LS_Net_FlashDrop (Loop Start Net Flash Drop Signal)
- LS_Net_Abandon (Loop Start Net Abandon)
- LS_Net_RingOn (Loop Start Net Ring On Signal)
- LS_Net_RingOff (Loop Start Net Ring Off)
- GS_PBX_Ground (Ground Start PBX Ground Signal)
- GS_PBX_Answer (Ground Start PBX Answer Signal)
- GS_PBX_Release (Ground Start PBX Release Signal)
- GS_PBX_Drop (Ground Start PBX Drop Signal)
- GS_PBX_FlashDrop (Ground Start PBX Flash Drop Signal)
- GS_PBX_Flash (Ground Start PBX Flash Signal)
- GS_Net_Ground (Ground Start Net Ground Signal)
- GS_Net_Drop (Ground Start Net Drop Signal)
- GS_Net_FlashDrop (Ground Start Flash Drop Signal)

- GS_Net_RingOn (Ground Start Net Ring On Signal)
- GS_Net_RingOff (Ground Start Net Ring Off Signal)

4.16.1. ProtocolType (Protocol Type)

Description: The **ProtocolType** parameter defines the type of T-1 protocol used on a channel.

NOTE: The **ProtocolType** parameter is also used when defining ISDN protocol variants.

Values:

- 1 = E&M wink
- 2 = Loop start-FXS
- 3 = Ground start-FXS
- 4 = Loop start-FXO (This value is not supported.)
- 5 = Ground start-FXO (Supported by HDSI and Dialogic Integrated Series boards only.)

4.16.2. Wink (Wink Flag)

Description: The **Wink** parameter enables Wink detection for outbound and Wink generation for inbound calls. This parameter is enabled only when using T-1 E&M wink protocols. For all other protocols, this parameter is disabled.

If Wink detection (generation) is enabled, up to three winks are supported in a sequence. Examples of CONFIG file settings for **ANI**, **Wink**, **DNIS**, and **R4Compatibility** parameters for 1, 2, and 3 wink sequences are as follows:

- 1 Wink Example: Seize (Wink) DNIS (Answer) ANI

```
Variant ANI    2    ! No=0, Pre=1, Post=2
Variant Wink   1    ! 1 wink sequence
Variant DNIS   y    ! Enable DNIS collection
SetParm=0x1310,1    ! R4 Compatibility Flag
                    ! 0=default, 1=enable, 2=disable
```

  In the 1 Wink example, ANI information is collected after the call is answered, there is one wink in the sequence, DNIS information is collected, and the R4 Compatibility Flag is turned on.

- 2 Wink Example: Seize (Wink) ANI (DNIS) (Wink) Answer

```
Variant ANI    1    ! No=0, Pre=1, Post=2
Variant Wink   2    ! 1 wink sequence
Variant DNIS   y    ! Enable DNIS collection
SetParm=0x1310,1    ! R4 Compatibility Flag
                    ! 0=default, 1=enable, 2=disable
```

  In the 2 Wink example, ANI information is collected before the call is answered, there are two winks in the sequence, DNIS information is collected, and the R4 Compatibility Flag is turned on.

- 3 Wink Example: Seize (Wink) DNIS (Wink) ANI (Wink) Answer

```
Variant ANI    1    ! No=0, Pre=1, Post=2
Variant Wink   3    ! 1 wink sequence
Variant DNIS   y    ! Enable DNIS collection
SetParm=0x1310,2    ! R4 Compatibility Flag
                    ! 0=default, 1=enable, 2=disable
```

  In the 3 Wink example, ANI information is collected before the call is answered, there are three winks in the sequence, DNIS information is collected, and the R4 Compatibility Flag is turned off.

Values:

- 0 or n = Disable wink
- 1 or y = Enable 2 winks in the sequence
- 2 = Enable 1 wink in the sequence
- 3 = Enable 3 winks in the sequence

4.16.3. StartTimeout (Start Timeout)

Description: The **StartTimeout** parameter is defined differently depending on the T-1 protocol used. Depending on the value of the **ProtocolType** parameter, the **StartTimeout** parameter is defined as follows:

- E&M wink - The amount of time that a call originator will wait for a wink once it has seized the line for an outbound call. The outbound call will fail if this time is exceeded.
- Ground start - The amount of time that a call originator will wait for the dial tone signal once it has seized the line for an outbound call. The outbound call will fail if this time is exceeded.

- Loop start - The call originator will wait the specified amount of time after seizing the line for an outbound call, and then proceed to sending digits (that is, can be used to for dial tone before dialing).

Values: Time (in milliseconds)

4.16.4. FarEndAnswer (Far End Answer)

Description: The **FarEndAnswer** parameter defines whether or not the network will provide far end (remote) answer signaling. This parameter is enabled when using T-1 loop start protocols only. For all other T-1 protocols, this parameter is disabled.

Values:

- y = Enable far end answer support
- n = Disable far end answer support

4.16.5. AnswerTimeout (Answer Timeout)

Description: The **AnswerTimeout** parameter defines the maximum amount of time allowed to answer a call once a remote party has sent its last wink. If the time is exceeded, the call fails. Otherwise, the timer is reset once the call is answered. This parameter is enabled only when using T-1 E&M Wink protocols. For all other T-1 protocols, this parameter is disabled.

Values: Time (in milliseconds)

4.16.6. ReconnectTimeout (Reconnect Timeout)

Description: The **ReconnectTimeout** parameter defines the maximum amount of time allowed between a local disconnect (*MsgDropCall*) and a reconnect (*MSgReconnectCall*).

Values: Time (in milliseconds)

4.16.7. DisconnectTimeout (Disconnect Timeout)

Description: The **DisconnectTimeout** parameter defines the time before a remote drop is considered to be a disconnect. If *MsgDropCall* is followed by a *MsgReconnectCall* within this time period, then the call will be reconnected and remain in the connected state.

NOTE: The **DisconnectTimeout** parameter is also used when defining ISDN protocol variants.

Values: Time (in milliseconds)

4.16.8. InterCallDelay (Inter-call Delay)

Description: The **InterCallDelay** parameter defines the minimum amount of time between outbound calls. This is the time the firmware will wait after a call is dropped and before making another call from the same channel.

NOTE: The **InterCallDelay** parameter is also used when defining ISDN Protocol variants, including E-1.

Values: Time (in milliseconds)

4.16.9. Dial (Outbound Dialing Flag)

Description: The **Dial** parameter enables or disables outbound dialing.

Values:

- y = Enable outbound dialing
- n = Disable outbound dialing

4.16.10. DialFormat (Dial Digits Format)

Description: The **DialFormat** parameter defines the format of the dial digits.

Values:

- 1 = DTMF - Dual Tone Multi-Frequency

- 2 = MF - Multi-Frequency
- 3 = DP - Dial Pulse

4.16.11. ANI (ANI Flag)

Description: The **ANI** parameter enables or disables the collection of ANI data. When the parameter is enabled, it also defines when the data is collected, before or after the call is answered.

Values:

- 0 = Disable ANI data collection
- 1 = Enable ANI data collection - pre-answer
- 2 = Enable ANI data collection - post-answer

4.16.12. ANIFormat (ANI Digit Format)

Description: The **ANIFormat** parameter defines the format of the ANI digits.

Values:

- 1 = DTMF - Dual Tone Multi-Frequency
- 2 = MF - Multi-Frequency
- 3 = DP - Dial Pulse
- 4 = FSK_US - Frequency Shift Keying, United States
- 5 = FSK_JP - Frequency Shift Keying, Japan
- 6 = FSK_UK - Frequency Shift Keying, United Kingdom (This value is not supported.)

4.16.13. ANICount (ANI Digit Count)

Description: The **ANICount** parameter defines the number of ANI digits to collect from the incoming call. If the parameter is set to 0, then digit collection will stop after the timeout period set by the **PreDigitTimeout** and **InterDigitTimeout** parameters defined in the [CHP] `Variant Define n` section of the CONFIG file. For details see:

- Section 1.3.10, "[CHP] Section", on page 25.
- Section 4.16.18, "InterDigitTimeout (Inter-digit Timeout)", on page 168.
- Section 4.16.17, "PreDigitTimeout (Pre-digit Timeout)", on page 168.

Values:

- 0 = Collect all the digits provided
- *n* = Number of digits to collect

4.16.14. DNIS (DNIS Flag)

Description: The **DNIS** parameter enables or disables DNIS data collection for inbound calls.

Values:

- y = Enable DNIS collection
- n = Disable DNIS collection

4.16.15. DNISFormat (DNIS Digit Format)

Description: The **DNISFormat** parameter defines the format of the DNIS digits.

Values:

- 1 = DTMF - Dual Tone Multi-Frequency
- 2 = MF - Multi-Frequency
- 3 = DP - Dial Pulse

4.16.16. DNISCount (DNIS Digit Count)

Description: The **DNISCount** parameter defines the number of DNIS digits to collect from the incoming call. If the parameter is set to 0, then digit collection will stop after the timeout period set by the **PreDigitTimeout** and **InterDigitTimeout** parameters defined in the [CHP] `Variant Define n` section of the CONFIG file. For details see:

- Section 1.3.10, "[CHP] Section", on page 25.
- Section 4.16.18, "InterDigitTimeout (Inter-digit Timeout)", on page 168.
- Section 4.16.17, "PreDigitTimeout (Pre-digit Timeout)", on page 168.

Values:

- 0 = Collect all the digits provided
- *n* = Number of digits to collect

4.16.17. PreDigitTimeout (Pre-digit Timeout)

Description: The **PreDigitTimeout** parameter defines the maximum amount of time that the protocol will wait to receive digits once a call has been initiated. E&M wink-start protocols start this time from the end of the wink.

Values: Time (specified in multiples of 10 milliseconds)

4.16.18. InterDigitTimeout (Inter-digit Timeout)

Description: The **InterDigitTimeout** parameter defines the maximum amount of time between digits. If a digit is not followed by another within this time limit, then digit collection is terminated.

Values: Time (specified in multiples of 10 milliseconds)

4.16.19. CallProgress (Call Progress Flag)

Description: The **CallProgress** parameter enables or disables call progress detection for call setup on outbound calls.

NOTE: The **CallProgress** parameter is also used when defining ISDN protocol variants.

Values:

- y = Enable call progress detection
- n = Disable call progress detection

4.16.20. CaRingingSet (Ringing Signal)

Description: The **CaRingingSet** parameter defines the signal set used to detect ringing for call progress analysis.

NOTES: **1.** Modifications to the **CaRingingSet** parameter is not supported.

2. The **CaRingingSet** parameter is also used when defining ISDN protocol variants.

Values: 0x024940

4.16.21. CaBusySet (Busy Signal)

Description: The **CaBusySet** parameter defines the signal set used to detect busy for call progress analysis.

NOTES: **1.** Modifications to the **CaBusySet** parameter is not supported.

2. The **CaBusySet** parameter is also used when defining ISDN protocol variants.

Values: 0x004de0

4.16.22. CaSitSet (SIT Signal)

Description: The **CaSitSet** parameter defines the signal set used to used to detect Standard Information Tones (SIT) for call progress analysis.

NOTES: **1.** Modifications to the **CaSitSet** parameter is not supported.

2. The **CaSitSet** parameter is also used when defining ISDN protocol variants.

Values: 0x02f240

4.16.23. CaFaxSet (Fax Signal)

Description: The **CaFaxSet** parameter defines the signal set used to detect Fax tones for call progress analysis.

NOTES:
1. Modifications to the **CaFaxSet** parameter is not supported.
2. The **CaFaxSet** parameter is also used when defining ISDN protocol variants.

Values: 0x014b80

4.16.24. CaPvdId (Voice Detection Signal)

Description: The **CaPvdId** parameter defines the signal to use for Positive Voice Detection (PVD) for call progress analysis.

NOTES:
1. Modifications to the **CaPvdId** parameter is not supported.
2. The **CaPvdId** parameter is also used when defining ISDN protocol variants.

Values: 0x01f4c1

4.16.25. CaPamdId (Answering Machine Signal)

Description: The **CaPamdId** parameter defines the signal to use for Positive Answering Machine Detection (PAMD) for call progress analysis.

NOTES:
1. Modifications to the **CaPamdId** parameter is not supported.
2. The **CaPamdId** parameter is also used when defining ISDN protocol variants.

Values: 0x01a041

4.16.26. CaSignalTimeout (Signal Timeout)

Description: The **CaSignalTimeout** parameter defines the maximum amount of time to wait to detect a call progress tone from one of the call analysis signal sets. For T-1 loop start and ground start protocols, if this time is exceeded, then the outbound call will fail with the reason being NoAnswer.

NOTE: The **CaSignalTimeout** parameter is also used when defining ISDN protocol variants.

Values: Time (specified in multiples of 10 milliseconds)

4.16.27. CaAnswerTimeout (Answer Timeout)

Description: The **CaAnswerTimeout** parameter defines the maximum amount of time that call analysis will wait for Ringback to stop. This is equivalent to the number of rings. For T-1 loop start and ground start protocols, if this time is exceeded, then the outbound call will fail with the reason being NoAnswer.

NOTE: The **CaAnswerTimeout** parameter is also used when defining ISDN protocol variants.

Values: Time (specified in multiples of 10 milliseconds)

4.16.28. CaPvdTimeout (Voice Detection Timeout)

Description: The **CaPvdTimeout** parameter defines the maximum amount of time that call analysis will wait to detect positive answering machine detection (PAMD) or positive voice detection (PVD) once ringback has ceased. If this time is exceeded, then connection type will be reported as “Unknown”, that is, not Fax, PAMD, or PVD.

NOTE: The **CaPvdTimeout** parameter is also used when defining ISDN protocol variants.

Values: Time (specified in multiples of 10 milliseconds)

4.16.29. DialToneId (Dial Tone Signal)

Description: The **DialToneId** parameter defines the signal to use for dial tone detection. This parameter is used for loop start protocols. If this is set to 0 (Null) by default to to disabled dial tone detection. This parameter is set at the boad level and enabled (or disabled) for all stations.

Values:

- 0 = Disable dial tone detection
- 0x00a261 = Enable dial tone detection

4.16.30. CaDialTimeout (Dial Timeout)

Description: The **CaDialTimeout** parameter defines the maximum amount of time that call analysis will wait to detect a tone, for example, busy, SIT tones, and ring back.

Values: Time (specified in multiples of 10 milliseconds)

4.16.31. BlindTransfer (Blind Transfer)

Description: The **BlindTransfer** parameter enables or disables blind transfer.

Values:

- 0 = Disable blind transfer
- 1 = Enable blind transfer

4.16.32. BtDialToneId (DTD Signal)

Description: The **BtDialToneId** parameter works together with the **DialToneId** parameter and defines the signal used for dial tone detection in a Blind Transfer. If **BtDialToneId** is 0, then the protocol will wait for a time period (**BtStartTimeout**) before sending digits after generating a Flash.

Values:

- 0 = Pause (the pause time is equal to the value of **BtStartTimeout**)

- *n* = Signal parameter number

4.16.33. BtStartTimeout (DTD Timeout)

Description: The **BtStartTimeout** parameter is used only when the **BtDialToneId** parameter is set to a value of 0 (zero). This parameter defines the maximum amount of time that the protocol will wait for detecting dial tone after a flash has been generated. Once the **BtStartTimeout** value has been reached, a transfer failure will occur with the reason being ProtocolError, and the call will return to the connected state.

If the **BtDialToneId** parameter is set to 0 (zero), **BtStartTimeout** is the time period that the protocol will wait after a Flash has been generated before sending digits

Values: Time (milliseconds)

4.16.34. BtAddressDef (Address Definition)

Description: The **BtAddressDef** parameter defines what addresses will be sent on a blind transfer, and the order in which they will be sent. Addresses are analogous to phone numbers dialed (destination = DNIS), or dialing from (origination =ANI).

Values:

- 1 = None
- 2 = Destination
- 3 = Origination
- 4 = Destination, Origination
- 5 = Origination, Destination

4.16.35. BtOrigFormat (Originator Address Digits)

Description: The **BtOrigFormat** parameter defines the format of the Originator address digits in a blind transfer. The the address is analogous to ANI (caller ID) information.

Values:

- 1 = DTMF - Dual Tone Multi-Frequency
- 2 = MF - Multi-Frequency
- 3 = DP - Dial Pulse

4.16.36. BtDestFormat (Destination Address Digits)

Description: The **BtDestFormat** parameter defines the format of Destination address digits in a blind transfer. The the address is analogous to DNIS information.

Values:

- 1 = DTMF - Dual Tone Multi-Frequency
- 2 = MF - Multi-Frequency
- 3 = DP - Dial Pulse

4.16.37. BtCancelDigitsFormat (Cancel Digits Format)

Description: The **BtCancelDigitsFormat** parameter defines the format of Cancel digits in a blind transfer.

Values:

- 1 = DTMF - Dual Tone Multi-Frequency
- 2 = MF - Multi-Frequency
- 3 = DP - Dial Pulse

4.16.38. BtCancelFlashCount (Cancel Flash Count)

Description: The **BtCancelFlashCount** parameter defines the number of flashes to be sent to cancel or abort a transfer.

Values: Number

4.16.39. BtCancelInterFlashDuration (Cancel Inter-flash Duration)

Description: The **BtCancelInterFlashDuration** parameter defines the time between flashes for canceling or aborting a transferred call.

Values: Time (milliseconds)

4.16.40. BtCancelDigits (Cancel Digits)

Description: The **BtCancelDigits** parameter defines the digits to dial after the flash sequence to cancel or abort a call transfer. For example, *69.

Values: Digits

4.16.41. BtDestSuffix (Destination Suffix Digits)

Description: The **BtDestSuffix** parameter defines the digits to be dialed immediately following the destination address (dialed number/DNIS) in a blind transfer. For example, *34.

Values: Digits

4.16.42. BtDestPrefix (Destination Prefix Digits)

Description: The **BtDestPrefix** parameter defines the digits to be dialed immediately before the Destination address (dialed number/DNIS) in a blind transfer. For example, *54

Values: Digits

4.16.43. BtOrigPrefix (Originator Prefix Digits)

Description: The **BtOrigPrefix** parameter defines the digits to be dialed immediately before the Originator address (calling number/ANI) in a blind transfer. For example, *94.

Values: Digits

4.16.44. BtOrigSuffix (Originator Suffix Digits)

Description: The **BtOrigSuffix** parameter defines the digits to be dialed immediately following the Originator address (calling number/ANI) in a blind transfer. For example, *64.

Values: Digits

4.16.45. PolarityDetection (Polarity Flag)

Description: The **PolarityDetection** parameter defines whether or not the CO reverses battery polarity as the first step before sending the call ring. In Japan, the CO (Nippon Telephone and Telegraph, NTT) reverses the loop polarity prior to sending a call. When **PolarityDetection** parameter is enabled, the polarity reversal sent from the CO in NTT is used when detecting the incoming call.

Values:

- 0 = Disable polarity reversal (normal polarity)
- 1 = Enable polarity reversal (reverse polarity)

4.16.46. EM_Offhook (E&M Off-hook Signal)

Description: The **EM_Offhook** parameter defines the T-1 E&M wink off-hook CAS transition signal id. This signal is sent to make an outbound call, or to answer an incoming call. Receiving this signal indicates that the network is offering a call.

Values: 0xc15ca001

4.16.47. EM_Onhook (E&M On-hook Signal)

Description: The **EM_Onhook** parameter defines the T-1 E&M wink on-hook CAS transition signal id. This signal is sent to drop a call.

Values: 0xc15ca002

4.16.48. EM_FlashOnhook (E&M Flash On-hook Signal)

Description: The **EM_FlashOnhook** parameter defines the T-1 E&M wink on-hook CAS transition signal used when flash hook transfer (blind transfer) is enabled. This signal is sent to drop a call.

defines the transition signal from an off-hook state to a flash on-hook state during a blind transfer.

Values: 0xc15ca003

4.16.49. EM_Wink (E&M Wink Signal)

Description: The **EM_Wink** parameter defines the T-1 E&M Wink CAS pulse signal id. This signal is used as part of the inbound call setup for wink-start E&M protocols. The signal is sent to tell the far end to proceed with the call in response to EM_Offhook.

Values: 0xc15ca011

4.16.50. EM_Flash (E&M Flash Signal)

Description: The **EM_Flash** parameter defines the T-1 E&M wink Flash CAS pulse signal id. This signal is used to initiate a blind transfer while the call is connected.

Values: 0xc15ca012

4.16.51. LS_PBX_Open (Loop Start PBX Open)

Description: The **LS_PBX_Open** parameter defines the T-1 loop start PBX Open CAS transition signal id. This signal is sent to drop a call.

Values: 0xc15a021

4.16.52. LS_PBX_FlashOpen (Loop Start PBX Flash Open Signal)

Description: The **LS_PBX_FlashOpen** parameter defines the T-1 loop start PBX Open CAS transition signal to use when flash hook transfer is enabled. This signal is sent to drop a call.

Values: 0xc15ca029

4.16.53. LS_PBX_Close (Loop Start PBX Close Signal)

Description: The **LS_PBX_Close** parameter defines the T-1 loop start PBX Close CAS transition signal id. This signal is sent to make an outbound call, or to answer an incoming call.

Values: 0xc15ca022

4.16.54. LS_PBX_Flash (Loop Start PBX Flash Signal)

Description: The **LS_PBX_Flash** parameter defines the T-1 loop start PBX Flash CAS pulse signal id. This signal is used to initiate a blind transfer while the call is connected.

Values: 0xc15ca031

4.16.55. LS_Net_Answer (Loop Start Net Answer Signal)

Description: The **LS_Net_Answer** parameter defines the T-1 loop start Net Answer CAS transition signal id. Receiving this signal indicates that the network has answered an outbound call.

Values: 0xc15ca023

4.16.56. LS_Net_Drop (Loop Start Net Drop Signal)

Description: The **LS_Net_Drop** parameter defines the T-1 loop start Net Drop CAS transition signal id. Receiving this signal indicates that the network has dropped the call.

Values: 0xc15ca024

4.16.57. LS_Net_FlashDrop (Loop Start Net Flash Drop Signal)

Description: The **LS_Net_FlashDrop** parameter defines the T-1 loop start Net Drop CAS transition signal to use when flash hook transfer is enabled. Receiving this signal indicates that the network has dropped the call.

Values: 0xc15ca02a

4.16.58. LS_Net_Abandon (Loop Start Net Abandon)

Description: The **LS_Net_Abandon** parameter defines the T-1 loop start Net Abandon CAS transition signal id. Receiving this signal indicates that the network has dropped an offered call, that is, ringing has stopped.

Values: 0xc15ca025

4.16.59. LS_Net_RingOn (Loop Start Net Ring On Signal)

Description: The **LS_Net_RingOn** parameter defines the T-1 loop start Net Ring On CAS transition signal id. Receiving this signal indicates that the network is ringing the line, and an inbound call is offered. This is the leading edge of the ring.

Values: 0xc15ca026

4.16.60. LS_Net_RingOff (Loop Start Net Ring Off)

Description: The **LS_Net_RingOff** parameter defines the T-1 loop start Net Ring Off CAS transition signal id. Receiving this signal indicates that the network is “ringing” the line, and an inbound call is offered. This is the trailing edge of the ring.

Values: 0xc15ca027

4.16.61. GS_PBX_Ground (Ground Start PBX Ground Signal)

Description:

The **GS_PBX_Ground** parameter defines the T-1 ground start PBX Ground CAS transition signal id. The signal sent by the station to make an outbound call, or to answer an incoming call (off-hook). From the station side, it is the GS-FXS transmitting a generic seize, and from the corresponding office or network side, it is the GS-FXO receiving a generic seize.

Values: 0xc15ca041

4.16.62. GS_PBX_Answer (Ground Start PBX Answer Signal)

Description: The **GS_PBX_Answer** parameter defines the T-1 ground start PBX Answer CAS transition signal id. This signal is sent by the station (GS-FXS) when an inbound call is answered.

Values: 0xc15ca042

4.16.63. GS_PBX_Release (Ground Start PBX Release Signal)

Description: The **GS_PBX_Release** parameter defines the T-1 ground start PBX Release CAS transition signal id. This signal is sent by the station (GS-FXS) to release an outbound call.

Values: 0xc15ca043

4.16.64. GS_PBX_Drop (Ground Start PBX Drop Signal)

Description: The **GS_PBX_Drop** parameter defines the T-1 ground start PBX Drop CAS transition signal id. This signal is used by the station (GS-FXS) to know that the network (GS-FXO) has dropped the call. The network generates a GS_Net_Drop signal.

Values: 0xc15ca044

4.16.65. GS_PBX_FlashDrop (Ground Start PBX Flash Drop Signal)

Description: The **GS_PBX_FlashDrop** parameter defines the T-1 ground start PBX Drop CAS transition signal id. This signal is sent to drop a call when flash hook transfer is enabled.

Values: 0xc15ca049

4.16.66. GS_PBX_Flash (Ground Start PBX Flash Signal)

Description: The **GS_PBX_Flash** parameter defines the T-1 ground start PBX Flash CAS pulse signal id. This signal is used by the station to initiate a blind transfer while the call is connected.

Values: 0xc15ca051

4.16.67. GS_Net_Ground (Ground Start Net Ground Signal)

Description: The **GS_Net_Ground** parameter defines the T-1 ground start Net Ground CAS transition signal id. Receiving this signal indicates that the network has answered an outbound call, or that the network is offering an inbound call.

Values: 0xc15ca045

4.16.68. GS_Net_Drop (Ground Start Net Drop Signal)

Description: The **GS_Net_Drop** parameter defines the T-1 loop start Net Drop CAS transition signal id. Receiving this signal indicates that the network has dropped the call.

Values: 0xc15ca046

4.16.69. GS_Net_FlashDrop (Ground Start Flash Drop Signal)

Description: The **GS_Net_FlashDrop** parameter defines the T-1 ground start Net Drop CAS transition signal to use when flash hook transfer is enabled. Receiving this signal indicates that the network has dropped the call.

Values: 0xc15ca04a

4.16.70. GS_Net_RingOn (Ground Start Net Ring On Signal)

Description: The **GS_Net_RingOn** parameter defines the T-1 ground start Net Ring On CAS transition signal id. Receiving this signal indicates that the network (GS-FXO) is “ringing” the line, and an inbound call is offered to the station (GS-FXS). This is the leading edge of the ring.

Values: 0xc15ca047

4.16.71. GS_Net_RingOff (Ground Start Net Ring Off Signal)

Description: The **GS_Net_RingOff** parameter defines the T-1 ground start Net Ring Off CAS transition signal id. The signal generated by the network (GS-FXO) to stop the “ring” on a line when a call is offered to the station (GS_FSO). This is the trailing edge of the ring.

Values: 0xc15ca048

4.17. [CHP] ISDN Protocol Variant Definitions

The CHP parameters define line configurations. Within the [CHP] section of the CONFIG file, ISDN protocol variants are defined using the `Variant Define n` command. For a detailed description of the `Variant Define n` command, see Section 1.3.10, “[CHP] Section”, on page 25. The ISDN protocol variant parameters include:

- ProtocolType (Protocol Type)
- InterCallDelay (Inter-call Delay)
- DisconnectTimeout (Disconnect Timeout)
- Layer1Protocol (Layer 1 Protocol)
- InfoTransferRate (Information Transfer Rate)
- InfoTransferCap (Information Transfer Cap)
- CalledNumberType (Called Number Type)
- CalledNumberPlan (Called Number Plan)

- CalledNumberCount (Called Number Count)
- CallingNumberType (Calling Number Type)
- CallingNumberPlan (Calling Number Plan)
- CallingNumberPresentation (Calling Number Presentation)
- CallingNumberScreening (Calling Number Screening)
- CallingNumberCount (Calling Number Count)
- CallProgress (Call Progress)
- CaRingingSet (Ringing Signal)
- CaBusySet (Busy Signal)
- CaSitSet (SIT Signal)
- CaFaxSet (Fax Signal)
- CaPvdId (Voice Detection Signal)
- CaPamdId (Answering Machine Signal)
- CaSignalTimeout (Signal Timeout)
- CaAnswerTimeout (Answer Timeout)
- CaPvdTimeout (Voice Detection Timeout)

4.17.1. ProtocolType (Protocol Type)

Description: The **ProtocolType** parameter defines the type of ISDN protocol used on a channel. The value of the parameter is dependent on the firmware being downloaded and the CONFIG files used. For example, when downloading the *ml2_qsa_4ess.config* file, **ProtocolType** should be set to a value of 1.

NOTE: The **ProtocolType** parameter is also used when defining T-1 protocol variants.

Values:

- 1 = 4ESS and NI2
- 2 = 5ESS
- 3 = DMS100 and DMS 250
- 4 = NTT
- 7 = NET5 and QSIG

4.17.2. InterCallDelay (Inter-call Delay)

Description: The **InterCallDelay** parameter defines the minimum amount of time between outbound calls.

NOTE: The **InterCallDelay** parameter is also used when defining T-1 protocol variants.

Values: Time (in milliseconds)

4.17.3. DisconnectTimeout (Disconnect Timeout)

Description: The **DisconnectTimeout** parameter defines the time delay between proceeding and alert/connect. The call will transition to idle after this time period (sooner if ClearConf is received).

NOTE: The **DisconnectTimeout** parameter is also used when defining T-1 protocol variants.

Values: Time (in milliseconds)

4.17.4. Layer1Protocol (Layer 1 Protocol)

Description: The **Layer1Protocol** parameter defines the User Layer 1 Protocol.

Values:

- 0x00 = Protocol not present
- 0x01 = CCITT
- 0x02 = G.711 Mu Law
- 0x03 = G.711 A Law
- 0x04 = G.721 ADPCM
- 0x05 = G.721 7KHz
- 0x06 = 384 KHz Video
- 0x07 = NS Rate Adaption
- 0x08 = V120 Rate Adaption
- 0x09 = X.31 HDLC

4.17.5. InfoTransferRate (Information Transfer Rate)

Description: The **InfoTransferRate** parameter defines the information transfer rate.

Values:

- 0x00 = Rate undefined
- 0x10 = 64 Kbps
- 0x11 = 128 Kbps
- 0x13 = 384 Kbps
- 0x15 = 1536 Kbps
- 0x17 = 1920 Kbps
- 0x18 = Multirate

4.17.6. InfoTransferCap (Information Transfer Cap)

Description: The **InfoTransferCap** parameter defines the information transfer capability.

Values:

- 0x00 = Speech
- 0x08 = Unrestricted digital
- 0x09 = Restricted digital
- 0x10 = 3 KHz
- 0x11 = 7 KHz
- 0x18 = Video

4.17.7. CalledNumberType (Called Number Type)

Description: The **CalledNumberType** parameter defines the type of outbound calls (Called Party Numbers).

Values:

- 0x00 = Unknown

- 0x01 = International
- 0x02 = National
- 0x03 = Network specific
- 0x04 = Network subscriber
- 0x06 = Network abbreviated

4.17.8. CalledNumberPlan (Called Number Plan)

Description: The **CalledNumberPlan** parameter defines the numbering plan to use for outbound calls (Called Party Numbers).

Values:

- 0x00 = Unknown
- 0x01 = ISDN
- 0x02 = Telephony
- 0x03 = Date X.121
- 0x04 = Telex F.69
- 0x08 = National standard
- 0x09 = Private

4.17.9. CalledNumberCount (Called Number Count)

Description: The **CalledNumberCount** parameter defines the number of digits to collect from an incoming call.

Values:

- 0 = Collect all the digits provided
- *n* = Number of digits to collect

4.17.10. CallingNumberType (Calling Number Type)

Description: The **CallingNumberType** parameter defines the type of outbound call (Calling Party Number).

Values:

- 0x00 = Unknown
- 0x01 = International
- 0x02 = National
- 0x03 = Network specific
- 0x04 = Network subscriber
- 0x06 = Network abbreviated

4.17.11. CallingNumberPlan (Calling Number Plan)

Description: The **CallingNumberPlan** parameter defines the numbering plan to use for outbound calls (Calling Party Numbers).

Values:

- 0x00 = Unknown
- 0x01 = ISDN
- 0x02 = Telephony
- 0x03 = Date X.121
- 0x04 = Telex F.69
- 0x08 = National standard
- 0x09 = Private

4.17.12. CallingNumberPresentation (Calling Number Presentation)

Description: The **CallingNumberPresentation** parameter defines the presentation for calling number (outbound calls).

Values:

- 0x00 = Allowed
- 0x01 = Restricted
- 0x02 = Not available

4.17.13. CallingNumberScreening (Calling Number Screening)

Description: The **CallingNumberScreening** parameter defines the screening for calling number (outbound calls).

Values:

- 0x00 = User provided
- 0x01 = Verified and passed
- 0x02 = Verified and failed
- 0x03 = Network provided

4.17.14. CallingNumberCount (Calling Number Count)

Description: The **CallingNumberCount** parameter defines the number of Calling Party Number digits to collect from incoming call.

Values:

- 0 = Collect all the digits provided
- *n* = Number of digits to collect

4.17.15. CallProgress (Call Progress)

Description: The **CallProgress** parameter enables or disables call progress detection for call setup.

NOTE: The **CallProgress** parameter is also used when defining T-1 protocol variants.

Values:

- y = Enable call progress detection
- n = Disable call progress detection

4.17.16. CaRingingSet (Ringing Signal)

Description: The **CaRingingSet** parameter defines the signal set used to detect ringing for call progress analysis.

NOTE: The **CaRingingSet** parameter is also used when defining T-1 protocol variants.

Values: 0x024940

4.17.17. CaBusySet (Busy Signal)

Description: The **CaBusySet** parameter defines the signal set used to detect busy for call progress analysis.

NOTE: The **CaBusySet** parameter is also used when defining T-1 protocol variants.

Values: 0x004de0

4.17.18. CaSitSet (SIT Signal)

Description: The **CaSiteSet** parameter defines the signal set used to detect Standard Information Tones (SIT) tones for call progress analysis.

NOTE: The **CaSiteSet** parameter is also used when defining T-1 protocol variants.

Values: 0x02f240

4.17.19. CaFaxSet (Fax Signal)

Description: The **CaFaxSet** parameter defines the signal set used to detect Fax tones for call progress analysis.

NOTE: The **CaFaxSet** parameter is also used when defining T-1 protocol variants.

Values: 0x014b80

4.17.20. CaPvdId (Voice Detection Signal)

Description: The **CaPvdId** parameter defines the signal to use for positive voice detection in call progress analysis.

NOTE: The **CaPvdId** parameter is also used when defining T-1 protocol variants.

Values: 0x01f4c1

4.17.21. CaPamdId (Answering Machine Signal)

Description: The **CaPamdId** parameter defines the signal to use for positive answering machine detection in call progress analysis.

NOTE: The **CaPamdId** parameter is also used when defining T-1 protocol variants.

Values: 0x01a041

4.17.22. CaSignalTimeout (Signal Timeout)

Description: The **CaSignalTimeout** parameter defines the maximum amount of time to wait to detect a call progress tone from one of the call analysis signal sets. For T-1 loop start and ground start protocols, if this time is exceeded, then the outbound call will fail with the reason being NoAnswer.

NOTE: The **CaSignalTimeout** parameter is also used when defining T-1 protocol variants.

Values: Time (specified in multiples of 10 milliseconds)

4.17.23. CaAnswerTimeout (Answer Timeout)

Description: The **CaAnswerTimeout** parameter defines the maximum amount of time that call analysis will wait for Ringback to stop (equivalent to the number of rings). If this time is exceeded, then the outbound call will fail with the reason being NoAnswer.

NOTE: The **CaAnswerTimeout** parameter is also used when defining T-1 protocol variants.

Values: Time (specified in multiples of 10 milliseconds)

4.17.24. CaPvdTimeout (Voice Detection Timeout)

Description: The **CaPvdTimeout** parameter defines the maximum amount of time that call analysis will wait to detect positive answering machine detection (PAMD) or positive voice detection (PVD) once ringback has ceased. If this time is exceeded, then the call state will transition to "Connected" with the reason being Normal. If PAMD or PVD is detected within this time period, then the "Connected" reason will be PAMD or PVD respectively.

NOTE: The **CaPvdTimeout** parameter is also used when defining T-1 protocol variants.

Values: Time (specified in multiples of 10 milliseconds)

4.18. [TSC] Parameters

The parameters in the [TSC] section of the CONFIG file are associated with the B channel sets. The parameters associated with all the sets include:

- Encoding (Encoding Method)

4.18.1. Encoding (Encoding Method)

Parameter Number: 0x1209

Description: The **Encoding** parameter defines the encoding method used on a line.

Values:

- 1 = A Law
- 2 = Mu Law

4.19. [TSC] defineBSet Parameters

The parameters defined by the `defineBSet` command in the [TSC] section of the CONFIG file are associated with the B channel sets. The syntax of the `defineBSet` command is:

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

NOTE: The [TSC] defineBSet parameters do not have parameter numbers explicitly defined within the CONFIG file.

- SetId (Set Identifier)
- LineId (Line Identifier)
- StartChan (Start Channel)
- NumChans (Number of B Channels)
- BaseProtocol (Base Protocol)
- Inbound (Inbound Variant)
- Outbound (Outbound Variant)
- DChanDesc (D Channel Identifier)
- Admin (Admin)
- Width (Width)
- BChanId (B Channel Identifier)
- SlotId (Slot Identifier)
- Direction (Direction)
- Count (Count)

4.19.1. SetId (Set Identifier)

Description: The **SetId** parameter is an arbitrary identifier set by the user that identifies the B channel set in which the B channels are a member. The only requirement is that each B channel set must have a unique identifier.

For example, for each line on a QuadSpan board, **SetId** can be set sequentially to a value that is a multiple of 10 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,2,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,3,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,4,20,1, 1,1,3,23,0
```

Values: Number

4.19.2. LineId (Line Identifier)

Description: The **LineId** parameter defines the T-1 or E-1 line that carries all of the B channels in the set.

For example, on a T-1 QuadSpan board with four network interfaces, the value of **LineId** is set to 1 for line 1, 2 for line 2, and so on for each line as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

Values: 1 to 4

4.19.3. StartChan (Start Channel)

Description: The **StartChan** parameter defines the first B channel in the set. This parameter is used in combination with the **NumChans** parameter to define a contiguous set of B channels.

For example, on a T-1 line where 23 of the 24 channels are used as B channels, the value of **StartChan** is set to 1 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

Values:

- 1 to 24 (T-1)

- 1 to 30 (E-1)
- 1 to 31 (E-1 clear channel)

4.19.4. NumChans (Number of B Channels)

Description: The **NumChans** parameter defines the total number of B channels in the set. This parameter is used in combination with the **StartChan** parameter to define a contiguous set of B channels.

For example, on a T-1 line, a value of 1 for **StartChan** and a value of 23 for **NumChans** defines 23 B channels numbered from 1 to 23:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,2,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,3,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,4,20,1, 1,1,3,23,0
```

Values:

- 1 to 24 (T-1)
- 1 to 30 (E-1)
- 1 to 31 (E-1 clear channel)

4.19.5. BaseProtocol (Base Protocol)

Description: The **BaseProtocol** parameter defines the base protocol on which the B channel set will run. For T-1 CAS, ISDN, and PDK protocols, each firmware load supports only one base protocol, so this parameter will be set to 0 for these protocols.

For example, on T-1 ISDN lines, **BaseProtocol** is set to a value of 0 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

This parameter is also set to 0 for clear channel. Clear channel is the ability to access telephony channels in the system and configure them to a user defined call control protocol, or to simply leave the lines 'clear'. The resources should have access to the telephony bus for media routing purposes, as well as signal detection, signal generation, and tone generation capabilities, if desired.

Values:

- 0 = T-1 CAS, ISDN or PDK protocols (where the default protocol is defined by the firmware) or clear channel
- 7 = Circa Analog - Supports Circa L feature phones (Dialogic Integrated Series boards)

4.19.6. Inbound (Inbound Variant)

Description: The **Inbound** parameter selects one of the protocol type variant parameter sets defined in the [CHP] section of the CONFIG file to use for inbound calls. The protocol variant defines the type of protocol running on the set of B channels. This is set to 0 to clear the channels or disable inbound calls.

For example, on T-1 ISDN lines, **Inbound** is set to a value of 1 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

This parameter is also set to 0 for clear channel. Clear channel is the ability to access telephony channels in the system and configure them to a user defined call control protocol, or to simply leave the lines 'clear'. The resources should have access to the telephony bus for media routing purposes, as well as signal detection, signal generation, and tone generation capabilities, if desired.

Values:

- 0 = Clear channel (disable inbound calls)
- *n* = Variant identifier as defined in the [CHP] section of the CONFIG file

4.19.7. Outbound (Outbound Variant)

Description: The **Outbound** parameter selects one of the protocol type variant parameter sets defined in the [CHP] section of the CONFIG file to use for outbound calls. The protocol variant defines the type of protocol running on the set of B channels. This is set to 0 to clear the channels or disable outbound calls.

For example, on T-1 ISDN lines, **Outbound** is set to a value of 1 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

This parameter is also set to 0 for clear channel. Clear channel is the ability to access telephony channels in the system and configure them to a user defined call control protocol, or to simply leave the lines ‘clear’. The resources should have access to the telephony bus for media routing purposes, as well as signal detection, signal generation, and tone generation capabilities, if desired.

Values:

- 0 = Clear channels (disable outbound calls)
- *n* = Variant identifier as defined in the [CHP] section of the CONFIG file

4.19.8. DChanDesc (D Channel Identifier)

Description: The **DChanDesc** parameter is an ISDN parameter used to select the D channel that the B channels in the set will use. A value of 0 selects the default D channel. This parameter is ignored for T-1 CAS, clear channel and PDK protocols.

For example, on T-1 ISDN lines, **DChanDesc** is set to a value of 1 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

Values:

- 0 = Use default channel (24 for T-1, 16 for E-1)
- 1 to 24 (T-1)
- 1 to 30 (E-1)

4.19.9. Admin (Admin)

Description: The **Admin** parameter is an arbitrary 32-bit value set by the user that is exported to the TSC_AttrAdminGroup attribute of the TSC cluster for each B channel in the set. This attribute can be used to find and/or allocate TSC clusters.

For example, on a T-1 line, **Admin** is set to a value of 20 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

Values: 0 to 0xffffffff

4.19.10. Width (Width)

Description: The **Width** parameter specifies the number of time slots used by each B channel. Currently, only one time slot per channel is used.

NOTE: This Width should not be modified by the user.

For example, on a T-1 line, **Width** is set to a value of 1 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

Values: 1

4.19.11. BChanId (B Channel Identifier)

Description: The **BChanId** parameter defines the initial B channel in the set to which the TSC instance is associated. It is also the channel to which the initial time slot, defined by **SlotId**, will be mapped. B channels are then sequentially mapped to time slots for a count of **Count**.

For example, on a T-1 board where the D channel is mapped to time slot 24 on all four lines, **BChanId** and **SlotId** are set to a value of 1 and **NumChans** is set to a value of 23. This defines 23 B channels numbered 1 to 23 mapped to time slots 1 to 23.

```
defineBSet=10,1,1,23,0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23,0,1,1,2,20,1, 1,1,3,23,0
defineBSet=30,3,1,23,0,1,1,3,20,1, 1,1,3,23,0
defineBSet=40,4,1,23,0,1,1,4,20,1, 1,1,3,23,0
```

For E-1 ISDN lines that usually contain a D channel mapped to time slot 16, the mapping of channels to time slots occurs in two sets of **BChanId**, **SlotId**, **Direction** and **Count** definitions. The first set of definitions map time slots before the D channel and the second set maps time slots after the D channel.

For example, on an E-1 ISDN board with four network interfaces, where time slot 16 is used for signaling on all four lines, **BChanId** would be defined on each line as follows:

```
defineBSet=10,1,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=20,2,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=30,3,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=40,4,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
```

In this example, channels 1 to 15 are mapped to time slots 1 to 15 and channels 16 to 30 are mapped to time slots 17 to 31.

For E-1 clear channel lines where the time slot 16 is not used for signaling, additional `defineBSet` commands are added to clear channel 31. Both **StartChan** and **BChanId** are set to a value of 31, **NumChans** and **Count** are set to a value of 1, and **SlotId** is set to 16 as follows:

```
defineBSet=50,1,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=60,2,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=70,3,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=80,4,31,1, 0,0,0,1,21,1, 31,16,3,1,0
```

Values:

- 1 to 24 (T-1)
- 1 to 31 (E-1)

4.19.12. SlotId (Slot Identifier)

Description: The **SlotId** parameter defines the logical time slot the initial B channel, defined by **BChanId**, is using. B channels are then sequentially mapped to time slots for a count of **Count**.

For E-1 ISDN, the mapping of channels to time slots occurs in two sets of **BChanId**, **SlotId**, **Direction** and **Count** definitions. The first set of definitions map the time slots before the D channel, and the second set maps the slots after the D channel.

For example, on an E-1 ISDN board with four network interfaces, where time slot 16 is used for signaling on all four lines, **SlotId** for all four lines would be as follows

```
defineBSet=10,1,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=20,2,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=30,3,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=40,4,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
```

For all lines in this example, channels 1 to 15 are sequentially mapped to time slots 1 to 15 and channels 16 to 30 are mapped to time slots 17 to 31.

For E-1 clear channel lines where time slot 16 is not used for signaling, additional `defineBSet` commands are added to clear channel 31 and to map time slot 16. Both **StartChan** and **BChanId** are set to a value of 31, **NumChans** and **Count** are set to a value of 1, and **SlotId** is set to 16 as follows:

```
defineBSet=50,1,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=60,2,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=70,3,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=80,4,31,1, 0,0,0,1,21,1, 31,16,3,1,0
```

Values:

- 1 to 24 (T-1)
- 1 to 30 (E-1 ISDN)
- 1 to 31 (E-1 clear channel)

4.19.13. Direction (Direction)

Description: The **Direction** parameter defines the direction in which the data can be sent: inbound, outbound or both.

For example, on an T-1 line where data is transferred both inbound and outbound, **Direction** is set to a value of 3 as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,2,20,1, 1,1,3,23,0
defineBSet=30,3,1,23, 0,1,1,3,20,1, 1,1,3,23,0
defineBSet=40,4,1,23, 0,1,1,4,20,1, 1,1,3,23,0
```

Values:

- 1 = Inbound
- 2 = Outbound
- 3 = Both

4.19.14. Count (Count)

Description: The **Count** parameter defines the number of time slots that are being mapped to B channels. This value is limited to the value of **NumChans** since only the number of channels that exist on a line can be mapped to a time slots.

For example, on a T-1 line containing two network interfaces, where time slot 24 is used as a D channel on both lines, the **Count** for both lines would be as follows:

```
defineBSet=10,1,1,23, 0,1,1,1,20,1, 1,1,3,23,0
defineBSet=20,2,1,23, 0,1,1,1,20,1, 1,1,3,23,0
```

For an E-1 line, **Count** is set to a value of 30 for lines that contain only B channels. For lines that contain a single D channel, the mapping of channels to time slots occurs in two sets of **BChanId**, **SlotId**, **Direction** and **Count** definitions. The first set of definitions map the time slots before the D channel, and the second set maps the slots after the D channel. For example, on an E-1 board with four network interfaces, where time slot 16 is used for signaling on all four lines, the **Count** for all four lines would be as follows:

```
defineBSet=10,1,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=20,2,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=30,3,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
defineBSet=40,4,1,30, 0,1,1,1,20,1, 1,1,3,15, 16,17,3,15,0
```

For all lines in this example, channels 1 to 15 are mapped to time slots 1 to 15 and channels 16 to 30 are mapped to time slots 17 to 31.

For E-1 clear channel lines where the time slot 16 is not used for signaling, additional `defineBSet` commands are added to clear channel 31 and mapped time slot 16. **Count** is set to a value of 1 (also the value of **NumChans**) as follows:

```
defineBSet=50,1,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=60,2,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=70,3,31,1, 0,0,0,1,21,1, 31,16,3,1,0
defineBSet=80,4,31,1, 0,0,0,1,21,1, 31,16,3,1,0
```

Values: 1 to **NumChans**

4.20. [0x1b] Parameters

The [0x1b] section only appears in CONFIG files that are associated with IPLink boards. The following parameters only apply to IPLink boards:

- PrmAGCActive (AGC Enable)
- PrmAGCnf_attfast (Fast Attack Filter Coefficient)
- PrmAGCnf_attslow (Slow Attack Filter Coefficient)
- PrmAGCgain_inc_speech (Maximum Gain Attach Rate)
- PrmAGCmax_gain (Maximum Gain Limit)
- PrmAGCMEM_max_size (Maximum Memory Size)
- PrmAGCMEM_sil_reset (Memory Size Reset)
- PrmAGClow_threshold (Noise Floor Estimate)
- PrmAGCk (Target Output Level)
- PrmCEDCadence (CED Cadence)
- PrmCNGCadenceMin (CNG Minimum Cadence)
- PrmCNGCadenceMax (CNG Maximum Cadence)
- PrmCNGCadenceSilence (CNG Silence)
- PrmDTMFGainCtrl (DTMF Gain Control)
- PrmDTMFVolCrtl (DTMF Volume Control)
- PrmDTMFDurationDflt (DTMF On Time)
- PrmDTMFOffTimeDflt (DTMF Off Time)
- PrmDTMFXferMode (DTMF Transmission Mode)
- PrmECActive (Echo Cancellation)
- PrmECOrder (Number of Taps)
- PrmECNLPActive (NLP Enable)
- PrmECMu (Convergence Rate)
- PrmECResSpFlagEnableDisable (Residual Speech Flag)
- PrmECSuppressGain (Suppress Gain)
- PrmGainCtrl (Gain Control)
- PrmVolCtrl (Volume Control)
- PrmOptLatPktsTx (PLR Optimal Latency)

- PrmMaxLatPktsTx (PLR Maximum Latency)
- PrmRedDepth (Redundancy)
- PrmFaxEnable (Fax Enable)
- PrmT38ECOverride (ECM Override)
- PrmT38DFOverride (Limit Image Encoding Method)
- PrmT38BROverride (Limit Modulation and Bit Rates)
- PrmT38TCFThrshld (Local Training Maximum Error Tolerance)
- PrmT38IndSecBlocks (Fax IND Redundancy Factor)
- PrmT38V21SecBlocks (Fax V21 Redundancy Factor)
- PrmT38HSECMSecBlocks (Fax HSECM Redundancy Factor)
- T38HSSecBlocks (Fax HS Redundancy Factor)
- PrmT38HSEOFSecBlocks (Fax HSEOF Redundancy Factor)
- PrmT38HSEOTSecBlocks (Fax HSEOT Redundancy Factor)
- PrmT38TxThrshld (Transmit Hold Back Threshold)
- PrmT38TCFMethod (Local or End-to-End Training)
- PrmT38TxPower (Transmit Power Level)
- PrmT38SpoofLevel (Spoofing Level)
- PrmHPFActive (HPF Enable)

4.20.1. PrmAGCActive (AGC Enable)

Parameter Number: 0x1b1c

Description: The **PrmAGCActive** parameter allows you to enable or disable the automatic gain control (AGC) of the IPLink. The AGC maintains a uniform signal power level as the signal is retrieved from the bus.

Values:

- 0 = Disable AGC
- 1 = Enable AGC

4.20.2. PrmAGCnf_attfast (Fast Attack Filter Coefficient)

Parameter Number: 0x1b61

Description: The **PrmAGCnf_attfast** parameter sets the noise floor fast attack filter coefficient.

Values: 0x0 to 0xffffff

4.20.3. PrmAGCnf_attslow (Slow Attack Filter Coefficient)

Parameter Number: 0x1b62

Description: The **PrmAGCnf_attslow** parameter sets the noise floor slow attack filter coefficient.

Values: 0x0 to 0xffffff

4.20.4. PrmAGCgain_inc_speech (Maximum Gain Attach Rate)

Parameter Number: 0x1b60

Description: The **PrmAGCgain_inc_speech** parameter sets the maximum attack rate.

Values: 0x20c4 to 0x66666

4.20.5. PrmAGCmax_gain (Maximum Gain Limit)

Parameter Number: 0x1b5e

Description: The **PrmAGCmax_gain** parameter defines the limit of maximum gain allowed in decibels.

Values: 0x0 to 0x7e7db9

4.20.6. PrmAGCMEM_max_size (Maximum Memory Size)

Parameter Number: 0x1b64

Description: The **PrmAGCMEM_max_size** parameter sets the maximum memory size in milliseconds.

Values: 0 x6d3a to 0xaec

NOTE: This parameter should not be adjusted.

4.20.7. PrmAGCMEM_sil_reset (Memory Size Reset)

Parameter Number: 0x1b63

Description: The **PrmAGCMEM_sil_reset** parameter sets the memory size reset during silence intervals (measured in milliseconds) between words or sentences. It controls how fast the gain varies at the beginning of each sentence except for the first one.

Values: 0 x28f5c to 0x20c4

NOTE: This parameter should not be adjusted.

4.20.8. PrmAGClow_threshold (Noise Floor Estimate)

Parameter Number: 0x1b5d

Description: The **PrmAGClow_threshold** parameter sets the upper limit for the noise floor estimate. Any signal above this threshold is considered as speech.

Values: 0x20c4 to 0x732ae

4.20.9. PrmAGCk (Target Output Level)

Parameter Number: 0x1b5f

Description: The **PrmAGCk** parameter defines the output level at which the AGC will attempt to maintain the signal.

Values: 0x2061 to 0xd5f1

4.20.10. PrmCEDCadence (CED Cadence)

Parameter Number: 0x1b53

Description: The **PrmCEDCadence** parameter is used to adjust the duration in milliseconds of the Called Station Identification (CED) signal.

IPLink uses the cadence mechanism to determine if a tone is a fax tone or a DTMF tone, based on the signal's cadence. This mechanism can be adjusted using the **PrmCEDCadence**, **PrmCNGCadenceMin**, **PrmCNGCadenceMax**, and **PrmCNGCadenceSilence** parameters.

Values: 0 to 400 milliseconds

4.20.11. PrmCNGCadenceMin (CNG Minimum Cadence)

Parameter Number: 0x1b54

Description: The **PrmCNGCadenceMin** parameter is used to adjust the minimum duration in milliseconds that the cadence mechanism will recognize the CalliNG (CNG) Tone signal as a fax calling tone (also known as an auto fax tone).

IPLink uses the cadence mechanism to determine if a tone is a fax tone or a DTMF tone, based on the signal's cadence. This mechanism can be adjusted using the **PrmCEDCadence**, **PrmCNGCadenceMin**, **PrmCNGCadenceMax**, and **PrmCNGCadenceSilence** parameters.

Values: 0 to 700 milliseconds

4.20.12. PrmCNGCadenceMax (CNG Maximum Cadence)

Parameter Number: 0x1b55

Description: The **PrmCNGCadenceMax** parameter is used to adjust the maximum duration in milliseconds that the cadence mechanism will recognize the CalliNG (CNG) Tone signal as a fax calling tone (also known as an auto fax tone).

IPLink uses the cadence mechanism to determine if a tone is a fax tone or a DTMF tone, based on the signal's cadence. This mechanism can be adjusted using the **PrmCEDCadence**, **PrmCNGCadenceMin**, **PrmCNGCadenceMax**, and **PrmCNGCadenceSilence** parameters.

Values: 500 to 1000 milliseconds

4.20.13. PrmCNGCadenceSilence (CNG Silence)

Parameter Number: 0x1b56

Description: The **PrmCNGCadenceSilence** parameter is used to adjust the silence period between consecutive CNG tones so that the cadence mechanism will recognize the (CNG) tone signal as a fax calling tone (also known as an auto fax tone). Otherwise, an arbitrary 1100 Hz tone can trigger a fax transmission.

IPLink uses the cadence mechanism to determine if a tone is a fax tone or a DTMF tone, based on the signal's cadence. This mechanism can be adjusted using the **PrmCEDCadence**, **PrmCNGCadenceMin**, **PrmCNGCadenceMax**, and **PrmCNGCadenceSilence** parameters.

Values: 0 to 5000 milliseconds

4.20.14. PrmDTMFGainCtrl (DTMF Gain Control)

Parameter Number: 0x1b4b

Description: The **PrmDTMFGainCtrl** parameter is used in conjunction with the **PrmDTMFVolCrtl** parameter to govern the overall signal power level of the signals before they are transmitted to the SCbus in the direction of the PSTN interface.

The **PrmDTMFGainCtrl** parameter is a gain factor by which the signal is amplified, and can be any value between 1 and 8 inclusive. A Gain Control value of 1 results in no amplification. If the volume of the signal seems weak, increase this

value. If the volume seems too strong, decrease this value. It is recommended that this parameter be modified one integer at a time and the results evaluated.

The product of the **PrmDTMFGainCtrl** setting and the **PrmDTMFVolCrtl** attenuation setting, together determine the overall amplification of the DTMF signal.

Values: 0x1 to 0x8

4.20.15. PrmDTMFVolCrtl (DTMF Volume Control)

Parameter Number: 0x1b4a

Description: The **PrmDTMFVolCrtl** parameter is used in conjunction with the **PrmDTMFGainCtrl** parameter to govern the signal power level of the DTMF signals before they are transmitted onto the SCbus in the direction of the PSTN interface.

The **PrmDTMFVolCrtl** parameter is essentially a scale (attenuation) factor by which the signal is multiplied. It allows you to fine tune the signal level. If the volume seems too weak, increase the value. If the volume seems too strong, decrease the value.The equivalent decimal values range between 0 and 0.999999.

To determine the hexadecimal value, multiply the desired decimal value by 2^{23} and then convert that number into the corresponding hexadecimal value. For example, to define a volume control value of 0.7, multiple 0.7 by 2^{23} (8388608) and convert the product to hexadecimal:

```
0.7 * 8388608 = 5872025.6 (5872026) = 0x599D1E
```

Values: 0x0 to 0x7fffff

4.20.16. PrmDTMFDurationDflt (DTMF On Time)

Parameter Number: 0x1b0f

Description: The **PrmDTMFDurationDflt** parameter is used when no on-time and off-time is passed to the coder (only the digit itself is passed). Dual Tone Multi-frequency (DTMF) tones are comprised of two parts: on-time and off-time. The duration of each can be set separately, to ensure transmission of DTMF tones of sufficient length to be recognized by the receiving side.

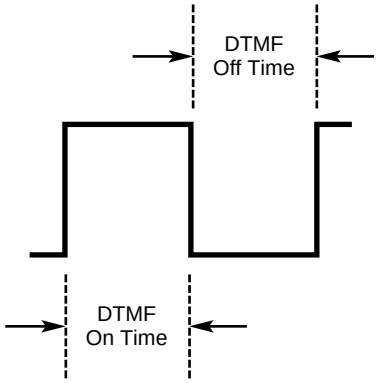


Figure 7. DTMF Tone Generation (DTMFOnTime)

The **PrmDTMFDurationDflt** parameter sets the length of time in milliseconds that the DTMF tone is on.

Values: 0x3c to 0xf0

4.20.17. PrmDTMFOffTimeDflt (DTMF Off Time)

Parameter Number: 0x1b10

Description: The **PrmDTMFOffTimeDflt** parameter is used when no on-time and off-time is passed to the coder (only the digit itself is passed).Dual Tone Multi-frequency (DTMF) tones are comprised of two parts: on-time and off-time. The duration of each can be set separately, to ensure transmission of DTMF tones of sufficient length to be recognized by the receiving side.

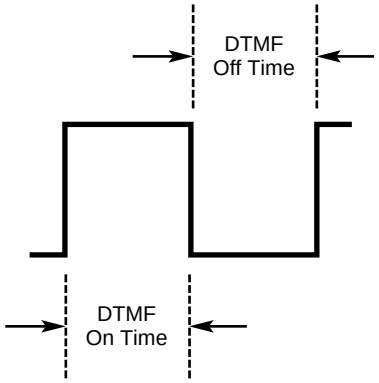


Figure 8. DTMF Tone Generation (DTMFOffTime)

The **PrmDTMFOffTimeDflt** parameter sets the length of time in milliseconds that the DTMF tone is off following the DTMF On Time.

Values: 0x3c to 0xf0

4.20.18. PrmLimit1SIDActive (Insert SID Frame)

Parameter Number: 0x1b6a

Description: The **PrmLimit1SIDActive** parameter lets the application send a single SID frame during silence.

Values:

- 0 = No limit
- 1 = Send one SID frame

4.20.19. PrmDTMFXferMode (DTMF Transmission Mode)

Parameter Number: 0x1b06

Description: The **PrmDTMFXferMode** parameter allows you to select the method for transmitting the DTMF signals. Dual Tone Multi-frequency (DTMF) signals can be transmitted over the IP network either in-band as a standard voice packet or over the H.245 channel as a special out-of-band DTMF packet.

NOTE: If out-of-band DTMF transmission is selected, the application may not use User Input Indication.

Values:

- 0 = No DTMF signals are transmitted
- 1 = DTMF signals are transmitted in-band
- 2 = DTMF signals are transmitted out-of-band
- 3 = Reserved

The following table describes each of the DTMF options:

DTMF Comes from…	**DTMF goes to…**	
	Out Of Band DTMF Enabled	**Out Of Band DTMF Disabled**
DTMF RTP voice from Internet	to local PSTN	to local PSTN
H.245 UII command from Internet	to local PSTN	UII command to local host application
DTMF voice from local PSTN	H245 UII command to Internet	DTMF RTP voice to Internet
UII command from local host application	Ignored (not passed to remote site)	H245 UII command to Internet

4.20.20. PrmECActive (Echo Cancellation)

Parameter Number: 0x1b12

Description: The **PrmECActive** parameter enables or disables echo cancellation in the IPLink gateway. In most cases the echo canceler should be enabled. For applications designed to run over a completely digital network, however, you might consider disabling the echo canceler to free up MIPS.

Because IP telephony is based on placing long distance calls using local central office circuits, the end points do not receive echo canceled signals. The IPLink gateway, therefore, must provide the equivalent of Telco grade, long distance echo cancellation. The IPLink echo canceler provides this capability by removing the echo from the remote channel arriving from the public switched telephone network (PSTN) interface. Parameters used to control the operation of the echo canceler are: **PrmECOrder**, **PrmECMu**, and **PrmECNLPActive**.

Values:

- 0 = Disable echo cancellation
- 1 = Enable echo cancellation

4.20.21. PrmECOrder (Number of Taps)

Parameter Number: 0x1b13

Description: The **PrmECOrder** parameter defines the number of taps (1 tap = 0.125 millisecond) that the echo canceler samples.

Because IP telephony is based on placing long distance calls using local central office circuits, the end points do not receive echo canceled signals. The IPLink gateway, therefore, must provide the equivalent of Telco grade, long distance echo cancellation. The IPLink echo canceler provides this capability by removing the echo from the remote channel arriving from the public switched telephone network (PSTN) interface. Parameters used to control the operation of the echo canceler are: **PrmECOrder**, **PrmECMu** and **PrmECNLPActive**.

Values: 48 to 128 taps

4.20.22. PrmECNLPActive (NLP Enable)

Parameter Number: 0x1b1b

Description: The **PrmECNLPActive** parameter allows you to enable or disable the echo canceler's nonlinear processor (NLP). When the NLP is enabled, the echo canceler uses its comfort noise estimation and generation mechanism to suppress any echo present at the local end.

Because IP telephony is based on placing long distance calls using local central office circuits, the end points do not receive echo canceled signals. The IPLink gateway, therefore, must provide the equivalent of Telco grade, long distance echo cancellation. The IPLink echo canceler provides this capability by removing the echo from the remote channel arriving from the public switched telephone network (PSTN) interface. Parameters used to control the operation of the echo canceler are: **PrmECOrder**, **PrmECMu**, and **PrmECNLPActive**.

Values:

- 0 = Disable NLP
- 1 = Enable NLP: Residual echo is substituted for background-matched white noise. The spectrum is not matched.
- 2 = Enable NLP: Residual echo is suppressed using the value in ECSuppressGain. The active suppression remains until there is double talk or detected near end speech.

4.20.23. PrmECMu (Convergence Rate)

Parameter Number: 0x1b16

Description: The **PrmECMu** parameter defines the rate of convergence of the echo path estimation algorithm, that is, the adaption gain factor of echo cancellation. The higher the value, the faster the convergence rate. Setting the value too low causes convergence to take too long. Setting the value too high causes echo canceller divergence.

NOTE: This parameter should not be adjusted without, first, consulting Customer Engineering.

Values: 0x80 to 0x40000

4.20.24. PrmECResSpFlagEnableDisable (Residual Speech Flag)

Parameter Number: 0x1b65

Description: The **PrmECResSpFlagEnableDisable** parameter enables both Echo Return Loss (ERL) and other proprietary double-talk mechanisms. If this parameter is disabled, then only ERL is enabled.

NOTE: When using NetMeeting, disable this parameter if therre is a noticeable echo.

Values:

- 0 = Enable ERL and other proprietary double-talk mechanisms
- 1 = Disable other proprietary double-talk mechanisms (ERL only is enabled)

4.20.25. PrmECSuppressGain (Suppress Gain)

Parameter Number: 0x1b66

Description: The **PrmECSuppressGain** parameter is used only when echo cancellation is enabled, that is, when the **prmECNLPActive** parameter is set to a value of 2.

Values: 0x0 - 0x7fffff

4.20.26. PrmGainCtrl (Gain Control)

Parameter Number: 0x1b39

Description: The **PrmGainCtrl** parameter is used in conjunction with the **PrmVolCtrl** parameter to govern the overall signal power level of the decoded PCM data before it is transmitted to the SCbus in the direction of the PSTN interface.

The **PrmGainCtrl** parameter is a gain factor by which the signal is amplified, and can be any value between 1 and 8 inclusive. A **PrmGainCtrl** value of 1 results in no amplification. If the volume of the signal seems weak, increase this value. If the

volume seems too strong, decrease this value. It is recommended that this parameter be modified one integer at a time and the results evaluated.

The product of the **PrmGainCtrl** setting and the **PrmVolCtrl** attenuation setting, together determine the overall amplification of the signal.

Values: 0x1 to 0x8

4.20.27. PrmVolCtrl (Volume Control)

Parameter Number: 0x1b14

Description: The **PrmVolCtrl** parameter is used in conjunction with the **PrmGainCtrl** parameter to govern the signal power level of the decoded PCM data before it is transmitted onto the SCbus in the direction of the PSTN interface.

The **PrmVolCtrl** parameter is essentially a scale (attenuation) factor by which the signal is multiplied. It allows you to fine tune the signal level. If the volume seems too weak, increase the value. If the volume seems too strong, decrease the value. This parameter must assume a hexadecimal value corresponding to the digital signal processor's (DSP's) 24-bit fractional representation. The equivalent decimal values range between 0.0 and 0.999999.

To determine the hexadecimal value, multiply the desired decimal value by 2^{23} and then convert that number into the corresponding hexadecimal value. For example, to define a volume control value of 0.7, multiple 0.7 by 2^{23} (8388608) and convert the product to hexadecimal:

```
0.7 * 8388608 = 5872025.6 (5872026) = 0x599D1E
```

Values: 0x0 to 0x7fffff

4.20.28. PrmOptLatPktsTx (PLR Optimal Latency)

Parameter Number: 0x1b07

Description: The **PrmOptLatPktsTx** parameter defines the amount of Packet Loss Recovery (PLR) latency (delay) that can be introduced by defining the number

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

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

Each new frame is then stored in an elastic buffer before sending it to the decoder. This is done to allow packets arriving out of order to be inserted in the queue in the correct order. The size of this elastic buffer is defined by the number of frames stored and is controlled by both the **PrmOptLatPktsTx** parameter and the **PrmMaxLatPktsTx** parameter.

Values: 0x1 to 0x6 frames

4.20.29. PrmMaxLatPktsTx (PLR Maximum Latency)

Parameter Number: 0x1b08

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

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

Each new frame is then stored in an elastic buffer before sending it to the decoder. This is done to allow packets arriving out of order to be inserted in the queue in the correct order. The size of this elastic buffer is defined by the number of frames stored and is controlled by both the **PrmOptLatPktsTx** parameter and the **PrmMaxLatPktsTx** parameter.

Values: 0x1 to 0x10 frames

4.20.30. PrmRedDepth (Redundancy)

Parameter Number: 0x1b15

Description: The **PrmRedDepth** parameter sets the number of times that a frame is re-transmitted. This complies with RFC 2198 specification for redundancy.

Values: 1 to 4

4.20.31. PrmFaxEnable (Fax Enable)

Parameter Number: 0x1b01

Description: The **PrmFaxEnable** parameter enables or disables the IPLink support for fax transmission. The IPLink gateway supports the T.30 fax handshake protocol over the T.38 IP network.

Values:

- 0 = Disable fax support
- 1 = Enable fax support

4.20.32. PrmT38ECOverride (ECM Override)

Parameter Number: 0x1b3c

Description: The **PrmT38ECOverride** parameter allows you to override the Error Correction Mode (ECM) for all calls. If the **PrmT38ECOverride** parameter is enabled, then the gateway will limit the negotiated fax session to only non-ECM calls. This parameter enables the receiving fax device to request missing or erroneous scan lines.

Values:

- 0 = Disable (Do not override the ECM)
- 1 = Enable (Override the ECM)

4.20.33. PrmT38DFOverride (Limit Image Encoding Method)

Parameter Number: 0x1b3d

Description: The **PrmT38DFOverride** parameter limits the type of image encoding method used by the IPLink. For most images, MMR encoding is smaller than MR, and MR is smaller than MH. If the Encoding Method is set to MH_ONLY, then the gateway will limit the negotiated image encoding to only MH. If the Encoding Method is set to MR_BEST, then the gateway will limit the negotiated image encoding to either MH or MR.

Values:

- 0 = Disable (Enables all image encoding methods)
- 1 = MH_ONLY (Disables MR+ image encoding)
- 2 = MR_BEST (Disables MMR+ image encoding)

4.20.34. PrmT38BROverride (Limit Modulation and Bit Rates)

Parameter Number: 0x1b3e

Description: The **PrmT38BROverride** parameter limits the modulation and bit rates for all fax calls. Each modulation has its own range and baud rate that it supports. For V.27, the rates are 2400 and 4800 baud. For V.29, the rates are 7200 and 9600 baud. For V.17, the range is 7200 to 14400. Setting this parameter to V27_ONLY limits the negotiates fax session to just V.27. Setting this parameter to V29_BEST limits the negotiate fax session to V.27 or V.29.

Values:

- 0 = Enable all modulations and bit rates
- 1 = V27_ONLY (disables V.29+)
- 2 = V29_BEST (disables V.17+)

4.20.35. PrmT38TCFThrshld (Local Training Maximum Error Tolerance)

Parameter Number: 0x1b3f

Description: The **PrmT38TCFThrshld** parameter sets the local modem training maximum error tolerance. If the error rate is higher than the value set, the call will retrain at a lower bit rate. The tolerance is set as a percentage of the bit rate.

Values: 0 to 100 (%)

4.20.36. PrmT38IndSecBlocks (Fax IND Redundancy Factor)

Parameter Number: 0x1b4d

Description: ThePrmT38IndSecBlocks parameter defines the number of secondary blocks to send with each fax packet for IND events.

Values: 0 to 15

4.20.37. PrmT38V21SecBlocks (Fax V21 Redundancy Factor)

Parameter Number: 0x1b4e

Description: The **PrmT38V21SecBlocks** parameter defines the number of V.21 secondary blocks to send with each fax packet.

Values: 0 to 15

4.20.38. PrmT38HSECMSecBlocks (Fax HSECM Redundancy Factor)

Parameter Number: 0x1b4f

Description: The **PrmT38HSECMSecBlocks** parameter defines the number of high speed Error Correction Mode (ECM) secondary blocks to send with each fax packet.

Values: 0 to 3

4.20.39. T38HSSecBlocks (Fax HS Redundancy Factor)

Parameter Number: 0x1b50

Description: The **T38HSSecBlocks** parameter defines the number of High Speed secondary blocks to send with each fax packet.

Values: 0 to 3

4.20.40. PrmT38HSEOFSecBlocks (Fax HSEOF Redundancy Factor)

Parameter Number: 0x1b51

Description: The **PrmT38HSEOFSecBlocks** parameter defines the number of high speed End of Frame (EOF) secondary blocks to send with each fax packet.

Values: 0 to 3

4.20.41. PrmT38HSEOTSecBlocks (Fax HSEOT Redundancy Factor)

Parameter Number: 0x1b52

Description: The **PrmT38HSEOTSecBlocks** parameter defines the number of high speed End of Text (EOT) secondary blocks to send with each fax packet.

Values: 0 to 15

4.20.42. PrmT38TxThrshld (Transmit Hold Back Threshold)

Parameter Number: 0x1b40

Description: The **PrmT38TxThrshld** parameter allows you to define the transmit hold back threshold. Increasing the time (in milliseconds) permits the system to better handle timing jitter without generating image errors, but reduces the tolerance for total end-to-end network delay.

Values: 0 to 100 milliseconds

4.20.43. PrmT38TCFMethod (Local or End-to-End Training)

Parameter Number: 0x1b41

Description: The **PrmT38TCFMethod** parameter specifies whether the modem training is local or end-to-end across the network.

Values:

- 0 = Local training
- 1 = End-to-end training

4.20.44. PrmT38TxPower (Transmit Power Level)

Parameter Number: 0x1b42

Description: The **PrmT38TxPower** parameter specifies the transmit power level gain for all modes. The value is set in terms of -dBm0.

Values: -3 to -60 dBm0

4.20.45. PrmT38SpoofLevel (Spoofing Level)

Parameter Number: 0x1b5b

Description: The **PrmT38SpoofLevel** parameter enables proactive algorithms in the T.38 coder to compensate for excessive round trip delay in the IP network. Level 0 (default) is for networks with a maximum 2 seconds round trip delay. Level 1 is for networks with delays up to 3 seconds.

NOTE: Only level 0 is currently supported.

Values:

- 0 = None
- 1 = Level 1

4.20.46. PrmHPFActive (HPF Enable)

Parameter Number: 0x1b1d

Description: The **PrmHPFActive** parameter allows you to enable or disable the high pass filter (HPF). When enabled, the HPF removes DC and very low frequency corruption from the data.

Values:

- 0 = Disable HPF
- 1 = Enable HPF

4.21. [NetTSC] Parameters

The [NetTSC] section only appears in the CONFIG files that are associated with IPLink boards. The following parameters only apply to IPLink boards:

- PrmG711MultyFPP (G.711 Frame Size Overload)
- PrmIPLinkMode (IPLink Mode)
- PrmDebugLevelRAS (Gatekeeper Module)
- PrmDebugLevelH245 (H.245 Channel)
- PrmDebugLevelMsg (Message Module)
- PrmDebugLevelQ931 (Q.931 Channel)
- PrmDebugLevelRVModule (Radvision Module)
- PrmDebugLevelStack (Stack Module)
- PrmDebugLevelRVSTACK (Radvision Stack Module)
- PrmDebugLevelStates (State Machine Module)
- PrmDebugLevelStream (Stream Module)
- PrmDebugLevelTimer (Timer Module)
- PrmDebugLevelUtil (Utilities Module)
- PrmDataDbgLvl (Data Module)
- PrmDebugLevelMNTI (MNTI Module)
- PrmEventsDbgLvl (Events Module)
- PrmExitNotifyDbgLvl (Exit Notify Module)
- PrmInitDbgLvl (Initial Module)
- PrmParamDbgLvl (Parameter)
- PrmRecvMsgDbgLvl (Receive Message)

- PrmSendMsgDbgLvl (Send Message)
- PrmRTCDbgLvl (Run Time Control)
- PrmStandardDisconnectReasonEnabled (Disconnect Reason)
- PrmStateMachineDbgLvl (State Machine)
- PrmNumOfWriteDTMFBuf (Number of Write DTMF Buffers)
- PrmDebugFacility (Facility)
- PrmDebugRVFacility (Radvision Facility)
- PrmDebugMsgSize (Message Size)
- PrmDebugRVMsgSize (Radvision Message Size)
- PrmDebugMode (Mode)
- PrmDebugRVMode (Radvision Mode)
- PrmDebugNumMsg (Message Number)
- PrmDebugRVNumMsg (Radvision Message Number)
- PrmNumOfReadStreamBuf (Number of Read Stream Buffers)
- PrmNumOfWriteStreamBuf (Number of Write Stream Buffers)
- PrmDialogicEnable (Dialogic Enable)
- PrmAsyncStrmTimeOut (Asynchronous Reply Timeout)
- PrmRTCPTimeInterval (RTCP Time Interval)
- PrmCallDurationComput (Compute Call Duration)

4.21.1. PrmG711MultyFPP (G.711 Frame Size Overload)

Parameter Number: 0x1a07

Description: The **PrmG711MultyFPP** parameter instructs the VSR to use a 10 millisecond task for all G.711 Rx regardless of frame size.

Values:

- 0 = Use frame size broadcast in FSE
- 1 = Use 10 millisecond task

4.21.2. PrmIPLinkMode (IPLink Mode)

Parameter Number: 0x1e38

Description: The **PrmIPLinkMode** parameter defines whether applications will use the H.323 protocol stack supplied by Intel or the Internet Protocol Voice Streaming (IPVS) resource. Selecting IPLink will provide embedded H.323 signaling using the H.323 component that uses the RadVision H.323 stack on PPC. Selecting IPVS will enable the application to implement its own protocol stack on the host and use IPVS only for the RTP/RTCP protocol stack.

Values:

- 0 = IPLink (Use the H.323 stack supplied by Intel)
- 1 = IPVS (Use the IPVS resource H.323 stack)

4.21.3. PrmDebugLevelRAS (Gatekeeper Module)

Parameter Number: 0x1e39

Description: The **PrmDebugLevelRAS** parameter sets the H.323 print level for the Gatekeeper Module.

Values:

- 0 = Off
- 1 = Fatal errors only
- 2 = Error (Adds non-fatal error printouts)
- 3 = Warning (Adds warning printouts)
- 4 = Info (Adds trace printouts)
- 5 = Expand (Adds expanded printouts)

4.21.4. PrmDebugLevelH245 (H.245 Channel)

Parameter Number: 0x1e31

Description: The **PrmDebugLevelH245** parameter sets the H.323 print level for the H.245 Channel.

Values:

- 0 = Off
- 1 = Fatal errors only
- 2 = Error (Adds non-fatal error printouts)
- 3 = Warning (Adds warning printouts)
- 4 = Info (Adds trace printouts)
- 5 = Expand (Adds expanded printouts)

4.21.5. PrmDebugLevelMsg (Message Module)

Parameter Number: 0x1e0f

Description: The **PrmDebugLevelMsg** parameter defines the H.323 print level for the Message Module.

Values:

- 0 = Off
- 1 = Fatal errors only
- 2 = Error (Adds non-fatal error printouts)
- 3 = Warning (Adds warning printouts)
- 4 = Info (Adds trace printouts)
- 5 = Expand (Adds expanded printouts)

4.21.6. PrmDebugLevelQ931 (Q.931 Channel)

Parameter Number: 0x1e32

Description: The **PrmDebugLevelQ931** parameter sets the H.323 print level for the Q.931 Channel.

Values:

- 0 = Off
- 1 = Fatal errors only
- 2 = Error (Adds non-fatal error printouts)

- 3 = Warning (Adds warning printouts)
- 4 = Info (Adds trace printouts)
- 5 = Expand (Adds expanded printouts)

4.21.7. PrmDebugLevelRVModule (Radvision Module)

Parameter Number: 0x1e2a

Description: The **PrmDebugLevelRVModule** parametertracks the Radvision Model. This is a bit-flag parameter, where the value is calculated according to the sum of the Radvision stack module values.

For example, to track the CM, CMAPI, and CMAPICB modules, add their values together (0x1 + 0x2 + 0x400 = 0x403) and enter this value for prmDebugLevelRVModule.

The following Radvision stack modules are the most used for debugging IPLink.

Table 60. Radvision Stack Modules - Common

Module Name	Description	Bit Number	Value
TPKTCHAN	To view the content of the H.323 messages	7	0x80
UDPCHAN		8	0x100
CM	Used for debugging purposes	0	0x1
CMAPI		1	0x2
CMAPICB		10	0x400

The following Radvision stack modules should not be used without first consulting technical support.

Table 61. Radvision Stack Modules - Technical

Module Name	Bit Number	Value
LI	2	0x4
LIINFO	3	0x8
SELI	4	0x10
PDLSRC	5	0x20
PDLCHAN	6	0x40
Q93	9	0x200
DEBUG	11	0x800
PI	12	0x1000
TIMER	13	0x2000
SOCKIO	14	0x4000
PDLAPI	15	0x8000
VT	16	0x10000
PDLMISC	17	0x20000
PDLLIST	18	0x40000
PDLCOMM	19	0x80000
PDLPRINT	20	0x100000
PDLFNERR	21	0x200000
PDLCONF	22	0x400000
PDLMTASK	23	0x800000
PDLTIMER	24	0100000
PDLSM	25	0x200000
PER	26	0x400000
CHANNELS	27	0x800000

Table 61. Radvision Stack Modules - Technical

Module Name	Bit Number	Value
NAMECHAN	28	0x1000000

Values: 0 to 0xffffff

4.21.8. PrmDebugLevelStack (Stack Module)

Parameter Number: 0x1e0e

Description: The **PrmDebugLevelStack** parameter defines the H.323 print level for the Stack Module.

Values:

- 0 = Off
- 1 = Fatal errors only
- 2 = Error (Adds non-fatal error printouts)
- 3 = Warning (Adds warning printouts)
- 4 = Info (Adds trace printouts)
- 5 = Expand (Adds expanded printouts)

4.21.9. PrmDebugLevelRVSTACK (Radvision Stack Module)

Parameter Number: 0x1e1e

Description: The **PrmDebugLevelRVSTACK** parameter sets the H.323 print level for the Radvision Stack Module.

Values:

- 0 = Off
- 1 = Info (Adds trace printouts)
- 2 = Statistics
- 3 = Statistics

- 4 = Statistics

4.21.10. PrmDebugLevelStates (State Machine Module)

Parameter Number: 0x1e11

Description: The **PrmDebugLevelStates** parameter defines the H.323 print level for the State Machine Module.

Values:

- 0 = Off
- 1 = Fatal errors only
- 2 = Error (Adds non-fatal error printouts)
- 3 = Warning (Adds warning printouts)
- 4 = Info (Adds trace printouts)
- 5 = Expand (Adds expanded printouts)

4.21.11. PrmDebugLevelStream (Stream Module)

Parameter Number: 0x1e10

Description: The **PrmDebugLevelStream** parameter defines the H.323 print level for the Stream Module.

Values:

- 0 = Off
- 1 = Fatal errors only
- 2 = Error (Adds non-fatal error printouts)
- 3 = Warning (Adds warning printouts)
- 4 = Info (Adds trace printouts)
- 5 = Expand (Adds expanded printouts)

4.21.12. PrmDebugLevelTimer (Timer Module)

Parameter Number: 0x1e12

Description: The **PrmDebugLevelTimer** parameter defines the H.323 print level for the Timer Module.

Values:

- 0 = Off
- 1 = Fatal errors only
- 2 = Error (Adds non-fatal error printouts)
- 3 = Warning (Adds warning printouts)
- 4 = Info (Adds trace printouts)
- 5 = Expand (Adds expanded printouts)

4.21.13. PrmDebugLevelUtil (Utilities Module)

Parameter Number: 0x1e13

Description: The **PrmDebugLevelUtil** parameter defines the H.323 print level for the Utilities Module.

Values:

- 0 = Off
- 1 = Fatal errors only
- 2 = Error (Adds non-fatal error printouts)
- 3 = Warning (Adds warning printouts)
- 4 = Info (Adds trace printouts)
- 5 = Expand (Adds expanded printouts)

4.21.14. PrmDataDbgLvl (Data Module)

Parameter Number: 0x1e26

Description: The **PrmDataDbgLvl** parameter sets the NetTSC debug level for the Data Module.

Values:

- 0 = Off
- 1 = Information
- 2 = Warning
- 3 = Error
- 4 = Fatal

4.21.15. PrmDebugLevelMNTI (MNTI Module)

Parameter Number: 0x1e14

Description: The **PrmDebugLevelMNTI** parameter sets the NetTSC debug level for the MNTI Module.

Values:

- 0 = Off
- 1 = Off
- 2 = Off
- 3 = Off
- 4 = Fatal

4.21.16. PrmEventsDbgLvl (Events Module)

Parameter Number: 0x1e27

Description: The **PrmEventsDbgLvl** parameter sets the NetTSC debug level for the Events Module.

Values:

- 0 = Off
- 1 = Information

- 2 = Warning
- 3 = Error
- 4 = Fatal

4.21.17. PrmExitNotifyDbgLvl (Exit Notify Module)

Parameter Number: 0x1e25

Description: The **PrmExitNotifyDbgLvl** parameter sets the NetTSC debug level for the Exit Notify Module.

Values:

- 0 = Off
- 1 = Information
- 2 = Warning
- 3 = Error
- 4 = Fatal

4.21.18. PrmInitDbgLvl (Initial Module)

Parameter Number: 0x1e35

Description: The **PrmInitDbgLvl** parameter sets the NetTSC debug level for the initial module.

Values:

- 0 = Off
- 1 = Information
- 2 = Warning
- 3 = Error
- 4 = Fatal

4.21.19. PrmParamDbgLvl (Parameter)

Parameter Number: 0x1e37

Description: The **PrmParamDbgLvl** parameter sets the NetTSC debug level for the initial module.

Values:

- 0 = Off
- 1 = Information
- 2 = Warning
- 3 = Error
- 4 = Fatal

4.21.20. PrmRecvMsgDbgLvl (Receive Message)

Parameter Number: 0x1e33

Description: The **PrmRecvMsgDbgLvl** parameter sets the NetTSC debug level for Receive Messages.

Values:

- 0 = Off
- 1 = Information
- 2 = Warning
- 3 = Error
- 4 = Fatal

4.21.21. PrmSendMsgDbgLvl (Send Message)

Parameter Number: 0x1e34

Description: The **PrmSendMsgDbgLvl** parameter sets the NetTSC debug level for Send Messages.

Values**:**

- 0 = Off
- 1 = Information
- 2 = Warning
- 3 = Error
- 4 = Fatal

4.21.22. PrmRTCDbgLvl (Run Time Control)

Parameter Number: 0x1e36

Description: The **PrmRTCDbgLvl** parameter sets the NetTSC debug level for Run Time Control (RTC).

Values:

- 0 = Off
- 1 = Information
- 2 = Warning
- 3 = Error
- 4 = Fatal

4.21.23. PrmStandardDisconnectReasonEnabled (Disconnect Reason)

Parameter Number: 0x1e40

Description: The **PrmStandardDisconnectReasonEnabled** parameter enables the application to select how to implement the disconnect reason mechanism. The application, in compliance with the ITU Recommendation H.225.0, can define the ReleaseCompleteReason by selecting either the Q.931 Cause or the H.225.0 ReleaseCompleteReason as its disconnect reason mechanism.

NOTES: 1. When the parameter is disabled the application uses the disconnect reasons defined in the TSCdefs.h file.

2. When the parameter is enabled the application uses the disconnect reasons defined in the NTSCdefs.h file.
3. A copy of the TSC disconnect values is stored using Disconnect InternalBusy as its prefix instead of CallStateR.
4. The H.225.0 ReleaseCompleteReason values are stored using the prefix Disconnect H2250Reason.
5. The Q.931 Cause values are stored using the prefix Disconnect Q931.

Values:

- 0 = Disable (internal TSC disconnect only)
- 1 = Enabled

4.21.24. PrmStateMachineDbgLvl (State Machine)

Parameter Number: 0x1e24

Description: The **PrmStateMachineDbgLvl** parameter sets the NetTSC debug level for the State Machine.

Values:

- 0 = Off
- 1 = Information
- 2 = Warning
- 3 = Error
- 4 = Fatal

4.21.25. PrmNumOfWriteDTMFBuf (Number of Write DTMF Buffers)

Parameter Number: 0x1e22

Description: The **PrmNumOfWriteDTMFBuf** parameter defines the number of out-of-band DTMF packets buffered between the Voice Stream Resource (VSR) component and the H.323 component.

Values: 0x1 to 0xa

4.21.26. PrmDebugFacility (Facility)

Parameter Number: 0x1e28

Description: The **PrmDebugFacility** parameter determines which facility the output of the H.323 debugging data will be directed to.

Values:

- 0 = STDIO (output is directed to the screen)
- 1 = Stream (output is directed to a file)
- 2 = UDP Socket

4.21.27. PrmDebugRVFacility (Radvision Facility)

Parameter Number: 0x1e29

Description: The **PrmDebugRVFacility** parameter determines which facility the output of the Radvision Stack Module debugging data will be directed to.

Values:

- 0 = STDIO (output is directed to the screen)
- 1 = Stream (output is directed to a file)
- 2 = UDP Socket

4.21.28. PrmDebugMsgSize (Message Size)

Parameter Number: 0x1e2b

Description: The **PrmDebugMsgSize** parameter defines the maximum size of the H.323 debugging printout in bytes. It is used when using an external debugging facility (see **PrmDebugFacility**) instead of sending printouts to stdio, that is, sending printouts to Dialogic\bin\iptdebug.exe utility via GStream or via UDP socket.

Values: 0x0 to 0x100

4.21.29. PrmDebugRVMsgSize (Radvision Message Size)

Parameter Number: 0x1e2c

Description: The **PrmDebugRVMsgSize** parameter defines the maximum size of the RadVision debugging printout in bytes. It is used when using an external debugging facility (see **PrmDebugRVFacility**) instead of sending printouts to stdio, that is, sending printouts to Dialogic\bin\iptdebug.exe utility via GStream or via UDP socket.

Values: 0x0 to 0x100

4.21.30. PrmDebugMode (Mode)

Parameter Number: 0x1e2d

Description: The **PrmDebugMode** parameter defines the mode of the H.323 debugging printouts. In synchronous mode, the printouts are sent to the debug facility (stdio, GStream or UDP see **PrmDebugFacility**). In asynchronous mode, the printouts are written to a large buffer in memory to avoid affecting the system performance. The printouts are read from the buffer using Tornado WindShell tool or iptdebug.exe utility after the problem has been reproduced.

Values:

- 0 = Synchronous mode
- 1 = Asynchronous mode

4.21.31. PrmDebugRVMode (Radvision Mode)

Parameter Number: 0x1e2e

Description: The **PrmDebugRVMode** parameter defines the mode of the RadVision debugging printouts. In synchronous mode, the printouts are sent to the debug facility (stdio, GStream or UDP see **PrmDebugFacility**). in asynchronous mode, the printouts are written to a large buffer in memory to avoid affecting the

system performance. The printouts are read from the buffer using Tornado WindShell tool or iptdebug.exe utility after the problem has been reproduced.

Values:

- 0 = Synchronous mode
- 1 = Asynchronous mode

4.21.32. PrmDebugNumMsg (Message Number)

Parameter Number: 0x1e2f

Description: The **PrmDebugNumMsg** parameter defines the number of printouts kept in the cyclic buffer to be allocated for the H.323 debugging printouts in asynchronous mode. When selecting asynchronous mode (see **PrmDebugMode**) a cyclic buffer is allocated at system initialization. The buffer width is **PrmDebugMsgSize** and the buffer height is **PrmDebugNumMsg.**

Values: 0 to 200000

4.21.33. PrmDebugRVNumMsg (Radvision Message Number)

Parameter Number: 0x1e30

Description**:** The **PrmDebugRVNumMsg** parameter defines the number of printouts kept in the cyclic buffer to be allocated for the RadVision debugging printouts in asynchronous mode. When selecting asynchronous mode (see **PrmDebugRVMode**) a cyclic buffer is allocated at system initialization. The buffer width is **PrmDebugRVMsgSize** and the buffer height is **PrmDebugRVNumMsg.**

Values: 0 to 50000

4.21.34. PrmNumOfReadStreamBuf (Number of Read Stream Buffers)

Parameter Number: 0x1e15

Description: The **PrmNumOfReadStreamBuf** parameter sets the number of packets buffered between the Voice Stream Resource (VSR) component and the

H.323 component. If there is excessive load on the host, or numerous interruptions on the IP network, you may need to adjust this parameter to a higher value.

Values: 0x1 to 0xa

4.21.35. PrmNumOfWriteStreamBuf (Number of Write Stream Buffers)

Parameter Number: 0x1e16

Description: The **PrmNumOfWriteStreamBuf** parameter sets the number of packets buffered between the H.323 component and the Voice Stream Resource (VSR) component. If there is excessive load on the host, or numerous interruptions on the IP network, you may need to adjust this parameter to a higher value.

Values: 0x1 to 0x28

4.21.36. PrmDialogicEnable (Dialogic Enable)

Parameter Number: 0x1e19

Description: The **PrmDialogicEnable** parameter enables Dialogic specific features. This parameter must be configured to use the *TSC_MsgNonStandardCMd* message.

Values:

- 0 = Standard Gateway (disables Intel® Dialogic® specific features)
- 1 = Dialogic Gateway (enables Intel® Dialogic® specific features)

4.21.37. PrmAsyncStrmTimeOut (Asynchronous Reply Timeout)

Parameter Number: 0x1e20

Description: The **PrmAsyncStrmTimeOut** parameter sets the timeout used for IPLink replies to H.323 messages, when they are sent asynchronously.

Values: 0x1 - 0xffff

4.21.38. PrmRTCPTimeInterval (RTCP Time Interval)

Parameter Number: 0x1e3b

Description: The **PrmRTCPTimeInterval** parameter allows the change of the RTCP interval between two adjacent RTCP packets. The default is set at 5 seconds.

Values: 0x3e8 to 0x2710 (1 to 10 seconds)

4.21.39. PrmCallDurationComput (Compute Call Duration)

Parameter Number: 0x1e1a

Description: The **PrmCallDurationComput** parameter specifies the method used to compute the call duration. It is computed starting from the Connected state or from the Initiated state.

NOTE: This parameter must be manually added to the FCD file.

Values:

- 0 = Compute call duration from the Connected state
- 1 = Compute call duration from the Initiated state

4.21.40. PrmIPTDebugIPLeft (IP AddressLeft)

Parameter Number: 0x1e3f

Description: The **PrmIPTDebugIPLeft** parameter sets the left half of the IP address of the debugging station (to which debug information is to be sent). The value of this parameters represents the the first two segements of the dotted decimal IP address converted to hexidecimal. The **PrmIPTDebugIPRight** parameter sets the right half of the IP address.

For example,assume that debugging information is to be sent to the IP address 146.152.172.142. Divide that address into two parts (left and right): 146.152 (first two segments) and 172.142 (last two segments). Convert each number (in both parts) to its hexadecimal representation: 92.98 (left) and AC.8E (right). Remove the

dots (".") leaving 9298 and AC8E. These two hexadecimal numbers are the values for **PrmIptDebugIPLeft** and **PrmIptDebugIPRight**, respectively.

Values: Hexidecimal value

4.21.41. PrmIPTDebugIPLRight (IP Address Right)

Parameter Number: 0x1e3f

Description: The **PrmIPTDebugIPRight** parameter sets the right half of the IP address of the debugging station (to which debug information is to be sent). The value of this parameters represents the the last two segements of the dotted decimal IP address converted to hexidecimal. The **PrmIPTDebugIPRight** parameter sets the right half of the IP address.

For example,assume that debugging information is to be sent to the IP address 146.152.172.142. Divide that address into two parts (left and right): 146.152 (first two segments) and 172.142 (last two segments). Convert each number (in both parts) to its hexadecimal representation: 92.98 (left) and AC.8E (right). Remove the dots (".") leaving 9298 and AC8E. These two hexadecimal numbers are the values for **PrmIptDebugIPLeft** and **PrmIptDebugIPRight**, respectively.

Values: Hexidecimal value

4.22. [0x1d] Parameters

The [0x1d] section only appears in the CONFIG files that are associated with IPLink boards.

The following parameter only applies to IPLink boards:

- PrmTOS (Type of Service)

4.22.1. PrmTOS (Type of Service)

Parameter Number: 0x1d01

Description: The **PrmTOS** parameter sets the Type of Service (TOS) byte in the IP header of transmitted datagrams in order to improve the mobility of the UDP/TCP packets. The TOS byte is set on a per call basis. The **prmTOS** parameter can be set in the FCD file or dynamically by sending a Std_MsgSetParm to the NetTSC component instance.

NOTE: Not all routers recognize a precedence higher than routine (default) and may drop the packets if the default setting is changed.

The IP protocol defines the TOS byte (8 bits) as follows:

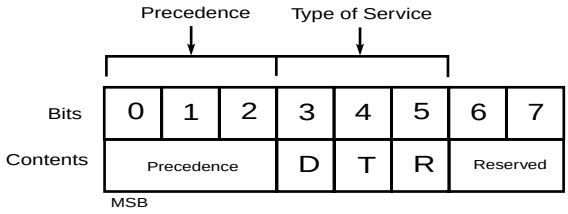


Figure 9. Type of Service (TOS) Byte

The **Precedence** field consists of 3 bits and defines 8 levels of precedence:

Table 62. TOS Precedence Field

Decimal Value	Binary Value	Precedence Level
0	000	Routine (Default)
1	001	Priority
2	010	Immediate
3	011	Flash
4	100	Flash Override
5	101	CRTIC/ECP (Emergency Control Procedures)
6	110	Internetwork Control

Table 62. TOS Precedence Field

Decimal Value	Binary Value	Precedence Level
7	111	National Network Control

The **Type of Service (TOS)** field consists of 3 bits and (Bits 6-7 are reserved and should be zero):

Table 63. TOS Field

Bit	Binary Value	Type of Service	Description
3	0 = Normal 1 = Low (Default)	**D**elay	Select a minimum delay link or circuit for the datagram
4	0 = Normal 1 = High (Default)	**T**hroughput	Select a high throughput link or circuit link for the datagram
5	0 = Normal 1 = High (Default)	**R**eliability	Select a high reliability link or circuit for the datagram

Values: 0 to 255

4.23. [0x1c] Parameters

The [0x1c] section only appears in the CONFIG files that are associated with IPLink boards.

The following parameter only applies to IPLink boards:

- PrmForcesSlowStart (Forces Slow Start)

4.23.1. PrmForcesSlowStart (Forces Slow Start)

Parameter Number: 0x1c05

Description: The **PrmForcesSlowStart** parameter specifies whether or not coder and port information will be provided in H.225 Q.931 setup messages or as separate H.245 messages. Setting this parameter to 0 (FastStart) will cause the coder and port information to be included in the setup message, thus reducing call setup time. Setting this parameter to 1 (SlowStart) will cause the coder and port information to be sent as separate H.245 messages.

NOTE: The **prmForceSlowStart** parameter may be added to IPLink CONFIG files that do not, by default, contain an [0x1c] section. For details about adding the [0x1c] section and the **prmForceSlowStart** parameter to IPLink CONFIG files, see Section 1.3.15, "[0x1c] Section", on page 32.

Values:

- 0 = FastStart (Use a single message)
- 1 = SlowStart (Use many messages)

Glossary

This glossary contains terms specific to the DM3 Configuration File Reference.

0x1b: Section of the CONFIG files that defines parameters relating to data received from the network. The parameters associated with this section have parameter numbers that start with 0x1b and only apply to IPLink technologies.

0x3b: Section of the CONFIG file that defines parameters relating to conferencing.

4ESS: A T-1 protocol switch primarily used for switching digital voice, but it also supports ISDN protocols.

5ESS: A T-1 protocol switch used for switching digital voice and data channels, and supports both basic rate and primary rate ISDN.

AGC: Automatic Gain Control is an encoding process that attempts to maintain a constant volume during voice recording.

alternate mark inversion: See AMI.

AMI: Alternate mark inversion is a form of bipolar signaling in which each successive mark is of the opposite polarity and spaces have zero amplitude.

Automatic Gain Control: See AGC.

base protocol: The protocol implemented by the CHP component. Protocol variants are derived from this base. Compare with *protocol variant*.

B channel: An ISDN bearer channel that carries voice, fax and compressed video.

CAS: Channel Associated Signaling is the DM3 component responsible for managing the generation and detection of digital line signaling functions required to manage voice channels. Channel Associated Signaling also applies to a signaling

method in which the signaling for that channel is directly associated with the channel.

CCS: Common Channel Signaling is the DM3 component that applies to technologies such as ISDN that use common channel signaling. Common Channel Signaling also applies, in general, to a signaling method in which the signaling for a group of channels is carried on a separate (common) channel.

CDP: Country Dependent Parameters file defining parameters necessary for configuring products to different country requirements. This file has a *.cdp* extension.

CEPT: European Conference of Postal and Telecommunications Administrations. A group of European countries organized for the purpose of setting telecommunications standards in Europe.

CFA: Carrier-Failure Alarm.

CHP: Channel Protocol is the DM3 component responsible for implementing the telephony communication protocol that is used on each network interface.

clear channel: A signaling configuration where none of the line's bandwidth is used for signaling. Clear channel signaling is the ability to access telephony channels in the system and configure them to a user defined call control protocol, or to simply leave the lines 'clear'. The resources should have access to the telephony bus for media routing purposes, as well as signal detection, signal generation, and tone generation capabilities, if desired. NFAS is an example of clear channel signaling.

cluster: A collection of DM3 component instances that share specific TDM time slots on the network interface or the SCbus, and which therefore operate on the same media stream data. The cluster concept in the DM3 architecture corresponds generally but not exactly to the concept of a "group" in S.100 or to a "channel" in conventional Dialogic architectural terminology. Component instances are bound to a particular cluster and its assigned time slots in an allocation operation.

CONFIG : A text-input configuration file containing component-specific parameters. This file has a *.config* extension and is used to create an FCD file.

configuration file: See CONFIG file.

configuration file set: A set of files associated with a specific board configuration. All the files in the set have the same name, but different extensions. The set includes the CONFIG, FCD, and PCD files.

Country Dependent Parameters: See CDP.

CRC: Cyclic Redundancy Check.

D channel: An ISDN channel that carries signaling information.

D4: A T-1 protocol switch that supports T-1 robbed bit signaling and provides D4 framing, but does not support ISDN protocols.

DCM: Dialogic Configuration Manager - a software program that allows you to configure system-level and certain board-level parameters. (Windows only).

DM3: An architecture on which a whole set of Dialogic products are built. The DM3 architecture is open, layered, and flexible, encompassing hardware as well as software components.

DMS: A T-1 protocol switch (DMS-100) for primary rate ISDN applications.

DTD: Dial Tone Detection.

E&M: Two-way telephony signaling that uses an “E” (far end) lead and an “M” (near-end) lead. Signaling is accomplished by applying -48 volts dc to the leads.

encoder: The DM3 component responsible for performing an encoding process on a media stream.

FCD: Feature Configuration Description file that lists any non-default parameter settings that are necessary to configure a DM3 hardware/firmware product for a particular feature set. This file has a *.fcd* extension.

FCDGEN: A command-line application that enables you to generate an FCD file from a base CONFIG file.

Feature Configuration Description: See FCD.

fixed routing: A routing configuration where the resource devices (voice/fax) and network interface devices are permanently coupled together in a fixed configuration. Only the network interface time slot device has access to the CT Bus

flash: While the phone is off-hook, quickly pressing and releasing the flash hook to signal the central office or PBX that you are requesting special processing, for example, call waiting.

flash hook: The plunger the phone's handset rests on while on-hook.

flexible routing: A routing configuration where the resource devices (voice/fax) and network interface devices are independent, which allows exporting and sharing of the resources. All resources have access to the CT Bus.

FXO: Foreign Exchange Office - a device at a central site that permits extending PBX services to remote sites. The FXO emulates a phone to the PBX.

FXS: Foreign Exchange Station - a device located remotely from a PBX that permits extending PBX services to remote sites. The FXS emulates a PBX to the remote phone.

ground start: A two-way, two-wire (tip and ring) signaling method similar to loop start in which the current flows in a circuit. Ground start is normally between a PBX and central office and seizure of the line is accomplished by momentarily grounding one of the circuit wires, usually the ring of the tip and ring circuit.

HDB3: A modified AMI signaling code that only applies to E-1 and is used to preserve one's density on the line.

high density bipolar three zero: See HDB3.

in-band signaling: A signaling scheme where both the data and the signaling information for the data are carried over the same channels.

instance: A component instance is an addressable unit within the DM3 software architecture; it represents a single thread of control. The DM3 system resource management and messaging services operate at the instance level. A set of

component instances that make up a resource instance communicate with one another using the DM3 system messaging services. A set of component instances is usually associated with a channel of call processing.

IPVS: IP Voice Streaming

ISDN: Integrated Services Digital Network. See primary rate ISDN.

LAPD: Link Access Protocol for the D channel.

Layer 1: Physical layer of the OSI model that address the physical aspects of network access.

Layer 2: Data Link layer of the OSI model that address data transfer and routing.

Layer 3: Network layer of the OSI model that addresses line communication procedures.

LCON: See LineAdmin.

LineAdmin: Line Administration component responsible for managing line devices.

LOF: Loss of frame.

LOS: Loss of signal.

loop start: A two-way, two-wire (tip and ring) signaling method in which the current used for signaling flows in a circuit (loop) between a telephone and PBX or a telephone and central office. Seizure of the line is accomplished by going off-hook which causes current to flow in a circuit (loop).

LOF: Loss of Frame.

media loads: Pre-defined, numbered sets of features supported by certain DM3 boards.

MLM: DM3 Load Module.

Net5: An E-1 protocol switch. Net5 is a European ISDN primary rate switch.

NFAS: Non-Facility-Associated Signaling is a form of out-of-band signaling where a single ISDN primary rate D channel provides signaling and control for up to 10 ISDN primary rate lines.

NI2: National ISDN-2. A U.S. standard software interface that can be installed on most switch types, providing maximum inter operability with ISDN lines.

NTT: A T-1 protocol switch (INS-Net 1500) that is used by Nippon Telephone and Telegraph (NTT) for primary rate ISDN.

on-hook: The signaling state that occurs when a handset is sitting on the phone (the phone's inactive state) and the flash hook is depressed. Compare with *off-hook*.

off-hook: The signaling state that occurs when the handset is removed from the phone and the flash hook is released. When a phone is taken off-hook it signals the central office or PBX that it needs attention, for example, to make a call or to answering an incoming call. Compare with *on-hook*.

OSI: Open Standards Interconnections. ISO-developed open standards-based framework for inter-system communications. The OSI model categorizes the communication process into seven layers. Layers 1 to 4 address network access and Layers 5 to 7 address messaging.

out-of-band signaling: A signaling scheme where the signaling is carried over channels separate from the channels carrying the data.

PAMD: Positive answering machine detection.

PBLM: Processor Boot Load Module.

PBX: Private Branch Exchange.

PCD: Product Configuration Description file that contains product or platform configuration description information. This file has a *.pcd* extension.

PDK: Protocol Developer Kit.

PDK protocols: CAS protocols.

PDKManager: A command-line application that enables you to download and configure PDK protocols on QuadSpan and DualSpan boards. These protocols are configured separately from the board-level parameters defined in the CONFIG file and are downloaded after system initialization.

PLM: Processor Load Module.

port: A logical entity that represents the point at which PCM data can flow into or out of a component instance or interface in a cluster. The port abstraction provides a high-level means of defining potential data flow paths within clusters and controlling the actual data flow using simple protocols. Ports are classified and designated in terms of data flow direction and the type of entity that provides the port.

primary D channel: the D channel that provides the signaling and control in an NFAS configuration.

primary rate ISDN: An application that uses a single channel to carrie the signaling for all other channels on a line. On a T-1 line, the application uses channels 1 through 23 (B channels) to carry data, digital voice, and compressed video. Channel 24 (D channel) carries the signaling for all 23 B channels. On an E-1 line, the application uses channels 1 through 15 and 17 through 31 (B channels) to carry data, digital voice, and compressed video. Channel 16 (D channel) carries the signaling for all 30 B channels.

Product Configuration Description: See PCD.

protocol variant: A version of the base protocol that has been customized by a set of parameters. This parameter set configures a CHP component to support a particular T-1 telephony protocol. Features such as wink start, DTMF DNIS and MF ANI are enabled and tuned by the parameters in a protocol variant. Compare with *base protocol*.

pulse: A temporary state change from the current signal state to a new signaling state, and then back to the original signaling state. Compare with *sequence*, *train* and *transition*.

PVD: Positive voice detection.

Q.931: Primary rate ISDN D channel signaling protocol standard. (ITU-T Recommendation I.451). The protocol defines the signaling packet, including message type and content, and allows for voice and data transfer on a single trunk.

QSIG: A T-1 and E-1 protocol switch. QSIG is an ISDN signaling and control protocol used for communications between two or more Private Integrated Network Exchange applications (PSS1). The signaling protocol for this standard is defined by Q.931.

R2MF: An E-1 protocol switch. R2MF is and inband common channel signaling protocol that uses channel 16 to convey the signaling for the 30 voice channels. This international signaling system is used mostly in Europe and Asia in non-ISDN applications to permit the transmission of numerical and other information relating to the called and calling subscriber lines.

RAI: Remote Alarm Indication.

Rate Adaption: Conversion of digital data into a different transfer speed (rate) and form.

recorder: The DM3 component responsible for a resource's message exchanges with the host, as well as media stream management and encoder component control functions.

red alarm: An alarm generated by the device at the receiving end of a T-1 or E-1 line to report a loss of signal or frame alignment (synchronization) in the signal being received (incoming data).

resource: A DM3 conceptual entity that provides a specific functionality to a host application. A resource contains a well defined interface or message set, which the host application utilizes when accessing the resource.

Resource firmware consists of multiple components that run on top of the DM3 core platform software (which includes the platform-specific DM3 kernel and device driver).

The DM3 GlobalCall resource is an example of such a resource, providing all of the features and functionality necessary for handling calls on the DM3 platform.

SCD: System Configuration Description file defining the physical parameters of a platform (Linux only). This file has a *.scd* extension.

SCR: Silence Compressed Record is an encoding process that compresses silence during voice recording.

sequence: A set of train signals. Compare with *pulse, train* and *transition.*

Silence Compressed Record: See SCR.

SIT: Standard Information Tones

System Configuration Description: See SCD.

TDM: Time division multiplexing.

TEI: Terminal Endpoint Indentifier. TEI defines which device(s) attached to a BRI ISDN line is communicating with the CO.

time division multiplex: A multiplexing scheme in which a number of low speed digital signals are incorporated onto a high speed line in a byte-interleave pattern.

train: A set of transitions from one signaling state to another in a predefined pattern (set of pulses). Compare with *pulses, sequence* and *transition*.

transition: A permanent state change from the current signal state to a new signaling state. Compare with *pulse, sequence* and *train*.

TS16: An E-1 protocol switch. TS16 is a type of clear channel signaling which allows time slot 16 to be used for data instead of signaling.

TSC: Telephony Service Component is the DM3 component responsible for managing the B channel sets.

yellow alarm: An alarm generated by the device at the receiving end of a T-1 or E-1 line and sent to the device at the transmitting (remote) end to signify that a red alarm condition exists at the receiving (local) end. The yellow alarm is sent to the transmitting device as long as the red alarm condition exists at the receiving end.

wink: A single pulse used sent from a phone, central office, or PBX as part of protocol hand-shaking.

Index

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z

Numerical Index of Parameters