Dialogic

Dialogic® Converged Services Platform Release 8.4.1 Engineering Release 3

Developer's Guide: Overview

February 2008

Copyright and Legal Disclaimer

All contents of this document are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries. Reasonable effort is made to ensure the accuracy of the information contained in the document. However, due to ongoing product improvements and revisions, Dialogic Corporation and its subsidiaries do not warrant the accuracy of this information and cannot accept responsibility for errors or omissions that may be contained in this document.

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

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

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic Corporation or its subsidiaries may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic Corporation or its subsidiaries do not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic Corporation or its subsidiaries. More detailed information about such intellectual property is available from Dialogic Corporation's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. The software referred to in this document is provided under a Software License Agreement. Refer to the Software License Agreement for complete details governing the use of the software.

Dialogic Corporation encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

Dialogic, Dialogic Pro, Brooktrout, Cantata, SnowShore, Eicon, Eicon Networks, Eiconcard, Diva, SIPcontrol, Diva ISDN, TruFax, Realblocs, Realcomm 100, NetAccess, Instant ISDN, TRXStream, Exnet, Exnet Connect, EXS, ExchangePlus VSE, Switchkit, N20, Powering The Service-Ready

Dialogic Product Line Warranty

Unless otherwise stated in an applicable product purchase agreement between the Customer and Dialogic, Dialogic warrants that during the Warranty Period, products will operate in substantial conformance with Dialogic's standard published documentation accompanying the product. If a product does not operate in accordance therewith during the Warranty Period, the Customer must promptly notify Dialogic. Dialogic, at its option, will either repair or replace the product without charge. The Customer has the right, as their exclusive remedy, to return the product for a refund of purchase price or license fee if Dialogic is unable to repair or replace it.

Warranty Period

In the event that you have no signed agreement setting out a warranty period, the Warranty Period shall be the standard warranty period set out on www.dialogic.com on the date of your purchase of the product.

The Warranty Period begins on the date of shipment of any products or software by Dialogic.

The Warranty Period for repaired, replaced or corrected products and software shall be coterminous to the Warranty Provided for the original products or software purchased.

To report warranty claims, Customer may contact Dialogic via email at techsupport@cantata.com or call (781) 433-9600.

Warranty Provisions

A. During the Warranty Period, Dialogic warrants to Customer only that:

(i) Products manufactured by Dialogic (including those manufactured for Dialogic by an original equipment manufacturer) will be free from defects in material and workmanship and will substantially conform to specifications for such products;

(ii) software developed by Dialogic will be free from defects which materially affect performance in accordance with the specifications for such software. With respect to products or software or partial assembly of products furnished by Dialogic but not manufactured by Dialogic, Dialogic hereby assigns to Customer, to the extent permitted, the warranties given to Dialogic by its vendors of such items.

B. If, under normal and proper use, a defect or non conformity appears in warranted products or software during the applicable Warranty Period and Customer promptly notifies Dialogic in writing during the applicable warranty period of such defect or non conformance, and follows Dialogic's instructions regarding return of such defective or non conforming Product or Software, then Dialogic will, at no charge to Customer, either:

(i) repair, replace or correct the same at its manufacturing or repair facility or

(ii) if Dialogic determines that it is unable or impractical to repair, replace or correct the product or software, provide a refund or credit not to exceed the original purchase price or license fee.

C. No product or software will be accepted for repair or replacement without the written authorization of and in accordance with instructions from Dialogic. Removal and reinstallation expenses as well as transportation expenses associated with returning such product or software to Dialogic shall be borne by Customer. Dialogic shall pay the costs of transportation of the repaired or replaced product or software to the destination designated in the original Order. If Dialogic determines that any returned product or software is not defective, Customer shall pay Dialogic's costs of handling, inspecting, testing and transportation. In repairing or replacing any product, part of product, or software medium under this warranty, Dialogic may use new, remanufactured, reconditioned, refurbished or functionally equivalent products, parts or software media. Replaced products or parts shall become Dialogic's property.

D. Dialogic makes no warranty with respect to defective conditions or non conformities resulting from any of the following: Customer's modifications, misuse, neglect, accident or abuse; improper wiring, repairing, splicing, alteration, installation, storage or maintenance performed in a manner not in accordance with Dialogic's or its vendor's specifications, or operating instructions; failure of Customer to apply Dialogic's previously applicable modifications or corrections; or items not manufactured by Dialogic or purchased by Dialogic pursuant to its procurement specifications. Dialogic makes no warranty with respect to products which have had their serial numbers removed or altered; with respect to expendable items, including, without limitation, fuses, light bulbs, motor brushes and the like; or with respect to defects related to Customer's data base errors. Improper packaging of product for repair will not be covered under this warranty agreement. No warranty is made that software will run uninterrupted or error free.

E. Warranty does not include:

a) Dialogic's assistance in diagnostic efforts;

b) access to Dialogic's Technical Support web sites, databases or tools;

c) product integration testing;

d) on-site assistance; or

e) product documentation updates.

These services are available either during or after warranty at Dialogic's published prices.

F. THE FOREGOING WARRANTIES ARE EXCLUSIVE & ARE GRANTED IN LIEU OF ALL OTHER EXPRESS & IMPLIED WARRANTIES (WHETHER WRITTEN, ORAL, STATUTORY OR OTHERWISE), INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. CUSTOMER'S SOLE AND EXCLUSIVE REMEDY AND DIALOGIC'S SOLE OBLIGATION HEREUNDER, SHALL BE TO REPAIR, REPLACE, CREDIT OR REFUND AS SET FORTH ABOVE.

G. IN NO EVENT SHALL DIALOGIC, ITS DIRECTORS, OFFICERS, EMPLOYEES, AGENTS OR AFFILIATES, BE LIABLE FOR ANY COSTS OR DAMAGES ARISING DIRECTLY OR INDIRECTLY FROM YOUR USE OF ANY PRODUCT INCLUDING ANY INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, MULTIPLE, PUNITIVE OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, WHETHER BASED ON CONTRACT, TORT (INCLUDING NEGLEGENCE), STRICT LIABILITY OR OTHER LEGAL THEORY, EVEN IF DIALOGIC, OR ANY OF ITS DIRECTORS, OFFICERS, EMPLOYEES, AGENTS OR AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY EVENT, DIALOGIC'S CUMULATIVE LIABILITY TO YOU FOR ANY AND ALL CLAIMS RELATING TO THE USE OF ANY PRODUCT SHALL NOT EXCEED THE TOTAL AMOUNT OF THE PURCHASE PRICE OR LICENSE FEES PAID TO DIALOGIC FOR SUCH PRODUCT.

H. CUSTOMER AND DIALOGIC HEREBY WAIVE THEIR RIGHT TO TRIAL BY JURY TO THE FULLEST EXTENT PERMITTED BY LAW IN CONNECTION WITH ALL CLAIMS ARISING OUT OF OR RELATED TO THIS WARRANTY, THE PRODUCTS COVERED HEREBY OR THE PERFORMANCE OF ANY PARTY HEREUNDER.

I. THIS WARRANTY SHALL BE CONSTRUED UNDER AND GOVERNED BY THE LAWS OF THE COMMONWEALTH OF MASSACHUSETTS WITHOUT GIVING EFFECT TO ANY CHOICE OR CONFLICT OF LAW PROVISION OR RULE (WHETHER OF THE COMMONWEALTH OF MASSACHUSETTS OR ANY OTHER JURISDICTION) THAT WOULD CAUSE THE APPLICATION OF THE LAWS OF ANY JURISDICTION OTHER THAN THE COMMONWEALTH OF MASSACHUSETTS. CUSTOMER SPECIFICALLY AND IRREVOCABLY CONSENTS TO THE PERSONAL AND SUBJECT MATTER JURISDICTION AND VENUE OF THE FEDERAL AND STATE COURTS OF THE COMMONWEALTH OF MASSACHUSETTS AND SUCH COURTS SHALL HAVE EXCLUSIVE JURISDICTION WITH RESPECT TO ALL MATTERS CONCERNING THIS WARRANTY OR THE ENFORCEMENT OF ANY OF THE FOREGOING.

J. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION.

About this Publication

Purpose

This documentation provides guidelines for using the Dialogic® CSP.

Safety Labels

The following Safety labels may appear in this information product to alert customers to avoidable hazards. The following are in the order of priority:

DANGER

Danger indicates the presence of a hazard that will cause death or severe personal injury if the hazard is not avoided.

WARNING

Warning indicates the presence of a hazard that can cause death or severe personal injury if the hazard is not avoided.

CAUTION

Caution indicates the presence of a hazard that will or can cause minor personal injury or property damage if the hazard is not avoided. Caution can also indicate the possibility of data loss, loss of service, or that an application will fail.

Conventions used

This information product uses the text conventions explained below. In addition, hexadecimal numbers are preceded by a zero and small “x.” For example, the decimal number 15 is represented in hexadecimal as 0x0F.

Convention	Description
. . .	A horizontal ellipsis in an API message indicates fields of variable length.
:	A vertical ellipsis in an API message indicates that a block of information is repeated or is variable.
n	The letter *n* is a generic placeholder for a number.
`Sans serif mono space`	Indicates a command name, option, input, output, non-GUI error, and system messages.
`Sans serif monospace italic`	Indicates a parameter name in an input message. Example: `move *.dot a: c:` *`-s`* The *`-s`* is the parameter.
Serif italic	Indicates the name of a book, chapter, path, file, or API message. Example: *`UserDirectory/Config.exe`*
Boldface	Indicates keyboard keys, key combinations, and command buttons Example: **Ctrl+Alt+Del**
Sans serif boldface	Identifies text that is part of a graphical user interface (GUI). Example: Go to the **Configuration** menu and select **Card**->**Span Configuration**

Contents

4 EXS API Application Development

5 Layer 4 Call Control

9 Configuring the DSP-ONE Card

Overview

Tones

Voice Recorded Announcements

1 Introduction to the CSP

This chapter is an introduction to the CSP.

Converged Services Platform

Overview

Today's telecommunications market is looking for proven technical solutions that can generate incremental service revenue, quickly and profitably. The CSP features an open and multi-functional design that enables developers to quickly create and deliver new revenue-generating services for both legacy and next-generation networks.

The CSP provides a high-performance applications platform that delivers carrier-grade reliability. The CSP also provides best-in-class call control and signaling support in public switched telephone networks (PSTN), packet network, or converged network environments.

The implementation of the following Internet Protocols (IP) with enhanced hardware and software delivers unparalleled features and functionality.

- Session-Initiation Protocol (SIP)
- H.323
- Real-Time Transport Protocol (RTP)

Refer to the *Developer's Guide: Internet Protocol (IP)* for complete information.

Products and Services

The flexible and open programmable architecture allows you to create powerful and differentiated products and services including the following:

- IP Service Node
- Media Services
- IP Centrex/PBX
- Class 4/Toll Bypass
- Protocol Converter/Transcoder
- Single Number Services
- Voice Messaging
- Prepaid Services

IP Service Node

Uses SIP or H.323 protocols to communicate with a softswitch to provide services such as unified messaging, voice recognition, presence, Web, and e-mail applications.

Media Services Provides Digital Signal Processor (DSP) resources for tones, prompts, announcements, conferencing, and interfaces with external resources such as interactive voice response (IVR) or speech recognition.

IP Centrex/PBX Uses circuit and packet interfaces to connect enhanced services applications with enterprises and the PSTN.

Class 4/Toll Bypass Lowers operating costs by eliminating new trunks and point codes with SS7-over IP.

Protocol Converter/ Transcoder Leverages a wide variety of PSTN/IP signaling and IP vocoder standards such as G.711, G723, and G729.

Single Number Services Enables a caller to dial just one number and lets the system use a predetermined set of parameters to reach the subscriber. The CSP software is used in a number of single number service applications.

Voice Messaging Allows you to leave a message when you cannot reach someone. The CSP incorporates new and existing enhanced voice messaging services into a single application platform. These services include interactive voice response, voice achieved dialing, single number, fax-on-demand, and paging.

Prepaid Services Allows customers to pay in advance for long distance and cellular services. Providers can use the CSP to track the time and money used by the subscriber.

System Overview

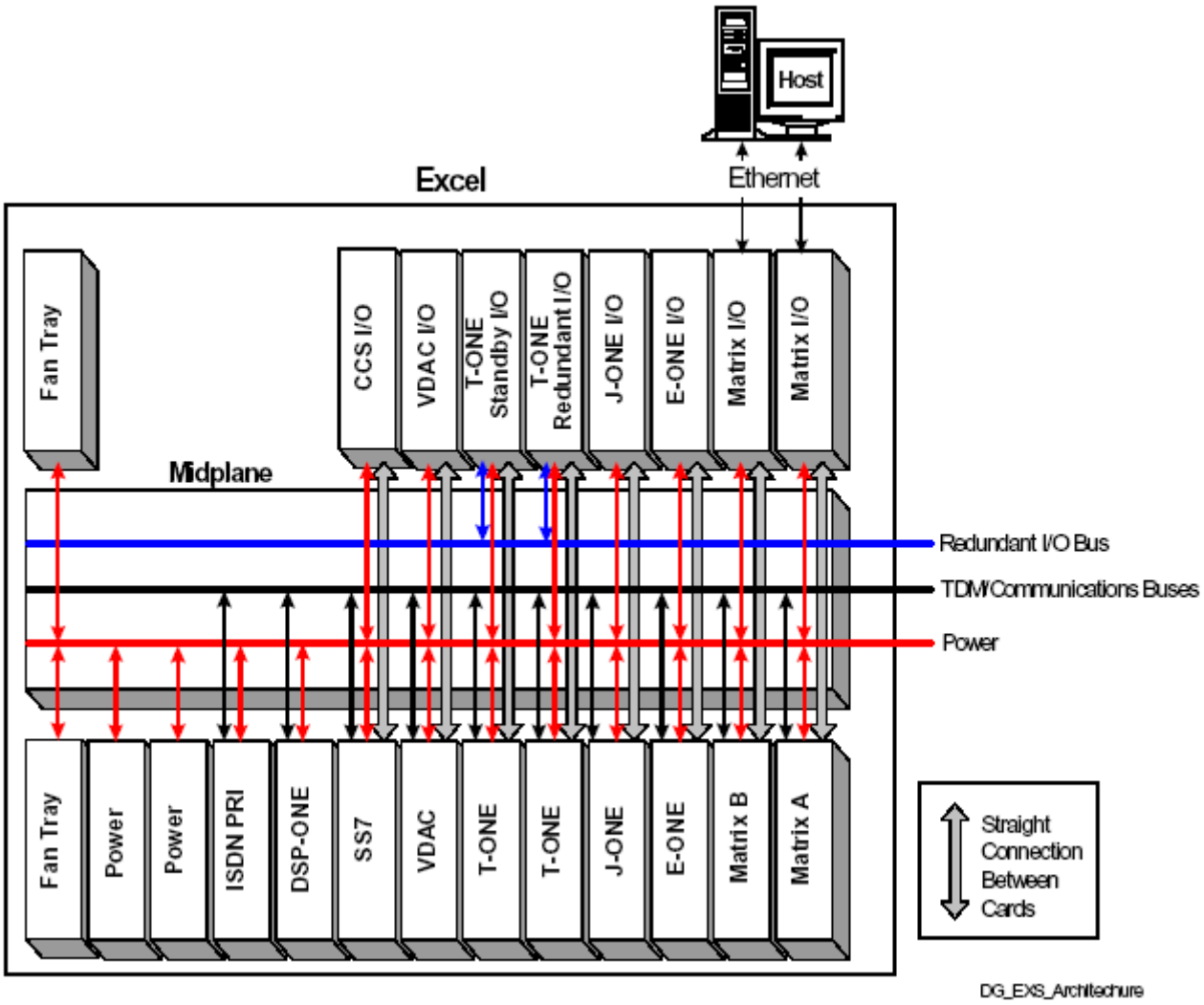
Design of the CSP

The CSP architecture is a marriage of a distributed hardware environment and an open, programmable software environment.

A Single Logical Switch

With the CSP, you can integrate up to seven nodes into a single logical switch.

Figure 1-1 An Overview of the CSP Architecture



Open Architecture for Software Development

The open architecture of the CSP provides the following advantages:

- integrates into a wide variety of networks
- supports a wide variety of signaling protocols
- complies with a wide variety of standards
- supports the host computer and a wide variety of operating environments

The CSP has been designed with an open architecture to operate with any host system, any operating system, and any application language. You can write applications independent of network protocols. And you can use the newest, best switching applications from any number of vendors. CSP software includes a rich set of standard Application Programming Interface (API) messages, so no matter how many different applications you use, you receive CSP information in a consistent format.

Open Architecture for Connectivity

The CSP's rich software environment enables you to:

- Bring new services to market quickly
- Customize your system to support new enhanced services as they emerge
- Adapt to requirements that are specific to a country or region

The CSP's open software architecture lets you program each port individually. So you can integrate a wide variety of applications with Integrated Services Digital Network (ISDN) and Digital Signal 1(DS1) in many different configurations. The system software also supports rate-compatibility with the Conference of European Postal and Telecommunications (CEPT).

The CSP also lets you connect unrelated resources into the logical switch. For example, you can integrate external voice resources for Interactive Voice Response, Voice Recognition, and Voice over Internet Protocol. The CSP can adapt to changing network protocols and standards worldwide.

Modular, Scalable Design

The CSP has a modular design, so you can reuse existing components and upgrade to newer technologies without redesigning or scrapping your existing technology. You can start out with a relatively small system, then as your business needs and revenues grow, increase your capacity incrementally. The CSP uses card sub-assemblies so that new chip technologies can be introduced without re-engineering the underlying main board. This modular approach minimizes redesign costs, which we can pass on to our customers.

DSP Resources

The DSP Series 2 card is a high-performance media processing resource that is fully integrated into the CSP, providing a consistent and easy-to-manage integrated telecommunications and media services environment.

The DSP Series 2 eliminates the need for separate voice response units (VRU) by providing powerful media processing services and resources within the DSP Series 2 environment. This also reduces T1/E1 communications, saving service providers both capital expense costs and on-going operating costs.

The DSP Series 2 enables the CSP to operate as a service node or intelligent peripheral. Configured with DSP Series 2 resources, the CSP can be programmed to inter-operate with speech recognition and/or bulk storage to provide a comprehensive media server solution. The CSP supports the Network File System (NFS) with on-board cache for voice file storage.

> **Important!** The CSP supports Network File System (NFS) Version 2.

See the DSP Series 2 Cards Product Description chapter.

Fault Tolerance through Redundancy

Introduction

The term “redundancy,” as used in this guide, is the ability to configure one or more backup components (or cards) to take over for a component that fails. Redundancy provides system-wide fault tolerance, helping to ensure that the CSP continues to process calls despite a hardware or software fault.

The CSP has a dual bus structure and dual host connections. It can be configured with dual power supplies, and it supports redundant Matrix Controller cards. The CSP also supports N+1 configuration for both line cards and ISDN cards.

Redundant Matrix Controller Cards

In a CSP with redundant Matrix Controller cards, the cards themselves are physically identical. It is only their designation by the host, through a process known as “hardware arbitration,” that determines which is “active” and which is “standby.” The Matrix Controller card designated “active” has full access to all hardware resources, while the card designated “standby” is ready to assume all functions of the active Matrix Controller card under the following conditions:

- The host requests a switchover.
- Software is downloaded to the switch.
- The active Matrix Controller card is removed or physically reset.
- The active Matrix Controller card is reconfigured.
- A hardware or software failure occurs after the system starts up and stabilizes.

Whenever the “standby” card takes over for the “active” card, it becomes the new “active” card, and is recognized as such by the CSP and host.

When system software is downloaded to one of the Matrix Controller cards, the host assigns a timestamp. The timestamp determines which Matrix Controller has the most recently downloaded system software. The system software with the most recent timestamp becomes resident and runs on both Matrix Controller cards.

The two Matrix Controller cards communicate with each other over a reserved, High-Level Data Link Control (HDLC) link. Pulse Code Modulation (PCM) data travels back and forth on the midplane bus between the Matrix Controller cards and line cards.

Redundant Line Cards

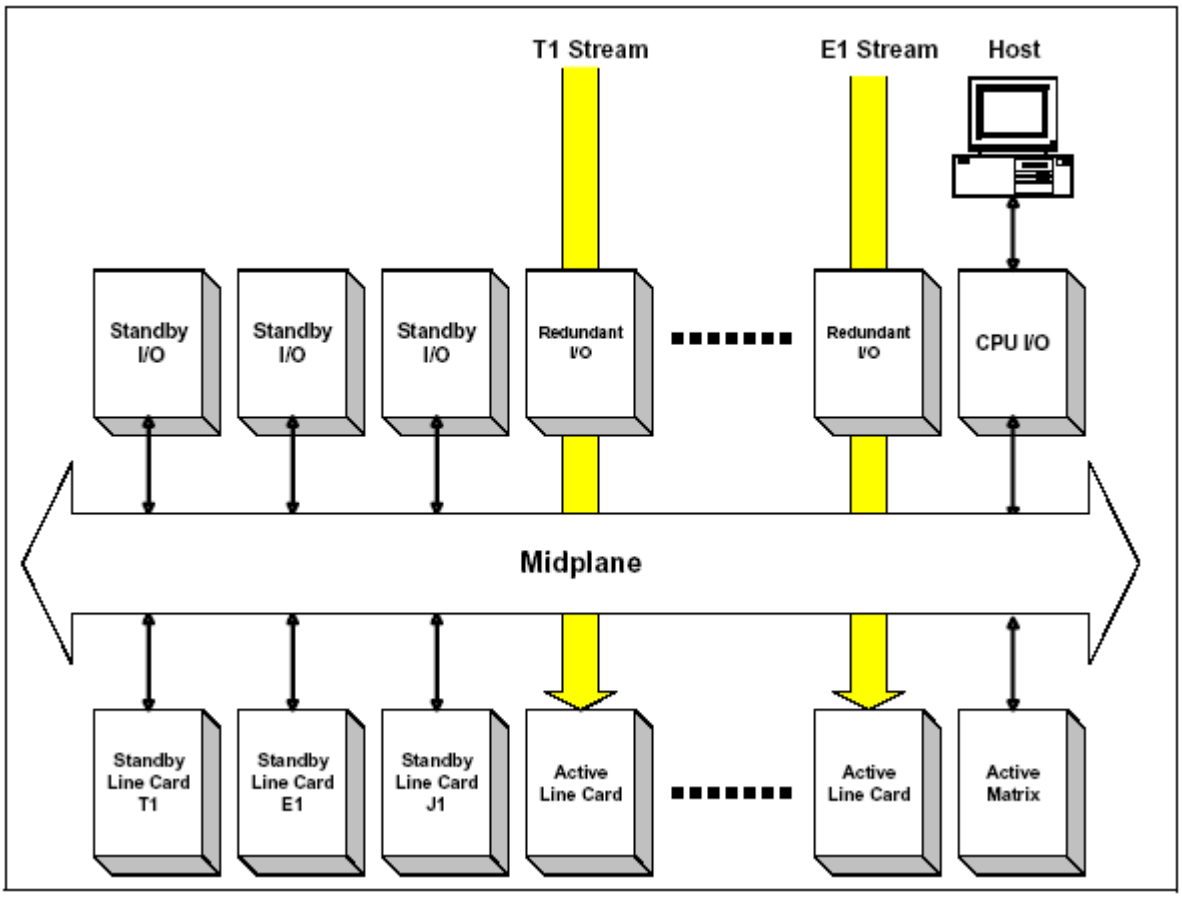
You can provide a CSP with N+1 redundancy for its line cards if you configure one extra line card, together with its I/O card, for each card type. N is the number of line cards of the same type in the CSP, and 1 is the one extra card that serves as a backup in the event of a hardware or software failure. Each redundant line card backs up several cards, but it can only back up a single card type.

For example, if your CSP has three T-ONE cards, you can insert a fourth T-ONE card to serve as a standby card in case one of the active T-ONE cards fails. If your CSP also has multiple E-ONE line cards, you need an E-ONE card in standby mode to achieve E1 redundancy. When a line card fails, the Matrix Controller notifies the host of the failure and the host sends a *Line Card Switchover* (0x0024) message to the Matrix Controller to redirect signaling from the failed line card to the standby line card.

> **Important!** Only one standby line card can switch to active mode at a time.

The Redundant I/O card redirects traffic to the redundant bus which moves traffic to the Standby I/O card. Note that standard I/O cards do not have this capability. The figure below depicts configuration of redundant line cards.

Figure 1-2 Configuration for Line Card Redundancy



Redundant Common Channel Signaling Cards

You can also configure redundancy for Common Channel Signaling (CCS) cards, which include SS7 and ISDN cards. Please see the *Developer's Guide: Common Channel Signaling* book for more information.

Network Interfaces

The CSP supports the following network interfaces:

- E1
- T1
- J1
- DS3

Common Channel Signaling

The CSP supports the following common channel signaling:

- SS7

- ISDN PRI and BRI
- DASS2/DPNSS

Software Overview

This section is an overview of the CSP software, including the following:

- Software layers
- Host/CSP communication
- API messages

Host/CSP Communication

Ethernet is the default communications link between the host and the CSP.

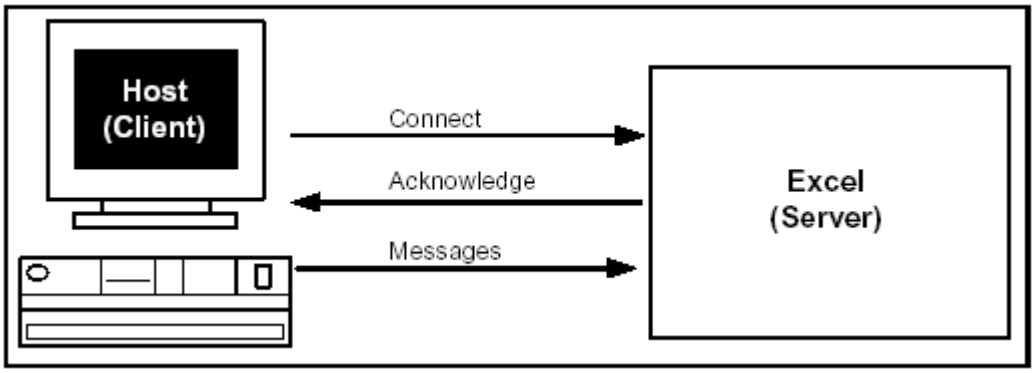
Communication Model

Communication between the CSP and the host is similar to the standard client/server model. In this case, the CSP acts as the server and the host acts as the client. Ethernet communication occurs through the stream transport service TCP. The host issues connect requests to the CSP before issuing commands.

API Messages

The Application Programming Interface (API) messages are a set of high-level, easy-to-use messages that provide host/CSP communication. The host uses the API messages to configure the CSP and to perform call processing functions.

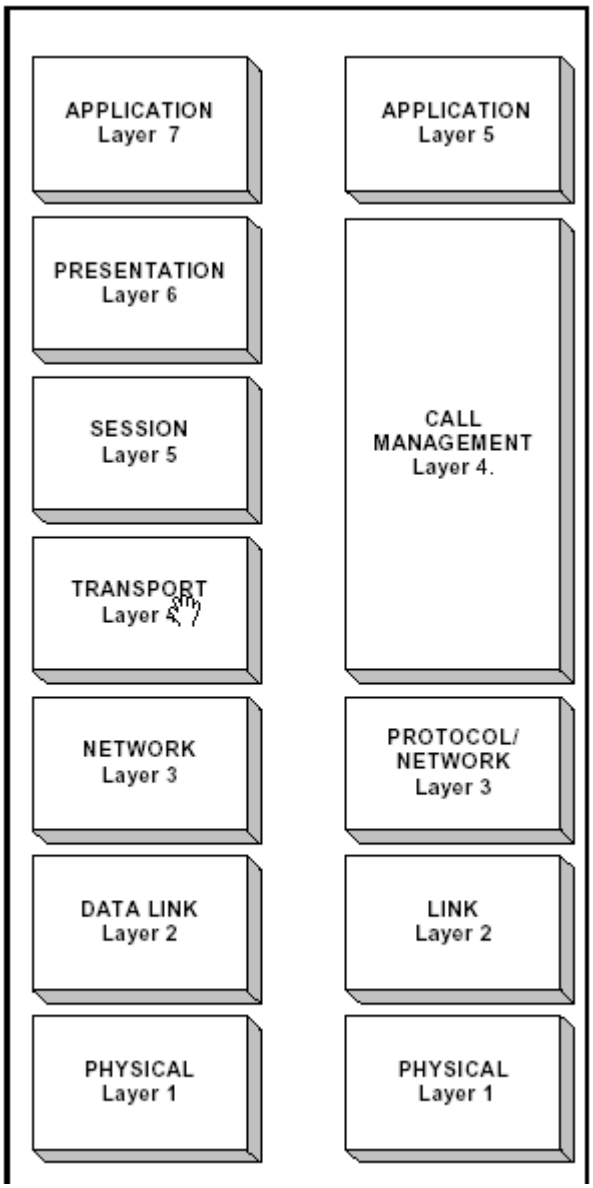
Figure 1-3 Communications Mode



Software Layers

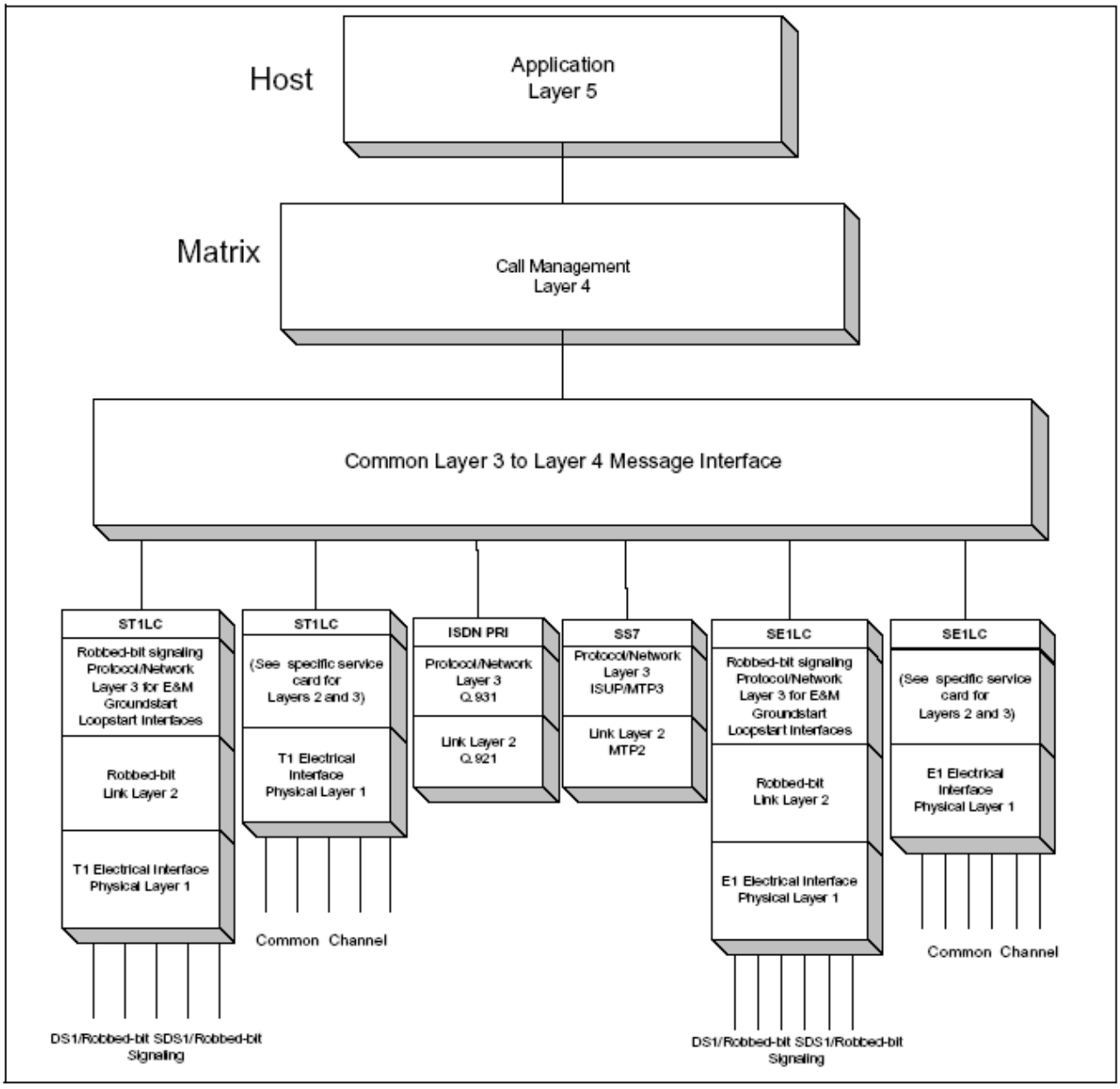
The software is modeled on the standard, seven-layer OSI network protocol stack, but there are important differences between the two. Note that the Call Control layer is a combination of the Presentation, Session and Transport Layers of the OSI network protocol stack, as shown in the figure below:

Figure 1-4 Compare of the OSI stack (left) to CSP software layers



The software layers are shown in the figure below. Call control occurs on the Matrix Controller card and front end protocol processing occurs on the line cards. This distribution ensures accurate, real-time signaling analysis and response within the Protocol/Network layer, with one processor dedicated to 192 ports.

Figure 1-5 Distribution of CSP Software Layers



Layer 5: Host Applications

The Host Applications layer provides internal or external control for high-level, application-specific call processing such as network call routing, hunting, and queueing. Dialogic's high-level application interface lets host applications operate independently of network protocols, so you can develop host applications quickly and easily.

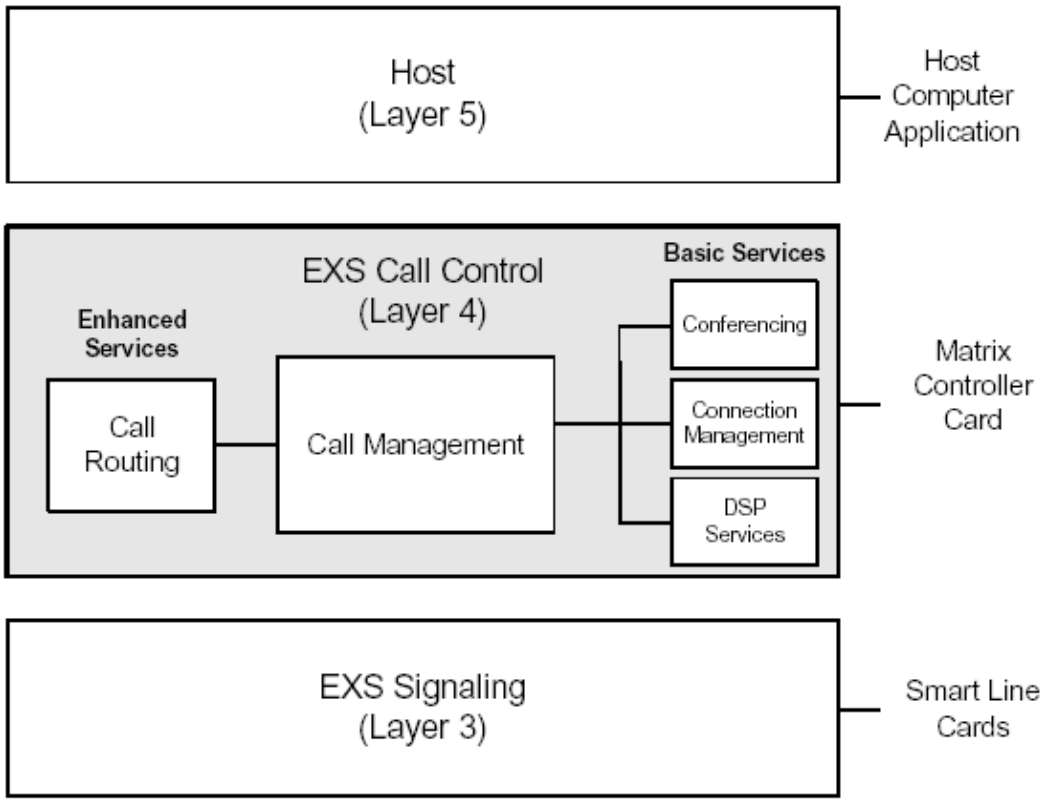
Layer 4: Call Control

The Call Control layer is a combination of the Presentation, Session and Transport Layers of the OSI network protocol stack. The Call Control layer resides on the Matrix Controller card. The host (Layer 5) communicates with the Call Control layer to orchestrate routing, connecting, releasing, parking, and other types of call management.

There are multiple call appearances per channel for flexible call features, such as hold, drop, transfer, conference, and broadcast. The Call Control layer is centralized, so it can also communicate with various network/protocol layers that reside on intelligent line cards.

The Call Control layer also provides network protocol independence to the application layer. Call Control is the default central call processing software for the CSP. It is based upon Dialogic's Programmable Protocol Language (PPL), and it lets you program advanced call processing applications. Call control is managed through the interaction of the host and the signaling layers within the CSP, as shown in Figure 1-6.

Figure 1-6 Call Control Architecture



Call Control provides the following features:

- Network Signaling Control
- Connection Control
- Call Routing
- Interactive DSP Services Management
- Call Control API Messaging
- Call Duration and Timestamp Reporting for generating Call Detail Records (CDR)

Using Call Control, you can quickly program simplified host applications using unique, programmable call models with different combinations of call routes and control parameters. Call Control supports multiple hosts, and the hosts can all use a single simultaneously.

Call Control offers internal call routing features that support user-defined routing, translation tables, and resource group tables. Resource group management simplifies host applications. It lets Call Control use criteria provided by the host to select an available channel for outseizure. In this way, it relieves the host from maintaining the idle/busy status of channels.

You can use the PPL to customize the default call model. You can also use the PPL to program entirely new call models into the Call Control. And you can use PPL generic data fields to pass additional information to and from Call Control. With the PPL, you can build PBX-like features into the call model to be executed automatically, such as one- and two-way connections, call parking, and call hunting. The entire call model can be downloaded to distributed nodes and implemented in an efficient, reliable, and modular way.

Layer 3 Plus

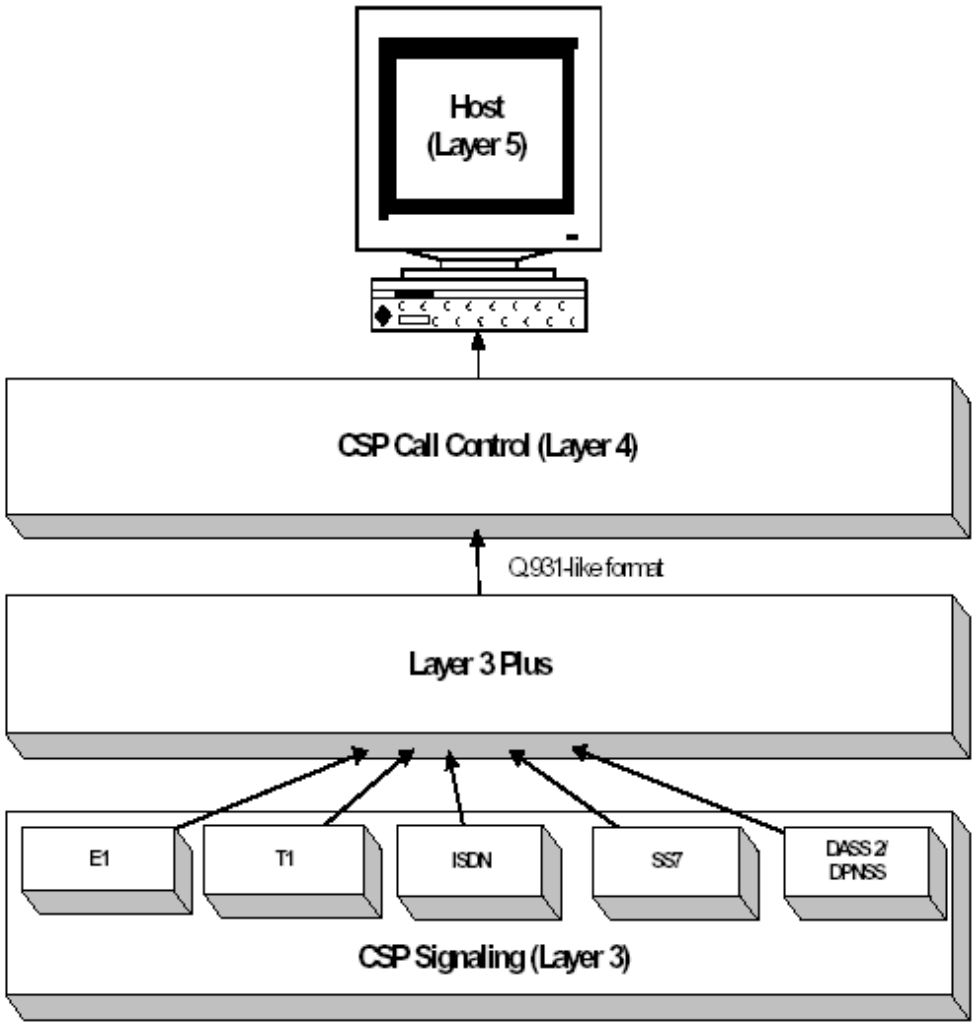
The CSP Layer 3 Plus binds together layers 3 and 4. Layer 3 Plus resides on smart line cards and converts information from layer 3 into a common message interface. Regardless of the signaling protocol residing on a line card, all communication between layer 3 and layer 4 uses this common message interface. Layer 3 Plus also manages application-specific variants for call control.

Layer 3: Network/Protocol Layer

Layer 3 is the line card Network/Protocol layer. Like Layer 3 Plus, Layer 3 resides on the CSP smart line cards. Layer 3 provides a simple interface to new signaling protocols. With the Programmable Protocol Language and this layered, distributed software architecture, the host can define and download custom protocols for signaling variants. The Network/Protocol layer analyzes in-band and out-of-band signaling control of call setup and teardown for incoming and outgoing calls.

The host can also download multiple Network/Protocol layers to address different network variants.

Figure 1-7 Signaling Architecture



Layer 2: Link Layer

The Link layer transfers signaling data between Layer 1 and Layer 3. The Link layer also performs frame synchronization, retransmission, and flow control, such as robbed-bit signaling, frame alarm control, and Q.921 LAPD functions for the ISDN interface.

Layer 1: Physical Layer

The Physical layer provides network electrical interfaces, which include those in the table below:

Table 1-1 Physical network interfaces on Layer

Interfaces	Ports	Standard
E1	2048	International Telecommunications Union (ITU) G.707
T1	1536	American National Standards Institute (ANSI)
J1	2048	Japan Approvals Institute for Telecommunications Equipment (JATE)

2 Licensing Overview

Overview

Purpose This chapter provides an overview of CSP product licensing including:

- General Licensing Information
- System Software Licensing
- Software Modules That You Can License
- Hardware Modules That You Can License

This section refers to other sections in the publication set that contain detailed information regarding licensing specific features.

General Licensing Information

Scalability

Product licensing lets you start with a more affordable set of core features. As your revenues and requirements grow, you can purchase licenses for additional hardware and software modules.

What You Can License

You can license features per CSP or in smaller units. Some licenses are installed on the host computer and remain on the host computer. Other licenses are downloaded from the host computer to the CSP using the API.

Licenses Installed on the Host Computer

- SwitchKit
  Refer *SwitchKit Software Licensing* in the *SwitchKit Installation and Maintenance Guide*.
- IN and Wireless protocol stacks
  Refer to *Licesning of IN Wireless Protocols* in the *IN Wireless Protocols Overview.*

Licenses Downloaded to the CSP

- System Software per chassis
- DSP Resources per chassis
- SS7 User Part Licenses:
  - SCCP/TCAP per chassis **and** active SS7 card
  - ISUP per chassis
  - TUP per chassis
  - IUP per chassis
- M3UA per chassis
- Number of links on an SS7 card
- SS7 Virtual Channels per chassis (2,000 - 10,000 channels in 2,000 channel increments)
- ISDN LAPD per chassis
- QSIG per chassis
- V5.2 per chassis
- IP Network Interface Series 2 channels
- SIP per chassis
- RFC2833 per chassis
- Call Agent with IP Virtual Channels per chassis
- Number of spans on an E-ONE or T-ONE card

Downloading License Keys to the CSP

When you register with Dialogic, we provide you with the product licenses that you purchased for specific hardware and software modules. To activate an additional licensed feature, you must buy a Product License Key. The key is unique and encrypted. You can purchase hardware and software Product License Keys to enable spans, links, protocols, DSP resources, and software modules. To request a Product License Key, you must provide Dialogic with the serial number from the chassis for a software license, or the serial number from the card for a hardware license.

The license key is provided as a text file: usually *LICENSE.CFG* or *SERIALNO.CFG*. For example, the file named *3452.CFG* indicates that this license file can only be downloaded to the CSP Matrix Series 3 Card in chassis serial number 3452.

Send the Product License Key in the Product License ICB (0x24), within the *Product License Download* message (0x0079). This message identifies both the key type and the license authorization code associated with a specific feature. The CSP verifies the key, then activates the feature.

The *Product License Download* message resets the configuration tags for cards. Please refer to the *Tag Configuration* message in the *API Reference*.

> **Important!** You must download the license key each time new system software is downloaded or when you cycle the power.

Querying Product Licenses

The system software keeps track of which features are enabled and reports them to the host in the *Product License Query* (0x007A) message. The message uses the Product License ICB.

Reporting and Querying Card Status

The CSP system software can allocate hardware resources only if they are installed and reported in the *Card Status Report* message. The host monitors *Card Status Report* messages so that it can allocate available resources when it starts the configuration.

The number of spans enabled on the T-ONE, E-ONE, and J-ONE cards is reported in the last byte of the 16-byte Hardware Configuration field in both the *Card Status Query* and *Card Status Report* messages.

The number of links enabled on the SS7 card is reported in Byte 19 in both the *Card Status Query* and the *Card Status Report* messages.

Error Conditions

Errors are reported during download of a product license for the following reasons:

- The product license code is typed incorrectly.
- A key is requested for the wrong card in the system.
- The card serial number is not currently running on the system.
- The key type specified in the API message is invalid or does not match the Product License ICB.
- There are insufficient licensed resources
- There is insufficient hardware

System Software Licensing

Overview

The System Software License is based on a combination of the chassis serial number and the System Software release number. For example, the System Software License for 8.2 is valid for releases 8.2.0, 8.2.1, and 8.2.2. For release 8.3.x, you must download a new license.

System Software Licensing is an evolution from existing Dialogic licensing models. Dialogic already licenses software functions, such as SIP, SS7 User Parts, and Call Agent Mode. Dialogic's licensing has always provided flexibility by allowing you to pay for only as much capacity as you currently need (such as SS7 links, T1/E1 Spans) with the ability to buy more capacity as your needs grow.

The System Software License applies to your base System Software load and to System Software upgrades.

Product License Download (0x0079)

Use this message to download the encrypted System Software Key. The new Key Type, Enable System Software (0x303E), is provided in the Product License ICB (0x24).

Product License Query (0x007A)

Use this message to query the installed System Software Key. The Key Type, Enable System Software (0x303E) appears in the Key Type field of the Response to this message.

Once the System Software is downloaded, the Poll message indicates to the host that the Matrix Controller card has a software load and is ready for configuration. The host then sends the System Software License.

If no license or an invalid license has been downloaded before configuration, the following Response Status Values appear in the responses to the *Product License Download* message (0x0079) or *Product License Query* message (0x007A):

0x4D01 Software Key Found Not Valid

0x4D02 Data Decrypted Not Valid

0x4D03 Product License-Insufficient Licensed Resources

0x4D3D Product License-Invalid ICB Data

0x4D74 Invalid Card Type

0x4DAA Insufficient Hardware

0x007F Software Module Locked

Assign Logical Span ID (0x00A8)

This messages has new logic to check that the System Software Key has been sent to the CSP. If a valid System Software Key has not been sent, the message is NACKed with the following Response Status Value:

Software Modules That You Can License

For software modules, you must provide the chassis serial number from the CSP backplane. After the software module is enabled, it is available to all system hardware. The exception is SCCP/TCAP, which is licensed for each active SS7 card or redundant card pair.

You can license the following software modules:

SS7 User Part Licenses

SCCP/TCAP

To enable SCCP/TCAP you must purchase a license for each active SS7 card.

For each CSP chassis, if the number of SS7 cards or card pairs configured to run the SCCP/TCAP stack exceeds the number licensed, the *SS7 Stack Configure* message (0x005C) is NACKed with the response status value Software Module Still Locked (0x007F).

TUP

To enable TUP on the SS7 card, you must purchase one license per chassis.

ISUP

To enable ISUP on the SS7 card, you must purchase one license per chassis.

IUP

To enable IUP on the SS7 card, you must purchase one license per chassis.

Message Sequence

The following is the message sequence to configure licensing for SS7 User Parts:

1. *Product License Download* (0x0079)
2. *Product License Download* ACK (0x0079)
3. *SS7 Signaling Stack Configure* (0x005C)
4. *SS7 Signaling Stack Configure* ACK (0x005C)

SS7 Virtual Channels You can purchase licenses for up to 10,000 CICs, in 2,000 CIC increments.

IN and Wireless Beginning with Release 8.2.2 CI, Dialogic licenses IN and Wireless protocol stacks on the host. You cannot use IN without TCAP, so you must also purchase a TCAP license to enable SCCP/TCAP.

As mentioned above, to enable SCCP/TCAP on the SS7 card, you must download one license per active SS7 card, even if the CSP was previously licensed for SCCP/TCAP. The IN/Wireless Codec license keys and the SCCP/TCAP licenses are generated together and must be used together.

Wireless keys and SCCP/TCAP licenses require the following information:

- Host ID
- Chassis ID
- Number of Active SS7 cards
- Specific IN and Wireless protocols (one or more of the following: GSMMAP, CAMEL, ANSI-41, WIN, INAP)

For each CSP chassis, if the number of SS7 cards or card pairs configured to run the SCCP/TCAP stack exceeds the number licensed, the *SS7 Stack Configure* message (0x005C) is NACKed with the response status value Software Module Still Locked (0x007F).

SIP To enable SIP on the Matrix Controller card, you must purchase one license per chassis.

Call Agent To use the Call Agent feature, you must configure virtual IP channels in the CSP.

The licensing of virtual channels, based on the CSP chassis serial number, can be from an initial 2,000 channels up to a maximum of 10,000 channels at 2,000 channel increments. To enable more than 2,000 virtual channels you require at least two licenses, one to enable the initial virtual 2,000 channels, and a second license to incrementally increase to greater than 2,000 virtual channels.

Refer to *Virtual IP Channels* in the *Developer's Guide: Internet Protocols* for more information.

You also must license SIP software to use the Call Agent feature.

The RFC 2833 Parser is a separately licensed feature that is bundled with the Call Agent feature.

QSIG To enable QSIG/PSS1 Basic Call Signaling, you must purchase one license per chassis.

ISDN LAPD To enable ISDN LAPD you must purchase one license per chassis.

V5.2 To enable V5.2 you must purchase one license per chassis.

Resource Points on the DSP-2 Card

The DSP Series 2 card uses a pooling scheme for dynamically allocating resources, through Resource Points. Resource Point licensing provides the DSP Series 2 with flexibility, scalability, and redundancy. The total available Resource Points used for Tone Reception, Conferencing, and File Playback/Record are managed as a resource pool.

When you insert the card in the chassis, the Resource Points are reported to the Matrix Controller card and added to the pool. Each function type has associated licensing per channel, measured in Resource Points.

Refer to DSP Resource Points for more information.

SwitchKit SwitchKit licenses are available for single-node and multi-node CSP configurations. Both types of licenses are matched to the chassis IDs of each node in the CSP. SwitchKit applications can connect to any LLC that is running with a valid software license. Independent software licenses are not required for SwitchKit applications.

Refer to *SwitchKit Software Licensing* in the *SwitchKit Installation and Maintenance Guide*.

Hardware Modules That You Can License

Overview

You can license the following hardware modules:

- SS7 Links on the SS7 Card
- Spans on the T1, E1, and J1 Line Cards

Licenses for Cards

For hardware modules on cards, you must provide the serial number on the card and the number of spans or links you want to license.

You can buy product licenses to enable hardware modules. You can buy a card initially enabled for 2, 4, 6, 8, 10, 12, 14, or 16 links (for an SS7 card) or spans (for a T-ONE, E-ONE, or J-ONE line card). As your needs grow, you can buy product licenses to upgrade the card in 2-link or 2-span increments as many times as you like, to the capacity of the card. You can upgrade a card in a CSP that is on-line, without interrupting service.

> **Important!** Hardware licenses are not cumulative. For example, if you have a six span card and you purchase two spans you will have eight spans total. To upgrade to a 12 span card you will receive a license for six spans (after having purchased four additional spans). You discard the two span license because it will not be used.

All modules that you buy originally and that ship with cards remain available at all times. For example, if an SS7, T-ONE, E-ONE, or J-ONE card is removed, if the configuration is lost, or if the system is upgraded, the licensed modules that shipped with the cards are immediately available when the system is brought back up. However, for modules that you buy licenses for later, this is not the case. You must re-download these added module licenses to re-instate those upgrades.

Licenses for Standby Line and Standby SS7 Cards

A standby line card must support the maximum number of spans of all the card it is backing up to provide total redundancy. Therefore, for a standby line card to back up all the spans on a card, it must be licensed for all the spans. For example, you might have two T-ONE cards and a standby line card, each licensed for eight spans. If you upgrade one of the line card licenses to 16 spans but you do not upgrade the standby card license to 16 spans, the standby line card will still back up only eight of the 16 spans on that card.

A Product License Key is required for both active and standby line cards. For example, if you purchase a license to upgrade a T-ONE line card by eight spans, you must also purchase a license to upgrade its corresponding standby line card by eight spans or already have the capacity in the standby card.

Before you can upgrade a standby card, it must be in an active state and the card's Product License Key must be validated.

Hardware upgrade information cannot be transferred to a standby card if the original active card has been removed. When the active card is re-installed, the information for module decryption must be downloaded again.

Licenses for Standby Matrix Controller Cards

When a Matrix Controller card changes from Standby to Active states, the following occur:

1. The newly active Matrix Controller card receives the download for the product license. Only the active Matrix Controller card can provide license information to the other cards.
2. The newly standby Matrix Controller card also receives the product license information.

3 Getting Started

Purpose This chapter presents tasks you need to perform to get your CSP to the point where you can begin running an application.

Installing and Configuring a TFTP Server

Overview You must install and configure a TFTP server to download system software files. The TFTP configuration file, which indicates the individual .bin files to download, is stored on the TFTP Server. For a full TFTP download, all software .bin files are stored on the TFTP Server.

UNIX systems have an embedded TFTP Server. If you are using a Windows NT® system, you must acquire third-party TFTP server software.

UNIX To enable the TFTP server, you must edit the inetd.conf file.

1. Open the file and locate the line shown below:

```
#tftp   dgram   udp     wait    root    /usr/etc/in.tftpd
   in.tftpd   -s   /tftpboot
```

2. Remove the pound symbol (#) to enable the TFTP Server.

```
tftp   dgram   udp6     wait    root    /usr/sbin/in.tftpd
   in.tftpd   -s   /tftpboot
```

3. Restart the in.inetd process by rebooting the host.
4. On a Sun Microsystems Operating System, you can use the KILL command to start the process by entering the following:

```
kill -process_id of the in.inetd -HUP
```

5. To locate the process ID, enter the following:

```
ps -axe | grep in.inetd
```

Installing and Configuring a BOOTP Server

To have system software downloaded automatically upon start-up, you must install a BOOTP Server. UNIX systems have an embedded BOOTP Server. If you are using an NT system, you must acquire third party BOOTP server software.

You must configure the BOOTP Server with the following:

- Location of the TFTP Configuration File
- The IP address and hardware address of the Matrix Controller card to which the system software .bin files are to be downloaded

UNIX

To configure the TFTP server on a UNIX system, you must modify the following files:

- ethers
- bootptab

Important! If you are using a Solaris OS, you must create these two files.

Modifying the Ethers File

You must enter the hardware address of the Matrix Controller card to which the software is to be downloaded. A sample entry to the ethers file is shown below.

00:20:1C:01:14:3C rack1

Modifying the bootptab File

In the bootptab file you must make entries for each CSP defining the following:

- Hardware Type (ht)
- Host Hardware Address (ha)
- Host IP Address (ip)
- Subnet Mask (sm)
- Boot File (bf)

Sample bootptab Entry

```
# Bootp clients
rack1:\
        :ht=ethernet:\
        :ha=0x00201C01143C:\
        :ip=10.1.219.101\
        :sm=255.255.0.0:\
        :bf=/tftpboot/tftpcfg:
```

Download Scenarios

Single Matrix Controller Card

For a CSP with a single Matrix Controller card, download the system software to storage memory. When the download finishes, the Matrix Controller card resets and copies the download from storage to executable memory.

Redundant Matrix Controller Card

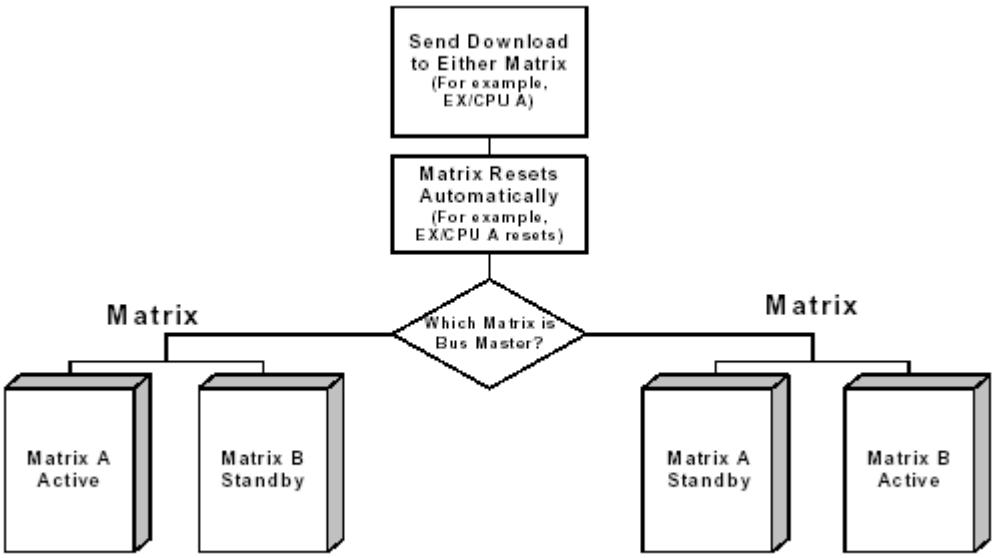
In redundant system (those with two Matrix Controller cards) download the system software to either Matrix Controller card. Depending on which Matrix Controller card has control of the bus, one of the following scenarios occurs:

If Matrix Controller card A in has control of the bus after the download finishes, the following occurs, as shown in Figure 3-1:

1. Card B remains in the Boot state. Card A enters the Initialize state from the Boot state.
2. Card B remains in the Boot state. Card A enters the standby state.
3. Card B remains in the Boot state. Card A enters the Switchover state, sends the system software to the line cards, and enters to the Active state to become the active Matrix Controller.
4. Card B requests the software load from Card A. Card A sends the software load to Card B.
5. Card B boots, resets, initializes, and enters the standby state to become the standby matrix controller.

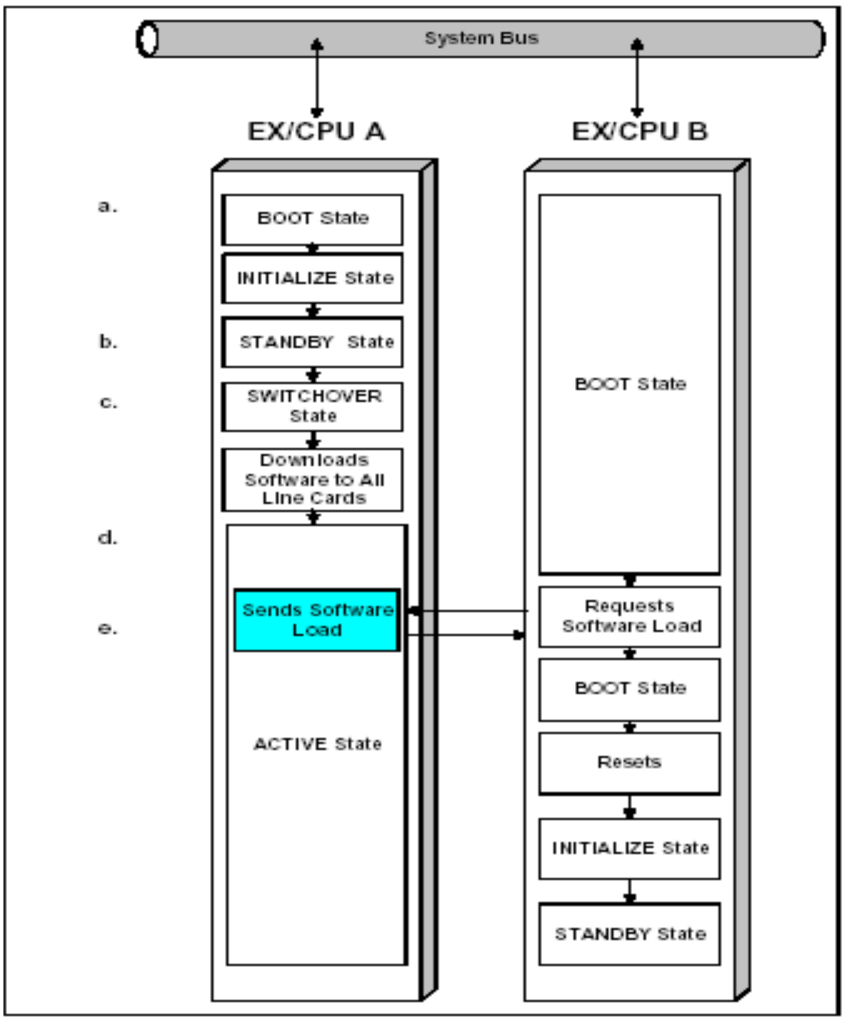
Important! As each Matrix Controller card transitions between states, it sends a *Poll* message to the host. The *Poll* messages do not indicate which Matrix Controller card has control of the system bus. You must look at the front panels on the Matrix Controller cards, where a green LED indicates the card with control of the system bus.

Figure 3-1 Downloading to Redundant Matrix Controller Cards



Dialogic recommends that you design your host application to stop polling before the download begins. But if you do so, you must send a *Poll Interval Configure* message when the download completes.

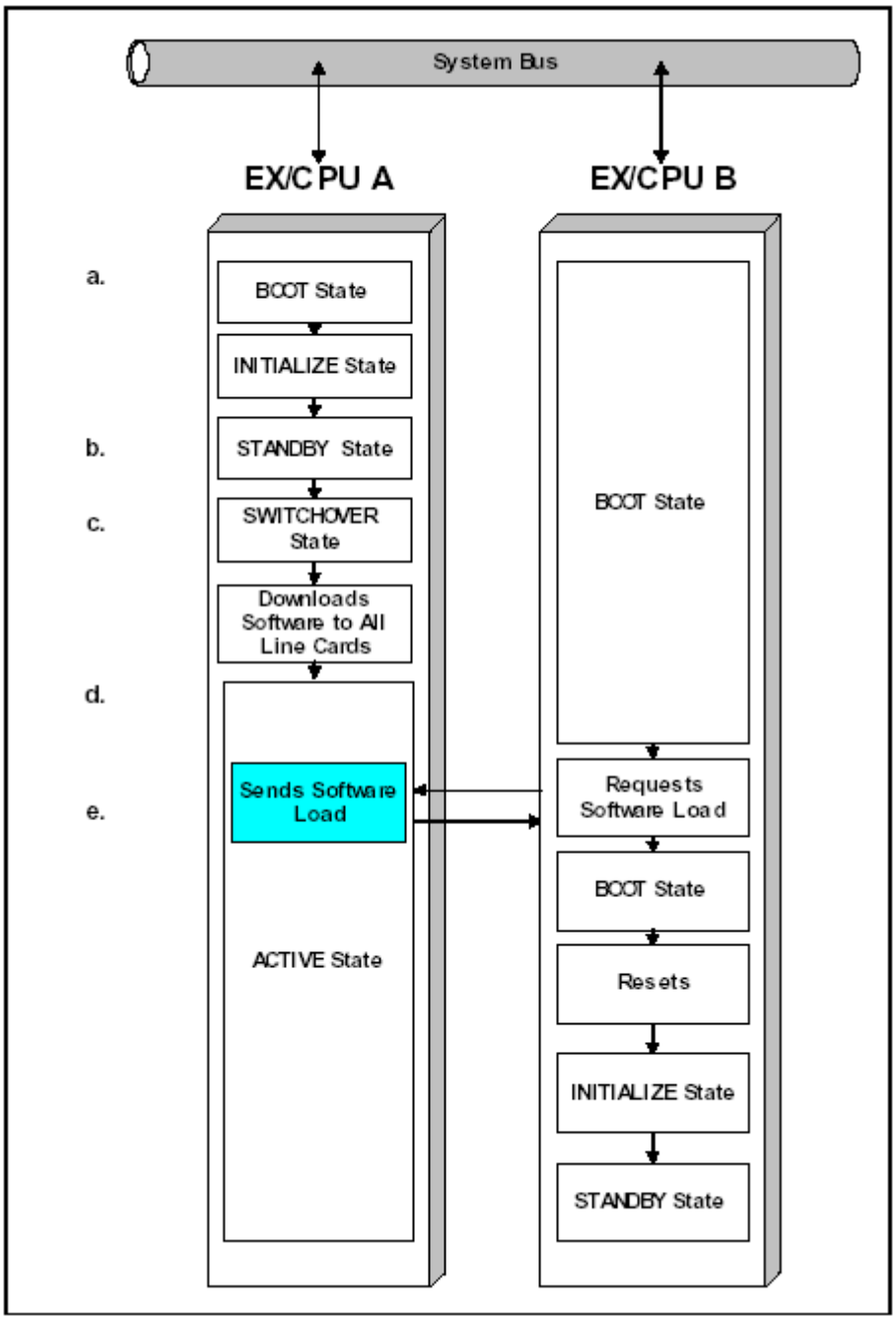
Figure 3-2 Download with Matrix Controller A Bus Control



If Matrix Controller B has control of the bus, the following scenario applies, as shown in Figure 3-3:

- Card A is in the Initialize state.
- Card B requests the system software load from Card A.
- Card A sends the system software load to Card B.
- After Card B receives the system software, it resets, initializes, enters the standby state, and progresses to the Switchover state. Card A remains in the Initialize state.
- Card B downloads the system software to all the line cards, enters the Active state, and becomes the active Matrix Controller. Card A enters the standby state and becomes the standby Matrix Controller.

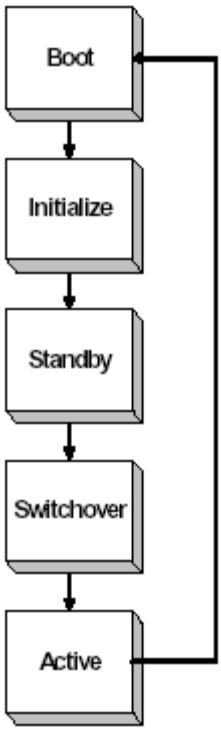
Figure 3-3 Download with Matrix Controller B Bus Control



Software Start-up States

The system software transitions through the states shown in the figure below.

Figure 3-4 System Software States



Boot State

The system software starts in the boot state. The ROM boot loader verifies an existing system software download. If you are downloading new system software from the host or from the adjacent Matrix Controller, the boot loader acquires and verifies the new software download. When the ROM boot loader verifies the download, it copies the download from storage space into execution space, then begins to execute it. Every time the Matrix Controller resets, it enters the Boot state.

Initialize State

After the system software initializes resources, it enters the Initialize state, and sends a *Poll* message to the host. In the Initialize state, it synchronizes its database with the active Matrix Controller or, if the adjacent Matrix Controller is not active, it resets the database.

Standby State

When the database is ready, the system software enters the standby state.

Switchover State

A Matrix Controller remains in the standby state until one of the following conditions occurs:

- The adjacent Matrix Controller is removed, reset, or experiences a failure.
- The host sends a *Become Active* message to the Matrix Controller.
- The adjacent Matrix Controller completes a download.

If any of the above conditions occurs, the Matrix Controller gains control of the system bus and enters the Switchover state. The Matrix Controller then gains control of the CSP and enters the Active state, and the system software immediately enters the Switchover state.

To perform a successful switchover, the active Matrix Controller must be in the Active state, and the standby Matrix Controller must be in the standby state.

If a switchover occurs when the standby Matrix Controller is in the Initialize state, (requesting information from the active Matrix Controller) the standby Matrix Controller resets after all tasks on the card are set to a known state. If the active Matrix Controller resets while the standby Matrix Controller is in the Boot state, the standby Matrix Controller progresses to the Active state but does not update any of the information held on the previous active Matrix Controller.

Active State

When the system software has completed its tasks in the Switchover state, it enters the Active state. The system software then sends a *Card Status Report* message to the host for each card in the CSP. The host must monitor the *Card Status Report* messages so that when it begins configuration, it can properly allocate resources.

When the system software is ready, the CSP sends a *Poll* message, with its *Ready for Configuration* bit set. The system software is now fully operational. The system software stays in the Active state until the Matrix Controller resets.

The Matrix Controller resets when one of the following occurs:

- The host sends a *Become Active* message to the standby Matrix Controller.
- The host sends a *Reset Matrix* or *Reset Configuration* message to the Matrix Controller.
- The Matrix Controller completes a download of system software with a more recent timestamp.

- The Matrix Controller experiences a hardware and software failure.

Configuration Guidelines

The system software cannot perform all actions in all states. The host must operate in accordance with the following restrictions of each state:

- If the system software is in any state other than Active, only the Core Message Set is available. The Core Message Set is a small subset of the API messages. Use the Core Message Set only for downloading system software and for tracking the status of the Matrix Controller.
- The system software can use hardware resources only if they are reported in the *Card Status Report* message.
- Before the host can configure the CSP, the Active system software must send a *Poll* message with the *Ready For Configuration* status bit set.
- The host monitors the *Poll* messages status bit, "Download this Matrix", and initiates a download if it is set. The system software cannot leave the Boot state until it has received a valid download.
- The host monitors all the status bits in the *Poll* message for indications of failure which prevents the CSP from operating with defective hardware.
- Only one Matrix Controller transitions to the Active state. The host can either allow the CSP to select which Matrix Controller becomes Active, or it can send a *Become Active* message to the Matrix Controller that it wants to address as the active Matrix Controller.

In a non-redundant system, the system software runs in exactly the same states and follows the same rules. The only difference is that the single Matrix Controller always runs in the active state.

Configuring a Shared IP Address Using BOOTP Server

This procedure explains how to configure a shared IP address using the BOOTP server.

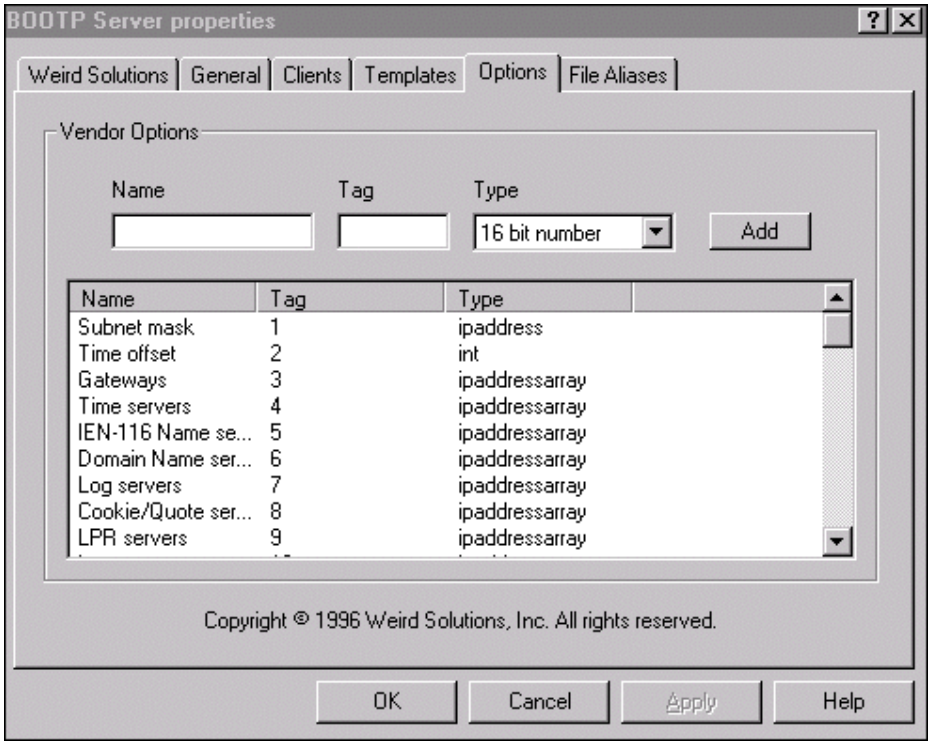
> **Important!** To enable SIP UA Redundancy, you must configure a shared IP address. The shared IP address is for redundancy only. All API messages should be sent directly to the active matrix card's IP address - not to the shared IP address.

You can also use the DHCP server to configure a shared IP address. See *Configuring a Shared IP Address Using DHCP Server.*

This procedure uses Weird Solutions BOOTP Server NT running on Windows NT® but you can use any BOOTP server running on any platform.

1 Double-click **BOOTP Server NT** from the Control Panel to display the Weird Solutions BOOTP Server NT dialog box.

2 If you have already defined the Shared IP address vendor, you can proceed to Step 5; otherwise, select the **Options** tab.



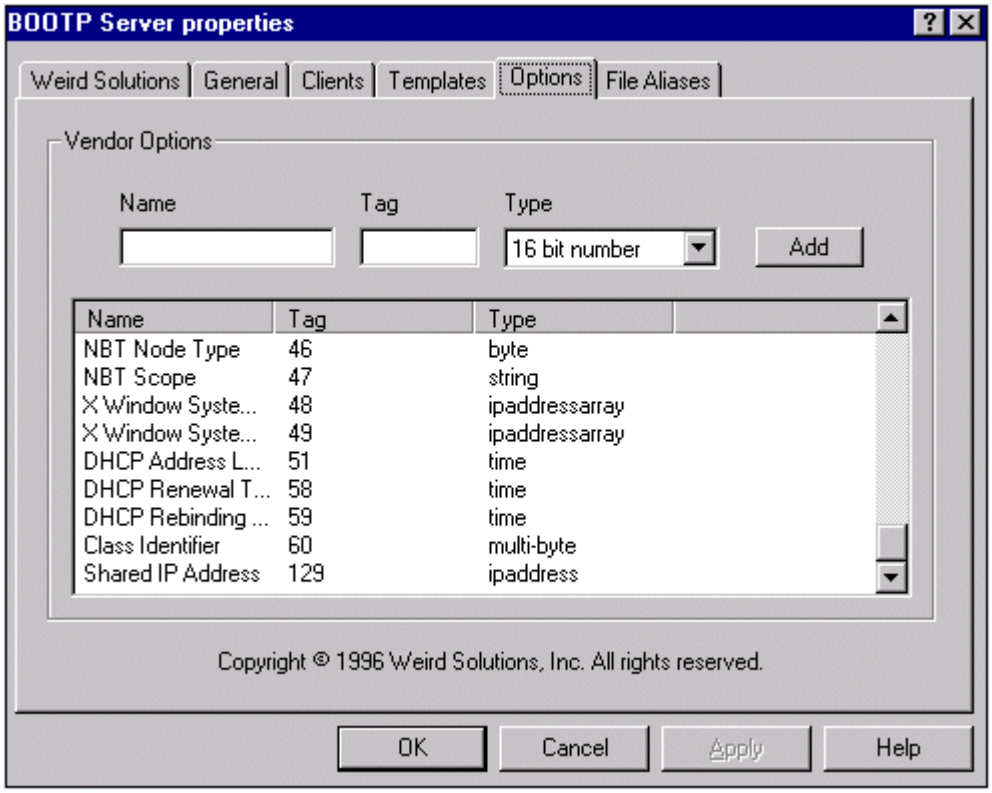
3 Complete this information as follows:

a. Type **Shared IP Address** in the **Name** box.

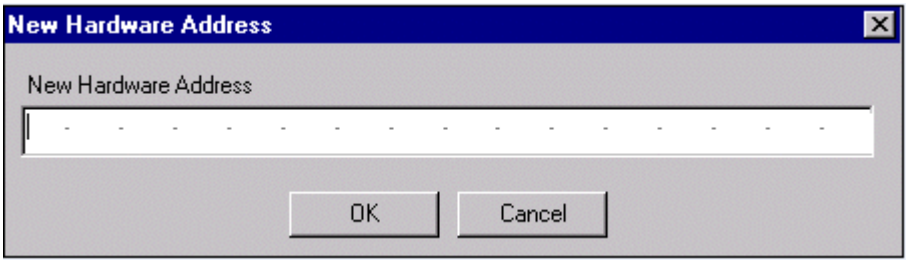
b. Type 129 in the **Tag** box.

c. Select **ipaddress** in the **Type** box. Click **Add**.

4 Scroll down the list of names on the **Options** page to verify that your entry appears with the correct information.

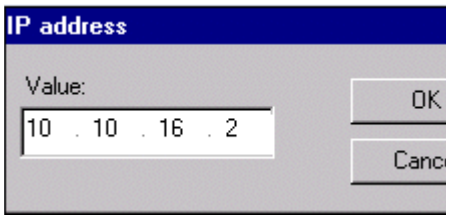


5 Select the **Clients** tab. Select the hardware address of the active Matrix Controller card of the redundant pair in the **Hardware Address** box. Click **New** and create the new hardware address if it is not listed.



6 If the IP Address for the Matrix Controller card is not already created, do the following:

If **IP address** is not listed in the **Configured options** window, select **IP address** from the **Available options** window and click the >> button to move it into the **Configured options** window.



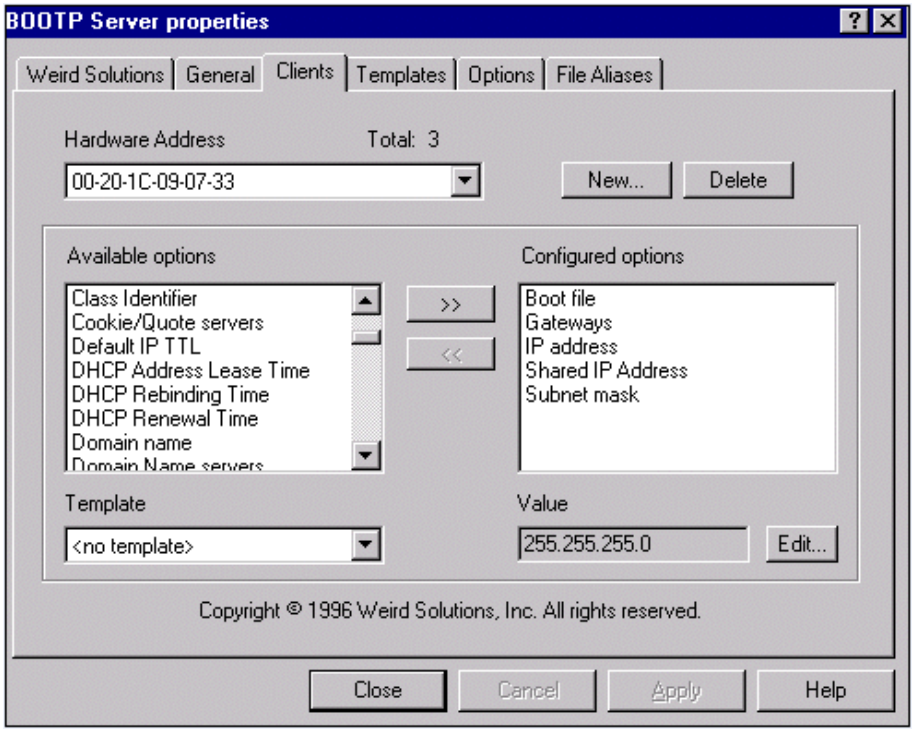
7 Configure the Shared IP Address. Select the **Shared IP Address** in the **Available options** window and click the >> button to move it into the **Configured options** window.

8 Select the option to change in the **Configured options** list and click **Edit** to set or change the value of the IP address or Shared IP Address. Type the new value and click **OK**.

9 Repeat Steps 5-9 to configure the standby Matrix Controller card of the redundant pair.

Configuring Subnet Mask, Gateways, and the Bootfile

You can also configure the subnet mask, gateways, and the bootfile using steps 5-9.



Configuring a Shared IP Address Using DHCP Server

This procedure explains how to configure a shared IP address using the DHCP server. This procedure is required to enable SIP UA redundancy.

> **Important!** To enable SIP UA Redundancy, you must configure a shared IP address. The shared IP address is for redundancy only. All API messages should be sent directly to the active matrix card's IP address - not to the shared IP address.

You can also use the BOOTP server for this procedure. See *Configuring a Shared IP Address Using BOOTP Server.*

This procedure uses the DHCP running on the Solaris system but you can use any DHCP server running on any platform.

1 If the Solaris system is already set up as a DHCP server, then proceed to Step 6.

2 Log in as `root` in a shell.

3 Create the DHCP directory by typing `mkdir /var/dhcp`

4 Configure DHCP with the command: `dhcpconfig`

- Select **1** for **Configure DHCP Service.**
- Select **Y** for **Would you like to stop the DHCP Service?**
- Enter files for **Enter datastore (files or nisplus).**
- Press **Enter** for the default for **Enter absolute path to datastore directory (/var/dhcp/).**
- Select **N** for **Would you like to specify nondefault daemon options?**
- Enter **0** for **Enter default DHCP lease policy (in days)?**
- Select **Y** for **Do you want to allow clients to renegotiate their leases?**

- Select **N** for **Enable DHCP/BOOTP support of networks you select.**
- Select **Y** for **Would you like to restart the DHCP service?**
- Select **4** to exit.

5 Edit the **/etc/init.d/dhcp** file. Change the line: `DHCPOPTIONS=" "` to: `DHCPOPTIONS="-n -b manual"`

6 If the shared IP address vendor option has already been defined, proceed to Step 8.

7 Add a *`SharedIP`* macro for the shared IP address in **/var/dhcp/dhcptab** with the following command:

```
dhtadm -A -s sharedIP -d 'Site,129,IP,1,1'
```

Type the following command to create a common macro called `call_server` to define common parameters for Matrix Controller cards. Note that this command will not fit on one line in this document and is actually one continuous command. Do not press **Enter** until you type all three lines:

```
dhtadm -A -m call_server -d ':Subnet=255.255.255.0:
BootSrvA=10.10.16.100:BootFile= "/tftpboot/
matrix.cfg":Router=10.10.16.200:'
```

Parameter	Description
call_server	Macro name for all the rest of these parameters
Subnet	Subnet value for a Matrix Controller card. Use the required subnet value in place of the 255.255.255.0 example shown.

Parameter	Description
BootSrvA	IP address of the TFTP server, which will usually be the IP address of this Solaris system. Use the required IP address value in place of the 10.10.16.100 example shown.
BootFile	TFTP configuration file for a Matrix Controller card. Use the required file name value in place of the /tftpboot/ matrix.cfg example shown.
Router	IP address of the Gateway for a Matrix Controller card. Use the required IP address value in place of the 10.10.16.200 example shown.

8 Create the DHCP network table for the 10.10.16.0 network that the Matrix Controller card uses. Type the command:

```
pntadm -C 10.10.16.0
```

Use the required value in place of the 10.10.16.0 shown in the example. The value to use is the IP address of the Matrix Controller “bit-wise anded” with the Subnet Mask of the Matrix Controller card.

9 Type the following command to add an entry for the active Matrix Controller of the redundant pair. Note that this command will not fit on one line in this document and is actually one continuous command. Do not press **Enter** until you type all the lines:

```
pntadm -A 10.10.16.2 -f 9 -m call_server -I
0100201C090733 10.10.16.0
```

Parameter	Description
`10.10.16.2`	IP address of the Matrix Controller card. Use the required value in place of the 10.10.16.2 in the example.
`9`	The value of the flags. The value `9` indicates the entry is permanent and BOOTP.
`call_server`	Use the *call_server* macro. This indicates that all values associated with the call_server macro will be used for this definition.
`0100201C090733`	The first two digits of 01 are always used, and indicate the hardware type, which is Ethernet. Use the hardware address of the Matrix Controller card in place of the 00201C090733 example shown. Note that this value is not case sensitive.
`10.10.16.0`	IP address of the Gateway for a Matrix Controller card. Use the required value in place of the 10.10.16.200 example shown. This is the same value used in Step 8.

10 Add the shared IP address for this entry as follows:

```
dhtadm -A -m 00201C090733 -d':sharedIP=10.10.16.9:'
```

Parameter	Description
00201C090733	Use the hardware address of the Matrix Controller card in place of the 00201C090733 example shown. This value is the same as the value used in Step 9, except the hardware type 01 is not used here, and all hexadecimal values which are letters must be uppercase.
10.10.16.9	IP address of the Matrix Controller card. Use the required value in place of the 10.10.16.2 example shown

11 Repeat Steps 10 and 11 for the standby Matrix Controller card of the redundant pair.

12 To restart the DHCP daemon, type the following two commands:

```
/etc/init.d/dhcp stop
/etc/init.d/dhcp start
```

13 To display the DHCP network table for the 10.10.16.0 network, use the command:

```
pntadm -P 10.10.16.0
```

Use the required value in place of 10.10.16.0 in the example. The value to use is the value used in Step 8.

14 To display the DHCP configuration table, use the command:

```
dhtadm -P
```

Configuring Dual Ethernet Ports on IP Signaling and Matrix Controller I/O Cards

Overview

You can configure the second Ethernet port on the IP Signaling Series 3 I/O card to separate the H.323 signaling traffic from the host control traffic. You can also configure the second Ethernet port on the Matrix Controller I/O card to separate the SIP signaling traffic from the host traffic.

For both cards, the host-to-CSP traffic is carried on Ethernet Port A and the signaling traffic is carried on the Ethernet Port B. Dialogic recommends this configuration.

Configuring the second Ethernet port is optional but if you enter an IP Address for Ethernet Port B you must enter an associated Subnet Mask.

> **Important!** Although a gateway IP Address may be configured for both Port A and Port B, only one is utilized. If two gateway IP addresses are defined, one for Port A and one for Port B the one specified for Port A will be used. Therefore, Dialogic recommends that you configure only one gateway IP address and that it be on Port B.

If the BOOTP server goes offline causing the CSP Matrix 2000 Card or IP Signaling Card to reboot, you get prompted for the Port A IP address, gateway IP address, and subnet mask if these were not stored. Enter these values for Port A. No other ports are configured.

Both of these features are fully explained in the *Developer Guide: Internet Protocol* as follows.

- *Dual Ethernet Port* in the *H.323 Software* chapter
- *Dual Ethernet Port* in the *SIP Software* chapter

This section explains how to configure the Dual Ethernet Port for either the IP Signaling Series 3 card or the Matrix Controller card.

The following are the Tag Definitions and values to configure in the procedure below.

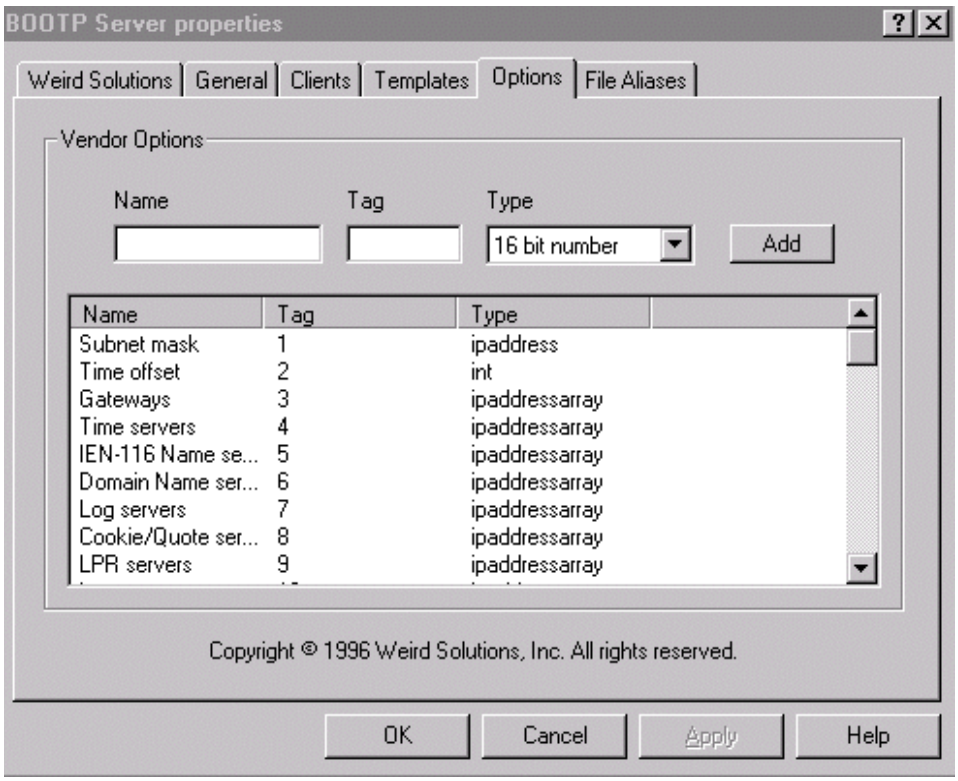
Tag Name	Value
Port B IP Address	130
Port B Subnet Mask	131
Port B Gateway IP Address	132
Port B Shared IP Address	133

When you enter the Tag Name in the procedure below, you can precede each name with “Dialogic” so they appear together in the list of options.

For example **Dialogic Port B Gateway IP Address**.

Procedure This procedure uses Weird Solutions BOOTP Server NT running on Windows NT® but you can use any BOOTP server running on any platform.

1 Double-click **BOOTP Server NT** from the Control Panel to display the Weird Solutions BOOTP Server NT dialog box. Select the **Options** tab.



2 Configure the Port B IP Address as follows:

- Type **Port B IP Address** in the **Name** box.
- Type 13Ø in the **Tag** box.
- Select **ipaddress** in the **Type** box. Click **Add**.

3 Configure the Port B Subnet Mask as follows:

- Type **Port B Subnet Mask** in the **Name** box.
- Type 131 in the **Tag** box.
- Select **ipaddress** in the **Type** box. Click **Add**.

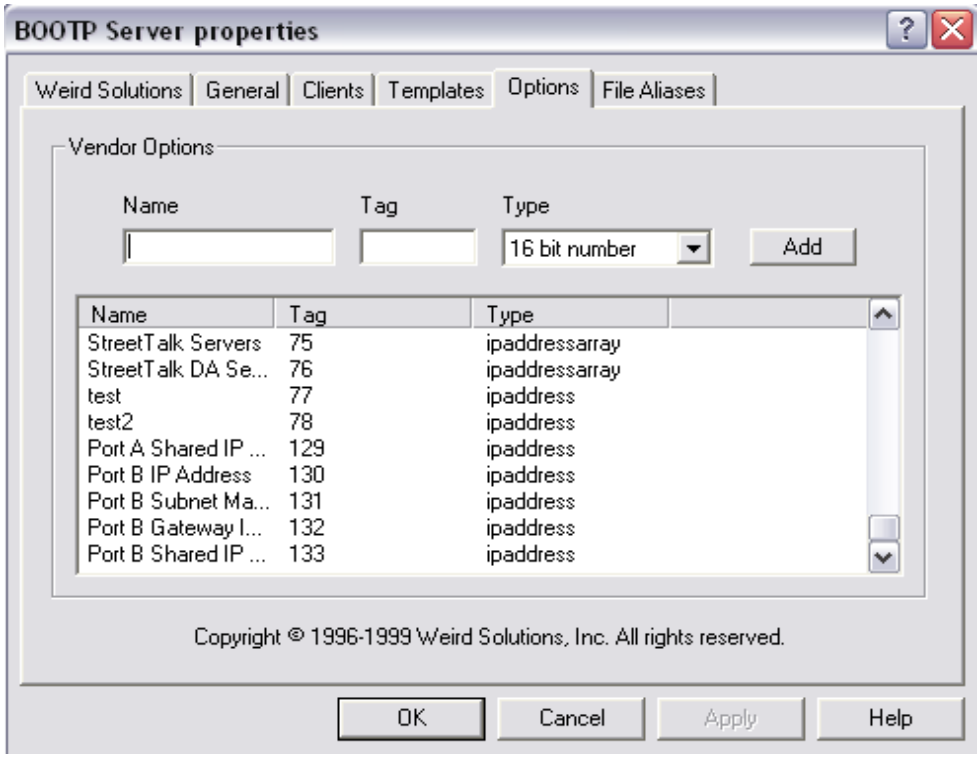
4 Configure the Port B Gateway as follows:

- Type **Port B Gateway IP Address** in the **Name** box.
- Type 132 in the **Tag** box.
- Select **ipaddress** in the **Type** box. Click **Add**.

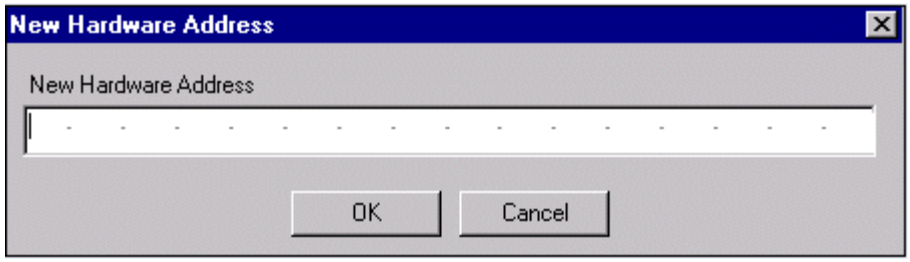
5 Configure the Port B Shared IP Address as follows:

- Type **Port B IP Address** in the **Name** box.
- Type 133 in the **Tag** box.
- Select **ipaddress** in the **Type** box. Click **Add**.

6 Click **OK** and scroll down the list of names on the **Options** page to verify that your entry appears with the correct information.

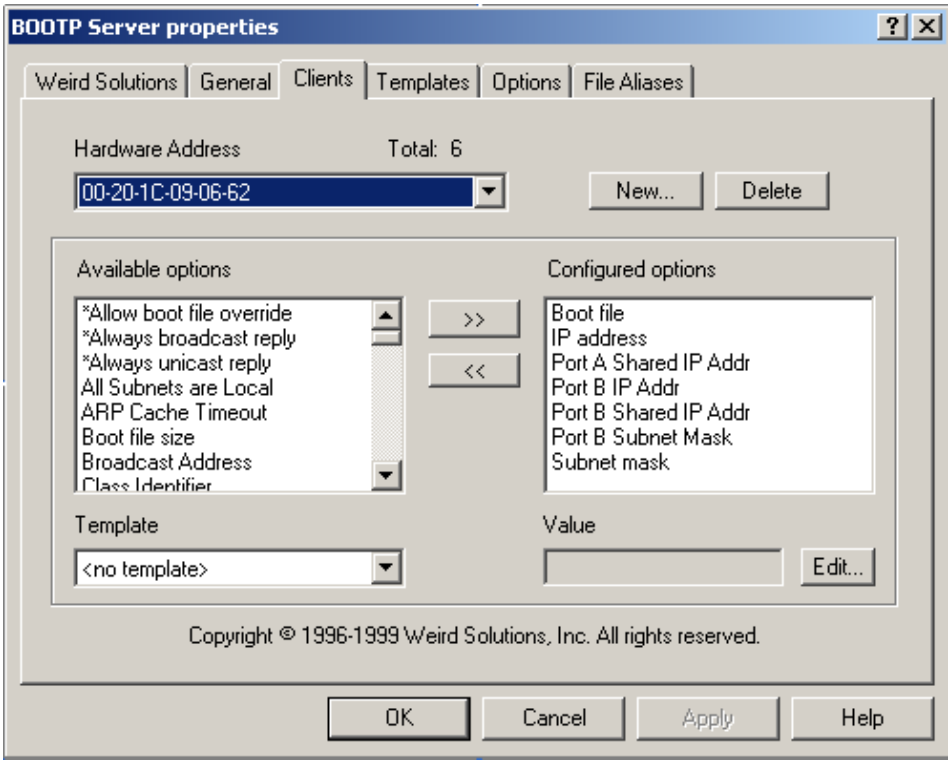


7 Select the **Clients** tab. Select the hardware address of the active IP Signaling Series 3 card in the **Hardware Address** box.
Click **New** and create the new hardware address if it is not listed.



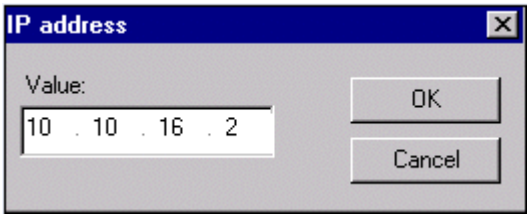
8 If the IP Address for the IP Signaling Series 3 card is not already created, do the following:

If each **IP address** is not listed in the **Configured options** window, select **IP address** from the **Available options** window and click the >> button to move them into the **Configured options** window.



9 Configure the IP Address. Select the **IP Address** in the **Available options** window and click the >> button to move it into the **Configured options** window.

10 Select the option to change in the **Configured options** list and click **Edit** to set or change the value of the IP address or Shared IP Address. Type the new value and click **OK**.



11 Repeat steps 8-10 for each Option that you created.

Robust Ethernet Redundancy

Alarm

This feature improves alarming and CSP reporting of Ethernet switch-overs. When the Ethernet ports are in redundant mode and an Ethernet switchover occurs, the CSP sends the Ethernet Switchover Occurred alarm (Card Alarm 0x4F) to the host. Refer to the *Alarm Message* (0x00B9) in the *API Reference*.

Unsolicited Reports

The CSP also send the following messages without the host requesting them:

- The active Ethernet interface is reported in every AB Poll. Refer the Card Status bit mask in the *Poll* (0x00AB) message in the *API Reference*.
- Whenever the CSP issues a *Card Status Report* for the Matrix Controller I/O, the report contains an indication of the active interface and each interface's link detect state.

Selecting Active Ethernet interface

Use the *IP Address Configuration* message to select the active Ethernet interface. Refer to this message in the *API Reference*.

CAUTION

Do not use this message while the CSP is passing data. Doing so could result in data loss

Querying

Use the *IP Address Configure Query* (0x00E6) message to query which Ethernet Interface is active.

CSA

Refer to the *CSA User's Guide* for instructions on Querying and Switching Over Ethernet Redundancy.

Multi-Host Control

Overview With Multi-host Control, the CSP can communicate on the LAN with up to six hosts simultaneously and you can run a different application on each host. For example, you can run a call control application on one host, a configuration and administration application on a second host, and an alarm management and status application on a third host.

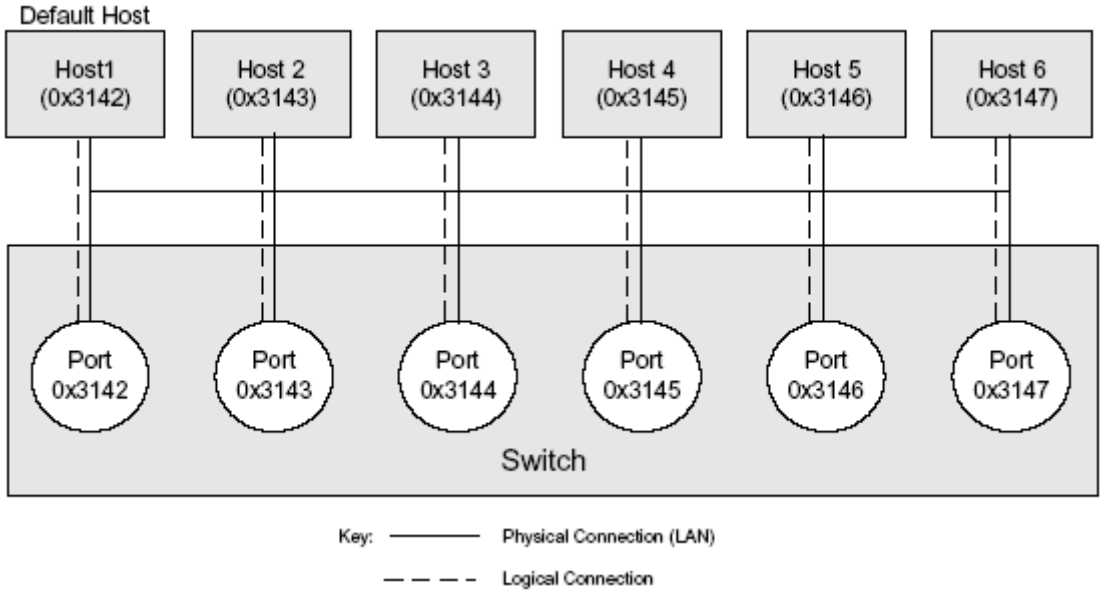
> **Important!** If the connection to a host is dropped, a *General Alarm* of Major severity is sent to indicate that a connection to one of the hosts has been dropped.

Benefits Multi-host Control provides the following benefits:

- Host redundancy
- Greater control of CSP management
- Greater bandwidth for the host port
- Faster response to time-critical messages such as *Request For Service*

Diagram The hosts connected to the CSP have host IDs between 0x3142 and 0x3147. The host connected on port 0x3142 is the default host.

Figure 3-5 Multi-Host Configuration



Multi-Host with Multiple Nodes

Only host nodes can be configured with Multi-host Control. If you send a *Multi-host Configure* message to a node that is not a host node, a negative acknowledgment (NACK) is returned, with a status of No Host Attached (0x500E). If a host node with Multi-host Control enabled is reconfigured as a slave node, Multi-host Control is automatically disabled on that node. If the node is subsequently reconfigured as a host node, the host application must re-enable Multi-host Control on that node.

Compatibility

To enable Multi-host Control, you must modify applications that use multiple connections to communicate with the CSP. You do not need to modify applications that communicate through a single connection. You can add new applications without changing existing applications. Opening new control ports and adding new applications does not affect an existing application that uses the original control port.

Performance

Multi-host Control creates extra demand for processing resources, so it may cause a slight delay in the host-to-CSP response time.

Requirements for TCP Connections

Each of the six host ports (0x3142 to 0x3147) is defined as an Ethernet TCP socket connection.

Port 0x3142 is the default host port. It is also the primary connection to the CSP. The host connected to this port is the default host for all CSP-initiated messages that cannot be routed to any of the hosts. Messages cannot be routed if no host registered for that particular message.

Ports 0x3143 to 0x3147 receive *Poll* messages from the CSP at an interval configured by the default host. These ports cannot unregister for Polls.

> **Important!** The memory buffer for host ports 0x3143 to 0x3147) is 4K. Host port 0x3142 has a larger send buffer of 32K. Applications with the most messaging should be connected to host port 0x3142 because it has much greater bandwidth. If an application with intensive messaging is connected to one of the four ports with the 4K bandwidth, a socket could be closed down.

Configure Software for Multi-Host

Multi-host Control is disabled by default, but if it is enabled, Multi-host Control remains enabled after any reset. The CSP-initiated message configuration is maintained throughout reset. The default host is the only host that can enable or disable Multi-host Control, and it does so by sending the *Multi-host Configure* message to the CSP.

When Multi-host Control is externally disabled, the CSP cannot communicate with the hosts connected to ports 0x3143–0x3147, even though those hosts are still physically connected to the CSP.

If you attempt to connect a host to a port that already has a host connection, the new host is connected and the original host is disconnected.

API Messaging

Processing Messages from the CSP

To receive messages from the CSP, host applications in multi-node must first send the *Multi-host Configure* message to register with the CSP.

The CSP can send messages to any individual host, to all of the hosts, or to any combination of hosts. You use the *Enable Switch-Initiated Messages* ICB in the *Multi-host Configure* message to enable messages to be sent from the CSP. If no host registers for a particular message type, messages of that type are sent to the default host. The table for each API message in the *API Reference* indicates whether a message is sent by the CSP or by the host.

You use the *Disable Switch-Initiated Messages* ICB in the *Multi-host Configure* message to disable sending messages from the CSP. By default, the *Poll* message is always sent to every host. The only time the *Poll* message is not sent to every host is when the default host disables the *Poll* message by setting the poll interval to 0 (zero).

For the CSP to send messages, the following setup is required:

- Each host must register for messages from the CSP. The resulting registration tables are protected against resets (warm starts) by battery back-up.
- Each host must use only the active Matrix Controller card to register for messages from the CSP. If a redundant Matrix Controller card is in the CSP, the active Matrix Controller card transfers a current image of registration tables to the standby Matrix Controller card. This procedure ensures that information is retained during a switchover.

Even if a host is not connected to the CSP, you can register and de-register the host on the active Matrix Controller for messages from the CSP. If a host is initially connected but then disconnects from the CSP, registration for CSP-initiated messages is not lost; it is sent to the default host on Port 0x3142. Even if the default host is not connected, you can still send messages to other hosts.

For all messages from the CSP that cannot be routed to any other host, the default host is the default destination. Messages cannot be routed when:

- There is no host registered for that particular message
- The connection is down to the host registered for the message

Multiple hosts can register for the same CSP-initiated message. In this case, the message is replicated and sent to all the hosts registered for that message type. The resend mechanism is initiated only if none of the hosts acknowledges the message. Broadcasting messages on multiple host ports has a performance impact.

Processing Messages from the Host

Any host can receive any message type. But only the host that sends a particular message receives an acknowledgment for that message. This is true even if other hosts have registered for the same message type. This arrangement prevents the other hosts from receiving unsolicited acknowledgments and messages.

For the host to send messages, the following steps are required:

- All responses to host-initiated messages are routed to the originating hosts.
- Call control messages that demand real-time processing have priority over other messages, such as configuration and query messages.
- The CSP can support only 256 outstanding messages from the host for a particular message type. Hosts sending additional messages receive a negative acknowledgment (NACK) with the error code, “System Congested.”
- Some messages from the host, such as *Fault Log Query,* produce multiple responses. Only one host can send such a message at any one time. If a second host sends such a message while the CSP is still processing the message from the first host, the second host receives a negative acknowledgment (NACK) with the error code, *Queue Full*.
- The hosts connected to the CSP on Ports 0x3143 to 0x3147 cannot send the following API messages:
  - *Poll Interval Configure*
  - *Reset Matrix*

Downloading System Software

Overview

To download system software Version 5.5 or above, you must use the TFTP Download method. There are two means of initiating a download.

- BOOTP: Initiated by sending a BOOTP Request to a BOOTP Server
- Host: Initiated by sending a *TFTP Manage* API message to the CSP.

Download the Matrix Controller load and all other card loads from a TFTP server.

Table 3-1 Download Method - Considerations

Method	TFTP Server	TFTP Config File	BOOTP Server	*TFTP Manage* Message	Storing of .bin files
Full TFTP via BOOTP	Yes	Yes	Yes	Only required to query TFTP configuration.	all required .bin files stored on TFTP Server
Full TFTP via Host	Yes	Yes	No	Required to initiate download and to query TFTP configuration.	all required .bin files stored on TFTP Server

Full TFTP Downloading

Full TFTP Downloading requires the Matrix Controller card.

With the Full TFTP Downloading method, all card loads are stored on a TFTP server and retrieved by the CSP via TFTP. You must modify the default TFTP Configuration File to indicate the individual card loads to be downloaded for your configuration.

Configuration

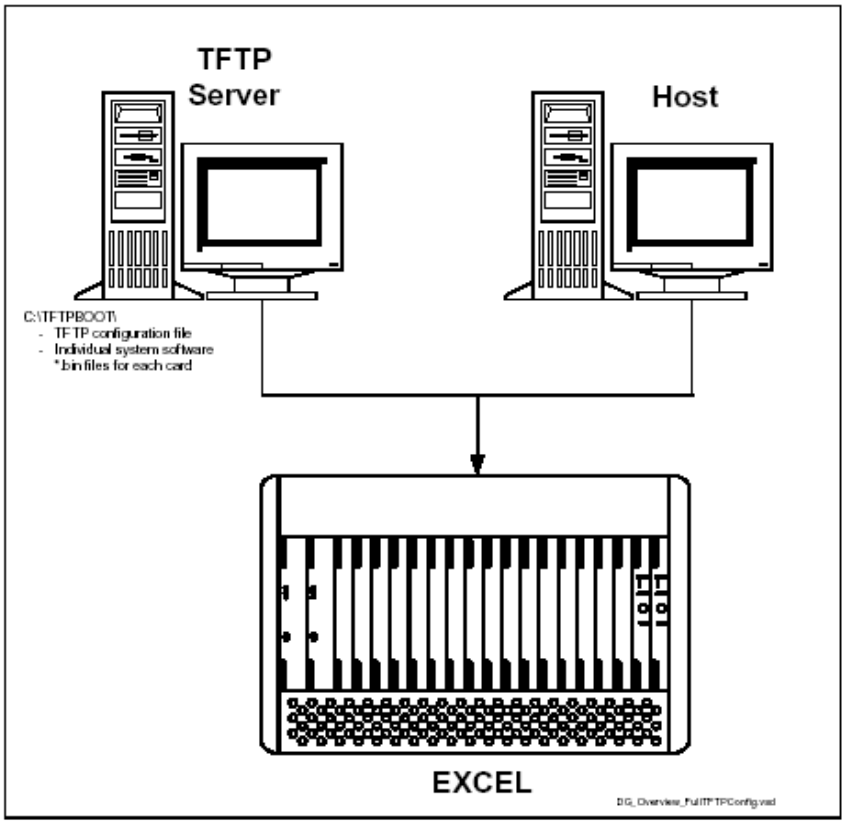
To implement Full TFTP downloading, follow the steps below.

1. Install and configure a TFTP Server (see *Installing and Configuring a TFTP Server.*
2. Modify the default TFTP configuration file according to your configuration (see *Creating a TFTP Configuration File).*
3. Copy the TFTP configuration file to the TFTP Server.

4. Copy the individual software .bin files for the cards you require to the TFTP Server.

You are now ready to initiate a download. See *Initiating a Download*.

Figure 3-6 Full TFTP Configuration



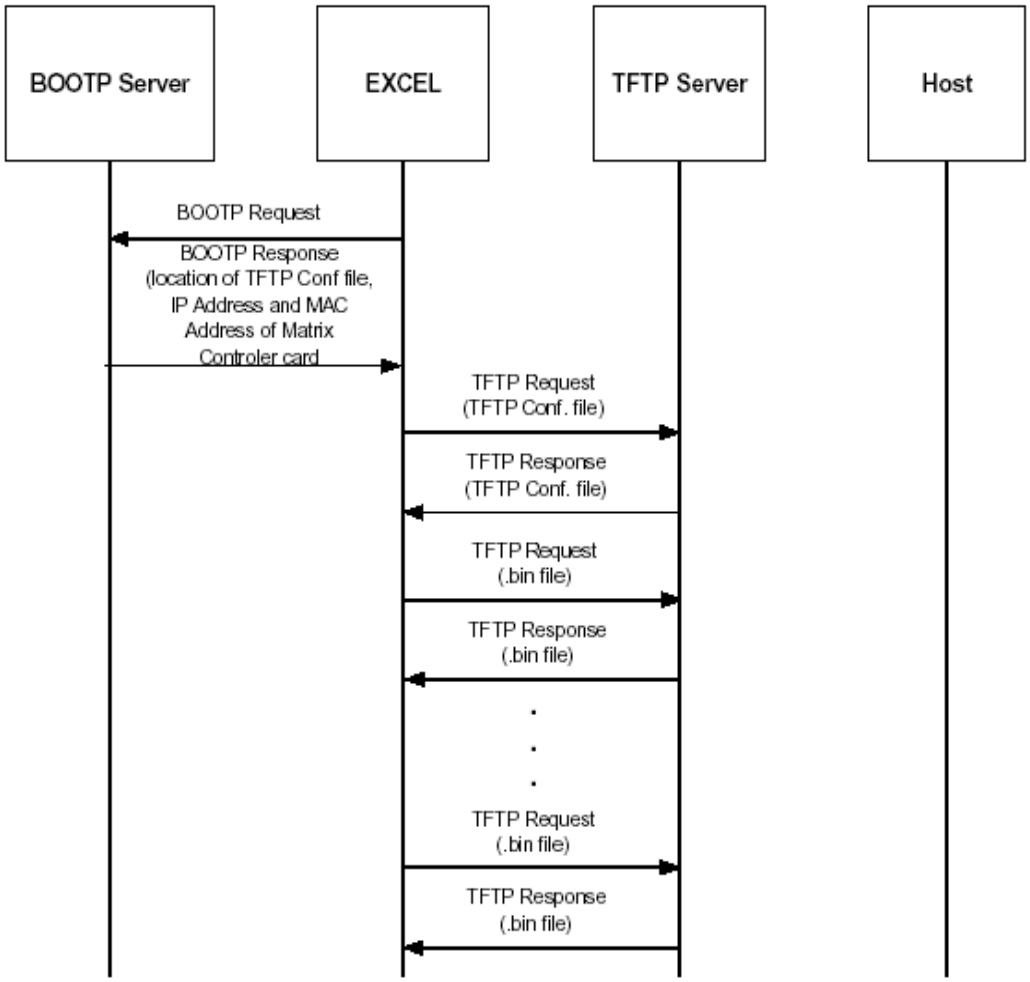
Initiating a Download

You can initiate a download using either a BOOTP server or the host.

BOOTP Server Download

Figure 3-7 shows the sequence for downloading system software files using the Full TFTP Downloading method and a BOOTP server.

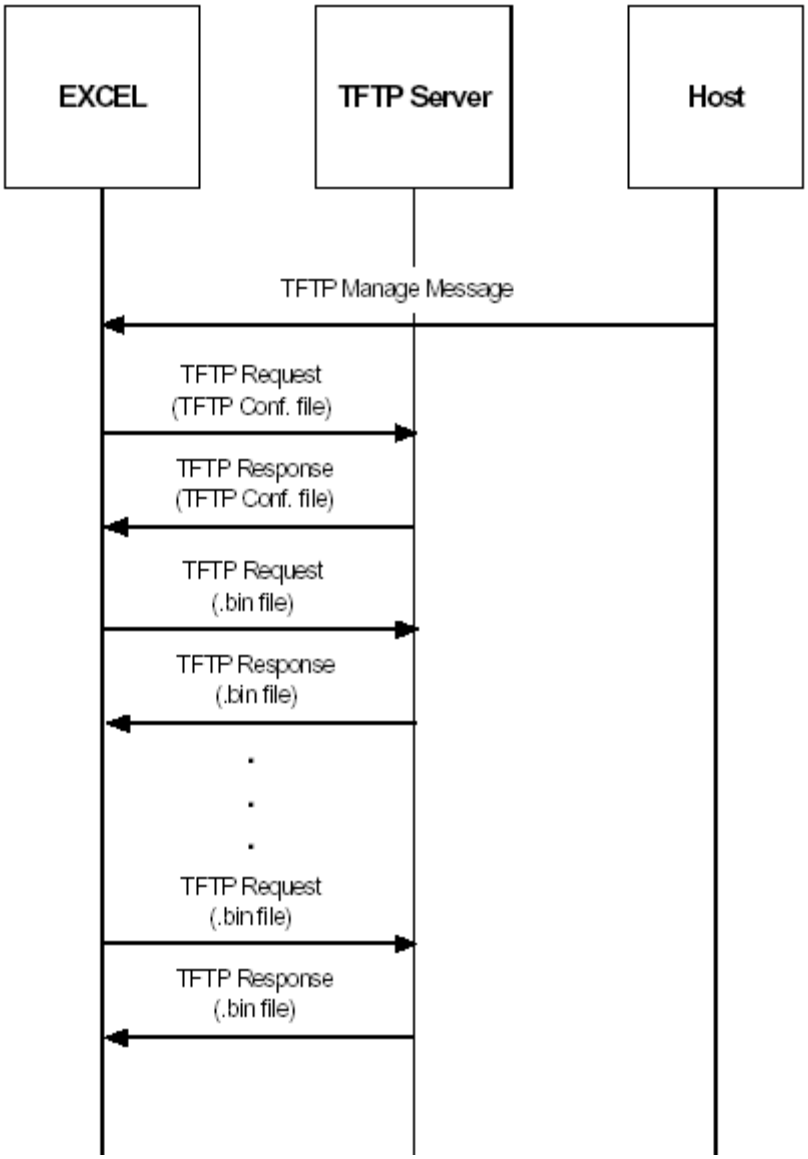
Figure 3-7 Full TFTP Downloading Using BOOTP Response message



Host-Initiated Download

Figure 3-8 shows a Full TFTP download initiated by the host using the *TFTP Manage* API message. See Using the TFTP Manage Message.

Figure 3-8 Full TFTP Downloading Using *TFTP Manage* API message



Downloading the Matrix Controller .bin File

Follow the step below to download the Matrix Controller .bin file from the host to the CSP using the download API messages.

1. Send the *Clear System Software* message to clear the existing software load, if any.
2. Reset the Matrix Controller by either pressing the reset button or sending the *Reset Matrix* message.
3. Send the *Download Begin BRecord*.
4. Use *Download BRecord* messages. Copy the file data into a series of *Download BRecord* messages. The host must send the B records in the exact order in which they exist in the file.
5. Send the *Download Complete* message. This message indicates that the host has sent all download messages. Once received, the CSP validates the software load and acknowledgments with either a success or error status. If the host receives an error status, repeat the system software download process again starting from the *Download Begin* message.

   If the download is successful, the Matrix Controller executes the download code.

   The *Download Complete* message includes a timestamp from the host. The host uses the timestamp to determine the most recent download in a redundant system.
6. Download other card loads using TFTP.

The host must follow each download message with another download message or a *Download Complete* message within five minutes, or the download aborts. For example, when a *Download Begin BRecord* is sent, a *Download BRecord* must follow within five minutes. The *Download BRecord* must be followed by either another *Download BRecord* or *Download Complete* within five minutes.

CAUTION

When downloading system software, do not have more than 25 outstanding (unacknowledged) messages at one time or a Matrix Controller fault will occur.

When the boot loader starts the system software, the CSP performs initialization and configuration tasks. These include downloading and validating the downloads on all the cards. As it performs this initialization, the CSP sends the host a *Card Status Report* message for each card it finds.

When initialization is complete, the CSP sends a *Poll* message to the host with the Ready For Configuration bit set and the host can begin configuration. If the CSP is part of an EXNET®configuration and uses fiber optic communications, only a *Node Status Report* message returns after initializing. The host must issue the *Assign Logical Node ID* message before polling begins.

The system software runs in several states that the CSP identifies and sends to the host in *Poll* messages. Even if polling is disabled, the CSP sends a *Poll* message to the host every time the system software changes states. The host must react appropriately to changes in the system software state in order to start the system and keep it running. In a redundant system, *Poll* messages are sent from both Matrix Controller cards.

BRecord Format

To read BRecords out of the *.bin file, perform the following steps:

1. Open the file as a binary file.
2. Read from the file with ten bytes (ten bytes is the minimum number needed to make a complete BRecord, with no *Data* field).
3. Read from the file the number of bytes specified in the *Length* field.
4. Add the bytes in the *Length* field to the first ten bytes. The combined data constitutes one complete BRecord.

Figure 3-9 BRecord Format

Content	Byte	Field Name
’B’	0	BRecord Identifier
tt	1	BRecord Type (See Below)
0x00	2	Reserved (Must be 0)
ll	3	BRecord Length (n)
aa	4	BRecord Address (MSB)
aa	5	BRecord Address
aa	6	BRecord Address
aa	7	BRecord Address (LSB)
dd	8	BRecord Data[0] (Optional)
dd	9	BRecord Data[1] (Optional)
:	:	:
dd	7+n	BRecord Data[n] (Optional)
cc	8+n	Checksum (MSB)
cc	9+n	Checksum (LSB)

BRecord Types

There are four types of BRecord (see table below). Each type is identified by a value. All BRecords include fields for *BRecord Identifier, Type, Length, Address* and *Checksum*. The *Header BRecord* and *Data BRecord* types also contain a variable number of data bytes.

Table 3-2 BRecords Type

Type	Field	Description
0x00	*Header*	Header record for a BRecord module. It contains data bytes.
0x05	*Record Count*	BRecord that verifies the number of data records preceding it; it contains no data bytes.
0x10	*Data BRecord*	BRecord that contains the actual download data; it contains data bytes.
0x11	*Terminator*	BRecord that marks the end of a BRecord module; it contains no data bytes.

Additional BRecord Fields

The remaining *BRecord* fields are shown in the table below.

Table 3-3 BRecord Fields

Field	Description
Length	A byte that represents the number of data bytes in the data of the BRecord. BRecords without data bytes have the *Length* field set to zero.
Address	A long word that specifies the data bytes location in memory for data BRecords. In Record Count BRecords, the *Address* field contains the number of data records preceding it. In Terminator Records, the *Address* field has no significance. The most significant byte of the *Address* field is sent first.
Data	The *Data* field contains the data bytes for the download. The first data byte will be placed at the address specified in the *Address* field. The *Length* field specifies how many data bytes are in the BRecord.
Checksum	The *Checksum* field is a word that contains the sum of all of the preceding bytes in the BRecord. The most significant byte of the *Checksum* field is sent first.

Creating a TFTP Configuration File

Overview

You must create a TFTP Configuration File that indicates the loads to be downloaded. Each software release comes with a default configuration file which contains information for all cards supported in that release. You can modify the default configuration file.

See the sample TFTP Configuration File below.

TFTP Configuration FIle Format

The entries in the configuration file do not need to be in a specific order, but they must follow this format:

<Card Label_LOAD>=<path/filename>:SAVE_LOAD_<TRUE or FALSE> ‘ Comment

No space is allowed on the line until after the SAVE_LOAD label. The path/filename can be up to 127 characters long.

Card Label

The table below shows the labels used in the TFTP Configuration File for each card.

Table 3-4 Labels for TFTP Configuration File

Card Name	Card Label
Matrix Controller Series 3	EXCPU
E-ONE, T-ONE, J-ONE	ONELC
ST1LC	ST1LC
SE1LC	SE1LC
SJ1LC	SJ1LC
DSP-ONE	DSPONE
SS7 PQ	SS7PQLC
SS7 Series 3	SS7HPLC
Subrate Controller	SRC
DSP Series 2	
ISDN PRI, ISDN BRI, DASS2/DPNSS	ISDN32
ISDN Series 3	ISDNHPLC
EXNET-ONE	EXNTONE
DS3	DS3
VDAC Module	VOIPMOD
VDAC Main Board	VOIPMBRD
IP Signaling Series 3 (H.323)	EXCPU

Matrix Controller Load

You must set the Matrix Controller load to be saved on the Matrix Controller card by including “TRUE” in the line for the matrix (for example, SAVE_LOAD_TRUE). The Save Load option is not supported for any other cards. The load for cards other than the Matrix Controller are retrieved as needed via TFTP.

Timestamp

If you are using BOOTP to initiate the download, you must provide a timestamp in the configuration file.

> **Important!** The configuration filename on the BOOTP server must match the .bin name on the TFTP server.

If you are using the *TFTP Manage* message to initiate the download, you can include the timestamp in either the TFTP configuration file or in the *TFTP Manage* message. The *TFTP Manage* response *Status* is 0x01 (Two Timestamps Provided).

The timestamp in the TFTP configuration file is formatted as follows:

<month (01-12)>/<day(01-31)>/<year(0000-9999)> <space> <hour(01-23)>:<minute(01-59)>:<second (01-59)>

or as a raw value, such as a hexadecimal number (beginning with "0x"), representing the number of seconds since 1/1/70.

If you are configuring the timestamp in the *TFTP Manage* message, see the message format in the *API Reference* for the format of the Timestamp ICB subtype.

Sample TFTP Configuration File

The following is an example of a TFTP configuration file.

> **Important!** If using the *TFTP Manage* message to initiate the download, the timestamp must be included in either the *TFTP Manage* message or the TFTP configuration file, but not both.

```
EXCPU_LOAD=c:\tftpboot\ELX0800073\MTX0800073.BIN=SAVE_LOAD_TRUE
ST1LC_LOAD=c:\tftpboot\ELX0800073\ST10800073.BIN=SAVE_LOAD_FALSE
SE1LC_LOAD=c:\tftpboot\ELX0800073\SE10800073.BIN=SAVE_LOAD_FALSE
SJ1LC_LOAD=c:\tftpboot\ELX0800073\SJ10800073.BIN=SAVE_LOAD_FALSE
SS7PQLC_LOAD=c:\tftpboot\ELX0800073\SPQ0800073.BIN=SAVE_LOAD_FALSE
ISDN32_LOAD=c:\tftpboot\ELX0800073\ISD0800073.BIN=SAVE_LOAD_FALSE
DS3_LOAD=c:\tftpboot\ELX0800073\DS30800073.BIN=SAVE_LOAD_FALSE
EXNTONE_LOAD=c:\tftpboot\ELX0800073\EX10800073.BIN=SAVE_LOAD_FALSE
DSPONE_LOAD=c:\tftpboot\ELX0800073\DSP0800073.BIN=SAVE_LOAD_FALSE
VOIPMOD_LOAD=c:\tftpboot\ELX0800073\VMD0800073.BIN=SAVE_LOAD_FALSE
VOIPMBRD_LOAD=c:\tftpboot\ELX0800073\MBR0800073.BIN=SAVE_LOAD_FALSE
ONELC_LOAD=c:\tftpboot\ELX0800073\TE10800073.BIN=SAVE_LOAD_FALSE
SS7HPLC_LOAD=c:\tftpboot\ELX0800073\SHP0800073.BIN=SAVE_LOAD_FALSE
TIMESTAMP=09/27/01 10:30:15
```

Using the *TFTP Manage* Message

The *TFTP Manage* message is used to initiate a download of system software, update download information, and query the TFTP Configuration.

Initiating a Download

Send the *TFTP Manage* message to the CSP with the following information:

- Local Matrix Controller configuration filename (Data ICB Subtype 0x37)
- Local Matrix Controller server IP address (Data ICB Subtype 0x38)
- Adjacent Matrix Controller configuration filename (Data ICB Subtype 0x39)
- Adjacent Matrix Controller server IP address (Data ICB Subtype 0x39)
- Timestamp (Data ICB Subtype 0x3B) (if not included in the TFTP Configuration File)
- Begin Download (Action ICB Subtype 0x00)

See the *API Reference* for the format of the *TFTP Manage* message.

Important! The term “Local” refers to the Matrix Controller to which you are downloading the files.

Updating Information

If you are simply updating information (such as configuration filename or Server IP address) but not downloading a new version of system software, send the relevant data ICBs but do not send the Begin Download action ICB.

Querying TFTP Configuration

You can use the *TFTP Manage* message to query the following information:

- Local Matrix Configuration Filename
- Adjacent Matrix Configuration Filename
- Local Matrix Server IP Address
- Adjacent Matrix Server IP Address
- Timestamp
- Save Options
- Load Version
- Load Filename

Each query option is performed by inserting the appropriate action ICB into the *TFTP Manage* message. The response contains an ICB with the relevant data for the query. See the *TFTP Manage* message in the *API Reference* for ICB values and a sample message trace.

Changing System Software Version

To download a new version of system software, you must send the new TFTP Configuration File name to the CSP.

BOOTP Server

To change a system software load using a BOOTP server, perform the following:

1. Configure the BOOTP server to issue the new TFTP Configuration File name for both the active and standby Matrix Controller in the BOOTP Response. The timestamp supplied must be a value higher than the timestamp of the current load.
2. Then send the *Reset Matrix* message to the standby Matrix Controller, which causes it to reset and retrieve the new TFTP Configuration File name from BOOTP. The standby Matrix Controller then retrieves the new load from the TFTP server. When the active Matrix Controller detects a newer load on the standby Matrix Controller, it requests it, reboots to run the new load, and receive the new filename from BOOTP.

TFTP Manage Message

Send the *TFTP Manage* message to the standby Matrix Controller with the following:

- The TFTP Configuration Filename and Server IP Address for the standby (local) Matrix Controller
- The TFTP Filename and Server IP Address for the active (adjacent) Matrix Controller
- The New Timestamp (if not included in the TFTP Configuration File)
- The timestamp supplied must be a value higher than the timestamp of the current load.
- The Begin Download Action ICB.

This message clears the system load and initiates a TFTP transfer of the new load to the standby Matrix Controller.

Redundancy

Redundant Matrix Controller configurations continue the automatic BOOTP load transfer without the need to reconfigure after switchover. After a switchover, the TFTP Configuration File name that is used by the new active Matrix Controller can be queried using the *TFTP Manage* message. At reset, the load with the most current timestamp is identified and transferred from either the server or the adjacent Matrix Controller.

Card DIP Switch Selectable Functionality

Overview

The CSP card DIP switch selectable functionality, listed below, are customer initiated by manually setting the DIP switch positions on the cards listed below.

- Ethernet Link, Force 100 Mbps/Full Duplex
- No Static IP Address
- Software Loading Using Second Ethernet Port

For detailed information on the DIP switch settings refer to the *Card Status Report* (0x00A6) message in the Hardware Information field.

Ethernet Link, Force 100 Mbps/Full Duplex

This feature enables the cards listed below to be able to force the Ethernet links to a 100 Mbps/full duplex mode instead of the default auto-negotiate mode. In certain cases, the auto-negotiation fails to achieve the highest possible rate when customers routers are in force full 100 configurations.

- CSP Matrix Series 3 Card
- IP Signaling Series 3 Card
- SS7 Series 3 Card
- ISDN Series 3 Card
- DSP Series 2 Card

Important! If you use the Ethernet Link, Force100 Mbps/Full Duplex mode with a hub/switch/router that is not in a forced full 100 Mbps configuration, the results can vary and a link connection may not be possible.

To set the Ethernet Link, Force100 Mbps/Full Duplex mode on any of the above cards, refer to the DIP switch settings in the Hardware Product Descriptions.

No Static IP Address

This feature allows the cards listed below to enable the No Static IP Address mode instead of using the default Static IP mode.

- CSP Matrix Series 3 Card
- IP Signaling Series 3 Card

In the No static IP Address mode, RARP or BOOTP must retrieve an IP address for the CSP Matrix Series 3 Card or IP Signaling Series 3 card. This feature is enabled by setting DIP switch S1 (SW1/IP Signaling) position 8 on either card to OFF to enable the following functionality:

- If RARP or BOOTP can not obtain an IP address, the static IP address will not be used.

This allows the Matrix Controller Series 3 or IP Signaling Series 3 cards to reset and try RARP or BOOTP again to obtain the correct IP address, rather than use the static IP address that is stored in RAM.

When DIP switch S1 (SW1/IP Signaling) position 8 is set to ON (default) the following functionality is enabled:

- If RARP or BOOTP can not obtain an IP address, the static IP address will be used.

To set the No Static IP Address mode on any of the above cards, refer to the DIP switch settings in the Hardware Product Descriptions.

Software Loading Using Second Ethernet Port

This feature allows the card listed below to use the second Ethernet port if the first port (default) cannot obtain the software load.

- CSP Matrix Series 3 Card

If you want to include the second Ethernet port in obtaining an IP address, set DIP switch S1, position 7, to OFF prior to booting up. To use this feature, refer to the sequence of events described below:

- The CSP Matrix Series 3 Card tries to retrieve an IP address from the I/O card default Ethernet port ETH1 to obtain the software load.
- If this attempt is unsuccessful, the CSP Matrix Series 3 Card then tries to obtain the software load through Ethernet port ETH2.
- If still unsuccessful, the CSP Matrix Series 3 Card uses the stored static IP address for ETH1 to obtain the software load.

Important! To enable the static IP address, ensure DIP switch S1, position 8 is set to ON.

To set the Software Loading Using the Second Ethernet mode on the above card, refer to the Hardware Product Descriptions.

4 EXS API Application Development

Purpose This chapter presents an overview of the software architecture and the application requirements for non-SwitchKit users.

Note to SwitchKit Users

If you are using SwitchKit you do not need to perform the application development tasks described in this chapter.

If you are using SwitchKit, refer to the *SwitchKit Product Description* in the *SwitchKit Installation and Maintenance Guide*.

Configuration Guidelines

The system software cannot perform all actions in all states. The host must operate in accordance with the following restrictions of each state:

- If the system software is in any state other than Active, only the Core Message Set is available. The Core Message Set is a small subset of the API messages. Use the Core Message Set only for downloading system software and for tracking the status of the Matrix Controller.
- The system software can use hardware resources only if they are reported in the *Card Status Report* message.
- Before the host can configure the CSP, the Active system software must send a *Poll* message with the *Ready For Configuration* status bit set.
- The host monitors the *Poll* messages status bit, "Download this Matrix", and initiates a download if it is set. The system software cannot leave the Boot state until it has received a valid download.
- The host monitors all the status bits in the *Poll* message for indications of failure which prevents the CSP from operating with defective hardware.
- Only one Matrix Controller transitions to the Active state. The host can either allow the CSP to select which Matrix Controller becomes Active, or it can send a *Become Active* message to the Matrix Controller that it wants to address as the active Matrix Controller.

In a non-redundant system, the system software runs in exactly the same states and follows the same rules. The only difference is that the single Matrix Controller always runs in the active state.

The Core Message Set

The Core Message Set is a subset of the API messages. When the CSP is shipped, it is not running system software. However, the Core Message Set is shipped on the CSP, residing in the firmware on the Matrix Controller card. The Core Message Set downloads system software from the host and queries the CSP.

When a valid download finishes, the CSP stops running the Core Message Set and starts running the System Software in RAM on the Matrix Controller card.

The following messages make up the Core Message Set:

- *Alarm*
- *Become Active*
- *Card Status Query*
- *Clear System Software*
- *Download Begin BRecord*
- *Download Begin SRecord*
- *Download BRecord*
- *Download Complete*
- *Download SRecord*
- *Fault Log Query*
- *Poll*
- *Poll Interval Configure*
- *Poll Request*
- *Reset Matrix*
- *TFTP Manage*
- *Version Request*

API Message Guidelines

Applications must conform to the following guidelines when using API messages:

Downloading System Software

A matrix fault will occur if you allow more than 25 outstanding messages remain unacknowledged simultaneously while you are downloading system software.

Matching Messages and Responses

Every message must receive a response, regardless of whether the CSP or the host sent the message. The response contains a status byte, and you should design your application to check the status bytes of host-initiated messages to determine if the message succeeded or failed.

The message flows between the host and the CSP are not synchronized. The host can send multiple messages, while the CSP simultaneously sends multiple messages to the host. Sequence numbers are used to match messages with responses. The originator of the message selects the sequence number. For host-initiated messages, the host communications driver typically selects the sequence number, and provides a counter for each message type. The host allows up to 255 outstanding messages for each message type.

Message Trace Format

When you see message traces in hexadecimal format, the first byte of each message (the framing byte, 0xFE) is not displayed.

Call Processing Messages

When a host application sends call processing messages to the CSP, the host must be aware of associations between channels and between spans and channels. The host can send only one call processing message at a time to a channel. Design your application to track channel information such as state and configuration.

Configuration Messages

Many of the configuration messages use a range to configure multiple channels. When you specify channel ranges, do not exceed eight spans. It is important to read the description for each message.

A positive acknowledgment (0x10) to a configuration message can indicate that the configuration is complete. But sometimes the CSP sends a separate message to indicate that the configuration is complete. For example, if the host sends a *Release Channel* message, the CSP sends back a *Channel Released* message.

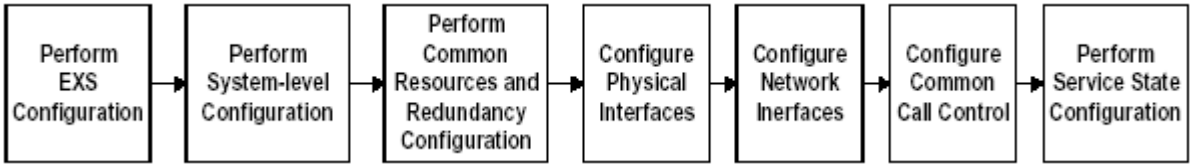
Configuring the CSP

Overview

You should design your host application to initialize the CSP through the configuration process. You should also design the host application to dynamically configure, deconfigure, and query all entities within the CSP. Examples of these dynamic functions include adding facilities, changing span formats, deconfiguring stack entities, and querying configuration entities.

A generic configuration for the CSP includes the steps shown below:

Figure 4-1 Generic Configuration Procedures



Configuration

The CSP is an expandable switching system, made up of two to seven nodes.You must initialize the EXNET® ring before you configure the nodes.

> **Important!** If you are configuring greater than 64 spans, please call Technical Support for assistance.

You should begin design of your system configuration with actions that affect the whole system, including network synchronization, system time, poll interval, and loop timing.

Network Synchronization

Introduction

Before running the CSP, synchronize T1, E1, and J1 spans to the network. Use the *Synchronization Priority List Configure* message to configure a prioritized list of clock sources. The CSP uses this list when it selects a synchronization clock source.

The default priority for synchronization resources is as follows:

1. External Reference Clock 1 (Primary)
2. External Reference Clock 2 (Secondary)
3. Loop Timing 1 (Primary)
4. Loop Timing 2 (Secondary)
5. Free Running clock (Internal)

The CSP continuously monitors all synchronization clock sources and uses the highest priority clock source currently available. For example, if you use the default configuration above, the CSP first determines if the Primary External Reference Clock is available. If it is not available, the CSP determines if the Secondary reference Clock is available. The CSP continues to check each source, in succession, until it finds an available source. When a higher priority source becomes available, the CSP automatically changes to the higher priority clock source.

Clock Sources

Descriptions of the synchronization sources are as follow:

External Reference Clock

Reference Timing is generated by an external clock source to the Matrix Controller card through the Matrix Controller I/O. One of the Matrix Controller I/O cards function is to take in a external reference clock and provide that timing to the Matrix Controller card which in turns uses that clock for system timing on the bus.The Matrix controller I/O card can sync to the following reference clocks 1.544 Mbps or 2.048 Mbps.

Loop Timing

Loop timing synchronizes the system to an incoming T1, E1, and J1 span. You must first designate which span you want to use to extrapolate the clock. Use the *Loop Timing Configure* message after you assign spans.

Free Running Clock

Free running timing is based on the internal clock source of 2.048 Mbps. Free running timing is for test environments only. Do not use this timing method for network operations. You can determine the current active timing by performing a synchronization priority list query.

System Time

You must use the *Time Set* message to set the system time. The CSP uses the system time to timestamp fault log entries.

Poll Interval

Two seconds is the default interval between *Poll* messages sent from the Matrix Controller to the host. You can use the *Poll Interval Configure* message to change the Poll interval.

Loop Timing

Configure Loop Timing to derive time from a T1 or E1 span.

Card Configuration

Card configuration includes the following:

- Line Cards
- DSP resources
- Common Channel Signaling Cards

Line Card Redundancy

Process

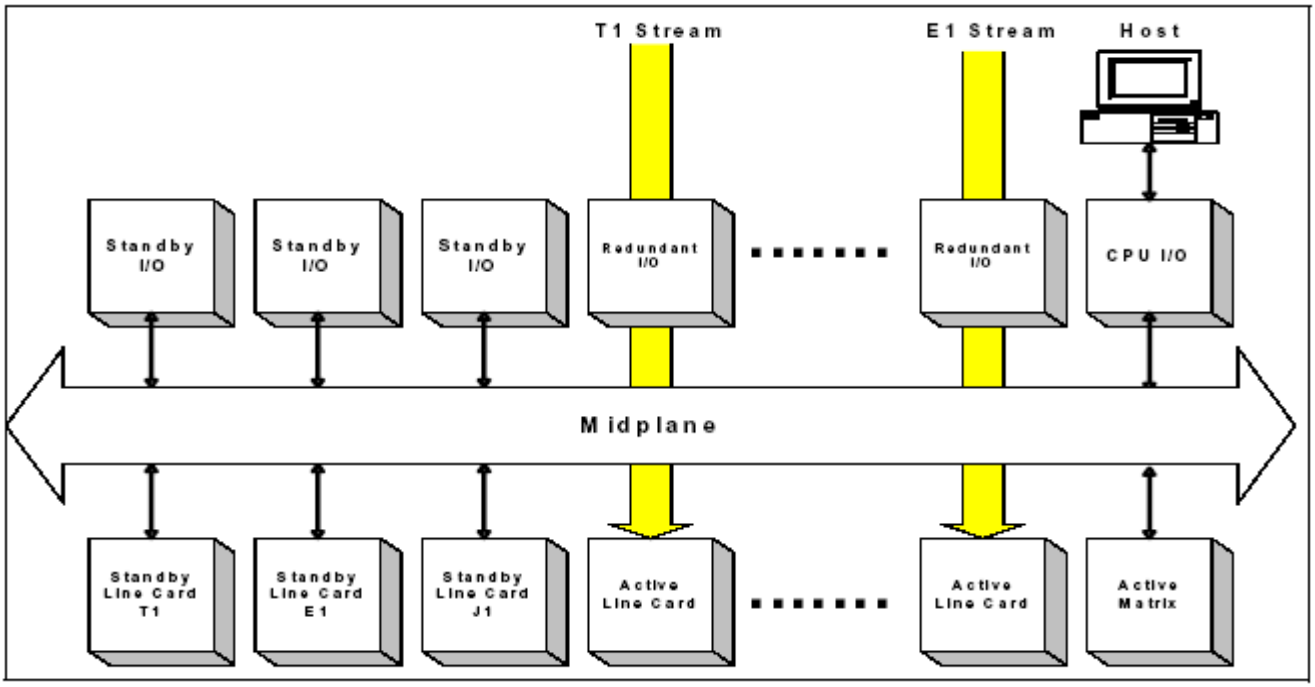
On a standby line card, the Span Status LEDs blink green, indicating that the card is in standby mode. The LEDs on the redundant I/O and the Standby I/O are also green. To configure the standby line card, the host must use the *Standby Line Card Configure* message that indicates the *Card Type* and *Action*.

If a line card fails, the Matrix Controller detects the failure and sends an *Alarm* message to the host, followed by a *Card Status Report* message. The *Card Status Report* message indicates the card type and slot number of both the failed line card (Originating Slot Number) and the standby line card (Destination Slot Number). The Matrix Controller confirms that the two cards are of the same type (if the cards are not of the same type, the host receives an error message).

The host then instructs the Matrix Controller to reroute the appropriate T1, E1, or J1 data stream. The Matrix Controller instructs the standby I/O card to connect to the redundant bus. The Redundant I/O card then switches the data stream of the failed line card onto the Redundant Bus. The data stream then passes from the Redundant Bus, through the Standby I/O card, and finally to the standby line card.

The LEDs on the card that was the standby line card then turn off, indicating that the card is now the Active line card. The LEDs on the Redundant I/O and Standby I/O turn red, indicating that they are in Switchover mode. The CSP sends a *Card Status Report* message to the host, indicating that the standby line card is in service. The host must then configure the new Active line card with the same configuration as the failed line card.

Figure 4-2 Network Interface Card Configuration

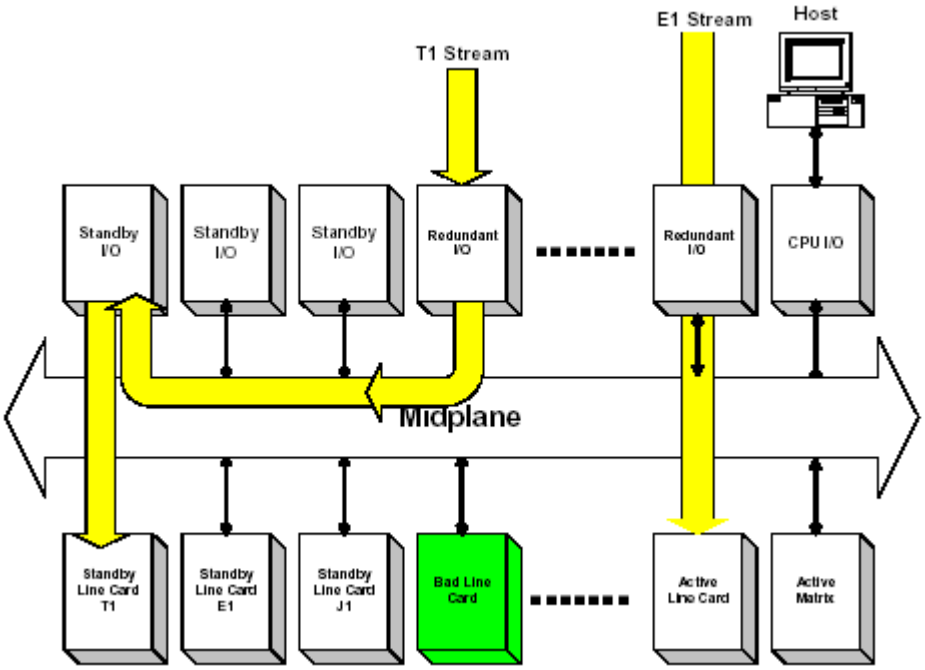


The Matrix Controller card controls and monitors Redundant I/O cards. The Matrix Controller tracks the location of Redundant I/O cards and standby line cards. Each standby line card must have a Standby I/O card. The Matrix Controller puts a line card into standby mode when it detects the associated Standby I/O card. The standby line card indicates that it is in standby mode by flashing all span lights green. The host can then dedicate that slot as a standby slot.

Rerouting Data

Figure 4-3 shows the incoming data stream being disconnected from the failed line card and reconnected through the midplane to the standby card.

Figure 4-3 Rerouting Data from a Failed Line card to a Standby Line card



Completing Switchover

To complete the switchover, the host configures the new standby line card. The host must send the *Standby Line Card Configure* message to designate a standby slot. The Matrix Controller connects the appropriate Standby I/O to the midplane, then resets the relays on the Redundant I/O card that is connected to the failing line card. The input to the Redundant I/O card is then rerouted to the Standby I/O card.

You can remove the failed line card without affecting operations. When you replace the failed line card with a new line card, the Matrix Controller immediately puts the new line card in standby mode. The LEDs on the card flash green to indicate that the data stream is still being routed to the standby line card. Then the host can send a *Line Card Switchover* message to put the new card in service and to place the standby line card back in standby mode.

When the new card is brought into service, the card's redundancy feature is reset, and the CSP is ready to use the standby card if another line card failure occurs. If the switchover is successful, the LEDs on the new line card go out (in-service) and the LEDs on the standby line card flash green (standby mode).

Cards in Switchover State

Only one N+1 switchover is allowed at any time. The example below makes this point and assumes the following hardware:

- T-ONE card and E-ONE card with Redundant I/O
- T-ONE card and E-ONE card with Standby I/O

Only one set of redundant/standby cards can be in a switchover state. For example, if an E-ONE card fails after a T-ONE card has performed a switchover, the T-ONE must switchback before an E-ONE card switchover can begin.

Important! When a Matrix Controller is reset, the line cards resume the states they were in, and the Redundant I/O cards maintain the states they were in before the reset. The host can reset the Redundant I/O cards or you can reset them manually by cycling power to the CSP.

DSP Redundancy

To duplicate the generation of a tone for redundancy, configure the tone on separate DSP cards.

For resources, you must determine the number of DSPs your CSP requires and configure the cards. See the *DSP Series 2 Cards Product Description* chapter for information.

Span Configuration

Perform Physical (Layer 1) configuration tasks prior to performing protocol-specific configuration.

Span configuration tasks include the following:

- Span assignment
- Span format parameter assignment

Before configuring spans, assign Logical Span IDs with the *Assign Logical Span ID* message. Then configure the span Layer 1 characteristics using the either the *T1 Span Format Configure, E1 Span Configure, J1 Span Configure* message. Layer 1 parameters include framing format, coding method, error checking, signaling method, line length, and other Layer 1 parameters specific to the interface.

Common Channel Signaling

Configure common channel signaling cards as your application requires. Please see the *Developer's Guide: Common Channel Signaling* for detailed information.

Important! If you use T1 or E1 for common channel signaling, only the span formats are relevant

Figure 4-4 Messages Required to Configure Span Formats.

<table>
<tr><th>T1</th><th>E1</th></tr>
<tr><td>T1 Span Configure</td><td>E1 Span Configure</td></tr>
<tr><td>Trunk Type Configure</td><td>*PPL Assign</td></tr>
<tr><td>* Start Dial Configure</td><td rowspan="4">*PPL Transmit Signal Configure</td></tr>
<tr><td>* Flash Timing Configure</td></tr>
<tr><td>* Transmit Signaling Configure</td></tr>
<tr><td>* Receive Signaling Configure</td></tr>
<tr><td colspan="2">* PPL Timer Configure / PPL Configure</td></tr>
<tr><td colspan="2">Tag Configuration</td></tr>
</table>

* = optional message

Channel Configuration

Configure channels using the following components:

- Answer Supervision
- Distant and Local End Release
- Call Control Instructions.
  Use the *Inseize Instruction List Configure* and *Outseize Instruction List Configure* messages to configure a list of call control instructions for the CSP to use during real-time call processing.
- Inpulsing Parameters.
  See also the *Inpulsing Parameters Configure* message.
- Busy Out (for certain T1 trunk type interfaces).
  The Busy Out feature lets the host control the incoming call rate by busying out a selected group of trunks, normally on the front end of the system. Each trunk in the system has a flag associated with it that determines whether the trunk can be busied out.

  You can set the flag with the *Busy Out Flag Configure* message. When the host issues a *Busy Out* message with the busy out action field set to Busy Out, all trunks with the busy out field enabled are put out of service. The CSP can Busy Out the following trunk types:

  - E&M
  - FXO Loop Start
  - FXO Ground Start

By default, all trunks are configured with the busy out flag disabled. Therefore, after you configure the trunk types, enable those trunks that are to be put out-of-service. To put the trunks back into service, issue a *Busy Out* message with a busy out action of *Release*.

Service State

Introduction

Use the *Service State Configure* message to bring spans into service. After you bring spans into service, you can then bring channels, signaling links, and D channels into service.

The host controls the service state of many entities in the CSP, including spans. When spans are initially assigned a Logical Span ID, the spans are out of service. Using the *Service State Configure* message, the host controls when spans and channels are brought into service and out of service.

When a span is out of service, it does not transmit valid signaling and ignores any signaling it receives (except for Loop Timing sources, where the span is enabled and framing is monitored). In either case, the LEDs on the line card are off.

When the host brings a span into service, the CSP enables the span to transmit signals, and begins to monitor framing. The LED indicates either red (red alarm), yellow (yellow alarm), or green (framed up).

> **Important!** Bring channels configured for outgoing calls (or connection) into service before you bring channels configured for incoming calls into service.

DS0 Status Change

The CSP uses the *DS0 Status Change* message to notify the host of channels in a channel's state.

Do not consider a channel in-service until the host receives a *DS0 Status Change* message with a status of In-service. When a channel is in-service, the software scans for and responds to inseizures and may generate outseizures. Until a channel is in-service, the host does not see any *Request for Service* or *Request for Service With Data* messages for that channel from the CSP.

Channel Out of Service

A channel that is out of service ignores inseizures, and transmits on-hook signaling according to its trunk type. To prevent false *Request for Service* messages from the CSP, take faulty channels out of service. The CSP automatically brings all associated channels out of service, when one of the following conditions is detected for a span:

- Not connected
- Contains a fault
- In red alarm condition

When the span becomes valid again, all of the affected channels are brought back in service, as long as the channels were previously brought in-service by the host.

Tag Configuration

After you bring all the channels back into service, use the *Tag Configuration* message to assign a tag to a card's configuration. The host uses this tag in the *Card Status Report* and *Card Status Query* messages.

Reconfiguration Considerations

You should design your host application to ensure correct reconfiguration after resets.

Power-up

When you return power to the CSP, it retains the system software load but resets the configuration to default values. The host must restore the customizations (such as custom protocols) but it must wait until the *Poll* message indicates that the CSP is ready to take configuration. Review all fields in the *Card Status Report* message to verify the contents of the CSP.

Voice Recorded Announcements are stored in non-volatile flash memory, and are preserved even during power failures.

Single Matrix Controller Reset

Use the *Poll* message to detect a single Matrix Controller reset. The Matrix Controller state changes from initialize, to standby, to active or switchover, to boot. If the Matrix Controller has a valid system software load, it retains the load. The *Poll* message also indicates the Matrix Controller state changing from boot, to initialize, to standby, and then to active.

The CSP sends *Card Status Report* messages for all cards. The CSP retains the configuration that was last sent by the host (except when the CSP is being powered down or when a card is being removed).

When a single Matrix Controller is reset, all calls that were previously connected are torn down and the information on all of the in-service channels reports to the host with the *DS0 Status Change* message. The host determines the cause of the reset.

If a software fault caused the reset, please send the Fault Log to Dialogic Technical Support. To obtain Fault Logs, send the *Fault Log Query* message to the CSP.

Standby Matrix Controller Reset

The host is notified of a standby Matrix Controller reset two ways:

- The *Poll* message from the standby Matrix Controller indicates that the Matrix Controller state has changed from Initialize or Standby to Boot.
- The *Poll* message from the active Matrix Controller indicates that the standby Matrix Controller has reset.

The system software load is retained. The *Poll* messages of the standby Matrix Controller indicate the return to the standby state. Any information shared by the active and standby Matrix Controller is sent to the standby from the active Matrix Controller.

The active Matrix Controller sends a *Card Status Report* message to the host reporting on the standby Matrix Controller. The *Poll* message reports that the adjacent Matrix Controller is not responding to the Polls. A Matrix Controller reset does not affect connected calls or calls that are in the process of setup.

Active Matrix Controller Reset

The active Matrix Controller sends *Poll* messages to the host to report a reset. The *Poll* messages report the Matrix Controller state changes as it transitions from Active, to Boot, to Initialize, to Standby. The standby Matrix Controller sends *Poll* messages that report the Matrix Controller state changes from Standby, to Switchover, to Active.

When the standby Matrix Controller changes to the active state, the CSP sends *Card Status Report* messages for all cards to the host applications. The Matrix Controller retains the configuration. We recommend creating your host application so that it verifies that the Matrix Controller has retained the configuration. You can use the configuration tags that are returned in the *Card Status Report* messages, as well as the tags that the application stores for each line card.

The application should prepare itself to communicate with the new active Matrix Controller. The CSP maintains connections for all channels that were connected before the switchover occurred. But calls that were in the process of setup are torn down. The CSP sends the *DS0 Status Change* message to the host to report all of the in-service channel states. If a software fault occurs, please notify Dialogic Technical Support.

Line Card Reset

When a line card resets, the CSP notifies the host with the *Card Status Report* message. The cards maintain their software load and configuration. Your application should check the configuration tag to confirm that configuration is intact.

All calls associated with the card, both connected and in the process of setup, are torn down. The *DS0 Status Change* message informs the host of the affected channel states. If the CSP is configured for line card redundancy, your application can configure the standby line card and can instruct it to become active.

DSP Card Reset

The CSP sends the *Card Status Report* message to the host to report a DSP card reset. The card maintains its software load and configuration. Use the configuration tag to verify the card configuration. The CSP removes the DSP resources from service during the reset. If a second card has the same configuration, DSP resources are maintained.

Channel Purge

If the CSP cannot appropriately correlate conditions relating to a channel, a channel purge occurs. The channel purge ensures that the CSP retains channels when an unexpected condition occurs, such as the following:

- Unexpected signaling conditions
- Unexpected results of related channels
- Timeouts

When a channel purges, the host receives a set of three *DS0 Status Change* messages with the following values:

- Channel Status: 0x01 (Out-of-Service)
- Channel Status: 0x00 (Purge)
  Purge Status: See Purge Status Values for purge reason
- Channel Status: 0x02 (In-Service)

If a channel purges, the host may not receive acknowledgments for outstanding call processing messages sent to the purged channel. Therefore, design all host applications to handle a channel purge.

Application Requirements

Overview

This section explains the modules you need in your application so that it will work effectively with the Converged Services Platform.

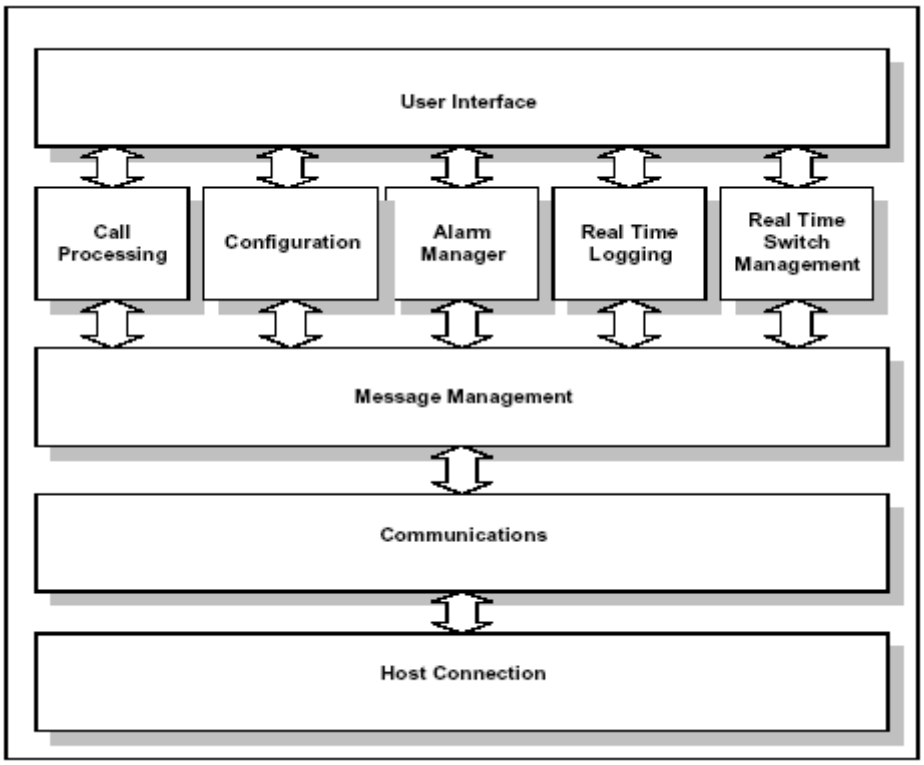
Create Core Modules First

When you begin developing your application, you must first create specific core modules. We recommend the following as a minimum set of core modules:

- Host Connection Module
- Communication Module
- Message Management Module
- Call Processing Module
- Configuration Module
- Alarm Manager Module
- Real-Time Logging Module
- User Interface Module
- Host Communication Module

After you create all of your core modules, you can begin creating modules to meet your specific needs. For example, you might require a module that manages a relational database, another module to perform number lookups, and a module with a graphical user interface that displays channel-specific information.

Figure 4-5 Application Modules



Host Connection Module

The host, as the client in the client/server model, maintains the connection. The host application can issue any number of requests to connect and disconnect. The CSP services each connection request instantaneously. Because each Matrix Controller card supports only one connection to one host, each additional connect request issued overrides the previous connection.

Communication Module

Overview

The Communication Module reads and writes bytes to the hardware link that connects the host to the CSP. This module can frame and queue messages, calculate checksums, process special characters, and queue host messages for CSP acknowledgment.

Sequence Number Logic

You should include in your Communication Module a mechanism to assign sequence numbers to messages as they are sent out the communications port, and for matching these sequence numbers to CSP responses, using the message type.

The Communication Module must monitor the *Poll* message constantly. The *Poll* message also relays the states of the active and the standby Matrix Controller cards.

Matrix Controller State

In a redundant CSP, the Communication Module recognizes when the active matrix is no longer active because:

- The Matrix Controller card is no longer sending *Poll* messages
- The standby Matrix Controller card sends *Poll* messages that indicate that no adjacent matrix is detected
- The standby Matrix Controller card sends *Poll* messages that indicate that its state has switched from standby to switchover, and then to active.

The Communication Module must also recognize the need to re-route messages intended for the original active Matrix Controller card to the standby Matrix Controller card (the new active Matrix Controller card). When the *Poll* message indicates that the new active Matrix Controller card is ready for configuration, the Communication Module should inform the Configuration Module.

System Software Downloaded

The *Poll* message informs the host application that the CSP needs software downloaded, and the Communication Module performs the download.

> **Important!** The fastest way to download software to the CSP is to use the binary downloading logic. In a redundant CSP, the Communication Module must also download the standby Matrix Controller card, and then the standby Matrix Controller card downloads the active matrix. This method causes the least amount of downtime for a CSP in the field.

During the development phase, you may be downloading different loads, but downloading on top of an existing download can be time-consuming. It is usually faster to clear the existing software load and then reset the Matrix Controller card or cards to download to the ROM code. The *Clear System Software* and *Reset Matrix* messages are useful for these processes.

Message Timeout Logic

You should include message timeout logic in the Communication Module. Timeout values can vary considerably, because with one configuration message, you can configure one channel or many channels that may span many line cards.

Timer Logic

The host determines whether to use multiple timers or a single, global timer. A global timer must be long enough so that a configuration message sent to the CSP would have enough time to complete. If the host uses multiple timers, some experimentation with the configuration messages is required.

Message Distribution Logic

The Communication Module should also distribute messages from the CSP to the modules that need them. The module may need to send a message to more than one other module.

Include the following basic functionality in your application:

- Framing
- Calculating checksums
- Handling special characters
- Downloading System Software

Message Management Module

Overview

Implement a mechanism to manage messages to:

- Assign sequence numbers to messages as they are sent out of the communications port.
- Match CSP responses to these messages using the sequence number and the message type.
- Support multiple CPUs to allow for message traffic to both active and standby matrices. Design your Message Management module so the host knows when to send messages intended for the formerly active Matrix Controller card to the now active Matrix Controller. Make this module inform the Configuration module when the *Poll* message indicates that the active Matrix Controller is ready for configuration.

Sequence Number Logic

You should include in your Message Management Module a mechanism to assign sequence numbers to messages as they are sent out the communications port, and for matching these sequence numbers to CSP responses, using the message type.

Timeout Logic

You should include message timeout logic in the Communication Module. Timeout values can vary considerably, because with one configuration message, you can configure one channel or many channels that may span many line cards.

Timer Logic

The host determines whether to use multiple timers or a single, global timer. A global timer must be long enough so that a configuration message sent to the CSP would have enough time to complete. If the host uses multiple timers, some experimentation with the configuration messages is required.

Message Distribution Logic

The Message Management module should distribute the messages from the CSP to the modules that need them. You may want to design your modules so that a message can be sent to more than one other module.

Call Processing Module

Important Note

Only **one** call processing message can be outstanding per channel at a time. The host must wait for the call processing message ACK before sending the next message for a given channel. The response status should also be checked to determine if the host-initiated message was processed successfully or not. These messages typically:

- Report incoming calls
- Pass address digit information
- Make connections and re-connections
- Manage DSP resources per call

Introduction

The call processing module contains the high level call management of the application. The logic within this module is determined by the call model required by the host application.

Implementation of a host-level call model involves sending/receiving a series of EXS API call control messages on a per channel basis. These messages are typically used to report incoming calls, pass address digit information, make connections/re-connections, and so on.

One way to implement the call processing module is with a software finite state machine. Upon sending a call control message to the CSP, a channel enters a "response wait" state. Upon receiving a positive ACK, the next operation is performed (that is, send another message, wait to receive a message). Each channel should be treated separately, allowing for channels to be in different call management states simultaneously.

DS0 Status Change message

The *DS0 Status Change* message is vital to the call processing module. The *DS0 Status Change* message should be handled in any state that the channel may be in. This message reports channel In/Out of Service as well as error information.

Channel State Queries

You may also want to provide support for channel state queries in the event of a failure at the Call Processing Module. To accomplish this, use the following messages:

- *Channel Connection Status Query*
- *Channel Parameter Query*
- *B Channel Query*
- *SS7 CIC Query*
- *Generic Report*

Incoming Call Setup

When the CSP receives an incoming call, it sends a *Request for Service* message to the host by default. To have the CSP perform internal routing without host intervention, please refer to Chapters 1 and 2 of the *Developer's Guide: Line Cards*.

When an incoming call is detected, the Signaling layer begins to execute the channel's preprogrammed instructions for inseize control. You can configure Signaling to send a single report with all collected address information, or intermediate setup events such as dialtone detection and digit strings.

The Signaling layer receives several stages of digit strings during incoming call setup.

First, configure each stage for the following with the *Inpulsing Parameters Configure* message:

- Address signaling type
- The number of strings in the stage
- The digit collection method per string

You can configure call processing with a variety of requirements for line signaling and address data exchange, including multiple stages of inpulsing and outpulsing digits.

Allow the CSP to perform as much of the setup on its own as possible. This helps to optimize call processing and simplify application development. When sending call processing messages, the host must wait for the acknowledgment of one message before sending the next call processing message for that channel.

Outgoing Call Setup

With a single command to the CSP, the host initiates an outseizure with outpulsing digit strings. The command includes preprogrammed instructions for completing outbound call setup.

The CSP accommodates multi-wink/stage signaling interfaces for Feature Group D by detecting multiple start dial signals, and by outpulsing multiple stages of address digits.

As part of the outbound instruction sequence, you can activate call progress analysis for each call or for all outbound calls over a specified channel.

The *Route Control* message with a seize instruction initiates the outseizure (a seize instruction cannot be preprogrammed). When the CSP receives a positive acknowledgment for the outseizure, it executes the outseize instructions that have been configured for that channel. You can also initiate an outseize using the *Outseize Control* message.

You can use the internal router in the CSP to find a terminating channel.

Preprogrammed Instructions

Instruction Lists

To reduce interactive messaging, you can preprogram instructions for a channel at the Signaling layer for faster call processing and simpler host applications. To preprogram instructions for a channel, use the *Inseize Instruction List Configure* and *Outseize Instruction List Configure* messages.

> **Important!** The ISDN, SS7, and IP calls do not support the following messages: *Inseize Instruction List Configure*, *Outseize Instruction List Configure*, and *Inseize Control*.

The host sends inseize and outseize instructions to the CSP using *Inseize Control* and *Outseize Control* messages. The instructions reside within these messages in blocks of data named Information Control Blocks (ICBs). You can insert multiple ICBs into these messages.

Customizing Preprogrammed Instructions

You can customize the default instruction lists for individual channels with the *Inseize Instruction List Configure* and *Outseize Instruction List Configure* messages. Your customized list can have up to 20 instructions. (The explanation of these messages in the *API Reference* contain examples to help you configure your own lists.)

Initiating Preprogrammed Instructions

To initiate a preprogrammed instruction list for a channel, you can send a *Use Instruction List* ICB in the *Inseize Control*, *Route Control*, or *Outseize Control* messages.

The Use Instruction List ICB allows the host to begin at any instruction number in the list. The Signaling layer begins executing at the specified instruction number and continues until it encounters a "Wait For Host Control" instruction.

This arrangement allows the host to program a channel’s instruction list in sections. The host can use different sections of the list for different scenarios, depending on the processing needs of a specific call.

Process

When an inseizure is detected, the Signaling layer executes the channel's instruction list. The host initiates outseize instructions by issuing a *Route Control* or *Outseize Control* message with an instruction of "Use Instruction List."

> **Important!** To initiate outseizures, use an instruction of Seize. You cannot include a seize instruction in a preprogrammed list.

Using preprogrammed instructions, the host determines the following:

- What information it receives from the CSP.
- When it receives information from the CSP.
- At what point during call setup it begins interactive processing.

In a preprogrammed list, the last instruction should be "Wait For Host Control" or "Wait For Host Control With Answer Supervision" (outseize only). The CSP then waits for more instructions from the host before it proceeds with call setup. The host can then manage the call interactively, or initiate the instruction list again at a different instruction number.

Configuration Module

Overview

The Configuration Module configures the CSP. You should design your Configuration Module with enough flexibility to reconfigure the entire CSP, as well as a single card. The module should also be able to take groups of channels in and out of service. The configuration script file contains ideas on how to organize a CSP configuration.

Your Configuration Module should be designed to receive information about the state of the CSP from the Communication Module. If the CSP has just been turned on, your Configuration Module must configure the entire CSP. The Configuration Module does not need to do anything if the CSP resets and sends a *Card Status Report* indicating that the battery-backed configuration is valid (see the *Card Status* field).

After it configures a card, the Configuration Module can use the *Tag Configuration* message to tag a configuration with a number. The *Card Status Report* message also returns a configuration tag. Design your host application to detect the presence of previously-configured tags. If previously configured tags are present, the host does not need to reconfigure the card.

> **Important!** Always send the *Tag Configuration* message last. For a list of messages that reset the tag configuration, see the *Tag Configuration* message in the *API Reference.* Any configuration message (other than the *Tag Configuration* message) sent to a card initializes the card's configuration tag to 0 (zero).

When you need to clear the configuration, use the *Reset Configuration* message in your Configuration Module. This message allows the host to initialize configuration of a line card (a card in slots 0x00 through 0x0F) or the entire CSP. In a redundant CSP, sending a *Reset Configuration* message to initialize the entire CSP causes a switchover to occur.

You should design your application for dynamic deconfiguration and reconfiguration. Provide for such changes as the following:

- Incremental configuration
- Adding facilities
- Changing the existing configuration

Host Connection Configuration

As the client in the client/server model, the host must maintain the connection. The host can issue any number of requests to connect and disconnect. The CSP services each connection request instantaneously. But because each Matrix Controller card supports only one connection to one host, each additional connect request overrides the previous connection. To establish the connection, the host sends a *Connect* message that specifies a port number of 0x3142, along with the IP address that was configured when the Matrix Controller card was turned on. You can encounter the following scenarios when disconnections occur both gracefully and ungracefully:

Graceful Disconnection

In the CSP, a server task runs continuously so that, it is always ready to send an *Accept* message as soon as it receives a connect request. If the host crashes and is unable to issue a *Close* message, it must issue a new connect request when it returns to service. The CSP instantly accepts this connect request.

Ungraceful Disconnection

The CSP software issues a *Close* message on a socket only if it detects that the socket has been closed by the host. The host software must have a strategy for handling connections that are broken by the CSP. Before sending the *Receive* message, the host should send a *Select* message on the receive socket descriptor, with a time-out value that is slightly greater than the poll rate. If the host is alerted of a timeout, it assumes that the CSP has broken the connection. The host must adjust the socket descriptors and issue a new *Connect* request.

Alarm Manager Module

Overview

The host must manage *Alarm* messages sent from the CSP.

Alarm Message

The *Alarm* message includes the following fields:

Severity

Dialogic has assigned the following severity levels to alarms, but you should become familiar with each alarm and determine how your application should treat them:

- Informative
- Minor
- Major

Alarm Type

Alarms are identified by the following types:

- General
- Card
- Span
- Channel
- DSP
- DSP SIMM
- CSP Node
- DS3
- VoIP Module

Alarm Number

Each alarm within a type has a unique number. See the *Alarm* message in the *API Reference* for alarm numbers.

Data Length

Some alarms have unique data associated with them. The *Data Length* field preceding the data indicates the length of the data to follow. If there is no data, the length is 0x00.

Data

Please refer to the specific *Alarm* message in the *API Reference* to see the data associated with that alarm, if any.

Real-Time Logging Module

Introduction

It is important for the host to enable a logging feature to capture host-to-CSP and CSP-to-host messages. If a fault occurs, you should immediately retrieve a fault log with the *Fault Log Query* message, and send this information to Dialogic Technical Support for analysis.

Fault Logs

The two types of fault logs are as follows:

- *Fault Log* (0x01). This is a NULL terminated ASCII string.
- *Fault Log and Stack Trace* (0x03). This is a NULL terminated ASCII string, followed by the stack pointer of when the fault occurred, followed by a trace of the stack.

Card Status Report

An *Alarm* messages is often accompanied by a *Card Status Report* message. Pay special attention to the *Card Status Report* message. It contains information such as the card's serial number and configuration. It could also contain other important information.

For example, in the message's *Card Status* field, the "faults logged" bit could indicate that a software fault has occurred on a specific slot. In a case such as this, it is important for your host application to immediately send a *Fault Log Query* message to the card in that slot. The *Fault Log Query* message retrieves the fault information from the card. Send this information to Dialogic Technical Support for analysis.

User Interface Module

Introduction

The User Interface module is able to perform many functions, such as downloading system software to the CSP, sending interactive configuration and query messages, and sending a configuration script file to configure the CSP. The module should be able to send a string of hex bytes to the CSP so that at any time you can send any message to the CSP and receive a response.

Create your User Interface Module so that you can disable the features of a message but still send the message and receive a response. That way, when Dialogic provides you with new messages as part of a software update, you can test the messages before you embed them in your applications.

Queries

Dialogic provides many messages that allow the host to query the CSP. Be sure to include all of the query messages that could apply to your application. Queries are important for debugging, because they retrieve a great deal of helpful information from the CSP. Queries are also important to ensure that new configurations are correct.

Query messages include the following:

- *E1 Span Query*
- *PPL Audit Query*
- *Card Status Query*
- *Fault Log Query*

For a complete listing of query messages, see the *API Reference*.

Host Communication Module

Overview

The host, as the client in the client/server model, maintains the connection to a Matrix Controller, SS7, ISDN, VDAC-ONE, or IP Network Interface Series 2 card. The host application can issue any number of requests to connect and disconnect. The CSP services each connection request instantaneously. Each subsequent connect request issued “steals” the previous connection.

API Messages

All API messages should be sent directly to the active matrix card’s IP address. Messages should not be sent to the matrix card’s shared IP address; the shared IP address is used for SIP redundancy only.

IP Address

Introduction

Before you can establish a connection link from the host to the CSP (Matrix Controller card) as described in the *CSP Hardware Installation and Maintenance Guide*, you must obtain an Internet Protocol (IP) address.

Acquire an IP Address

The IP address is a 32-bit address used in IP routing.

IP Addressing on Matrix Controller card

IP addresses and subnet masks can be assigned to a Matrix Controller card through Reverse Address Resolution Protocol (RARP), Bootstrap Protocol (BOOTP), or the debug console port. If the console is used, the IP address is saved in Electronically Erasable Programmable Read-Only Memory (EEPROM).

When a Matrix Controller card resets and finds an invalid IP address, subnet mask, or gateway address stored in the EEPROM, it issues five RARP requests. If the Matrix Controller card receives no reply, it issues five BOOTP requests. If there still is no reply, it waits 20 seconds for input from the console. If there still is no reply, it repeats RARP, BOOTP and console indefinitely until it gets an IP address. The Matrix Controller has to have an IP address before the host connects.

IP Addressing on ISDN or SS7 cards

For ISDN or SS7 cards, you can use RARP, BOOTP, or the *IP Address Configure* message to configure the IP address. If *IP Address Configure* is used, the IP address is saved in EEPROM.

If the host sends an *IP Address Configure* (0x00E7) message with an IP address and subnet mask that differ from the values stored in the EEPROM, the EEPROM is overwritten with the new values. For the new IP address and subnet mask to take effect, you must reboot the card.

If the host assigns the same IP address and subnet mask, the message is acknowledged and the card does not need to be reset, so the host can send the *IP Address Configure* message whenever it is necessary.

When an ISDN or SS7 card finds an invalid IP address or subnet mask stored in the EEPROM, issues five RARP requests. If the card receives no reply, it issues five BOOTP requests. If the card does not receive a reply from either a RARP or BOOTP server, then the card will start up without TCP/IP services being started.

To store an invalid IP address and subnet mask in the EEPROM, use an *IP Address Configure* message with the IP address set to 255.255.255.255 and any value for the subnet mask.

Set or Change an IP Address on a Matrix Controller

To manually set or change the IP address, use the debug port on the Matrix Controller card and then follow the steps below:

1. While the Matrix Controller card is running, enter `U` from the debug terminal. This deletes all stored IP address information. You are prompted to reset the Matrix Controller card.
2. Press **RESET**. You are prompted to enter a new IP address.
3. Enter the new IP address. You can use any class.
4. Enter the subnet mask. The subnet mask must be entered as an eight character hexidecimal value.

Subnet Masks

The subnet mask defaults to the values show in the following table. Make sure that cards are in the same subnet.

Class	IP Address	Subnet Mask
A	0.0.0.0 - 127.255.255.255	0xFF000000
B	128.0.0.0 - 191.255.255.255	0xFFFF0000
C	192.0.0.0 - 223.0.0.0	0xFFFFFF00

Set or Change an IP Address on an ISDN or SS7 card

If the host sends an *IP Address Configure* (0x00E7) message and the IP address and subnet mask in the message are different from the values stored in the EEPROM, the EEPROM is overwritten with the new values. The card will need to be rebooted for the new IP address and subnet mask to take effect.

If the host assigns the same IP address and subnet mask, the message is acknowledged and the card does not have to be reset. This allows the host to send the *IP Address Configure* message whenever it considers it necessary.

For SCCP/TCAP applications, you may configure whether the SS7 local host is used for each stack/subsystem. For SS7 stack/subsystems on the host that must communicate directly with the SS7 card, you must configure the local host as the host using the Config Type 0x08 in the *SS7 SCCP/TCAP Configure* (0x77) message.

Clearing IP Addresses

You can clear IP addresses by setting all of the IP address and subnet mask data to 0xFF in the *IP Address Configure* (0x00E7) message. If you plan to take a card with IP addresses from one chassis and insert it into another chassis, be sure to clear all IP addresses on the card first. Otherwise, there might be a conflict with the IP addresses of the cards in the new chassis.

More information

For more information about IP addressing for VDAC cards see the *Developer's Guide: Internet Protocol (IP)*. For more information about direct host communication with an SS7 card using TCP/IP, see the *Developer's Guide: Common Channel Signaling*.

System Maintenance

Design your host application to use API messages for maintenance functions, such as monitoring and troubleshooting.

Maintenance Features

The CSP supports the following maintenance features:

- Alarms
- Channel Status
- Card Status Report
- Poll Messages
- Fault Log
- Host Link Failure Detection
- Message Resend Logic
- System Busy Warning and System Busy Alarm
- Reports
- PPL Auditing
- Line Card Overload Logic
- Loopback Diagnostics

Alarms

When the CSP detects a possible problem, it sends the *Alarm* message. The message displays information specific to each type of alarm, as well as the severity of the possible problem. When the problem condition is resolved, the *Alarm Cleared* message is sent.

Channel Status

Your application must be able to process the *DS0 Status Change* message for in-service and out-of-service conditions. The CSP sends this message to inform the host that the status of a specific channel has changed. The message also informs the host of the current status of the channel.

Card Status Report

If a card fails, the CSP sends an *Alarm* message and a *Card Status Report* message. Design your application to generate a *Card Status Report* with the *Card Status Query* message.

Poll Messages

A *Poll* message checks whether an Matrix Controller is connected to the host with a functioning communication link. The host can use the *Poll Request* message to obtain information from the Matrix Controller or set up its own polling scheme. Use the *Poll Interval Configure* message to change the interval between *Poll* messages.

Fault Log

Design your application to send the *Fault Log* message when it receives the following messages:

- *Card Status Report*
- *Card Status Query*, with the Faults Logged bit set in the Card Status byte

The CSP sends separate responses for each fault logged on the specified slot. After the log is completed, you can clear the log by setting the Action field to 0x02. Collect all data in the fault log(s) and send it to Dialogic Technical Support for analysis.

Host Link Failure Detection

Introduction

If the CSP detects a failure in the host link, all spans/channels are brought out of service with the Host Link Failure Detection feature.

> **Important!** When the host reconnects all of the channels will be out of service and it is the host responsibilty to bring the channels back in service.

Determining HLF

The CSP determines the integrity of the host link by monitoring host acknowledgments to *Poll* messages. You must enable polling to use the Host Link Failure Detection feature.

If the host does not acknowledge two consecutive *Poll* messages, the CSP initiates a timer. If the CSP does not receive an acknowledgment from the host before the timer expires, the CSP resets its Ethernet receiver.

Configuration

You configure Host Link Failure options using the *System Configuration* message.

Data[1] configures the CSP response to Host Link Failure:

0x00 - Bring all channels out of service

0x01 - Bring all channels and spans out of service

Data[2,3] configures the Failure Confirmation Timer.

Process

When the CSP detects a host link failure, it responds according to one of two options set by the host and sends an *Alarm* message to the host to report the failure (Alarm Type: Card 0x02, Alarm Number: 0x10).

If the host link failure is related to CSP hardware, it is resolved when the Ethernet receiver is reset. If the host link failure is not related to CSP hardware, it may have been caused by slow host processing.

If this is the case, you should consider increasing the polling interval. Slow host processing is most evident during the boot-up cycle. This alarm may also indicate that the host is not sending messages to the CSP. If this is the case, the CSP will have stopped sending responses (because there are no host messages to respond to).

Message Resend Logic

Introduction

The Message Resend Logic (MRL) feature determines how the CSP responds to unacknowledged messages.

Configuration

You can configure the following MRL options for all messages:

- Do Not Resend
- Resend Once *
- Resend Once and Generate *Alarm*

  The CSP sends a *General* alarm of *Unacknowledged Message* to the host. The message includes the bytes of the message being resent.

  Important! An unacknowledged *Alarm* message is resent, but it does not generate an additional *Alarm* message.

* The system default is *Resend Once*. In the case of an Matrix Controller switchover, the Message Resend Logic resets to *Resend Once*.

Critical Message Resend

If you wish to have specific critical messages resent until they are acknowledged by the host, enable the *Resend Until Acknowledged* (0x03) option. This option applies to the following messages:

- *Alarm*
- *Alarm Cleared*
- *Request For Service*
- *Request For Service With Data*
- *Channel Released*

This option helps to ensure that the host receives critical messages that, because of a host switchover for example, it may not have received initially. To enable *Resend Until Acknowledged*, you must also enable *Resend Once*.

System Busy Warning and System Busy Alarm

Overview

The CSP sends a System Busy Warning in the *Alarm* message to indicate that the traffic volume on the CSP is approaching its capacity. This message is only a warning; it does not mean a System Busy condition will definitely occur. When the CSP is overloaded by a high volume of traffic, it sends a *System Busy* alarm to the host. The alarm indicates that the CSP cannot process any more incoming or outgoing calls until the condition clears.

Subsequent messages from the host may or may not be lost, but Dialogic recommends reducing the host's call processing until the System Busy condition clears.

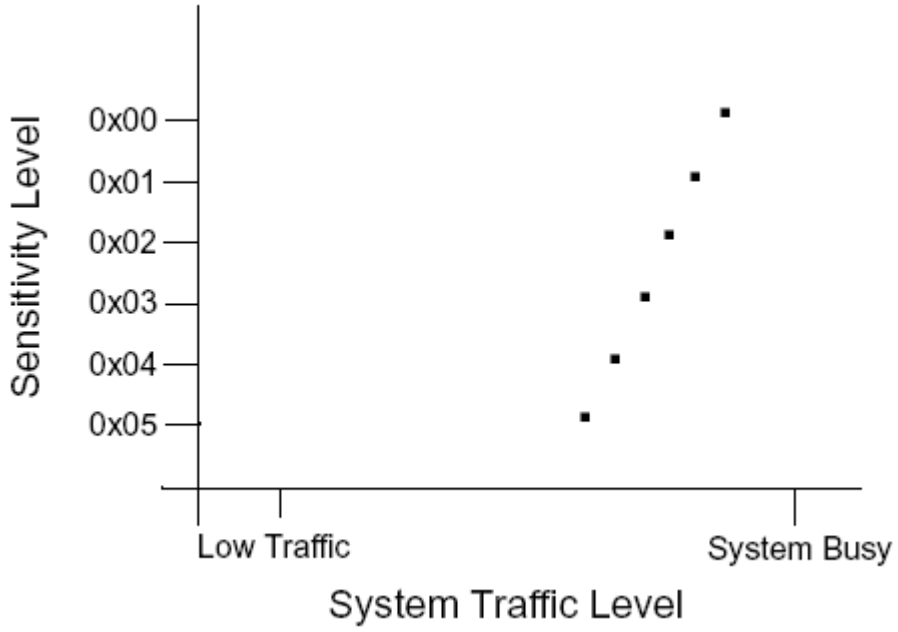
Configuration

Using the Data[0] byte in the *System Configuration* message, you can specify what volume will trigger the *System Busy Warning* message. This setting affects the amount of time the host has to respond to a potential System Busy condition. The least sensitive setting (0x00) gives the host the least amount of time between a System Busy Warning and a System Busy condition. The most sensitive setting (0x05) gives the host the most amount of time between a System Busy Warning and a System Busy condition.

The actual time between a System Busy Warning and a System Busy condition cannot be calculated precisely. We recommend testing the CSP to find an appropriate sensitivity level for each application.

Determining System Busy

The graph below shows the relationship between the sensitivity level and the amount of warning time as the system approaches a System Busy condition. The dots in the graph represent the point at which a *System Busy Warning* alarm is sent to the host.



System Busy Cleared

When the busy condition clears, the CSP sends the *Alarm Cleared* message to the host.

System Memory Low

Two alarms pertaining to system memory are reported with the *Alarm* (0xB9) message.

- 0x1F - *System Memory Low Alarm* (General/Major)

The CSP sends a *System Memory Low Alarm* when system memory has dropped below the minimum level of satisfactory performance. When the system is in this condition, it is also in a System Busy condition and the CSP sends a *System Busy Alarm* (0x01) to the host.

- 0x20 - *System Memory Satisfactory* (General/Informative)

When the system returns to the minimum level of satisfactory performance, the CSP sends a *System Memory Satisfactory Alarm* (0x20) to the host, along with an *Alarm Cleared* (0xC1) message.

Software Error Alarm

Software errors that are not severe enough to cause a system fault are reported as an informative card alarm in the *Alarm* message. If you

receive a *Software Error Alarm* (0x17), please forward the data to Dialogic Technical Support.

Reports

System Configuration Query

When the host requests information using the *System Configuration Query* message, the CSP sometimes responds with the *Generic Report* message.

Active Conference ID Report

The host queries the total number of active conference IDs by sending the *System Configuration Query* message with Active Conference IDs (0x04) as the Query Type.

The CSP responds with a *Generic Report* message that indicates the IDs of the active conferences. Each *Generic Report* message can report a maximum of 50 Conference IDs. So for example, if a system has 100 conference IDs, two *Generic Report* messages are required.

Channel State Report

The host queries information on specific channels by sending a *System Configuration Query* message with Channel State (0x05) as the Query Type.

The CSP responds with the *Generic Report* message that displays the state of each channel in Layer 3 and Layer 4. The CSP sends a separate *Generic Report* message for each span queried.

PPL Auditing

Introduction

You enable PPL Auditing for each card individually, using the *PPL Audit Configure* message. When PPL Auditing is enabled on a card, all state machine activity on the card is recorded in an audit log on either a per-PPL component basis or an individual entity basis.

The *PPL Audit Configure* message accepts and validates all AIBs that correspond to the provided component ID. A message received with a Slot AIB will be processed as in previous releases. A message received with an AIB, other than a Slot AIB, will be processed as an Individual Entity Audit configuration message.

The Audit Type field is a bit mask that provides bits 0 through 3 to enable or disable PPL Auditing features on a component or entity basis. For example, setting bit 0 enables PPL auditing. See the *PPL Audit Configure* message for the Audit Type information. If a card has PPL Auditing enabled, the activities of each PPL component are recorded in an audit log. The entries in the audit log represent a state transition on a specific entity (channel).

There are two new bit options in the Audit Type field. Bit 3 allows configuration of individual entities, and bit 2 enables the configuration. Both bits can be set at the same time or bit 2 can be enabled or disabled at any time to activate or deactivate individual auditing.

PPL Components

To support PPL Auditing on an Entity basis, the *PPL Audit Configure* message supports AIBs that are associated with the following PPL components:

- E1 (0x0001)
- T1 (0x0003)
- ISDN L3P Call Control (0x0005)
- ISDN L3 Call Reference (0x0008)
- SS7 L3P CIC (0x000F)
- SS7 ISUP CPC (0x0012)
- Channel Management (0x0061)
- Call Management (0x0062)

PPL Audit Query

The *PPL Audit Query* message supports Slot AIB, and all AIBs that are associated with the PPL components. This message supports the querying of all PPL Audit configurations. For example, when this occurs a single byte of Audit configuration data will be returned. This byte will contain the current settings for board level PPL Auditing, board level PPL Error Alarm settings, and Individual PPL Auditing.

If the AIB specifies an entity other than a Slot AIB, the byte will also contain entity level PPL Auditing and entity level Error Alarm settings. For example, this information will be queried by setting the Audit Type to 0x02 in the *PPL Audit Query* message. The *PPL Audit Query* message Audit Types are as follows:

- An Audit Type of (0x00) requests the PPL Audit information for the specified entity.
- An Audit Type of (0x01) requests a PPL ERROR audit on a per card basis.
- An Audit Type of (0x02) requests the current PPL Audit configuration. It will return a PPL Audit Query response containing the status, the AIB, the component ID, the Audit Type, and an Audit Configure byte containing the current Audit Configuration setting. The slot AIB will be supported when the Audit Type is Audit Configuration (0x02). The bits (0 through 4) of the Audit Configuration byte are listed in the *PPL Audit Query* message.

Audit Data

The length of the audit entries and the specific data logged depend upon the PPL component and upon the type of audit. See the *PPL Audit Query* message for the specific data returned. The following information is included in all audit entries:

- Protocol ID
- State Status
- Protocol-Specific Information
- PPL Event
- Initial State
- Next State
- Error Status
- Timestamp

Retrieving an Audit Log

To retrieve an audit log, send a *PPL Audit Query* message. There are two audit query types:

- PPL State Transitions
- PPL Errors

The host queries the CSP about PPL state transitions for each PPL component and entity. The CSP reports all state transitions for a specified PPL component on a specified entity. The host queries the CSP about PPL errors for each PPL component and card. The CSP reports all PPL errors for the specified PPL component on the specified card.

You cannot retrieve an entire log with one message. Audit entries are grouped into audit blocks. When you query a log, specify Audit Block 0 in the first message, Audit Block 1 in the second message, and so on until all audit entries have been returned. When there are no audit entries remaining, the response status to the *PPL Audit Query* message indicates 0xD7 (End of PPL Audit Data).

PPL Error Alarm

If you enable the PPL Error Alarm, the host is notified with an *Alarm* message when a PPL error occurs on an entity. The PPL Error Alarm is enabled by setting bit 1 of the Audit Type field in the *PPL Audit Configure* message. The *Alarm* message indicates Entity - **0x04** (Channel) and Alarm - 0x03 (PPL Error). The Alarm Information fields contain the audit entry data from the PPL Audit log.

Line Card Overload Logic

Introduction

Dialogic has developed this Line Card Overload Logic to detect, prevent, and react to overload (busy) conditions. Similar logic resides on the Matrix Controller, which detects the overload condition and sends *System Busy* messages to reduce the load. Similar overload detection logic is implemented directly on the 16-span line cards and SS7 PQ cards. The logic on these cards stops traffic from the host and network, and tries to bring the cards gracefully back into a normal state. Highlights of the Line Card Overload Logic feature are as follows:

- The detection logic is common to all line cards.
- Existing calls are preserved.
- Busy conditions are handled gracefully.
- The host can configure all thresholds and parameters for the logic.
- The logic minimizes processing, memory usage, and inter-task messages during a busy condition.

Purpose

The purpose of this feature is to prevent cards from faulting due to overload. The Line Card Overload Logic allows messages to be grouped for more efficient processing. If the card does become overloaded, the situation is handled gracefully, until the overload condition resolves.

When the card's traffic reaches the Approaching Busy threshold, the CSP sends an alarm to the host, signifying that the card is at risk of becoming overloaded. If the condition persists and progresses to "Real Busy," all new calls from the host and network are rejected, and idle channels are taken out of service. This way, existing calls remain intact and they are allowed to complete.

To reduce traffic that can lead to overloads, inter-task and inter-card messages (such as *DS0 Status Change* messages) are grouped for greater efficiency. The host can enable and disable message grouping, and it can configure various thresholds. A line card reports resource usage in response to a host request. A line card can also be configured to send this report at regular intervals.

The logic is common to all relevant cards, but the action differs, depending on the card. Timer 1 interrupt monitors the memory, Message Control Blocks (MCB), and CPU usage. To determine CPU usage, profiling on all cards is always ON. The average usage for the cards is sampled and checked against the configured thresholds.

When either of these tests shows usage above the Approaching Busy or Real Busy thresholds, an *Alarm* message is generated. When the resource usage falls back below Safe Level, the alarm is cleared. If the card usage drops from Approaching Busy down to Safe Level, the Approaching Busy alarm is cleared.

Configuring Thresholds

The host can configure two threshold levels for each line card. The Matrix Controller sends a message to the line card, with a value indicating the thresholds for CPU use and Resource use (MCB and Memory). The line card then sets values for the Approaching Busy and Real Busy thresholds.

Table 4-1 Threshold Parameters

Usage Designation	Meaning
Real Busy threshold	The maximum usage allowed
Approaching Busy threshold	The intermediate level. The card is approaching a busy condition
Safe Level	20% less usage than the Approaching Busy threshold. Usage is considered normal

Reporting Resource Usage

The host can turn the Resource Utilization Report ON and OFF (OFF is the default) using the *System Configuration* message. Within the message, the line card is specified in an Address Information Block (AIB) and the time interval for reporting is also specified. When the timer expires, the report is sent and it contains the value accumulated since the last report.

Grouping DS0 Status Changes

A channel and span's status can change because of hardware problems, or it can be changed by the host. For example, the host can bring up all channels on a span all at once, generating many simultaneous status change messages. To reduce this internal traffic, these messages are put into a single group for each span.

Line Cards

Every time a span on a 16-span line card goes out of service or comes back into service, a message is generated for each channel on the span. This is true whether the change is initiated by the CSP or by the host.

SS7 For this feature, SS7 CICs can be considered equivalent to DS0 channels on line cards. Now, *Channel Status* messages are grouped and handled the same way as the *DS0 Status Change* messages are handled.

Line Card Overload Logic Actions

Generating Alarms on Line Cards

When a 16-span line card reaches the Approaching Busy threshold, the CSP sends an Alarm to the host, and does nothing else. When the line card reaches the Real Busy threshold, it sends an alarm of Line Card Busy. All Idle channels are put into a new, well-defined PPL state in the Out Of Service protocol, and the response to all subsequent call requests is Access Denied.

When the resource usage falls back below the Safe Level threshold, the Real Busy condition has resolved. The card sends a message to clear the alarm and the channels are returned to Idle state. The line card generates a PPL Event indicating Line Card Not Busy and the Matrix Controller is informed about all of the channels that were taken out of service.

SS7 Approaching Busy

When Timer 1 detects that the card has reached the Approaching Busy threshold, it sends an Approaching Busy alarm to the host and sets a global flag. This state is handled in the ISUP and TUP call control modules as described below.

For ANSI/ITU ISUP

Atomic Function 140 tests for local or remote system overload conditions. It also tests for the flag, SS7 PQ Card Approaching Busy. If the flag is set, a PPL Event sends a Release for new calls to the network, and the network withholds additional traffic. Because the host has also been informed of the Approaching Busy condition, it stops setting up new calls.

For TUP

Similar changes were made to Atomic Function 142. If the SS7 Card Approaching Busy flag is set, a PPL event is generated and an Overload Message is sent to the network.

Approaching Busy Clear

When the Approaching Busy condition clears, the SS7 Card Approaching Busy flag is cleared and the SS7 Approaching Busy Clear alarm is sent. Both TUP and ISUP then start accepting new calls.

Real Busy If the overload condition persists and the card reaches the Real Busy threshold, the Matrix Controller in turn sends the alarm, SS7 Real Busy, and all Idle CICs are taken out of service. The CSP sends one *Grouped Channel Status* message for each span, containing all the timeslots on the span that was taken out of service. The CICs taken out of service remain out of service until the Real Busy condition clears. Because the CICs are in the Out Of Service protocol, all Outseizures from the Matrix Controller are returned as "Access Denied."

Real Busy Clear When the overload condition resolves, the SS7 Real Busy Clear alarm is sent to the host. All CICs that were taken out of service are brought back in service, and the CSP sends one *Grouped Channel Status* message for each span to the Matrix Controller.

API Messages You must use the following API messages to implement Line Card Overload Logic. These messages are documented in the *API Reference*.

- *Alarm 0x00B9*
- *System Configuration 0x00AF*
- *DS0 Status Change 0x0042*
- *System Resource Usage Query 0x008E*
- *Generic Report Generic Report 0x0046*

Loopback Diagnostics

Loop Back Configure/ Query message

The *Loop Back Configure/Query* message can help to determine whether the cause of a problem is inside or outside of the CSP. The message works by looping signals from the E-ONE, T-ONE, and DS3 cards back to the network. To enable remote loopback, first take the span you want to use out of service. In the *Loop Back Configure/Query* message, specify a logical span ID and set the action flag to enable loop back.

The E-ONE and T-ONE cards also include card diagnostics for carrier group alarm detection and generation, loss of frame and signal detection, and loop timing support. The host uses the *Span Status Query* message to retrieve the following statistics:

- Bipolar/Code violations
- Excessive zeroes
- Slips
- Cyclic Redundancy Check (CRC) errors
- Out Of Frame (OOF) defects
- Alarm Indication Signal (AIS) defects

5 Layer 4 Call Control

Overview Layer 4 Call Control is a PPL-based programmable software environment that provides programmable services for implementing advanced call processing applications. It is the default central call processing software for the Converged Services Platform (CSP).

Layer 4 Call Control Overview

Introduction

Call Control lets you create unique call models, with custom definitions of call routes and control parameters, and to implement them quickly. You can build into the model PBX-like features, such as one- and two-way connections, call parking, and call hunting, then automatically execute them.

Call Control allows many aspects of a call model to be executed in the CSP, simplifying host applications and improving performance.

Features

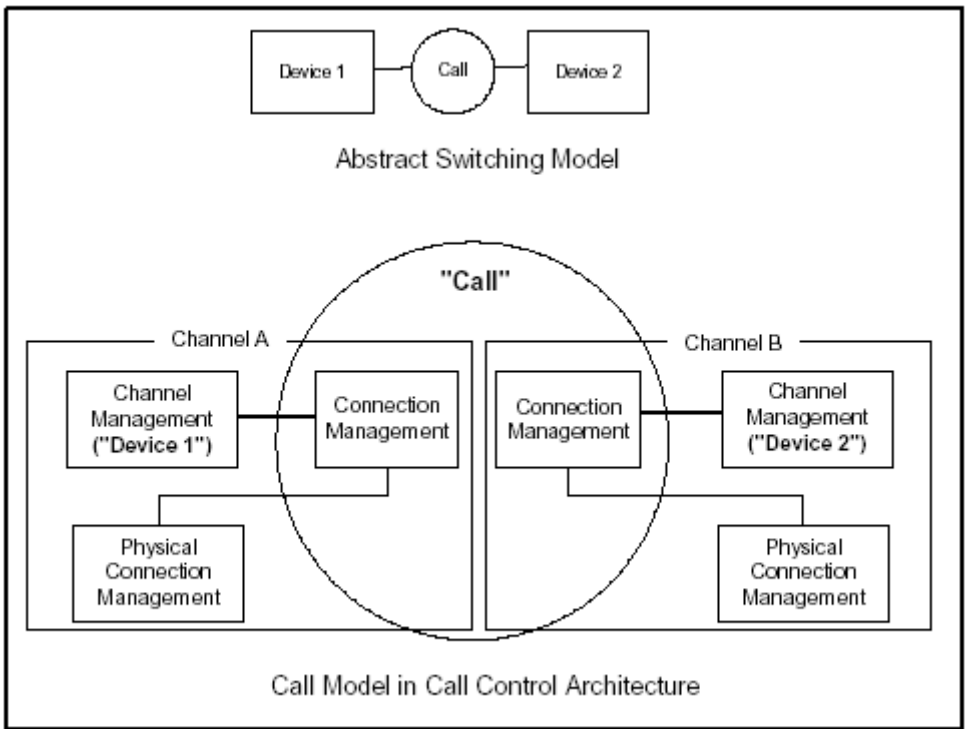
Call Control features include:

- Connection Management
- PPL Technology: Call Control uses Dialogic's Programmable Protocol Language (PPL) technology, simplifying application development, customization, and debugging
- DSP Resource Management: To reduce host interaction, you can preprogram Digital Signal Processors (DSP) resource management (except for conferencing) into the call model
- Call Routing: The Routing features allows the CSP to route calls based on defined criteria or preconfigured settings, without host intervention.

Call Control Software Model

Overview The diagram below shows how call and channel objects are separated in the Call Control software. The model also shows Connection Management (CM) state machines, each associated with a channel.

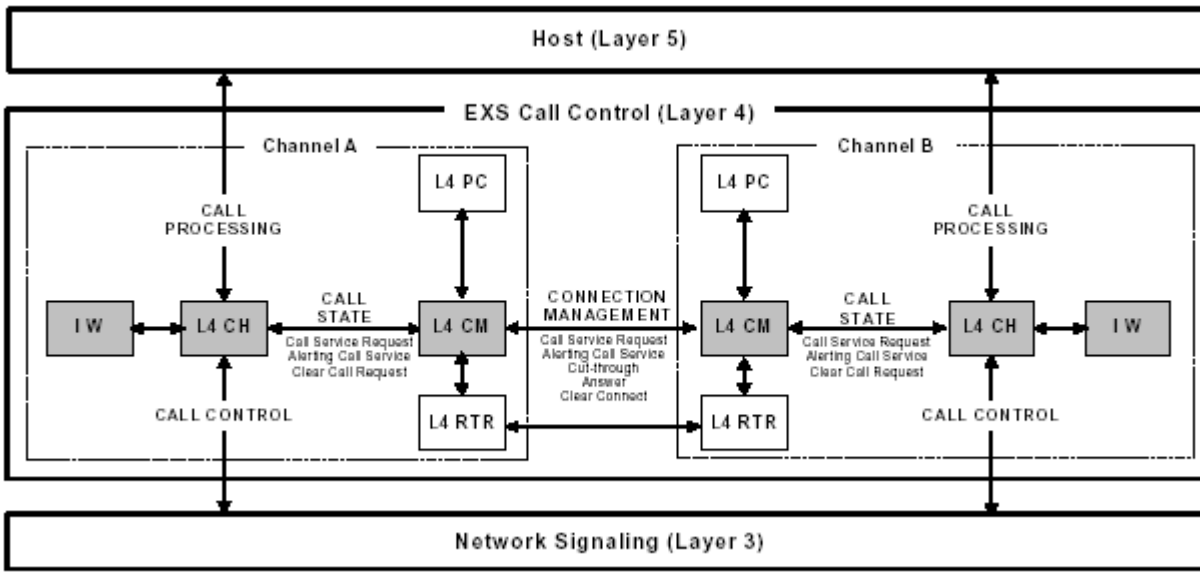
Figure 5-1 Call Control Design Model



Software Components The figure below shows the Call Control architecture, with four PPL-programmable software components:

- CH - Channel Management
- CM - Connection Management
- PC - Physical Connection Management
- RTR - Router
- IW -Interworking

Figure 5-2 Call Control Block Diagram



Channel Management (CH)

The CH component interfaces to the Signaling layer (Layer 3) for call control, and to the Host (Layer 5) for call processing. CH component functions include the following:

- Maintaining the call context for the channel, based on events from the Signaling layer and the host.
- Connection Management with the Call Control CM component
- DSP Service Management (except for conferencing)

All communication between the host and a channel passes through the CH component. A CH state machine is statically instantiated once for each channel.

Connection Management (CM)

The CM component manages calls between two channels, based on state associations and propagated events. Its functions include:

- Filtering incoming events from the remote channel
- Driving the remote channel into a call state appropriate for connection, including reconnections and non-traditional tandem calls
- Answer Supervision and Release

All messages between two channels pass through the CM component. A CM state machine is statically instantiated once for each channel.

Physical Connection Management (PC)

The PC component manages Pulse Code Modulation (PCM) information for a channel's connection. The PC checks and monitors which channels have active voice connections. Before a PCM connection can be made, both the local and remote channels must be in an appropriate state.

The local and remote CM components determine when their channels are ready for a PCM connection. The local CM notifies the local PC when the local channel is ready. The remote CM (using the local CM) notifies the remote PC when the remote channel is ready. When both channels are ready, the connection is made.

A PC state machine is statically instantiated once for each channel.

> **Important!** You can have the host bypass the PCM G.711 companding format conversion using the Companding Conversion Mode TLV (0x0118) in the *Route Control* (0x00E8) or *Connect with Data* (0x0005) messages. Use the 0x0D Channel AIB in the *Route Control* message to specify the channels to connect.
>
> If the host bypasses the conversion process, the B side of the call leg is forced to use the A side's companding format.
>
> Refer to the TLV and messages noted above in the *API Reference*.

Router (RTR)

The RTR component manages internal routing. You can configure multiple route tables in the CSP to allow calls to be routed without host intervention. Routing can be based on numerous methods, including:

- Resource Groups
- Called Party
- Calling Party
- Time of Day
- Span/Channel

An RTR state machine is dynamically instantiated as needed for each channel.

Interworking (IW)

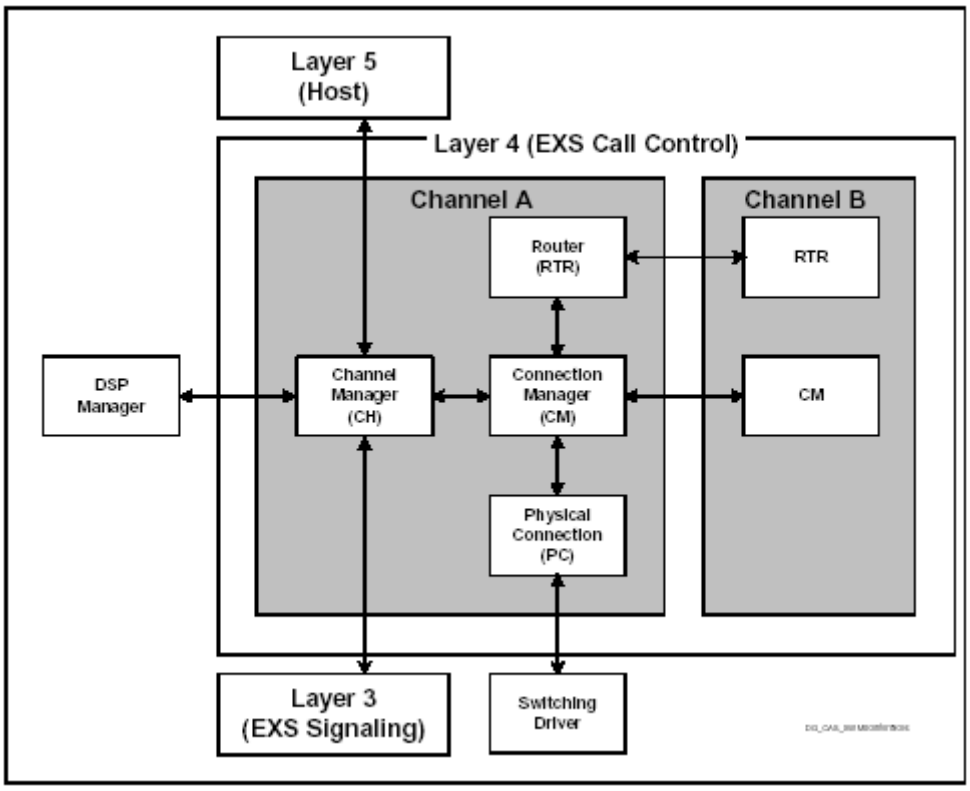
The Interworking component allows the interworking software feature to work within the call control layer. Interworking enables protocol conversion by analyzing and converting Network Signaling (NS) Layer 3 information.

Call Control Software Interfaces

Introduction

The Call Control components communicate through a set of structured interfaces with one another, other call processing components, and the host. The software interfaces shown below provide high level management of network signaling events and call processing events.

Figure 5-3 Software Module Interfaces



Within one instance of Call Control, the CH component may only communicate with the CM component. The CM component may communicate with the CH and PC components. All messages between channels must pass between the CM components of each channel.

The CM component may also communicate to the local portion of the distributed router. This interface is used to request a route and to invoke the Call Service Request event once a route is found. In this case, the Remote interface is not used, and instead the Call Service Request from A to B is handled through the distributed router.

Call Control Internal Interfaces

The interfaces within Call Control include the following:

- Channel Management (CH) to Connection Management (CM)
- Connection Management (CM) to Physical Connection Management (PC)
- Channel Management (CH) to Router (RTR)

CH to CM

Through the CH/CM interface, requests are from CH to CM, while indications are from CM to CH.

The CH component reports the following to the CM:

- Incoming channel events, such as Layer 3 alerting and answer
- Service requests such as one to make a call association with another channel

The CM reports to the CH appropriate incoming events from a remote CM, such as remote alerting, answer, or call request indications.

CM to PC

Through the CM/PC interface, the CM communicates changes in connection requests from both the local and remote CMs. PC uses these events to determine if PCM connections should be made or broken.

CM to RTR

The CM/RTR interface manages access to a distributed call router that gives the internal call routing capabilities. The purpose of this interface is to allow the Call Control layer to query whether routing information is available for an incoming call that it is trying to service.

To satisfy many applications, the CM component provides both blocking and non-blocking versions of the router atomic functions. This arrangement provides application developers greater control in handling call routing errors. Primitives are provided not only to request and acknowledge a router service, but also to communicate a channel's availability as a resource group member. Because the call router is distributed, calls cannot be queued if an entire resource group is busy.

A successful route is acknowledged by the CM component that terminates the route.

Call Control External Interfaces

Components within Call Control interface to the following software modules:

- Local CM (Channel A) to Remote CM (Channel B)
- Local Router (Channel A) to Remote Router (Channel B)
- Signaling Layer (Layer 3)
- Host (Layer 5)
- DSP Manager
- Switching Driver

Local CM to Remote CM

The Call Manager (CM) component of Call Control handles associations between channels (Call setup and teardown). CM manages the associations by sending messages between the PPL components on each side of the association. The messages use atomic functions in the CM state machine that are received as PPL events.

When channels are associated, call processing events are passed between them, such as Alerting, Answer, and requests for voice path connections. Information is also passed about each channel's PCM format, logical address, and physical address.

Local Router to Remote Router

Initiation of an internal router operation requires instantiation of a Router (RTR) component state machine for the channel requesting the route. For calls spanning multiple nodes, the local router initiates a router on the remote node, which continues processing the route on the remote node.

Interface to the Signaling Layer

The main call processing interface within the CSP is between Call Control and Signaling. Signaling resides on the line cards.

A protocol similar to Q.931 abstracts the network signaling layer from Call Control. It is used to accept incoming calls, make outgoing calls, answer, and release channels.

The CH component manages the Signaling interface. All events to and from the Signaling module are sent and received by the CH component. Events from Signaling are sent as PPL events into CH. Events from CH are sent to Signaling with PPL atomic functions in the CH state machine.

Interface to the Host

The Call Control interface to the host (Layer 5) communicates call processing events from the CSP to the host and allows the host to drive the call processing actions of the CSP. The host interface can be used to accept incoming calls, make outgoing calls, answer, and release channels from the host.

You customize the host interface using PPL. You can change the format of established API messages and add new messages using the *PPL Event Request* and *PPL Event Indication* messages. You can transfer new data to and from the host using the PPL Generic API feature. Call Control provides these features, which allow the host to create new call models with the CSP and to implement them using PPL.

By default, all host interface management within Call Control takes place in the CH component by default. The Router (RTR) component can terminate the *Route Control* message to perform certain host-initiated routing operations. The other components have a limited interface to the host, by which they can send and receive PPL events with the *PPL Event Request* and *PPL Event Indication* messages.

The following is a list of API messages used between the host and Call Control:

Host to CH

- *Connect*
- *Connect Wait*
- *Release Channel*
- *Outseize Control*
- *Inseize Control*
- *Connect Tone Pattern*
- *Connect With Pad*
- *Disconnect Tone Pattern*
- *Collect Digit String*
- *DSP Service Cancel*
- *Generate Call Processing Event*
- *Park Channel*
- *Connect With Data*
- *Release Channel With Data*
- *CPC Detection*
- *Connect To Conference*
- *Request For Service Response*

- *Outpulse Digits*
- *Connect One-Way Forced*
- *Recorded Announcement Connect*
- *Recorded Announcement Disconnect*
- *DSP Service Request*
- *PPL Event Request*
- *Route Control*

CH to Host

- Message Acknowledgment (a response to any of the host-initiated message listed above)
- *Request For Service*
- *Request For Service With Data*
- *Channel Released*
- *DS0 Status Change*
- *Call Processing Event*
- *Release Request*
- *Channel Released With Data*
- *Call Progress Analysis Result*
- *PPL Event Indication*

Interface to DSP Services

Call Control has an interface to the DSP Services module. The DSP Services module receives from various software modules requests for DSP services. DSP Services allocates DSP resources from a pool of available resources. The results of the DSP services (for example: collected digits, completion of outpulse) are reported to the requesting module.

The DSP Services module processes host-initiated DSP service requests and also initiates DSP service requests itself as a part of PPL modification. The CH state machine makes the full range of DSP services available to Call Control.

To satisfy many applications, CH provides both blocking and non-blocking versions of DSP feature requests through PPL atomic functions. This arrangement gives application developers greater control in handling errors in DSP allocation, including call failures caused by waiting for DSP service.

If a DSP resource allocation error occurs, the blocking versions automatically purge. The non-blocking versions allow the PPL to process the error.

Interface to the Switching Driver

The CSP contains a hardware/software switching subsystem that makes PCM connections between timeslots. The subsystem handles PCM connection issues such as PCM format conversion and power level (dB) output padding. Call Control simplifies this normally complex operation. The PC component controls when PCM connections are made and broken during the processing of a call.

The interface to the switching driver consists of three primitives that are accessed through atomic functions in the PC state machine. These primitives extract information from databases about the local and remote channel. Because the CSP is distributed, the PC state machine manages only half of a connection. To establish a full duplex connection, both channels must activate their default PC state machines.

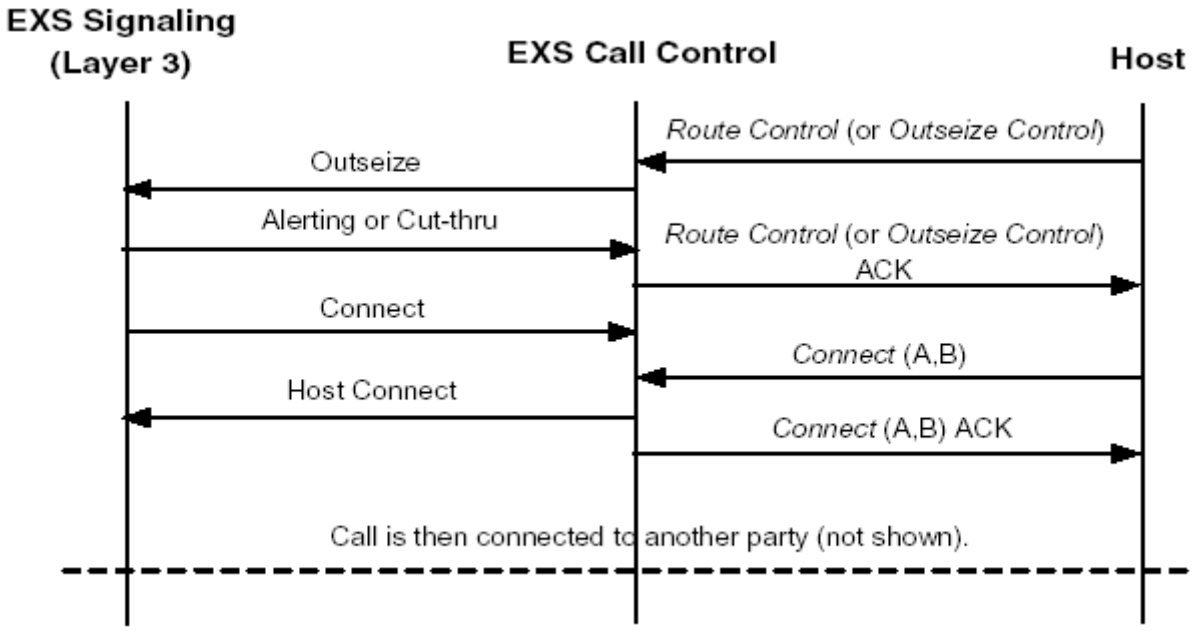
Call Control Call Flows

The following diagrams show basic call flows between Call Control and the Signaling layer. They illustrate basic concepts of the Signaling/ Call Control interface within the default, host-controlled call models.

The network signaling associated with each event is not shown, because it depends upon the specific network signaling protocol.

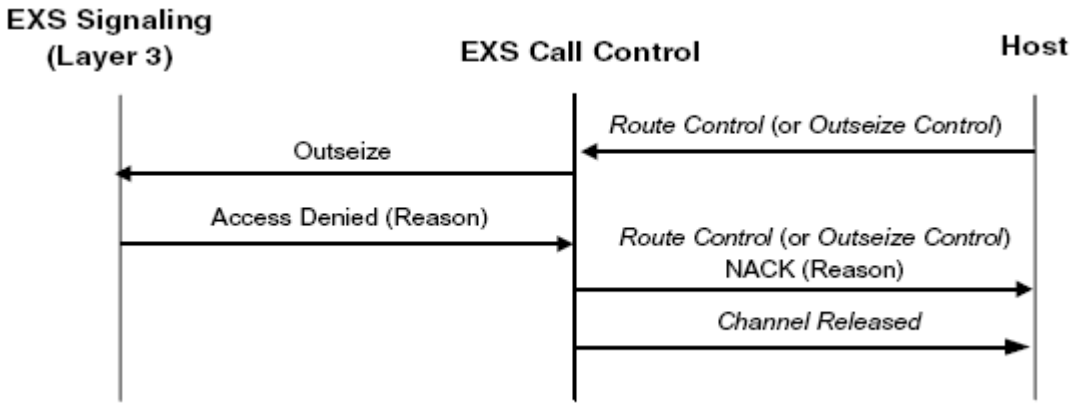
Outgoing call setup with call answering

The following call flow shows an example of outgoing call setup with call answering.



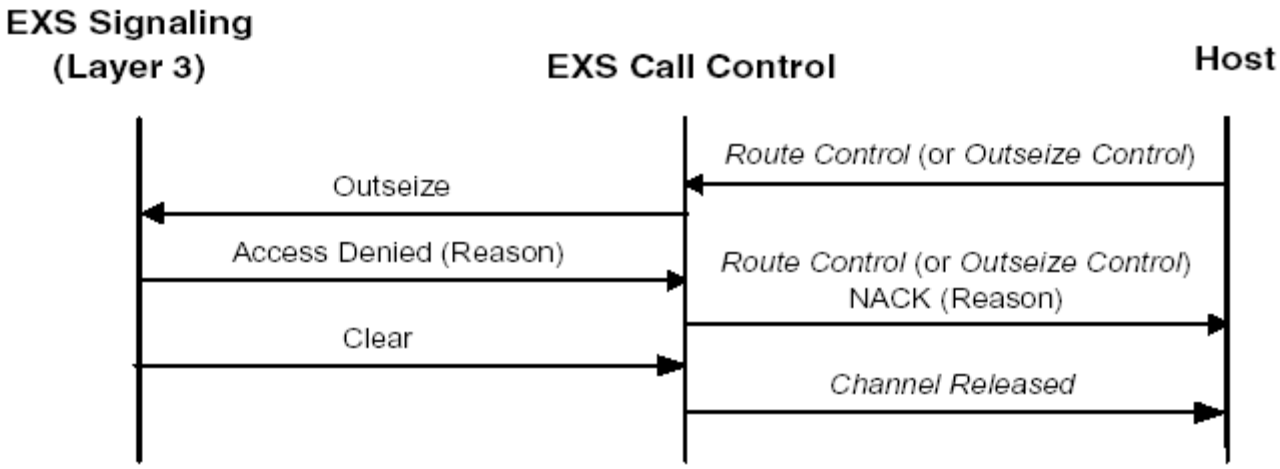
Outgoing call setup, rejected with Access Denied

The following call flow shows an example of outgoing call setup rejected by the Signaling layer using Access Denied.



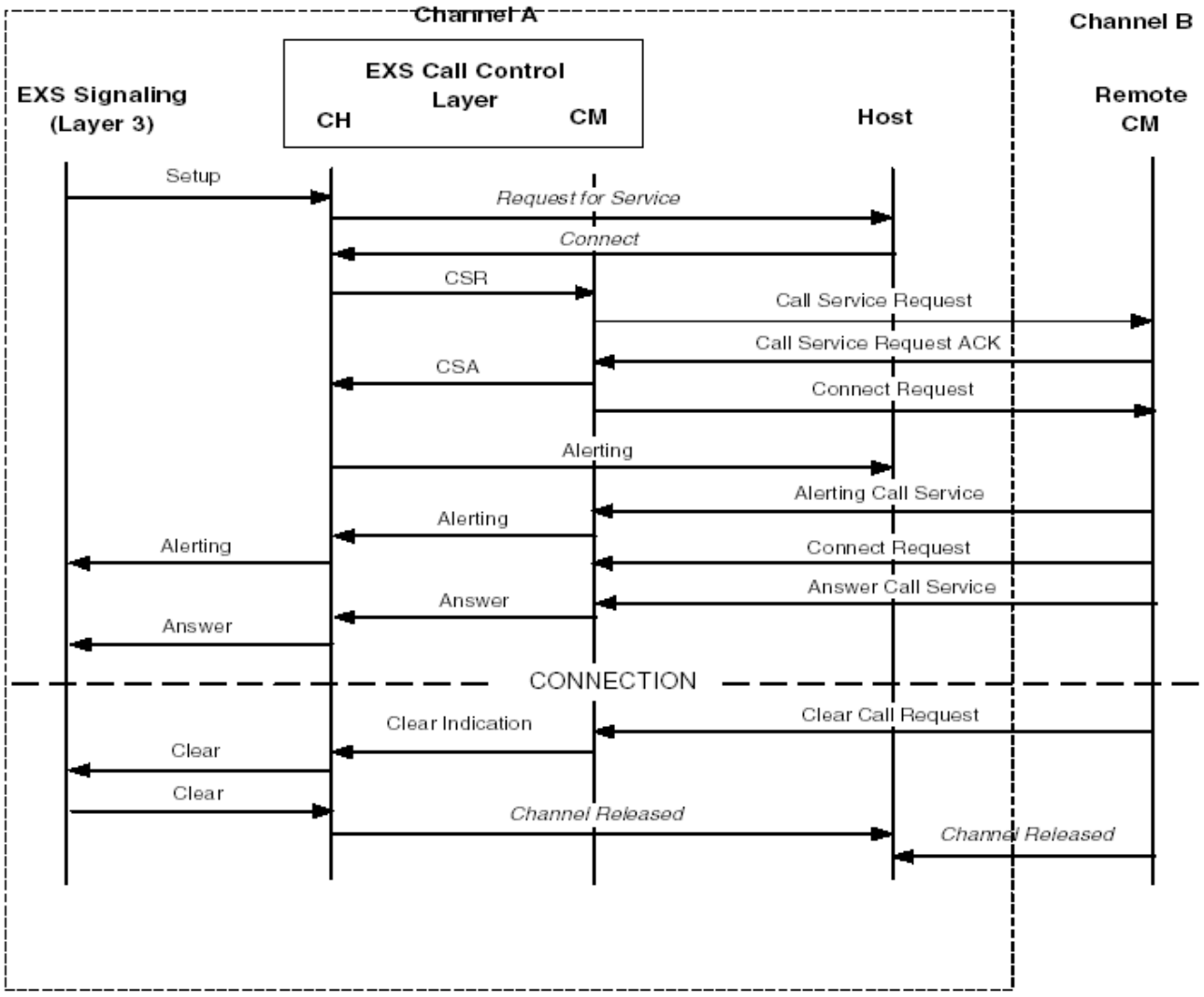
Outgoing call setup, rejected with Access Denied and Clear Pending Flag set

The following call flow shows an example of outgoing call setup rejected by the Signaling layer using Access Denied with the Clear Pending flag set.



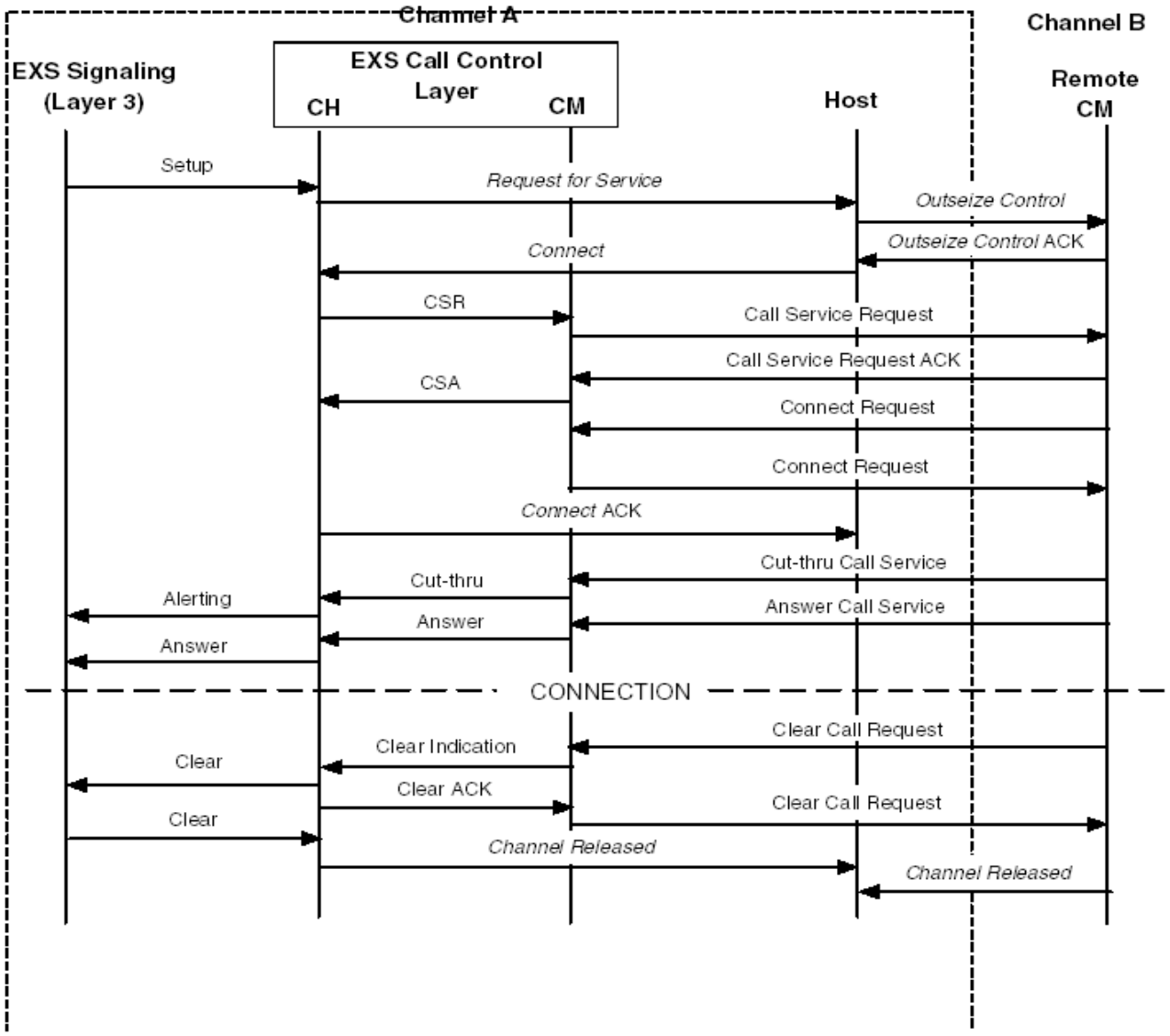
Incoming - Typical call flow, with messaging between CH and CM

The following call flow shows an example of a typical call flow. This call flow includes the messaging between the CH and CM components.



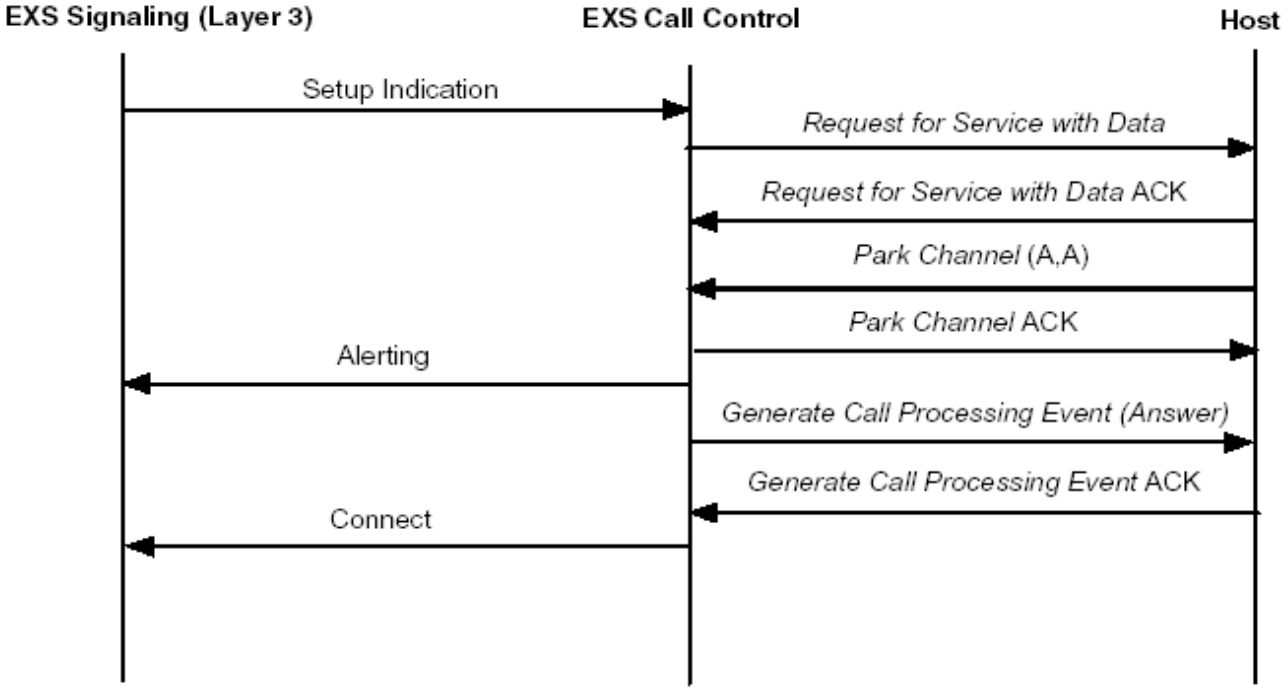
Request for Service, Outseize, and Cut-Through

The following call flow shows an example of a request for service followed by an outseize from the host and a subsequent cut-through.



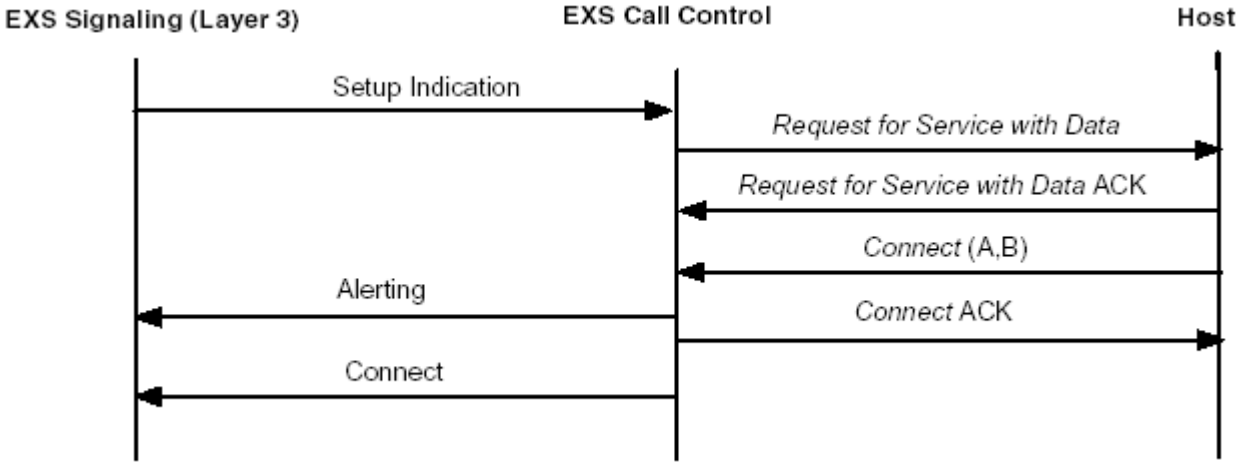
Incoming call with messages alerting and answering

The following call flow shows an example of an incoming call that is alerted and answered through API messages.



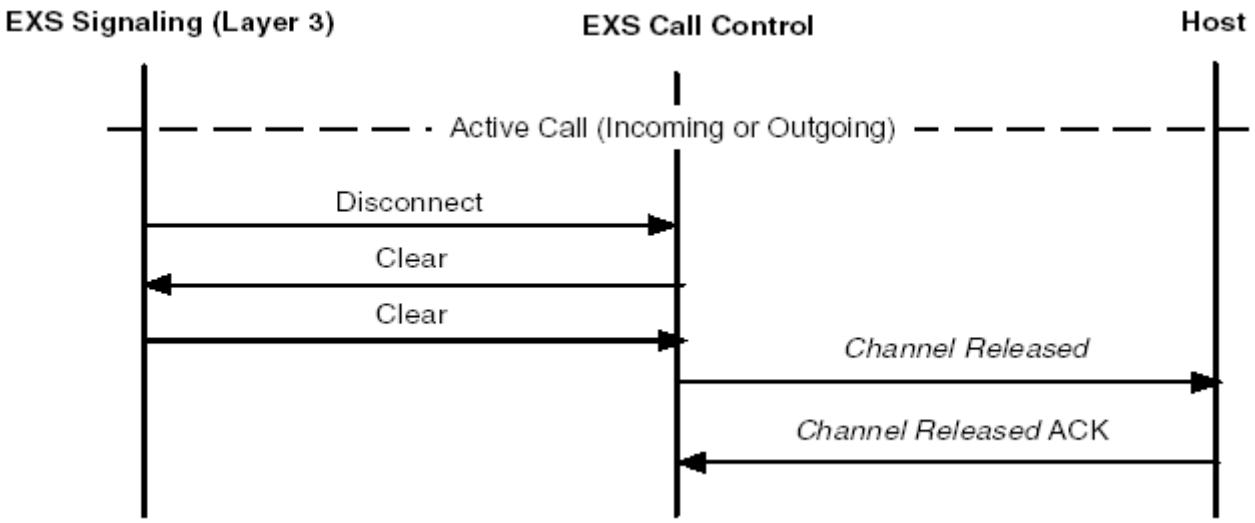
Incoming call with propagated alerting and answer signaling

The following call flow shows an example of an incoming call that is connected to another channel (not shown), where alerting and answer signaling is propagated through the connection.



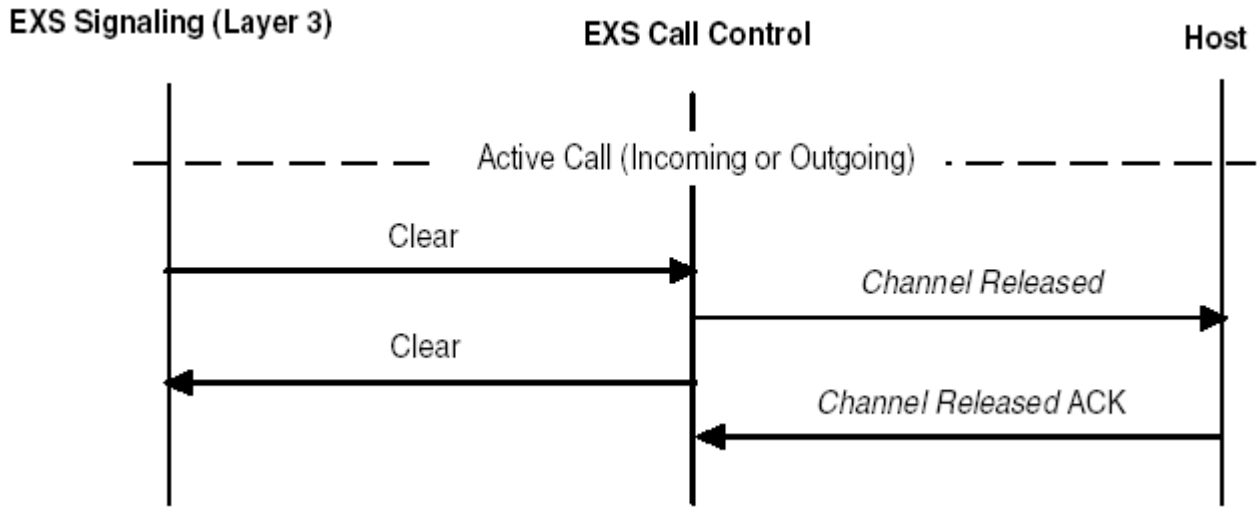
Network initiating release, with Disconnect event

The following call flow shows an example of a network-initiated release using the Disconnect event.



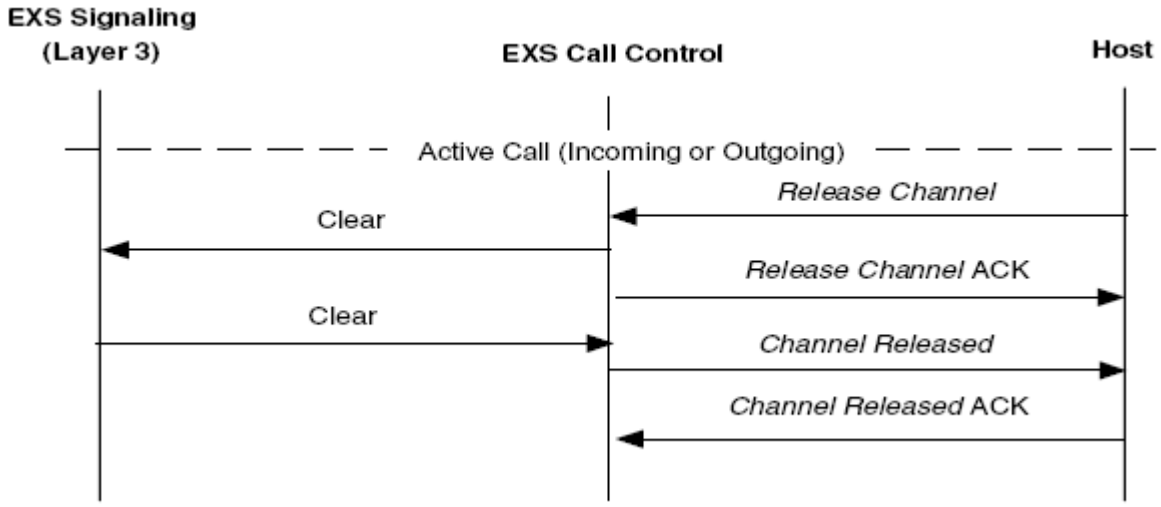
Network initiating release, with Clear event

The following call flow shows an example of a network-initiated release using the Clear event.



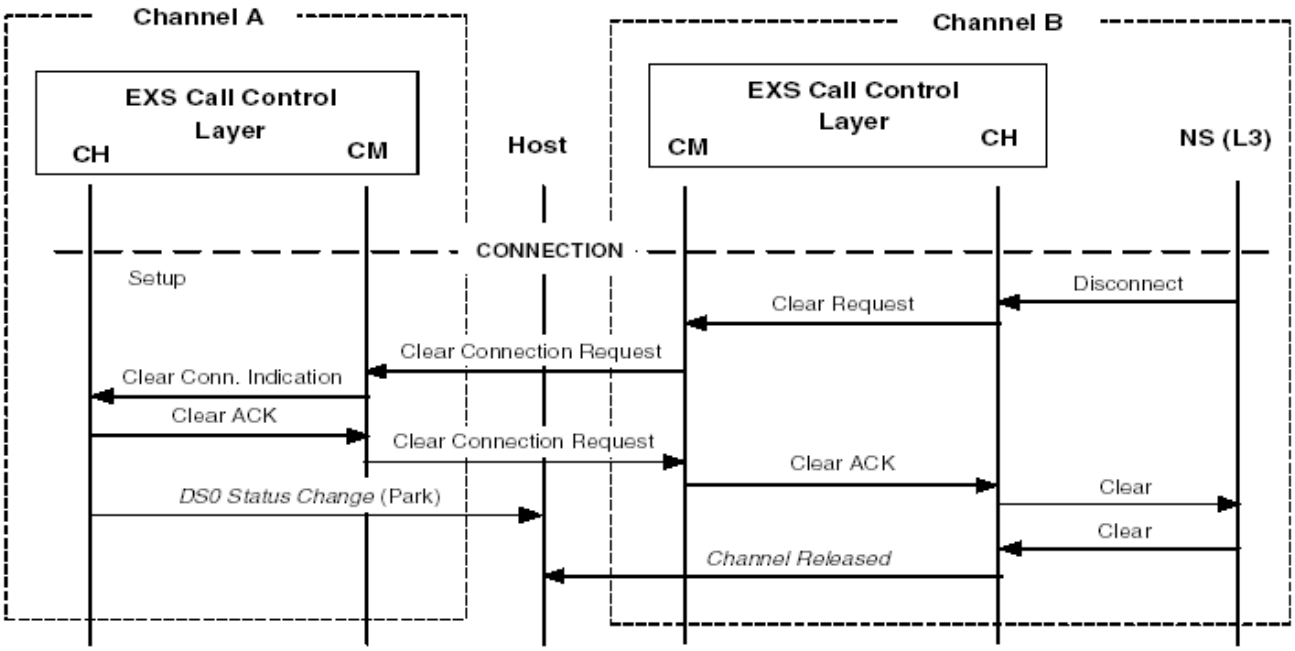
Host initiating release

The following call flow shows an example of a host-initiated release.



Release with Clear Connection Request from CM

The following call flow shows an example of a release with a Clear Connection Request sent by the CM component.



Redundancy

Redundancy for Answered But Non-Connected Channels

Calls that in the answer state but not connected to another channel can be restored when there is a matrix switchover.

The following are possible scenarios leading to a channel that is answered but not connected to another channel.

- When the CSP provides a media service (for example, play a tone/ announcement or collect digits) to a physical channel, the channel is slaving but is not connected to another node, conference, or broadcast. The IP Network Interface Series 2 card is in non-port consuming mode.

To enable this feature, set the CH component (0x61) configuration byte 0x1A to 0x01. See *CH - Channel Management (0x0061) (6-2)*.

The following is a sample *PPL Configure* message to allow span 00 07, channels 00-02 to restore answered calls upon a matrix switchover.

```
00 17 00 D7 00 00 01 01 02 0D 03 00 07 00 0D 03 00 07 02
   00 61 01 01 1A 01
```

Redundancy for One-Way Connections

You can set up a one-way connection and maintain it during a matrix switchover. The feature is for physical channels only. It does not apply to virtual channels.

To enable this feature, park the channels and then send the Connection Type TLV (0x0612) in the *Connect with Data* (0x0005) message. This TLV indicates whether a two-way connection (default) or one-way connection is set up.

The following is an example of the *Connect with Data* message with the Connection Type TLV that sets up a one-way connection between channels 00 and 01 on span 00 00:

```
00 20 00 05 00 10 01 00 02 0D 03 00 00 00 0D 03 00 00 01
   01 01 03 00 1E 00 08 00 01 06 12 00 02 00 01
```

Modifying a Connection Path Without Parking Channels

You can change the connection type of an established connection between two channels without having to park the channels during the transition.

To use this feature, send the Modify Connection TLV (0x0141) in the *Connect with Data* (0x0005) message as described in the scenarios below.

- To transition from a one-way connection to a two-way connection, send a *Connect with Data* message with the AIB containing the same channel values. Add the Modify Connection TLV (0x0141) in the message and set the Connection Type TLV (0x0612) to two-way connect. You could omit the 0x0612 TLV since two-way connection is the default.
- To transition from a two-way connection to a one-way connection, send a *Connect with Data* message with the AIB containing the same channel values. Add the Modify Connection TLV (0x0141) in the message and set the Connection Type TLV (0x0612) to one-way connect. The following is an example of the *Connect with Data* message to change a two-way connection to a one-way connection.

```
00 26 00 05 00 10 01 00 02 0D 03 00 00 00 0D 03 00 00 01
   01 01 03 00 1E 00 0E 00 02 06 12 00 02 00 01 01 41 00
   02 00 00
```

- To transition from a one-way connection to another one-way connection with opposite voice path, send a *Connect with Data* message with the AIB containing the same channel values in reverse order. Add the Modify Connection TLV (0x0141) in the message and set the Connection Type TLV (0x0612) to one-way connect. The following is an example of the Connect with Data message to change a one-way connection to another one-way connection.

```
00 26 00 05 00 10 01 00 02 0d 03 00 00 01 0d 03 00 00 00
   01 01 03 00 1e 00 0e 00 02 06 12 00 02 00 01 01 41 00
   02 00 00
```

The following applies to this feature regardless of the scenario:

- Do not use the *Connect* (0x0000) message to enable this feature.
- The TLV 0x0141 in a message is rejected if both channels are not already connected.

Call Control - Multi-Host Control

Basic Multi-Host Control lets you connect the with up to six hosts. One of the six connections is designated as the default. Each host can register to receive specific API messages from the CSP.

Call Control lets you customize the basic Multi-Host Control, to let you dynamically route API messages to the host.

Call Control components record which host sends a particular message, so that it can respond to that host with other messages for the call. The components can also test and report whether a host port is active or available.

Configuration

To enable Multi-Host Control, you must modify the PPL configuration, as described in the Customization section.

Customization

The CH component maintains information about the host port that can be used whenever a message is to be sent to the host. After each call, the host port is reset to the default: 0xFF, No Host Port Specified. So the host port must be set for every call. If the port is not set, the basic Multi-Host Control feature is used. Atomic functions 76–79 are used for host port information.

The valid values for the host ports are as follows:

0xFF: No Host Port Specified (default)

0x00: Port 3142

0x01: Port 3143

0x02: Port 3144

0x03: Port 3145

0x04: Port 3146

0x05: Port 3147

Atomic Functions

Using the four Atomic Functions below, you can modify the CH DSD to support Multi-Host.

- **AF 76**: Tests the real-time availability of the specified host port
- **AF 77**: Loads the current host port ID into the GPR. The PPL can then test the value.
- **AF 78**: Sets the current host port ID based on the value of the specified GPR. All messages initiated by Call Control for this channel are sent to this host port.
- **AF 79**: Stores the host port ID from the incoming host message (the host that sent it) into the GPR.

Call Control Connection Management

Overview

Connection Management supports two-party connections between channels involved in the same connection, and channels involved in different connections.

Both channels must be In Service, and Channel A must not be in the Idle state.

The Connection Management feature also allows two-party connections between channels that are involved in different connections. For example, if Channels A and B are connected, and Channels C and D are connected, a *Connect* (A,C) message results in A and C being connected while Channels B and D are released.

This feature also applies to conferences. If Channels A, B, and C are connected to a conference, the host sends a *Connect* (A, B) message, which results in A and B being connected (without using conference resources) and C remains connected to the conference.

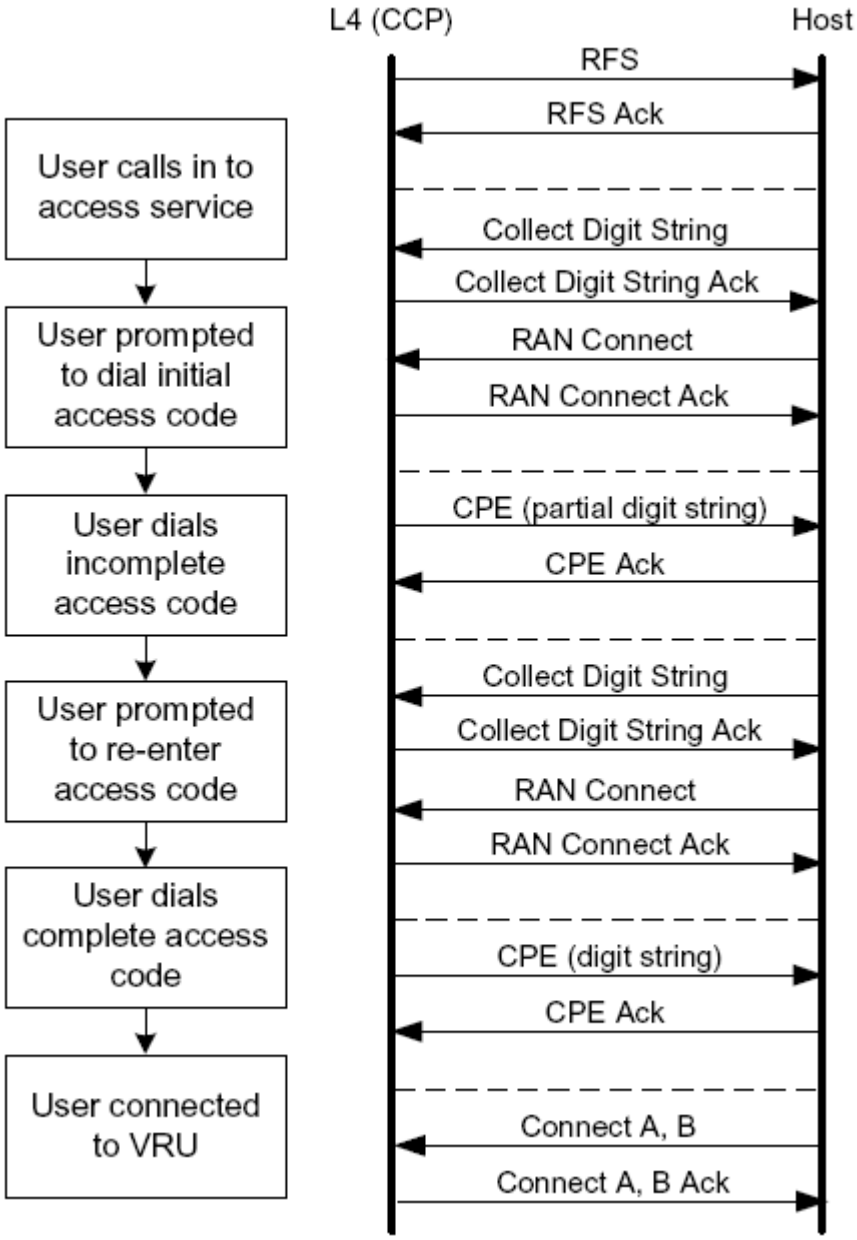
This feature also works if A and B are connected to different conferences. If Channel A is connected to Conference 1, Channel B is connected to Conference 2, and the host sends a *Connect* (A, B) message, Channels A and B are connected while both conferences continue with existing parties.

You can reduce host interaction of call control by off-loading some call processing to the CSP.

Standard Call Model

The figure below represents a call model that requires significant host/CSP interaction (eight API messages) to establish a connection.

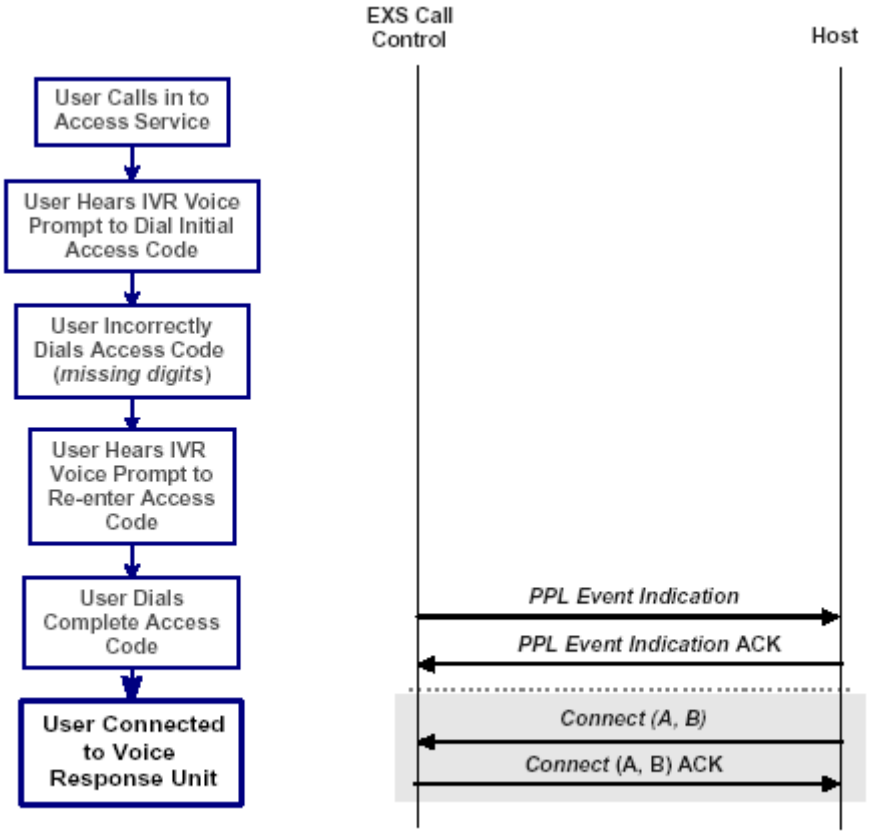
Figure 5-4 Standard Call Model



Call Control Model

The figure below represents the same call scenario, except that Call Control has been programmed to handle the initial stages of call setup, with no host intervention required. Instead of the eight API messages required in the preceding diagram, only two API messages are required when you use Call Control.

Figure 5-5 Call Control Call Model



Connection Management Messages

All connection features use the connection management messages shown below.

Connect

Establish a two-way connection between the two specified channels. If there is no active call up for the B channel, a seizure will be generated on that channel and the voice path will be connected once seize ACKs have been met.

Connect With Pad

Establish a two-way connection between the two channels, while maintaining the specified dB gain/loss adjustment for the duration of the connection.

Connect With Data

- Connect With Data (A, B) (2-Way)

Establish a two-way connection between two Common Channel Signaling (CCS) channels. If the terminating channel is not idle after seizure, the alerting and connect indications propagate to the incoming side, and the message's data specification is sent to the incoming CCS channel.

The following features are also supported:

- Canceling digit collection
- Playing call progress tones
- Voice path connection
- Dynamically configuring release modes

Park Channel

Park one or both channels involved in a connection. To park only one, set A and B to the same span/channel. When a channel is parked, it is not associated with any other channel and silence is transmitted. The channel may be used in a subsequent connection or it may be released later.

Release Channel

Release a single channel or a connected pair of channels. A *Channel Released* message is returned by each channel when the release is completed. If only one of the connected channels is released, the other channel is parked automatically.

Connect One-Way Forced

Establish a one-way connection between two channels. Channel A is the source channel (talk only) and channel B is the destination channel (listen only).

Channel A may be in any state except Idle, while Channel B can be in any state except Out of Service, and it cannot be in the process of clearing. This connection is transparent to Channel A, and it is torn down only if Channel B is released by the host or by the distant end.

Connect One-Way to Conference

Establishes a one-way (listen only) connection between the channel specified and the previously created conference specified. The channel can be in any state except Out of Service, and it cannot be in the process of clearing.

Connect to Conference

Adds the specified channel to the specified conference, which already exist. The channel to be added to a conference can be in any state except Out of Service, and it cannot be in the process of clearing.

Sub-Rate Connection Management

This message is used to establish and teardown subrate connections. Both channels must be out of service before the connection is established.

Cross Connections

Using the *Cross Connect Channel* or the *Cross Connect Span* message, the host connects two physical channels or spans. Both spans must already have a Logical Span ID. Dialogic recommends taking both ends of the connection out of service first. When either of these messages are used, the connection is made regardless of any line or trunk configurations or formats currently set.

Cross Connect Channel

Connect the data paths of the two channels. Both Channel A and Channel B should be out of service, except when cross connecting T1 and E1 channels. No signaling information can be used to detect either channel releasing. The host must use the *Cross Disconnect Channel* message to tear down the connection.

Cross connecting T1 and E1 channels causes automatic PCM conversion (A-law/µ-law). By default, channels on T1 spans are µ-law, and channels on E1 spans are A-law. To disable conversion, configure both channels with the same format by using the *PCM Encoding Format Configure* message. You must disable conversion when you are connecting through an ISDN D channel.

Cross Connect Span

Connect all data paths on Logical Span ID A to data paths on Logical Span ID B. You cannot cross connect a T1 span with an E1 span without PCM conversion. Multi-frame synchronization is not maintained between two cross-connected spans.

Cross Disconnect Channel

Disconnect the data paths of the specified channels.

Cross Disconnect Span

Use the *Cross Connect Span* message to disconnect all data paths of two spans that were previously connected.

Calling Party Control of Disconnect

Calling Party Control (CPC) is how the calling end notifies the called end that the established connection is no longer needed. The CSP provides this capability with the *CPC Detection* message. The CSP scans for dialtone from the distant end to indicate that it has been released.

CPC is used only on trunk interfaces that provide no release information through signaling. For example, the CSP commonly uses CPC Detection on T1 FXO-LS or analog FXO-LS lines that provide no loop current break to signal a release.

Modifying API Messages

Overview

You can modify the *PPL Event Request* message to send customized data to the CSP and to receive *PPL Event Indication* messages with customized data from the CSP. Dialogic recommends using these PPL Event messages, because you can increase performance by reducing messaging between the host and the CSP. You also save time that you would normally use to develop features for host applications.

You can modify the PPL Event messages using predefined data sets, named Tag/Length/Value (TLV) blocks. For example, one TLV calculates the billing duration with a specified granularity, so the host applications do not need to track timestamps and other billing information.

See the *API Reference* for information on TLVs and on the PPL Variable Data ICB that transports them.

Atomic Functions

The following atomic functions enhance the ability of the host to send and receive the TLV ICB.

- **AF 109**: Tests for the presence of a specified buffer.
- **AF 110**: Moves/copies data from an incoming to an outgoing buffer.
- **AF 111**: Clears the given outgoing buffer:
- **AF 356**: Performs tasks jobs depending upon the arguments passed.
- **AF 357**: Moves the TLV from the working buffer (with the Switch Tag corresponding to that Host Tag in Arg2) to the GPR indicated by the index in Arg1.
- **AF 358**: Makes a new TLV from the values of the GPR indicated by Arg1, and puts the TLV into the working buffer (with the Switch Tag corresponding to the Host Tag indicated by Arg2)
- **AF 359**: Puts the DSP data list TLV into the working buffer.

WARNING

The system time must be set correctly in the CSP to use the timestamp features.

The timestamp features do not work correctly during a Matrix Controller switchover. You may lose the start timestamp along with other TLVs.

Please note the maximum duration allowed is 1193.05 (approximately 50 days) hrs. If the duration requested is for more than 1193.05 hours, the duration indicator will reset and start from zero.

Use only the Tags defined in this document. Using an unlisted or reserved Tag may result in undesired operations.

Customization

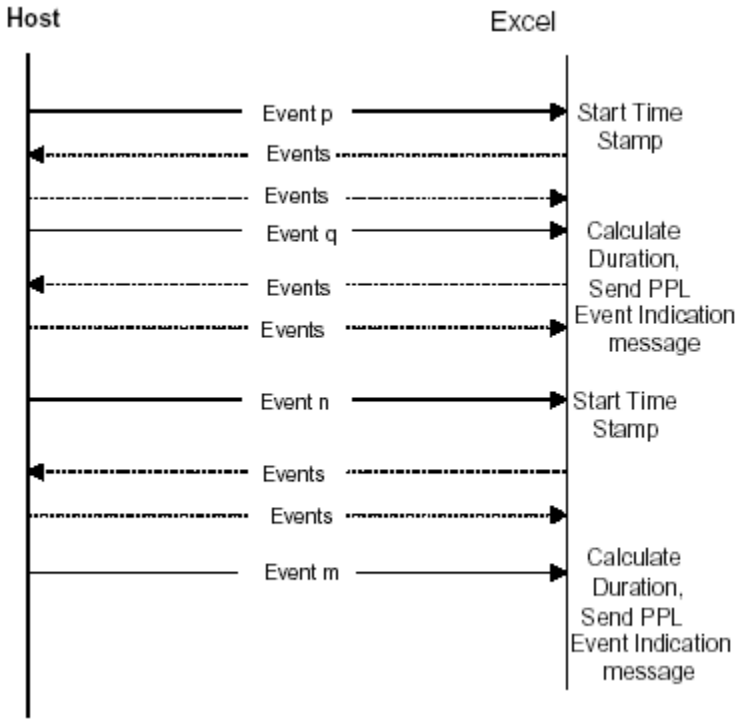
This section describes possible customizations that you can make, using the EXS API.

Start Timestamp TLV

You can use the Start Timestamp TLV to do the following:

- Calculate the call duration.
- Mark the beginning of a call using any event that the CH component can access
- To record an event. For example, you could use the Start Timestamp TLV to record Event p and report to the host, and to time the stages of any protocol ladder (without overlap).

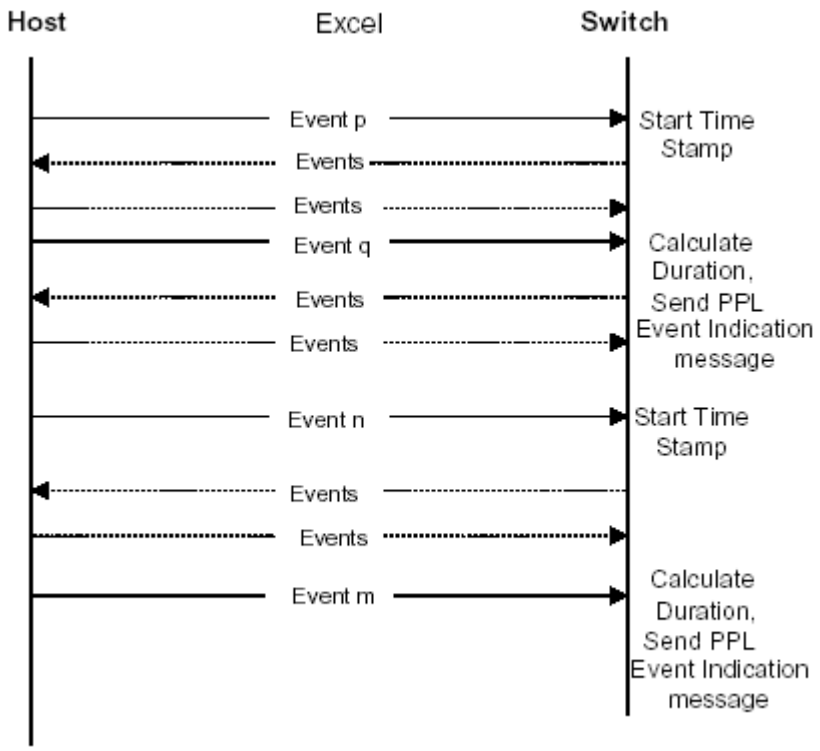
Figure 5-6 Using the Start Timestamp TLV



Current Timestamp TLV

You can use the Current Timestamp TLV with any combination of user-defined TLVs to report the time of that an event occurs (one at a time--for example, one timestamp in one *PPL Event Indication* message). The figure below shows a call flow of this scenario.

Figure 5-7 Current Timestamp with User-Defined TLV



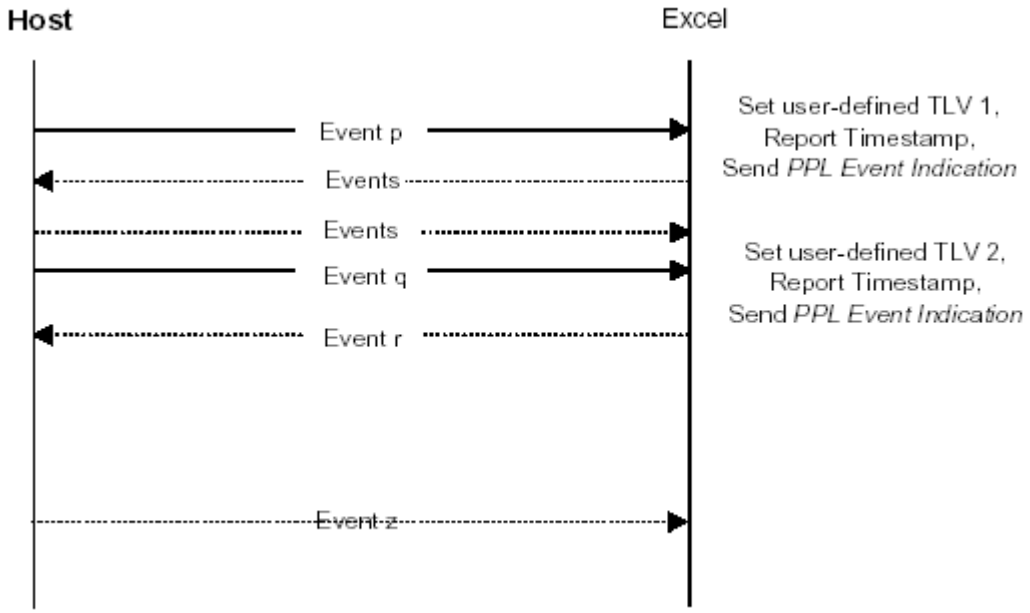
To concatenate a series of timestamps into a single report for the host, you can use custom TLVs with the Current Timestamp TLV in sequence.

In the next figure, the host receives a single *PPL Event Indication* message that contains the user-defined TLVs and the timestamps, in the same order in which they were created. Based on the user-defined TLVs, the host can match the timestamps to events.

For example, the timestamp after user-defined TLV 1 represents the time when Event P occurred and the timestamp after User-Defined TLV 2 represents the time when Event R is sent to the host.

Important! Do not concatenate so many timestamps into one message that the message exceeds its maximum size.

Figure 5-8 Multiple Timestamps with User-defined TLVs



User-Defined TLV This TLV lets you send and receive any data to and from the CSP. The following are a few of the possible uses for the User-defined TLV:

- Label a particular call, such as: important/secret/private/public/ from pay phone/from emergency and process the call based upon the label
- Temporarily store values for later use
- Inform the PPL state machine of an internal event. Based on the data in the user-defined TLVs, the state machine can take different branches.
- Use in combination with other TLVs which can have multiple instances in one message and identify them at the host with different meanings. For example, as described in the Current Timestamp TLV, you can use this to determine the events that triggered the multiple timestamps.
- Host application related sequencing.
- Host application related indexing.

Generic Counter TLV This TLV provides a counter that increments or decrements and returns values to the host.

Using Multiple TLVs

You can dramatically reduce the message volume from the CSP to the host by using multiple TLVs of the same type in a single message. The following are some guidelines for using multiple TLVs of the same type:

- Avoid using the same user-defined TLV tag more than once in a single message. It may be difficult for the host to determine the order of messages from the CSP.
- Use a unique user-defined TLV to indicate the following: in the Current Timestamp TLV example user-defined TLV 1 represents the time of Event p, and User-defined TLV 2 represents the time of sending Event r.
- Try to use only one start timestamp TLV in a message. A start timestamp remains unchanged until you overwrite it with the Store Start Timestamp atomic function (AF 356). The duration is always calculated with respect to the last start timestamp stored.
- Avoid using the Generic Counter and Generic Tag TLVs more than once. Instead, use the user-defined TLV with different tag values to achieve the same result.
- To time a sequence of events, use the Duration TLV repeatedly instead of the Current Timestamp TLV.

DSP Resource Management

You can modify the default CH component state machine to initiate DSP services without host intervention. You cannot manage conferencing DSP resources with Call Control.

Function	Atomic Function	API Message
Collect Digits	260-262	Collect Digit String
Attach DSP Receiver	267-269	DSP Service Request
Energy Detection	270-272	DSP Service Request
Call Progress Analysis	273-275	DSP Service Request
CPC Detection	280-283	CPC Detection
Cancel DSP Service	285-287	DSP Service Cancel
Connect Tone Pattern	290-292, 332-333	Connect Tone Pattern
Recorded Announcements	293-295, 330-331, 353	Recorded Announcement Connect Recorded Announcement Disconnect
Outpulse Digits	296-327	Outpulse Digits
Test	310-327	Not Supported

Call Control - Tones Generation and Reception

Address Signaling Tones

Configure the CSP to generate outbound DTMF, MFR1, or MFR2 address signaling. To initiate outpulsing, use the "Outpulse Stage N Address Digits" ICB in the *Route Control* or *Outseize Control* message. The host can supply outpulsed digits and inform the CSP of the signaling tone type and where to retrieve the digits.

Call Progress Tone Patterns

The *Connect Tone Pattern* message generates call progress tone patterns. A pattern is generated for an active call only, incoming or outgoing. The *Connect Tone Pattern* message then connects the channel to the pattern specified by the Pattern ID.

Tone Reception

The CSP provides tone reception for collecting digits and for analyzing call progress. The CSP software maintains separate tone receiver pools for DTMF, MFR1, MFR2 (Backward and Forward) and call progress reception, dynamically configured and verified upon start-up. These pools are shared by all channels.

Address Signaling Tones

This section describes the different types of Address Signaling tones.

Digit Collection During Call Setup:

During call setup, the CSP can collect DTMF, MFR1, and MFR2 (Forward and Backward) digits.

> **Important!** Any card with an MFR2 receiver must also have an MFR2 forward and backward transmitter configured. If a card has only receivers, an incoming call triggers a DSP Function Not Configured alarm.

You can program the CSP to automatically collect inpulsing address data, which identifies the Dialed Number Identification Service (DNIS) and/or the Automatic Number Identification (ANI). Applications can use the address data to authenticate subscribers and to identify services.

During call setup, the host uses a combination of the *DSP SIMM Configure, Inpulsing Parameter Configure,* and *Inseize Control* messages to program digit collection at the Signaling Layer 3. The message features are described below. Please refer to the *API Reference* for details:

- *DSP SIMM Configure* – Configures the function of a DSP. Tone reception functions include DTMF, MFR1, MFR2, and E1 Dial Pulse reception. (E1 Dial Pulse not available on the DSP Series 2 card).
- *Inpulsing Parameters Configure* – Configures parameters for collecting inpulse data at the Signaling Layer 3. Inpulsing parameters define the address signaling type, the number of digit strings, and the collection method used during call setup.

  The CSP supports four different inpulsing stages, each with one or two digit strings. When the host instructs the CSP to collect address signaling information (typically in-band dual frequency tones) it also specifies a preprogrammed inpulsing stage that describes how to perform the digit collection. Options for configuring the inpulsing stage include the following:

  - Address signaling type (DTMF, MFR1, MFR2)
  - Number of strings (1 or 2)
  - String collection method (fixed number of digits, KP/ST framed, compelled)
- *Inseize Control* – Sends inseize instructions to the CSP to control incoming call setup during real-time call processing at the Signaling Layer 3. The Receive Stage N Address Data instruction allocates and attaches a DSP to a channel to collect the incoming digit stream.

  A maximum of 100 digits can be collected within a single inpulsing stage. Use the *Inseize Instruction List Configure* message to preprogram instructions on a channel at the Signaling Layer.

  The *Report Incoming Call With Address Digits* instruction sends a *Request for Service With Data* message to the host, with the single-stage or multiple-stage digit streams collected.

Digit Collection After Call Setup

After a call is set up, the CSP performs interactive DTMF digit collection. The host collects additional information, either one digit at a time with the *DSP Service Request message*, or as a string of digits with the *Collect Digit String* message. These messages are processed by the CH component.

- *DSP Service Request* - allocates and attaches a digit receiver to a channel. The digit receiver remains attached and uses the *Call Processing Event* message to report digits as they are decoded. The receiver remains attached to the channel until the channel is released or until the host uses the *DSP Service Cancel* message to cancel digit collection.
- *Collect Digit String* - allocates and attaches a digit receiver to a channel, but unlike the *DSP Service Request* message, it directs the CSP to collect a string of digits until a termination condition is reached. The termination condition could be either:
  - the detection of a particular digit from a termination digit set
  - a fixed number of digits collected
  - a timeout

  When a termination condition occurs, the CSP reports all of the digits collected in the *Call Processing Event* message and the receiver is returned to the system receiver pool. When the CSP detects the first valid digit, it automatically cancels any prompting tone or recorded announcement being played out to that channel.

Call Control - Release

Release Modes

Release modes are managed by the CM component of Call Control. When a connection is terminated, the local and distant end release modes are used to determine what action to take on the released channels.

The terms *local* and *distant* are relative to the channel being configured. To specify whether a channel is released or parked when the channel in a connection releases, you configure the channel's local-end release mode. To specify whether the other channel in the connection is released or parked when that channel releases, you configure the channel's distant-end release mode.

When a channel terminates a connection, the CSP refers first to the local-end release mode of the other end of the connection, and then to the distant-end release mode of the channel that initiated the release. If either is set to park, the channel parks. Otherwise, it is released. The host is informed of the state of a channel with either a *Channel Released* or a *DS0 Status Change* message.

Customization

To configure release modes to either Park or Release, use the *PPL Configure* message with the PPL Config Bytes of the CH PPL component.

To find the default settings of the release modes, you can use the *Change Parameter Query, CIC Query* and *ISDN B Channel Query* messages. To change the defaults for the release mode, use the *Local End Release Mode Configure* and *Distant End Release Mode Configure* messages.

> **Important!** You can also change release modes using the *PPL Configure* message. But if you do, those changes override the settings made with *Local End Release Mode Configure* and *Distant End Release Mode Configure* messages, and you must then use the *PPL Data Query* message to find the current settings.

Distant End Release Mode

When you configure a channel, use the distant end release mode so that the distant end (the other channel in the connection) is not released when the connection is terminated. For example, you would not want to release an inbound channel that is connected to a Voice Recognition Unit's (VRU) *Please Wait* message before it is queued up for an available agent.

When the announcement is finished, the VRU channel is released for other calls and the inbound channel is parked. The distant-end release mode of the VRU channel is set to park so that when the VRU releases at the conclusion of the message, all inbound channels connected to the VRU park.

Local End Release Mode

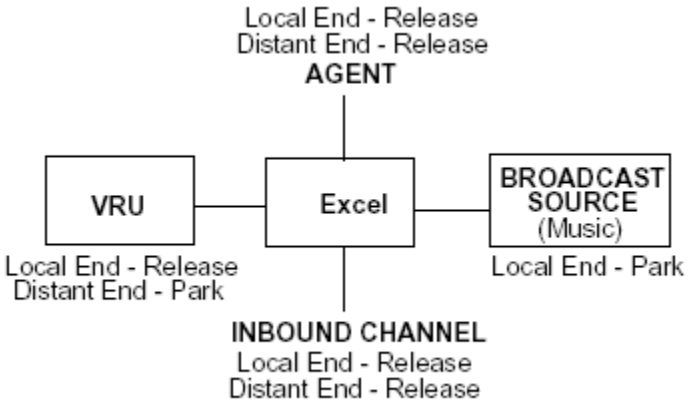
When you are configuring a channel, use local end release mode so that the channel is not released when a connection is terminated. Consider the previous example for distant end release mode. To configure a source to broadcast music to the channels waiting for an agent, the local-end release mode of the broadcast channel is set to park.

When there are no channels connected to the broadcast channel, it parks until another channel is connected. Otherwise, it releases and would have to be outseized after each connection is broken down.

The figure below illustrates the preceding examples. Both release modes for the inbound channel are set to release. Therefore, when the connection is torn down, it is parked only if the distant-end release mode of the other channel is set to park (as is the case for the VRU and the Broadcast Source).

If the distant-end release mode of the other channel is set to release (as is the case with the agent channel), the inbound channel is released when the connection is torn down

Figure 5-9 Release Mode Diagram



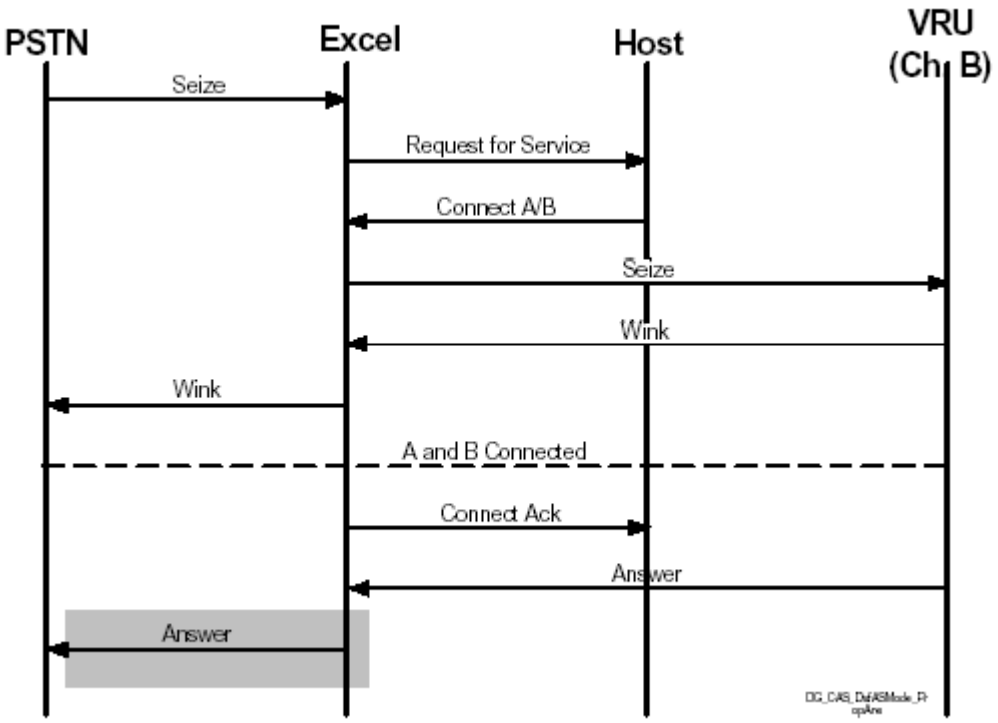
Answer Supervision

The CH and CM components of Call Control manage answer supervision.

Default Implementation

By default, the CSP propagates Answer to the distant end. No Answer notification is sent to the host.

Figure 5-10 Default Answer Supervision Mode - Propagate Answer (0x00)



Customization

You can change the Answer Supervision Mode for a channel by modifying PPL Config Byte 2 of the CH component. The mode options are listed below, along with modifications of the default call flow. Each modified call flow begins at the point where A and B are connected. The variation for each mode is shown in gray.

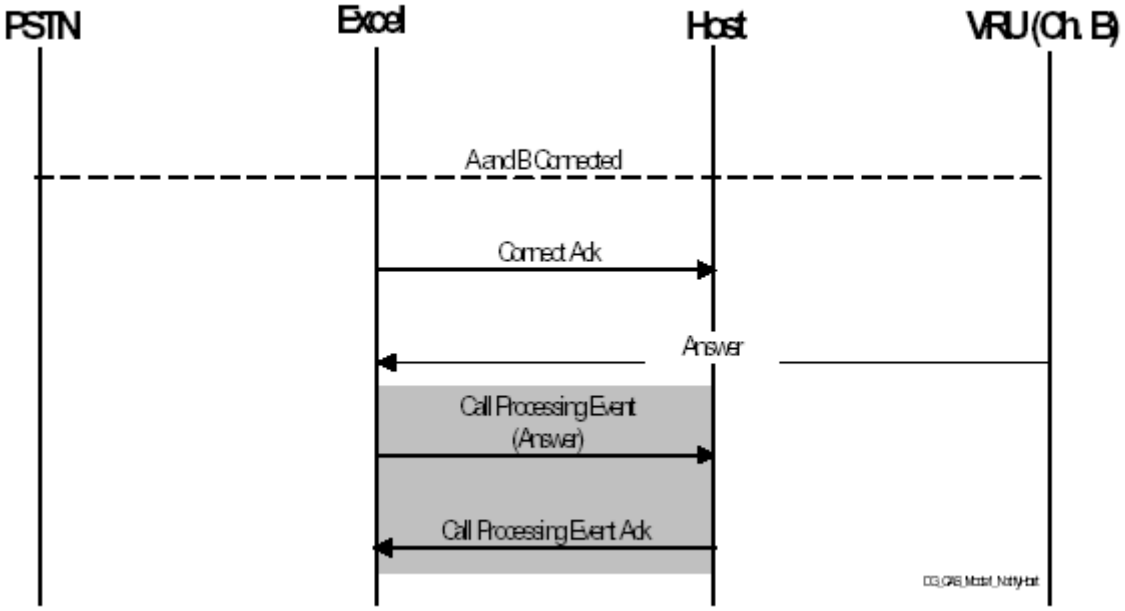
Important! You can also use the *Answer Supervision Mode Configure* message to configure Answer Supervision.

Notify Host of Answer

When Answer signaling is detected from the outseized channel, Call Control informs the host with a *Call Processing Event* message and does not propagate Answer to the inseized port.

Important! Required Modification: You must change PPL Config Byte 2 of the CH component to 0x01.

Figure 5-11 Mode 1 - Notify Host of Answer

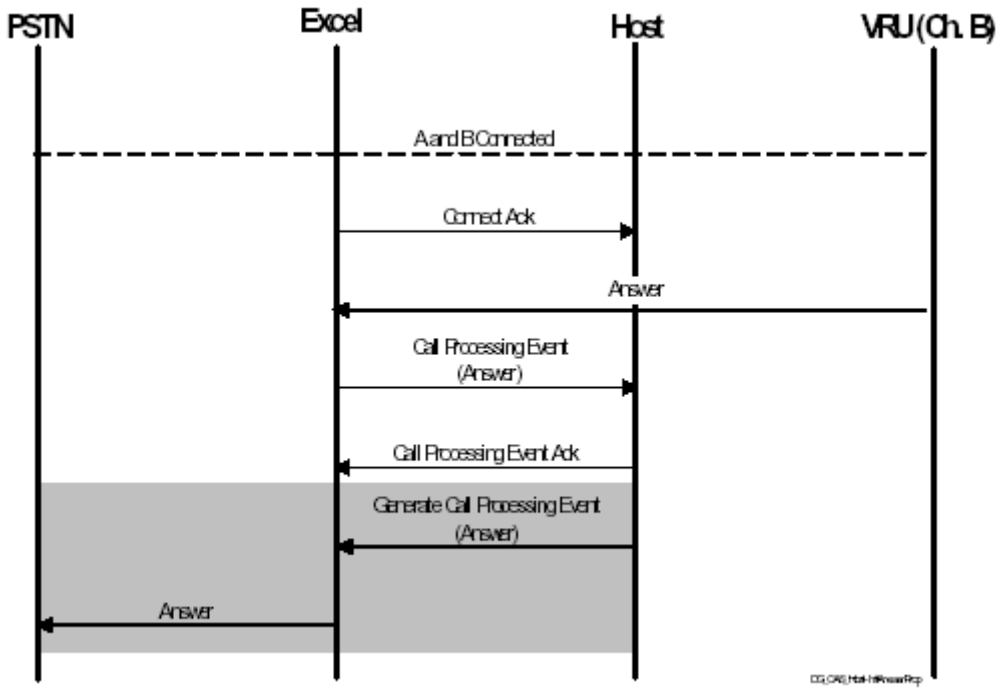


Host-Initiated Answer Propagation

At any time, the host can use the *Generate Call Processing Event* message to independently generate Answer to the inseized port. If Answer is not generated, the PSTN could unexpectedly terminate the connection.

Important! Required Modification: You must send a *Generate Call Processing Event* message of Answer to the CSP.

Figure 5-12 Host-Initiated Answer Propagation

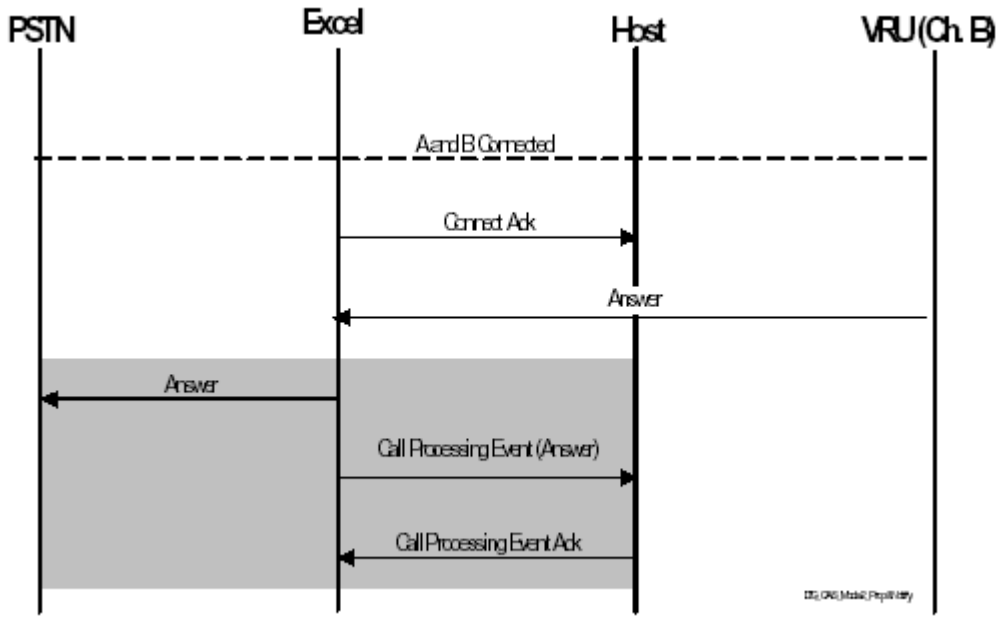


Propagate Answer to Distant End and Notify Host of Answer

When Answer signaling is detected from the outseized channel, Call Control propagates Answer to the inseized channel and notifies the host with a *Call Processing Event* message.

Important! Required Modification: You must change PPL Config Byte 2 of the CH component to 0x02.

Figure 5-13 Mode 2 - Propagate and Notify

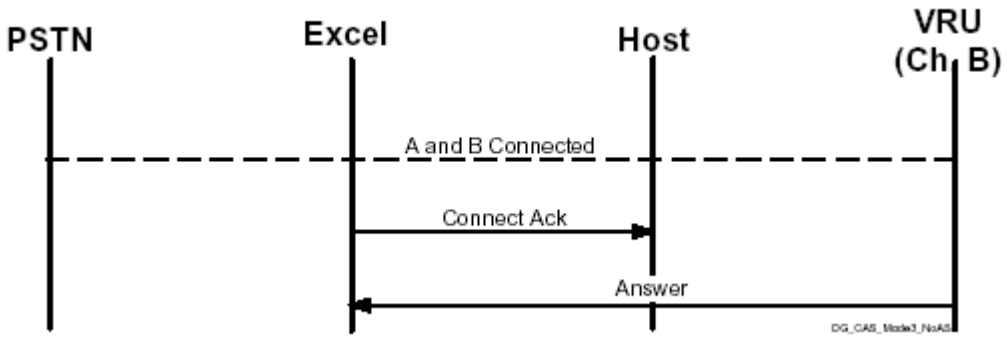


No Answer Supervision

When Answer signaling is detected from the outseized channel, Call Control does not propagate Answer to the inseized port and the host is not informed of the call being answered.

Important! Required Modification: You must change PPL Config Byte 2 of the CH component to 0x03.

Figure 5-14 Mode 3 - No Answer Supervision



Call Processing Guidelines

During call processing, real-world scenarios sometimes cause variations to the normal message flows designed by the application developer. The following are problem scenarios, with guidelines for developing the host application to accommodate these variations.

Glare

For bidirectional interfaces, the host and the network attempt to initiate an outseizure on a channel at the same time. Each network signaling protocol has its own means of determining which call is dropped. If the host initiates the outseizure with a *Route Control* (or *Outseize Control*) message, a response status of *Outseize Failure, Glare (0x1C)* is returned, followed by a *Channel Released* message (if the outgoing call is dropped). The CSP then sends a *Request for Service* or *Request for Service With Data* message to the host for the incoming call.

Positive ACK Wait

When sending multiple call control messages, the host should wait for a positive acknowledgment (ACK) to each message before sending another. Call Control allows multiple outstanding messages to a channel, but they are queued and processed in order until previous messages have been acknowledged.

If a channel purges, the CSP does not acknowledge outstanding host messages. You can configure the CSP to acknowledge outstanding host messages upon a purge by changing PPL Config Byte 19 (0x13) of the CH component to 0x01.

More Status

The More Status field is valid for some connection management messages. (See *Response Status Values* in the *API Reference*). For example, when the value “Invalid Event” (0x1D) is returned, the More Status value indicates the state of the CH state machine.

> **Important!** Some of the CH component call state values reported in the Status are not backward compatible with the values generated before System Software Release 5.3.

***DS0 Status Change* and *Channel Released* Messages**

Design your host application so that all channel states can accept and process the *DS0 Status Change* message from the CSP. All non-idle channel states must accept and process the *Channel Released* message.

Release Channel Race

When the distant end and the host initiate a release concurrently, a status of *Invalid Event* (0x1D) is returned, indicating that the call is

being released. The CSP sends a *Channel Released* message when the distant end releases.

Parking Channel Association

When the host initiates a *Park Channel* (A, B) message on a two-way connection, the acknowledgment message allows other messages to be directed to Channel A, and indicates only that Channel A has been parked. The host receives a *DS0 Status Change* message indicating that Channel B is parked.

Purge With No ACK

Sometimes the host initiates a message to the CSP, for example *Outseize Control*, to cause the purging sequence. Therefore, upon receiving the *DS0 Status Change* message of Out-of-Service, the CSP does not acknowledge the original message.

Call Control - Default Call Flows

This section includes default call flows for various call scenarios. You can modify these call flows to suit your application needs, using the PPL programmability of Call Control.

Simple Call Cut-Through

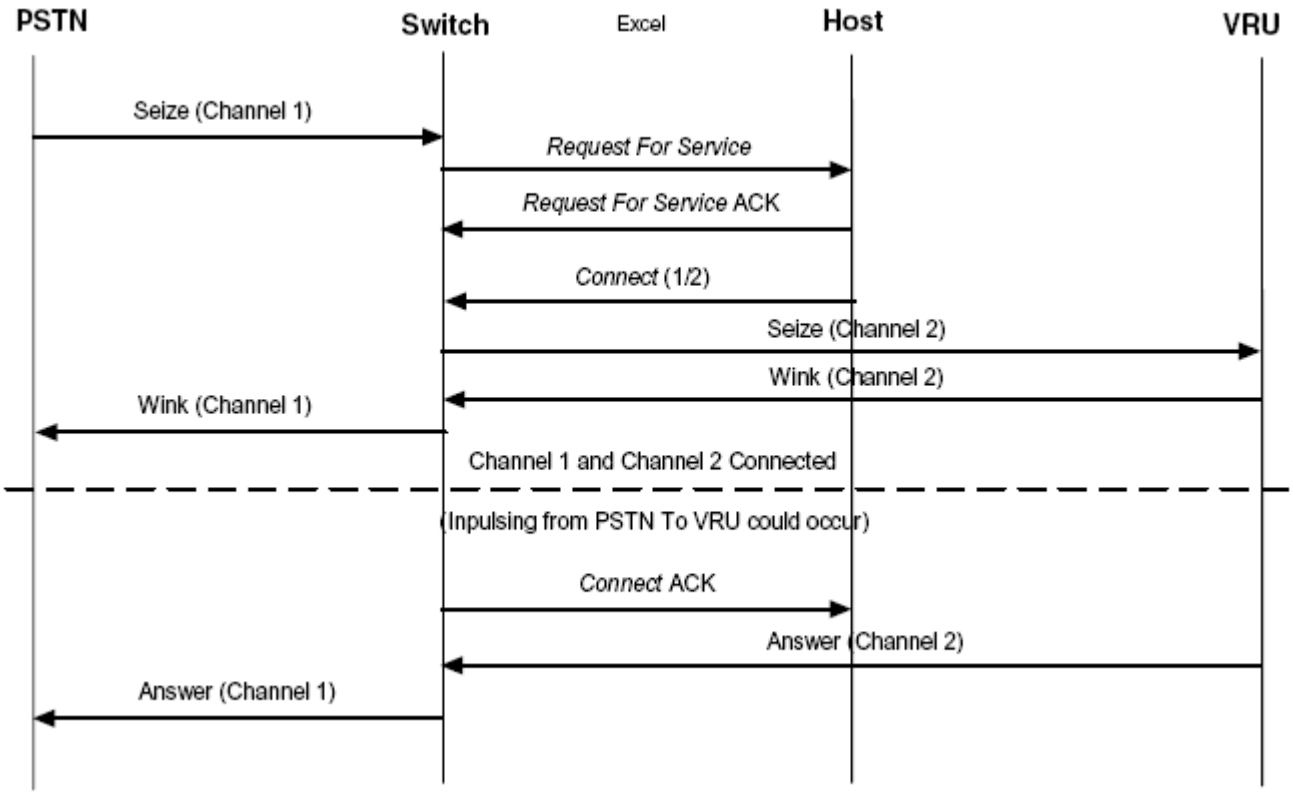
Trunk Type: E&M Wink Start

Inseize Instructions:

1. Report Incoming Call
2. Wait for Host Control

Sequence:

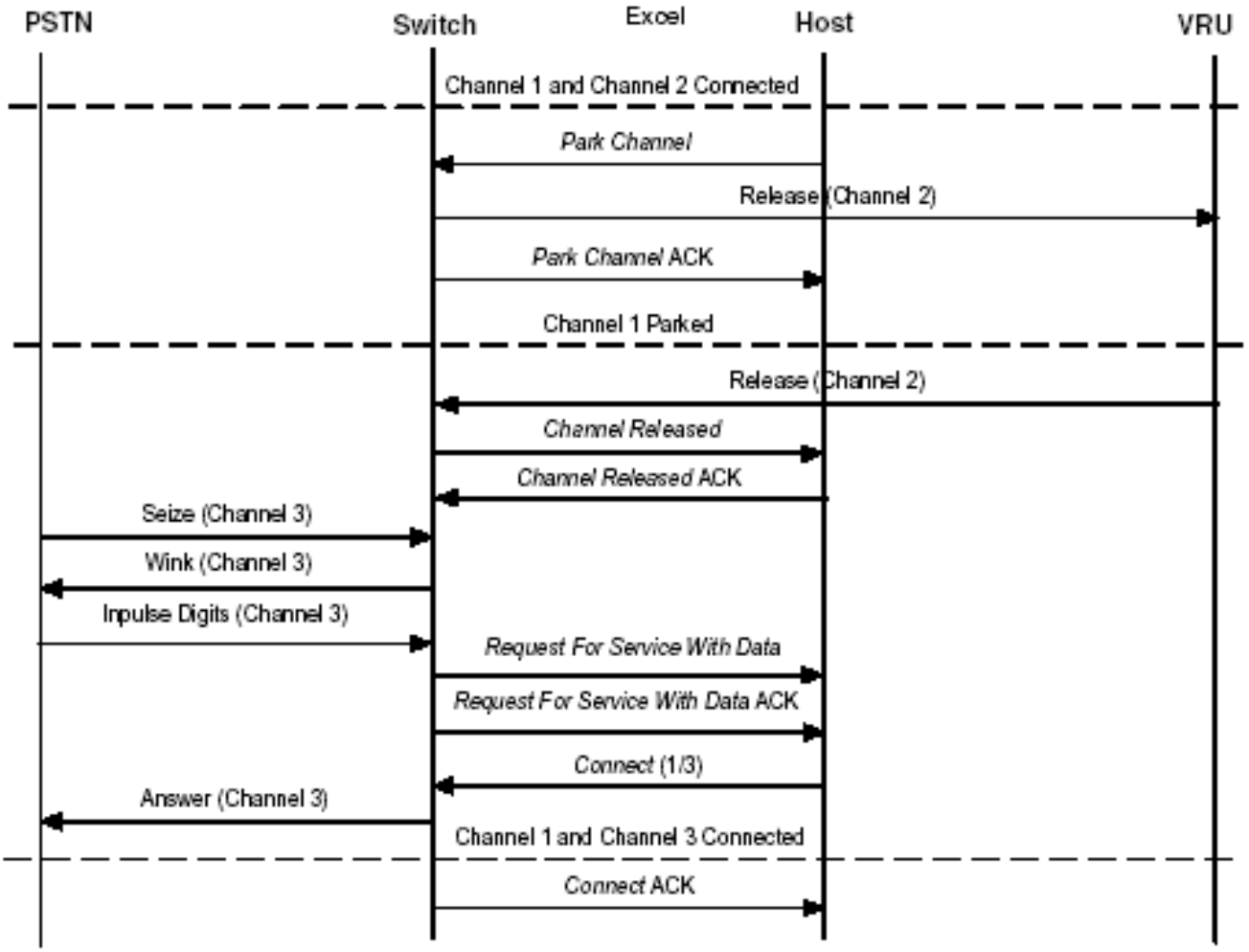
1. When the CSP detects an incoming call, the CSP reports the call along with the channel ID to the host in the form of a *Request for Service* message.
2. The host sends a *Connect* message to outseize the destination channel.
3. When the outseizure ACK (Wink) is detected, the inseize ACK (Wink) is generated.
4. The connection between the two channels is now established. The terminating equipment can now collect and analyze digits.
5. When answer signaling is detected, the CSP answers the incoming call.



Park with Reconnection to an Incoming Call

This call flow shows a parked connection with a re-connection to an incoming call.

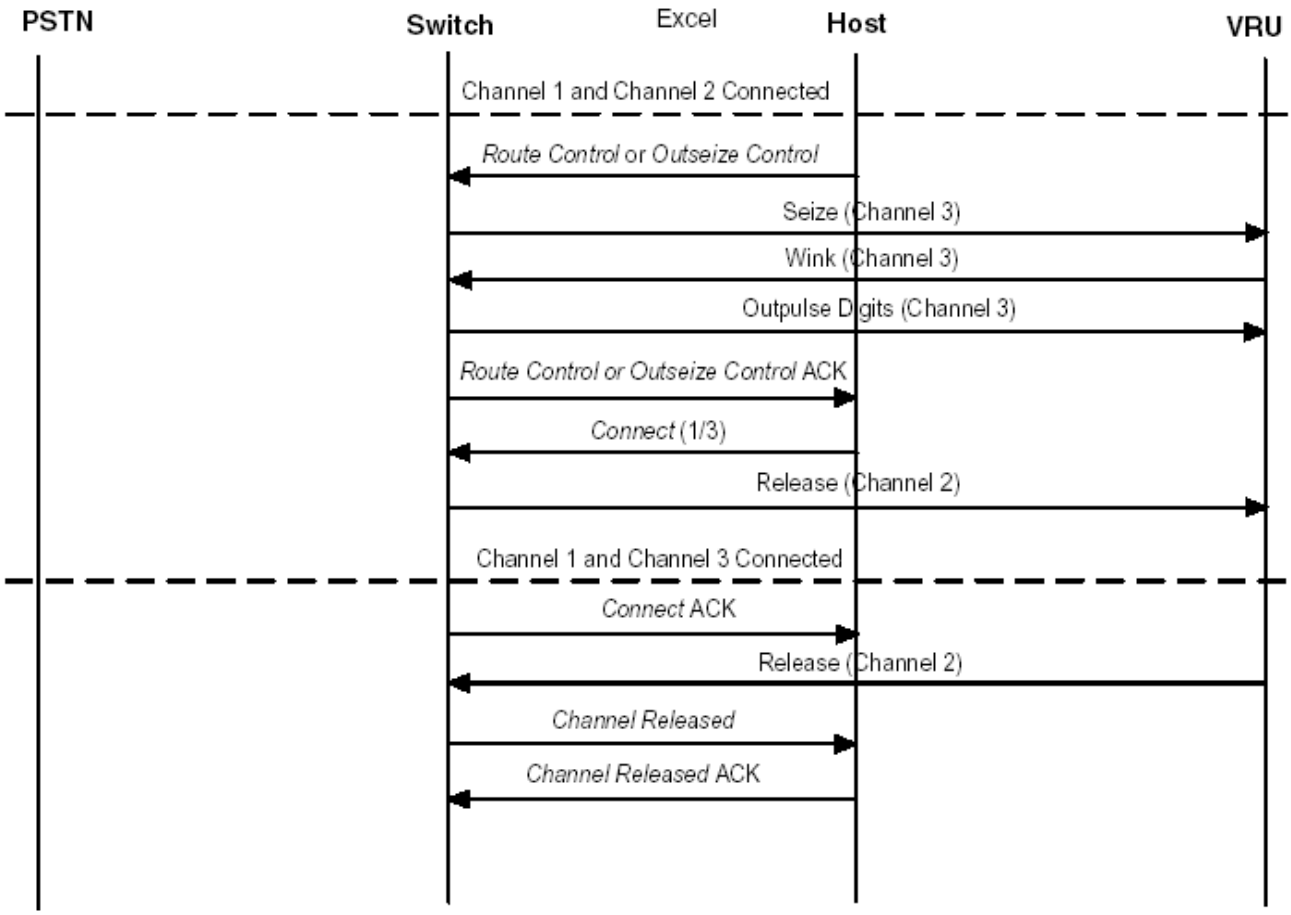
1. Channel 1 is connected to Channel 2.
2. The host issues a *Park Channel* message to park Channel 1.
3. The CSP parks Channel 1 and releases Channel 2.
4. The CSP processes an incoming call without host intervention.
5. This call is reported to the host along with all impulsed address digits.
6. The host sends a *Connect* message to connect Channels 1 and 3.
7. The CSP answers the incoming call if the parked call was previously answered, assuming that the CSP is controlling answer supervision.
8. Channel 1 is now connected to Channel 3.



Connecting a Connected Channel to an Outseized Channel

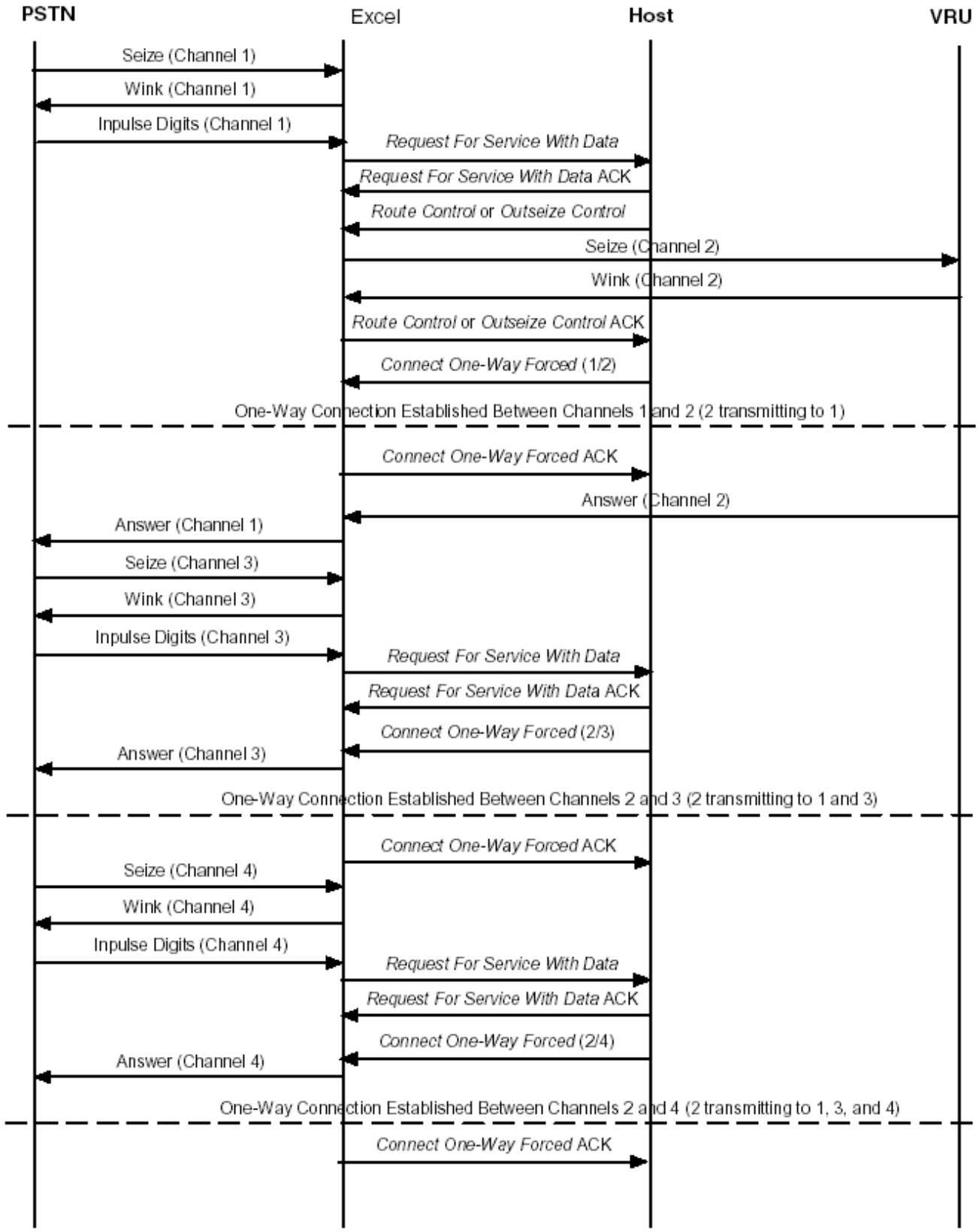
The following call flow shows a connection from a connected channel to an outseized channel.

1. Channel 1 and Channel 2 are connected.
2. The host initiates an outseizure with digit outpulsing.
3. Upon completion, the CSP returns an acknowledgment.
4. The host sends a *Connect* message to connect Channels 1 and 3.
5. Upon completion, the CSP sends an acknowledgment and releases Channel 2.



Broadcast with One-Way Connection

1. The CSP processes an incoming call over Channel 1.
2. The host initiates an outseizure over Channel 2.
3. Upon completion, the CSP returns an acknowledgment.
4. The host issues a *Connect One-Way Forced* message to the CSP, using Channel 2 as the one-way source.
5. The CSP makes the connection and returns an acknowledgment.
6. The CSP processes another incoming call over Channel 3 and reports it to the host.
7. The host again issues a *Connect One-Way Forced* message to the CSP, using Channel 2 as the one-way source.
8. The CSP makes the 1-way connection and returns an acknowledgment.
9. The CSP processes another incoming call over Channel 3 and reports it to the host.
10. The host issues another *Connect One-Way Forced* message to the CSP using Channel 2 as the one-way source.
11. The CSP makes the one-way connection and returns an acknowledgment.


PSTN
Excel
Host
VRU
Seize (Channel 1)
Wink (Channel 1)
Inpulse Digits (Channel 1)
Request For Service With Data
Request For Service With Data ACK
Route Control or Outseize Control
Seize (Channel 2)
Wink (Channel 2)
Route Control or Outseize Control ACK
Connect One-Way Forced (1/2)
One-Way Connection Established Between Channels 1 and 2 (2 transmitting to 1)
Connect One-Way Forced ACK
Answer (Channel 2)
Answer (Channel 1)
Seize (Channel 3)
Wink (Channel 3)
Inpulse Digits (Channel 3)
Request For Service With Data
Request For Service With Data ACK
Connect One-Way Forced (2/3)
Answer (Channel 3)
One-Way Connection Established Between Channels 2 and 3 (2 transmitting to 1 and 3)
Connect One-Way Forced ACK
Seize (Channel 4)
Wink (Channel 4)
Inpulse Digits (Channel 4)
Request For Service With Data
Request For Service With Data ACK
Connect One-Way Forced (2/4)
Answer (Channel 4)
One-Way Connection Established Between Channels 2 and 4 (2 transmitting to 1, 3, and 4)
Connect One-Way Forced ACK

6 Layer 4 Call Control PPL Information

Overview

This chapter contains information on configuration bytes, timers, and events for Call Control PPL components.

For information on PPL Component Addressing, refer to *PPL Component Addressing Information* in the *API Reference*.

CH - Channel Management (0x0061)

Configuration Bytes The table below shows the PPL Configuration Byte values for the CH component.

Byte	Description	Values
0x01	PCM Format	0x01 - u-law 0x02 - A-law (default = 0xFF, Use L3-provided value)
0x02	Answer Supervision	0x00 - Propagate Answer to Distant End 0x01 - Notify Host of Answer 0x02 - Propagate Answer to Distant End and Notify Host of Answer 0x03 - No Answer Supervision - no propagation of answer or notification (default = 0xFF, Use L3-provided value)
0x03	Local End Release Mode	0x00 - Release 0x01 - Park (default = 0xFF, Use L3-provided value)
0x04	Distant End Release Mode	0x00 - Release 0x01 - Park (default = 0xFF, Use L3-provided value)
0x05	Network Interface Type	0x00 - T1 or E1 CAS channel 0x01 - ISDN PRI channel 0x02 - Reserved 0x03 - ISDN PRI channel w/ raw data mode enabled 0x04 - SS7 channel
0x06	Flash Timing	0x00 - Flash Timing OFF 0x01 - Flash Timing ON / propagate to distant end 0x02 - Flash Timing ON / inform host 0x03 - Flash Timing ON / inform host and propagate
0x07	Local dB Padding	0x00: +3dB 0x01: 0dB (default) 0x02: -2dB 0x03: -3dB 0x04: -4dB 0x05: -6dB 0x06: -9dB

Byte	Description	Values
0x08	Outpulse Signal Type - Used only for Internal Routing.	0x01 - DTMF 0x02 - MFR1 (Host does not send KP ST) 0x03 - MFR2 0x04 - MFR1 (Host sends KP ST) 0x05 - Dial Pulse
0x09	Not Used	
0x0A	Turn on the L4 interworking feature	0x01 - turn on interworking 0x00 - turn off interworking (default)
0x0B-0x12	Not Used	
0x13	ACK Host Messages on Purge	0x00 - Do not ACK host messages on purge * 0x01 - ACK all queued host messages on purge
0x1A	Restore Answered Unconnected Channel After Matrix Switchover	0x00 - Purge channel (purge) 0x01 - Restore channel
0x14	Outseize Method	0x00 - Send Call Request to L4 0x01 - Send Outseize to L3
0x15	Routing Flag	0x00 - Send RFS to host 0x01 - Use Internal Router
0x16	Do not record an idle channel	0x00 - Record an idle channel (default) 0x01 - Do not record an idle channel
0x17	Not Used	
0x18	Two Purposes: 1. L4 Routing Fails 2. Host Involvement upon Release	 0x00 - Send RFS to host (Default) 0x01 - Clear the call 0x00 - Send host Channel Release Request (Default) 0x01 - Do not send host Channel Release Request
0x19	Not Used	
0x1A	Restore Answered Unconnected Channel After Matrix Switchover	0x00 - Purge channel (purge) 0x01 - Restore channel
0x1B-0x30	Not Used	
0x31	Purge Channel on Receiver Timeout	0x00 - Purge Channel on Receiver Timeout (Default) 0x01 - Receiver Timeout Will not Cause Channel to Purge.

Byte	Description	Values
0x32	TLV Count for Internal Routing TLVs	
0x33	TLVs for Internal Routing	
:	:	
0x7D	TLV Count for Outpulse Data TLVs	
0x7E	TLVs for Outpulse Data	
:	:	
0x97	TLV Count for IP Based Routing	
0x98	TLVs for IP Based Routing	
:		

PPL Timers The table below shows the PPL Timer values for the CH component.

Timer ID	Timer Name	Default Values (seconds)
1	*Request For Service* Retry Timer	5
2	Not Used	
3	L3 Clear Wait	150
4	L3 Outseize Wait	29
5	L3 Connect Wait	600
6	L5 Release Data Wait	20
7	TC Outpulse Complete Wait	60
8	Router Call Service ACK Wait	3 minutes
99	DSP Service Request Wait	3

External PPL Events The following PPL events are external and can be used by host application developers.

PPL Event ID	Description
0x00C9	Answer
0x00CA	Alerting

PPL Event ID	Description
0x00CB	Progress
0x00CC	Modify Bearer Services

Internal PPL Events

This section includes the PPL events sent between CH and other PPL components.

From Host to CH

The table below lists the PPL events sent from the host (L5) to CH (L4).

Event	Description
(1) L4CHevL5_CONNECT	Incoming L5 connect message (0x00)
(2) L4CHevL5_CONNECT_WAIT	Incoming L5 connect wait message (0x17)
(3) L4CHevL5_CLEAR	Incoming L5 release channel (0x08)
(4) L4CHevL5_OUTSEIZE_CONTROL	Incoming L5 outseize control message (0x2c)
(5) L4CHevL5_INSEIZE_CONTROL	Incoming L5 inseize control message (0x2b)
(6) L4CHevL5_CONNECT_WITH_PAD	Incoming L5 connect with pad message (0x03)
(7) L4CHevL5_GENERATE_CALLPROC_EV	Incoming L5 generate call processing event message (0xba)
(8) L4CHevL5_PARK_PORT	Incoming L5 park channel message (0xbf)
(9) L4CHevL5_CONNECT_WITH_DATA	Incoming L5 connect with data message (0x05)
(10) L4CHevL5_COLLECT_DIGIT_STRING	Incoming L5 collect digit string message (0xbc)
(11) L4CHevL5_REQUEST_DSP_SERVICE	Incoming L5 DSP service request message (0xbd)
(12) L4CHevL5_RELEASE_WITH_DATA	Incoming L5 release with data message (0x36)
(13) L4CHevL5_CPC_DETECTION	Incoming L5 CPC detection message (0x47)
(14) L4CHevL5_CONNECT_1WAY_FORCED	Incoming L5 connect 1-way forced message (0x50)
(15) L4CHevL5_FEATURE_REQUEST	Incoming L5 ISDN feature request message (0x22)
(16) L4CHevL5_RFS_RESPONSE	Incoming L5 RFS or RFS w/ data response (0x40 or 0x2d)
(17) L4CHevL5_CANCEL_DSP_SERVICE	Incoming L5 DSP service cancel (0xbe)
(18) L4CHevL5_CONNECT_TONE_PATTERN	Incoming L5 connect tone pattern message (0x2f)

Event	Description
(19) L4CHevL5_DISCONNECT_TONE_PATTERN	Incoming L5 disconnect tone pattern message (0x1e)
(20) L4CHevL5_DISCONNECT_RAN	Incoming L5 disconnect RAN message (0x56)
(21) L4CHevL5_CONNECT_RAN	Incoming L5 connect RAN message (0x55)
(22) L4CHevL5_OUTPULSE_DIGITS	Incoming L5 outpulse digits message (0x20)
(23) L4CHevL5_CONNECT_TO_CONFERENCE	Incoming L5 connect to conference message (0x4C)
(24) L4CHevL5_ROUTE_CONTROL	Route Control message from L5.

From Signaling to CH The table below lists the PPL events sent from the Network Signaling layer (L3) to CH (L4).

Event	Description
(30) L4CHevL3_CUT_THRU	L3 indication of ALERTING with a request to connect voice path.
(31) L4CHevL3_ALERTING	L3 indication of ALERTING
(32) L4CHevL3_PROGRESS	L3 indication of PROGESS message reception.
(33) L4CHevL3_FLASH	L3 indication of reception of FLASH.
(34) L4CHevL3_BUSY_OUT	L3 indication of Channel Busy Out.
(35) L4CHevL3_ACCESS_DENIED	L3 indication of operation failure.
(36) L4CHevL3_CHAN_STAT	L3 channel status, used to carry inservice and OOS information.
(37) L4CHevL3_SETUP_IND	L3 indication of an incoming call.
(38) L4CHevL3_DISCONNECT	L3 initiation of normal call release
(39) L4CHevL3_CLEAR_REQUEST	L3 indication of call teardown
(40) L4CHevL3_CONNECT	L3 indication of call answer.
(41) L4CHevL3_L4_CONTROLLED_SEIZE_ACK	L3 indication of successful Route Control Seize message execution.
(42) L4CHevL3_L4_CONTROLLED_SEIZE_REJECT	L3 indication of unsuccessful Route Control Seize message execution.

From CM to CH The table below lists the PPL events sent from CM to CH.

Event	Description
(45) L4CHevCFG_CHAN_STAT	CFG indication of channel status.
(50)L4CHevL4CM_CALL_SERVICE_REQUEST_INDICATION	Indication of a remote channel requesting an association to this channel.
(51)L4CHevL4CM_CALL_SERVICE_ACK_INDICATION	Indication that the remote channel has positively acknowledged the Call Service Request previously generated by this channel.
(52)L4CHevL4CM_CALL_SERVICE_REJECT_INDICATION	Indication that the remote channel has rejected the Call Service Request previously generated by this channel.
(53)L4CHevL4CM_ALERTING_CALL_SERVICE_INDICATION	Indication to send an ALERTING request to L3.
(54)L4CHevL4CM_CUT_THRU_CALL_SERVICE_INDICATION	Indication to send a CUT-THRU request to L3.
(55)L4CHevL4CM_ANSWER_CALL_SERVICE_INDICATION	Indication to send an ANSWER request to L3.
(56) L4CHevL4CM_CLEAR_CALL_SERVICE_INDICATION	Indication to send CLEAR request to L3.
(57) L4CHevL4CM_CLEAR_CONNECTION_SERVICE_INDICATION	Indication to Park this channel.
(58) L4CHevL4CM_FLASH_CALL_SERVICE_INDICATION	Indication to send a Flash request to L3.
(59) L4CHevL4CM_CLEAR_ACK_SERVICE_INDICATION	Indication that the Clear request has been processed.

From DSP Manager to CH The table below lists the PPL events sent from the DSP Manager to CH.

Event	Description
(80) L4CHevSYM_TONE_RCVR_DSP_SVC_ REQ_RESPONSE	Response to a DSP tone receiver request
(81) L4CHevSYM_CPA_RCVR_DSP_SVC_REQ_RESPONSE	Response to a DSP CPA receiver request
(82) L4CHevSYM_ENERGY_RCVR_DSP_SVC_REQ_RESPONSE	Response to a DSP energy receiver request
(83) L4CHevSYM_TRANSMITTER_DSP_SVC_REQ_RESPONSE	Response to a DSP transmitter request
(100) L4CHevTC_DIGITS	Indication of digits reported
(101) L4CHevTC_INPULSING_COMPLETE_TIMEOUT	Indication of an inpulsing complete timer expiration
(102) L4CHevTC_FIRST_DIGIT_TIMEOUT	Indication of a first digit timer expiration
(103) L4CHevTC_INTER_DIGIT_TIMEOUT	Indication of a inter-digit timer expiration
(104) L4CHevTC_OUTPULSING_COMPLETE	Indication of the completion of outpulsing reported
(105) L4CHevTC_CALL_PROGRESS_RESULT	Indication of a successful call progress analysis reported.
(106) L4CHevTC_TONE_COMPLETE	Indication that call progress tone generation is complete.
(107) L4CHevTC_CPC_RESULT	Indication of the detection of CPC from the attached CPA receiver
(108) L4CHevTC_FIRST_DIGIT	Indication of the first received digit
(109) L4CHevTC_RAN_STARTING	Indication of the beginning of the RAN chain from the attached RAN transmitter
(110) L4CHevTC_RAN_COMPLETE	Indication of the completion of the RAN chain
(111) L4CHevTC_ENERGY_DETECTED	Indication of energy detected
(112) L4CHevTC_ENERGY_DETECTION_TIMEOUT	Indication of an energy detection timeout
(120) L4CHevL4RTR_ROUTE_CONTROL	Indication from the Router of a Route Controlled Seize

CM - Call Management (0x0062)

Configuration Bytes

The table below shows the PPL Configuration Byte values for the CM component.

BYTE	Description	Values
1	Force propagate answer on forced 1-way connections	0x00 - Do not propagate answer 0x01- Propagate Answer

General Purpose Registers

The table below shows the default PPL General Purpose Register value for the CM component.

BYTE	Description
1	Local CH channel state

PPL Timers

The table below shows the PPL Timer values for the CM component.

Timer ID	Timer Name	Default Values (seconds)
1	Remote CM CSA Ack Wait	30
2	Remote CM Alerting Wait	60
3	Remote CM Answer Wait	640
4	Not Used	
5	Remote CM Channel Info Wait	30
6	Remote CM Clear Response Wait	160
7	Not Used	
99	Conference Message ACK Wait	12

PPL Events The table below shows the PPL Events sent to the CM component.

Event	Description
(1) L4CMevL4CH_ALERTING_SERVICE_REQUEST	Indication of CH receiving an alerting event from L3. This is a request by CH to propagate the event to the remote channel (if a call is active).
(2) L4CMevL4CH_CUT_THRU_SERVICE_REQUEST	Indication of CH receiving a cut-thru event from L3. This is a request by CH to propagate the event to the remote channel (if a call is active).
(3) L4CMevL4CH_ANSWER_SERVICE_REQUEST	Indication of CH receiving a connect event from L3. This is a request by CH to propagate the event to the remote channel (if a call is active).
(4) L4CMevL4CH_CALL_SERVICE_REQUEST	Indication of CH requesting an association with another channel. This is a request by CH to propagate the event to the remote channel.
(5) L4CMevL4CH_ROUTE_SERVICE_REQUEST	Indication of CH requesting an association with another channel determined through the routing function. This is a request by CH to propagate the event to the remote channel through the router.
(6) L4CmevL4CH_CALL_SERVICE_ACK_REQUEST	Positive acknowledgment by CH to a call service request.
(7) L4CmevL4CH_CALL_SERVICE_REJECT_REQUEST	Negative acknowledgment by CH to a call service request.
(8) L4CmevL4CH_CLEAR_CALL_SERVICE_REQUEST	Indication by CH to clear the call. This is a request by CH to propagate the event to the remote channel (if a call is active).
(9) L4CmevL4CH_CLEAR_CONNECTION_SERVICE_REQUEST	Indication by CH to clear the connection. This is a request by CH to propagate the event to the remote channel (if a call is active).
(10) L4CmevL4CH_CALL_SERVICE_1WAY_FORCED_REQUEST	Indication of CH requesting a 1 way forced association with another channel. This request will not establish any signaling propagation to the remote channel.

Event	Description
(11) L4CMevL4CH_FLASH_SERVICE_REQUEST	Indication of CH receiving a flash event from L3. This is a request by CH to propagate the event to the remote channel (if a call is active).
(12) L4CMevL4CH_CLEAR_ACK_REQUEST	Response from CH that the clear request sent has been processed.
(30) L4CmevL4CM_CALL_SERVICE_REQUEST_ACK	Message sent by remote CM acknowledging a successful call association.
(31) L4CmevL4CM_CALL_SERVICE_ REQUEST _REJECT	Message sent by remote CM acknowledging a failed call association.
(32) L4CmevL4CM_ALERTING_CALL_SERVICE_REQUEST	Indication by the remote CM that the remote channel is propagating an alerting event to the local channel.
(33) L4CmevL4CM_CUT_THRU_CALL_SERVICE_REQUEST	Indication by the remote CM that the remote channel is propagating a cut-thru event to the local channel.
(34) L4CMevL4CM_ANSWER_CALL_SERVICE_REQUEST	Indication by the remote CM that the remote channel is propagating an answer event to the local channel.
(35) L4CMevL4CM_CALL_SERVICE_REQUEST	Indication by the remote CM that the remote channel is requesting a call association to the local channel.
(36) L4CMevL4CM_CONNECT_REQUEST	Indication by the remote CM that the remote channel is requesting a voice path connection with the local channel.
(37) L4CMevL4CM_CLEAR_CALL_SERVICE_REQUEST	Indication by the remote CM that the remote channel is propagating a clear call event to the local channel.
(38) L4CMevL4CM_CLEAR_CONNECTION_SERVICE_REQUEST	Indication by the remote CM that the remote channel is propagating a clear connection event to the local channel.
(39) L4CMevL4CM_CHANNEL_INFO_REQUEST	Indication from a CM that the remote channel is requesting information on the channel (physical ID and PCM encoding).
(40) L4CMevL4CM_CHANNEL_INFO_ACK	Response from the remote CM indicating its physical information.

Event	Description
(41) L4CMevL4CM_FLASH_SERVICE_REQUEST	Indication by the remote CM that the remote channel is propagating a flash event to the local channel.
(60) L4CMevL4RTR_CALL_SERVICE_REJECT_INDICATION	Indication by the Router components that an attempt to route the call has failed.
(61) L4CMevL4RTR_CALL_SERVICE_REQUEST	Indication by the remote RTR that the remote channel is requesting an call association to the local channel.

PC - Physical Connection Management (0x0063)

PPL Events The table below shows the PPL Events sent to PC from the CM component.

Event	Description
(1) L4PCevL4CM_LOCAL_CALL_CONNECT_REQUEST	Request to connect voice path from the local channel
(2) L4PCevL4CM_REMOTE_CALL_CONNECT_REQUEST	Request to connect voice path from the remote channel
(3) L4PCevL4CM_DISCONNECT_REQUEST	Request to disconnect voice path from the remote channel
(4) L4PCevL4CM_LOCAL_CALL_FORCED_CONNECT_REQUEST	Request to immediately force connect the voice path

RTR - Router (0x0064)

Configuration Bytes

The table below shows the PPL Configuration Byte values for the Router component.

BYTE	Description	Values
0x01	Hunting Algorithm	0x00 = Reserved 0x01 = First Free 0x02 = Reserved 0x03 = Round Robin
0x02	Reserved	
0x03	Route entry on rejection (How to handle a rejection from a terminating object)	0x01 - Do Not Retry Routing (default) 0x02 - Retry Routing
0x04	Resource Group Table ID	The Resource Group Table to be used (User-defined)
0x05	Route Table ID	The Route Table to be used (User-defined)

General Purpose Registers

The table below shows the General Purpose Register values for the Router component.

GPR #	Description
1	Routing Method
2	Search Key
3	Route Table Entry row index
4	Entry Data Tag; Resource Group member counter
5	Entry Data TLV Count
6	Resource Group Member Count
7	Hunting Algorithm
8	Resource Group Address Element Type
9	Resource Group Member counter
10	Reserved
11	Index to the Entry Data TLVs
12	TLV Data

GPR #	Description
13	Error Codes for rejecting Route requests
14	Reserved
15	Reserved
16	Flag for Criteria/Route Group ID
17	Resource Group Tag length
18	Resource Group Data
19	Flag to track whether the Request initiator being CM or RTR
20	Reserved
21	Reserved
22	More Criteria
23	Num Digits to compare
24	Criteria Type
25	Route Group ID from Route Table

PPL Timers

The table below shows the PPL Timer values for the RTR component.

Timer ID	Timer Name	Default Value (seconds)
2	Remote CM/RTR CSA ACK Wait	60
3	Initiating RTR Route Operation Complete Event Wait	30

PPL Events The table below shows the PPL events sent to the Router component.

Event	Description
To Router from CM	
(1) L4RTRevL4CM_CALL_SERVICE_ROUTE_REQUEST	Indication of CM requesting a route derivation for an incoming call.
Router to Router	
(2) L4RTRevL4RTR_CALL_SERVICE_ROUTE_REQUEST	Indication of RTR requesting a route derivation of an incoming call on a different node.
(3) L4RTRevL4RTR_CALL_SERVICE_REQUEST_ACK	Indication of RTR successfully deriving a route for a CSR.
(5) L4RTRevL4RTR_CALL_SERVICE_REQUEST_REJECT	Indication of RTR being unsuccessful in deriving a route for a CSR.
(7) L4RTRevL4RTR_CALL_SERVICE_REQUEST	Indication of RTR requesting a route derivation to the RTR component at the other node.
(8) L4RTRevL4RTR_ROUTE_OP_COMPLETE	RTR Indicating to the other RTR about completion of an association among the two RTR components.

PPL Component for Interworking (0x0084)

Configuration Bytes

The table below shows the PPL Configuration Byte values for the Layer 4 Interworking component

Byte (Hex)	Function	Values
0x03	Local Network Protocol ID	0x01 - SS7
0x04	Local Network Protocol Flavor	SS7 0x01 - ISUP ANSI 0x02 - ISUP ITU
0x0E	Host Protocol ID	0x05 - Universal Protocol
0x0F	Host Protocol Flavor	0x01 - Universal Flavor

7 Configuring and Using Resources on the DSP Series 2 Cards

DSP Series 2 Card Configurations

The CSP supports the following DSP Series 2 cards and associated I/O cards:

- DSP Series 2 and associated Multi-Function Media I/O
- DSP Series 2 Plus and associated Multi-Function Media I/O Plus

The Overview section in Chapter 7 provides detailed information on the DSP Series 2 Plus card and a comparison to the DSP Series 2 card.

In this document, unless otherwise stated, **DSP Series 2** will be used for both the DSP Series 2 and DSP Series 2 Plus cards. Any software or hardware differences are indicated.

Basic DSP Configuration

Contents

DSP Configuration API Message Summary

Introduction The following table shows the API message used for DSP configuration.

GENERAL DSP CONFIGURATION		
Tasks	**Options/Notes**	**API Message(s)**
Assign Function Types	See Function Types	*DSP SIMM Configure*
Configure NFS	File Management Server Address Vocabulary Index File NFS User Group ID NFS Poll Retries	*Generic Card Configure*
Configure Overload Alarm Thresholds	See Configuring Overload Management	
Assign IP Address	Required for NFS	*IP Address Configure*
Bring DSP In Service		*Service State Configure*
FUNCTION-SPECIFIC CONFIGURATION		
Tasks	**Options/Notes**	**API Message(s)**
Conferencing		
Configure Card-level Conferencing Parameters	Output Gain Control Noise Gating Echo Suppression Auto. Gain Control Failure Behavior	*Generic Card Configure*
Record Files		
Record Files	Gain File Event Descriptor Beep Tone Parameters Initial Silence Timer Final Silence Timer Max. Record Timer Dual Chan. Rec. Option Silence Threshold Append or Replace	*Record File Start* *Record File Stop*

Modify Recording	Modify Gain Pause Resume	*Record File Modify*
Tones		
If required, modify default tones and Call Progress Analysis Pattern/Class.	See Customizing Tones.	*Call Progress Analysis Pattern Configure* *Call Progress Analysis Class Configure* *Tone Configure* *Inseize Instruction List Configure*
Implementing DSP Resources		
Conferencing	See Conferencing.	*Resource Create* *Resource Connect* *Resource Modify* *Resource Disconnect*
Play Files	See Playing a File.	*DSP Service Request* *DSP Service Cancel* *Play File Start* *Play File Modify* *Play File Stop*
Record Temporary Files	See Recording Files.	*Record File Start*
Tones	See Tones.	*DSP Service Request* *DSP Service Cancel* *Connect Tone Pattern* *Disconnect Tone Pattern*

Basic Card Configuration Sequence

Configuration API Messages

The following API messages are used for general DSP configuration.

- Generic Card Configure 0x0122

  Use this message to configure the DSP Series 2 card for NFS, alarms, statistics, and card-level defaults.

- DSP SIMM Configure 0x00C0

  Use this message to configure DSP chips for specific functions.

- IP Address Configure 0x00E7

  Use this message to assign an IP Address to the DSP card for NFS functionality.

- Service State Configure 0x000A

  Use this message to take DSP chips in and out of service.

Configuration is maintained through warm starts, resets, and faults.

Procedure

The following is a basic configuration sequence for the DSP Series 2 card.

1. Bring Card out of Service

   Use the Service State Configure 0x000A message to bring the card out of service.

2. Configure DSP Function Types

   Use the DSP SIMM Configure 0x00C0 message to assign function types on each SIMM. The default DSP configuration is shown in the message.

3. Download Additional Resource Points if required.

   If your DSP resource requirements exceed the standard resources points that come with the card, you must download a license for more points.

   See Downloading License Keys to the CSP

   Also see DSP Resource Points in the DSP Series 2 Card Product Description chapter for more information.

4. Configure NFS Servers (up to 8)

   Use the Generic Card Configure 0x0122 message with the Announcement Configuration TLV (0x05DC).

   a. Server IP Address (for NFS)

      The IP Address must be set on initial power up only.

   b. Configure Vocabulary Index File and Announcements

5. Configure Conferencing

   Use the Generic Card Configure 0x0122 message to set conferencing features and options on a per-card basis. All conferences established on the card will use these settings unless you override them on a per-conference basis with the Resource Create 0x0124 or Resource Modify 0x0125 message.

   See Configuring Conferencing Features.

   See Configuring Conferencing Options.

6. Customize Tones

   Perform any tone customization required, such as for Call Progress Analysis.

   See Digit Collection

7. Bring Card in Service

   Use the Service State Configure 0x000A message to bring the card back in service.

Call Processing Messages

Table 8-1 shows the API messages related to DSP resources.

Table 7-1 DSP-Related API Messages

Category	Message	Description
File Playback	*Play File Start 0x011B* *Play File Stop 0x011D* *Play File Modify 0x011C*	Use these messages to start, stop, or modify the playing of a file.
	Recorded Announcement Connect 0x0055	If you have DSP-ONE cards in your system, you can use this message to connect to a recorded announcement on the DSP Series 2 card (only RAN IDs below 4096), however, the enhanced functionality of the DSP Series 2 card will not be available. Announcements must be accessible via NFS.
DSP Services	*DSP Service Request 0x00BD* *DSP Service Cancel 0x00BE*	Use these messages to allocate or cancel a DSP service on channel.
Tones	*Call Progress Analysis Result 0x0034*	The CSP sends this message to report the result of Call Progress Analysis on a channel. Only one result is reported per message.
	Connect Tone Pattern 0x002F	Use this message to transmit a call progress pattern ID to the specified channel. In order for the host to be notified when tone transmission has stopped, it must set the *Generate Event Flag* field of this message.
	Disconnect Tone Pattern 0x001E	Use this message to terminate call progress tone transmission on the specified channel.

Category	Message	Description
Conferencing	*Resource Create 0x0124*	Use this message to create a conference.
	Resource Delete 0x0126	Use this message to delete a conference.
	Resource Modify 0x0125	Use this message to change the options or features of an existing conference.
	Resource Delete Indication 0x0129	The CSP sends this message to the host after a conference is deleted.
	Resource Connect 0x0127	Use this message to add a channel to a conference which has already been created.
	Connect One-Way To Conference 0x004F	Use this message to establish a one-way (listen only) connection between the channel specified and the previously created conference specified.
	Resource Disconnect 0x0128	Use this message to remove a channel from a conference.
General	*Alarm 0x00B9*	This message is sent by the CSP to indicate alarm conditions on the DSP Series 2 card. See DSP Alarms in the DSP Series 2 Card Product Description chapter for more information.
	Card Status Report 0x00A6	This message is sent by the EXS platform to report the number of DSP modules and their function types configured on the DSP card.

Administration API Messages

Feature Description See Administration.

API Messages The following messages are used for queries and alarms.

Queries	*Generic Card Query 0x0123*	Use this message to query the configuration of the DSP Series 2 card for NFS, alarms, statistics, and card-level defaults.
	Statistics Query 0x0121	Use this message to query cache and function usage statistics on the DSP Series 2 card.
	System Configuration Query	Use this message to query various configuration settings in the CSP, including the following for DSP: • 0x04 - Active Conference IDs • 0x07 - Resource Threshold • 0x09 - Resource Usage Reporting • 0x10 - Detailed Conference Information • 0x11 - DSP Resource Threshold • 0x13 - Conference Speakers • 0x14 - Conferencing Features • 0x15 Child Conference Information • 0x16 Child Conference IDs
	System Resource Usage Query	Use this message to query the following conferencing resource utilization information: - Timeslots in use - Timeslots available - Percent used
	Generic Report	This message is generated in response to the following DSP Query Types sent in the *System Configuration Query* message: • 0x04 - Active Conference IDs • 0x0A - Resource Usage • 0x10 - Detailed Conference Information • 0x13 - Conference Speakers • 0x14 - Conferencing Features • 0x15 - Child Conference Information • 0x16 - Child Conference IDs

	Tone Query 0x005A	Use this message to query the transmit and receive tones assigned to a DSP, as well as the frequency and power levels of the tones.
	Call Progress Analysis Configuration Query 0x008A	Use this message to query call progress analysis (CPA) configuration information.
	Transmit Cadence Pattern Query 0x0059	Use this message to query any of the configured transmit cadence parameters.
	Card Status Query 0x0083	Use this message to query the number of DSP modules and their function types configured on the DSP Series 2 cards. You can address the *Card Status Query* message to the I/O slot behind the DSP Series 2 cards. The following are possible results: • If the Multi-Function Media I/O card is present, the Card Type contains 0xCA. • If the Multi-Function Media I/O Plus card is present, the Card Type contains 0x92. • If an illegal I/O card is present the Card Type is 0xAA. • If no I/O card is present the CSP nacked the message with status 0x61-Invalid Slot
General	*Alarm 0x00B9*	This message is sent by the CSP to indicate alarm conditions on the DSP Series 2 card. See DSP Alarms in the DSP Series 2 Card Product Description chapter for more information.
	Card Status Report 0x00A6	This message is sent by the CSP platform to report the number of DSP modules and their function types configured on the DSP card.

Configuring Overload Management

Configuration

To configure the overload threshold values, the number of times that the threshold must be exceeded (M), and the size of the sample window (N), use the Generic Card Configure 0x0122 message (0x0122) with the Alarm Threshold TLV (0x05EF) and these Alarm types:

- 0x0A CPU Idle Time
- 0x0B VRA Process Time
- 0x0C VRA IO Queue Time

Query

To query the overload thresholds, use the Generic Card Query 0x0123 message (0x0123) with the 0x05FA Card Object TLV.

To query the overload statistics, use the Statistics Query 0x0121 message (0x0121) with a slot AIB and the 0x0616 DSP Series 2 Overload Statistics (0x0616).

TLVs for DSP Series 2 Overload Management

- 0x0616 DSP Series 2 Overload Statistics
- 0x0617 Alarm Threshold Query
- 0x0618 Alarm Threshold Enabled Report

Record/Play Files

Contents

Recording Files

Feature Description

See File Record And Playback.

Procedure

1. Record files using the Record File Start 0x011E and Record File Stop 0x0120 messages.

   You should use File IDs above 0x00100000 for longer files and for files you want to keep for a significant period.

   For temporary recordings, such as those used for to directory assistance requests, you should use File IDs below 0x00100000. You must play back the temporary file from the same DSP Series 2 card on which they are recorded.
2. Create a Vocabulary Index File (VIF) to track recordings and point to files for playback.

   See VIF Format in the DSP Series 2 Card Product Description chapter.
3. Download the VIF to the DSP card using the 0x05DF Vocabulary Index File TLV in the Generic Card Configure 0x0122 message.

Recording API Messages

- Record File Start 0x011E

  Use this message to start recording a file.
- Record File Stop 0x0120

  Use this message to stop recording a file.
- Record File Modify 0x011F

  Use this message to modify the gain of a file recording, as well as to pause or resume the recording.
- DSP Cache Modify 0x011A

  Use this message to delete a file from cache, or copy a file from cache to permanent storage.

Deleting Files

To delete files, send the DSP Cache Modify 0x011A message.

Recording a File to a Network File System (NFS) Server

Important! Dialogic recommends a file storage system that is dedicated to NFS.

When recording a file to a Network File System (NFS) server, the DSP Series 2 card checks the NFS server status by using the Server ID set in the 0x05E2 File Location TLV.

After verification that the Server Status byte has been set to 0x02, the DSP Series 2 card sends a NFS Create Call message to the NFS server by using the Filename set in the 0x05E2 File Location TLV.

The NFS server must reply to each NFS Write Call message that the DSP Series 2 card sends within 500 milliseconds (ms). When the NFS server replies within 500ms to the DSP Series 2 card, the file is created and opened. The DSP Series 2 card then begins sending NFS Write Call messages. Each block of data sent represents 8000 bytes.

If the NFS server does not reply within 500ms, the DSP Series 2 card sends a retransmission of that same write request. If the DSP Series 2 card receives the next reply to the retransmission within 500ms, the next write request continues. If on the retransmit, the DSP Series 2 card does not receive the reply within 500ms from the NFS server, the DSP Series 2 card stops the recording and reports the following to the host.

- 0x02 Card Alarm, 0x48 NFS Write Error
- Call Processing Event, 0x35 Record File Stopped with reason NFS Write Error

Important! If you attempt to record to the NFS server and the directory location does not exist on the server, the DSP-2 card will not create the directory. The file is not created and the CSP sends the following two alarms:

- 0x02 Card Alarm, 0x4C, NFS Open File Error
- 0x02 Card Alarm, 0x48, NFS Write Error

Disable Idle Channel Recording

You can prevent recording idle channels. This functionality is configurable on a per channel basis and is disabled by default. That is, recording idle channels is allowed by default.

Race conditions can occur between a request to record a channel and a channel releasing. These conditions can cause the CSP to record an idle channel. Below is a sample scenario.

Example: The host sends a *Record File Start* (0x011E) message and the channel releases (for example, ISUP REL is received) before the CSP processes the *Record File Start* message.

The host can receive a positive ACK for the *Record File Start* message and an idle channel is recorded.

If another inbound call immediately lands on the same channel, it will be recorded without the host requesting the recording. If a Record File Start message is sent for the second call, it is NACK'ed

Prior to this feature, when the host sent a Record File Start message to an idle channel it was positively ACK'ed and the channel was recorded. As mentioned above, it is possible that the CSP ended up recording an idle channel which you do not require.

Enabling this Feature

Using API

Use the *PPL Configure* (0x00D7) message with configuration byte 0x16 in L4 CH (Component 0x61) set to 1 to enable this feature. If the configuration byte is set to 0 this feature is disabled which is the default. The configuration is on a per channel basis.

The host must explicitly set this configuration byte to value 1 to prevent recording on an idle channel. In this case, the Record File Start message is NACK'ed with response 0x1D03 if the channel is idle. The response is as follows:

- MSB 0x1D - Invalid channel state
- LSB 0x03 - Channel state

The following shows the API message to enable and disable this feature.

Enabled

```
00 12 00 d7 00 00 ff 00 01 0d 03 00 00 00 00 61 01 01 16
   01
```

Disabled

```
00 12 00 d7 00 00 ff 00 01 0d 03 00 00 00 00 61 01 01 16
   00
```

Using CSA

You can set configuration byte 0x16 in the PPL Configuration window from the Channel Group Configuration menu.

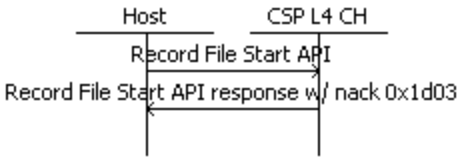
Querying the Feature

Use the PPL Data Query message to query the configuration byte as shown below.

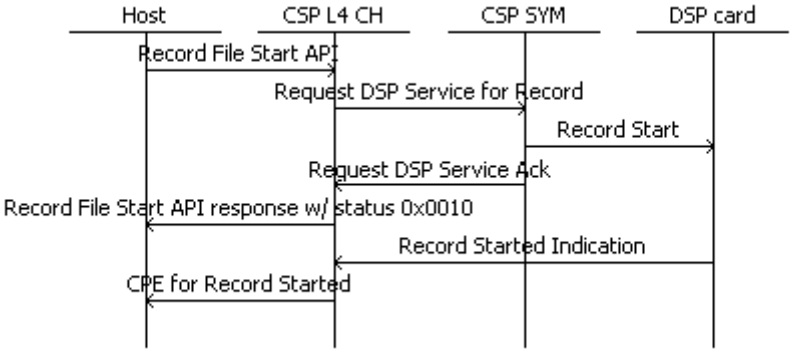
```
00 0f 00 de 00 00 ff 00 01 0d 03 00 11 00 00 61 01
```

Call Flows

The following call flow diagram shows the host sending the Record File Start message with this feature enabled. The call flow assumes the channel is in the idle state.



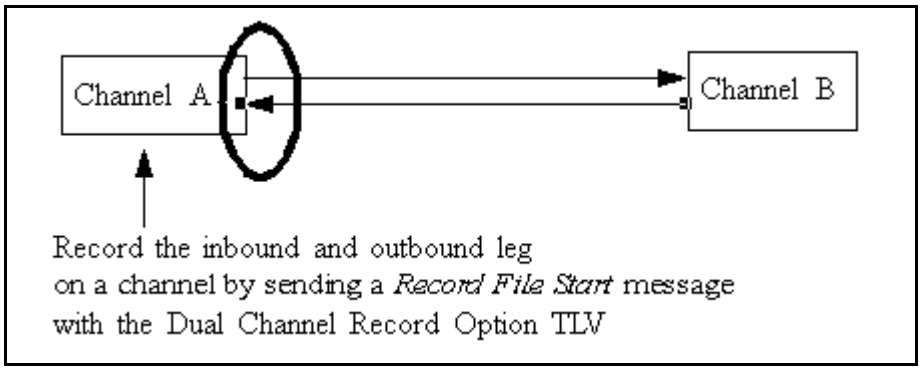
The following call flow shows the host sending a Record File Start message with the feature disabled. The call flow assumes the channel is in the idle state.



Full Duplex Channel Recording

Introduction

This feature allows you to record both the inbound and outbound legs of a single channel. You can use this feature to record both channels in a connection within a single node or across the EXNET® ring in a multi-node system.



Implementation

To invoke this feature, send a Record File Start 0x011E message that contains a single Channel AIB and the Dual Channel Recording TLV (0x5ED). In the TLV you can select whether the channels are recorded summed or independently.

TLV Format

The format of the Dual Channel Record Option TLV is shown below:

0x05ED Dual Channel Record Option

Use this TLV to record both sides of a connection. This TLV allows you to choose whether you want each channel to be saved to its own file (Independent) or both channels to be saved to the same file (Summed).

This TLV is optional for the Record File Start 0x011E message.

Byte	Description
0, 1	Tag: 0x05ED Dual Channel Record Option
2, 3	Length: 0x0001
4	Value[0]: 0x00 - Sum Channels Together (Default) 0x01 - Independent Channels (file location is the same, but filename is appended with “_2”)

Example Call Trace

```
Record File Start
H->X
00 78 01 1e 00 00 ff 00 02 0d 03 00 00 00 01 01 04
00 04 'Number of TLVs
05 e0 00 04 00 00 00 0a 'File ID
05 e1 00 02 00 01 'File Format
05 E2 00 15 00 01 00 02 5C 52 65 63 6F 72 64 5C 74 65
   73 74 2E 75 6C 61 77 'File Location
05 ED 00 01 00 'Dual Channel Record - Summed
```

Playing a File

Introduction

You use the *Play File Start* and *Play File Stop* messages to control Playback of files.

Play File API Messages

- Play File Start 0x011B

  Use this message to start playing a file.
- Play File Stop 0x011D

  Use this message to stop playing a file.
- Play File Modify 0x011C

  Use this message to modify (gain or speed), pause, resume, skip forward, or skip backward in a file that is currently playing.
- Recorded Announcement Connect 0x0055

  If you have DSP-ONE cards in your system, you can use this message to connect to a recorded announcement on the DSP Series 2 card (only RAN IDs below 4096), however, the enhanced functionality of the DSP Series 2 card will not be available. Announcements must be accessible via NFS.

File Barge-in Mode

Transmission of a file in barge-in mode is supported. You must use the *Play File Start* message with the Barge In TLV (0x05E5).

Specifying Offset and Length

You can specify the Offset and Length in one of two ways:

- in two fields within the Vocabulary Index File
- in the File Offset and Length TLV (0x0614) used in the *Play File Start* message. You can use the File Offset and Length TLV when queueing files, but you cannot use it when chaining files within a single message.

  If the Length plus the Offset is greater then the actual file size, the file plays until the end. If the Offset is greater then the actual file size, a File Open error occurs.

Queuing Files

You can send multiple *Play File Start* messages and the files will be queued so that when one ends, the next one starts. This functionality works for all files, cached or un-cached, and with or without using the VIF file. There is no limit to the number of files you can queue, except for the limits imposed by memory.

You can also queue a chain of files using the *Play File Start* message by entering the number of files in the Number of Files field, and using the Play File Queue TLV.

When the first file in the queue is started, a Call Processing Event 0x002E message of File Started is generated, and when the last file in the queue has been played, a File Complete event is generated.

Failure Conditions

- If a Play File request has more than one file descriptor, the request is NACKed.
- If a queued file cannot be opened, a CPE of *Play File Queue Failure* (0x33) is returned. All current requests in the queue are cleared and the DSP resource is freed.
- If queueing causes memory to be used up, a File Queue Failed alarm is returned.

Playing a .wav File

You indicate the WAV format in the File Format TLV (0x05E1) in the following API messages:

- Play File Start 0x011B
- DSP Cache Modify 0x011A

Alarm

If there is an error reading a .wav file, the switch will not NACK the API message, but will send an informational Card Alarm of 0x4E in the Alarm 0x00B9 message:

Example 1: No RIFF Chunk Found (Alarm Subtype 0x00)

```
X->H
   [00 19 00 b9 00 0c ff 01 02 4e 10 00 01 01 01 04 00 01
   00 00 17 70 00 42 41 44 20]
```

Example 2: RIFF type .wav not found (Alarm Subtype 0x01)

```
X->H
   [00 19 00 b9 00 0e ff 01 02 4e 10 00 01 01 01 04 00 01
   00 00 17 71 01 42 41 44 20]
```

Appending or Replacing Recorded Files

To append or replace a file, use the optional Append or Replace TLV (0x0615) in the Record File Start 0x011E message (0x011E).

Call Processing Events

There is a four-byte data field named Offset in the following Call Processing Events:

- Play File Started
- Play File Completed
- Record File Started
- Record File Completed

In the *Play File Started* and *Record File Started* events, the Offset represents the starting byte. In the *Play File Complete* and *Record File Complete* events, the Offset represents the ending byte.

The application is responsible for managing different encoding formats, and for designating how bytes represent time.

The Record File Started event is issued when the first block of data is written. The DSP Series 2 block size is 1 second, so this event occurs approximately 1 second after the *Record File Start* message is issued.

Configurable Barge-In for Play Files

This feature allows a single call session to have multiple announcements that can perform differently when subscribers press digits. This functionality increases the flexibility for application development.

This feature does not apply to conferences or child conferences.

API Call Control Messages

The following messages support this feature:

- Play File Start (0x011B)
- DSP Service Request (0x00BD)

Description

During a single call session, an Interactive Voice Response Service (IVRS) calls various announcements during the call flow. A single call session has some announcements that will terminate when the subscriber presses digits and other announcement that will not terminate when the subscriber presses digits.

Because each announcement can act differently during a single call session, this behavior is not controlled by the DSP Service Request message with service type 0xE1 (DTMF Receiver with Termination of Tone/Announcement).

Instead, the Play File Start (0x011B) message is used with the DSP Service Request (0x00BD) message to control the barge-in behavior for play file to channel.

Enabling this Feature

The Play Continue on Digit Detection (0x0751) TLV supports this feature. Refer to TLV chapter in the *API Reference*.

The *Play File Start* message for playing to a channel includes the Play Continue on Digit Detection (0x0751) TLV set to 1 (Continue play on digit detection). In addition, attach the DTMF receiver using the service type 0xE1. Even if the digits are detected, the play file is not cancelled.

If you do not want to enable this feature, set the Play Continue on Digit Detection TLV to 0 (Cancel on digit detection). Attach a DTMF receiver using service type 0xE1. The play file is cancelled when digits are detected.

Conferencing

Contents

Creating a Conference

Feature Description

See Unified Conference.

Pre-requisites

You must have DSP Chips configured for the conferencing type you are creating.

Procedure

1. Create the Conference

   Use the Resource Create 0x0124 message to create the conference.

   - AIB

     Use the DSP Chip AIB to indicate the DSP Chip to be used for the conference. The DSP Chip selected must be configured for the Unified Conference (0x22) Function Type.

   - Mandatory TLVs

     0x0602 Resource Type: 0x0100 - Conference

     0x0603 Channel/DSP Pool -select the appropriate pool of available DSP resources for the conference

   - Optional TLS

     Use optional TLVs to configure conference features and options.

     See Conferencing Features TLVs

     See Conferencing Options TLVs.

2. Add Conferees

   Add Conferees using the Resource Connect 0x0127 message.

Manual Input Volume Control

This feature enables a manual volume control on a parties inbound leg to a conference. The Automatic Gain Control for that party is automatically disabled, and the requested gain is added to the conferee's speech prior to being added to the conference.

You implement the Manual Volume Control feature using the 0x068B Input Gain Control TLV in the following messages:

- *Resource Create* (0x0124)
- *Resource Modify* (0x0125)
- *Resource Connect* (0x0127)

Querying Active Speakers

To query the active speakers of a conference, use the *System Configuration Query* message (0xB4) with the Conference ID AIB (0x55) and the Conference Speakers (0x13) query type. The *Generic Report* (0x46) message is returned with the conference data.

Creating a Child Conference

Feature Description

See Child Conference.

Dependencies and Considerations

- If a party is removed from the parent conference, it will also be removed from the child conference if it is part of any.
- A party can be in only one child conference at a time.
- Only parties that are connected 2-way in a parent conference can be moved into a child conference.
- A child conference inherits its conferencing features configuration from the parent conference at the time of its creation. You can modify the Conferencing Features configuration of the Child Conference with the Resource Modify 0x0125 message.
- The volume of parent conference getting recorded to a child conference is designed to be low. Playing the parent conference at a quieter volume allows monitoring of the main thread of conversation without it overwhelming the child conference.

Pre-requisites

You must already have a conference created (Parent Conference) from which to create the Child Conference.

Procedure

1. Create Child Conference

   Use the Resource Create 0x0124 message to create the Child Conference.

   - AIB:

     Use the Child Conference AIB. In the AIB you indicate the Parent conference ID. The response to the Resource Create 0x0124 message will indicate the new Child Conference ID, which you then use when adding conferees to the Child Conference using the Resource Connect 0x0127 message.

   - Mandatory TLV

     0x0602 Resource Type: Child Conference (0x0107)

   - Optional TLVs

     Use optional TLVs to override default conference parameters if necessary.

     See Conferencing Features TLVs

     See Conferencing Options TLVs.

2. Add Conferees

   Use the Resource Connect 0x0127 message to add conferees to a child conference.

Removing Party from the Child Conference

Use the Resource Disconnect 0x0128 message to move a channel from a child conference back into the parent conference. The *Resource Disconnect* message uses the Channel AIB (0x0D) and the Child Conference ID AIB (0x45).

Deleting a Child Conference

Use the Resource Delete 0x0126 message (0x0126) to delete a Child Conference. The *Resource Delete* message uses the Child Conference ID AIB (0x45) to address the child conference to be deleted.

The *Resource Delete Indication* message (0x0129) indicates that the Child Conference has been deleted.

- When a child conference is deleted forcibly, the channels connected to it are connected back to the parent conference and the child conference is immediately deleted.
- When a child conference is deleted gracefully, the CSP waits until all the parties connected to the child conference are either removed or reconnected back to the parent conference before deleting the child conference. The default behavior is graceful deletion.
- When a parent conference is deleted forcibly, all parties connected to the parent and the associated child conferences are removed, and the parent and all its child conferences are immediately deleted.
- When a parent conference is deleted gracefully, the CSP waits until all parties in the parent and the associated child conferences are removed. When all parties connected to a child conference have been removed or reconnected back to the parent conference, the child conference is deleted. When all the child conferences have been deleted and no parties are left in the parent conference, the parent conference is deleted.

Creating a Monitor Conference

Feature Description

See Monitor Conference.

Pre-requisites

You must have a DSP Chip configured for Monitor Conference.

Considerations

- Channels can be connected to the conference after call setup. It is not necessary to know at setup time all of the callers that will eventually connect to the conference.
- The call processing states of the source channels added to a monitor session are irrelevant. A source channel can be added to a monitor session with no effect on its current connection state. Subsequent state transitions of the channel have no effect on its association with the monitor conference.
- The source channels connected to a monitor conference do not have to be associated with one another in any way.

Procedure

1. Create the Conference

   Use the Resource Create 0x0124 message to create the conference.

   - Mandatory TLVs

     0x0602 Resource Type: 0x0100 - Conference

     0x0603 Channel/DSP Pool: Select desired option.

     0x0606 Monitor Conference Enable: 0x0001 - Enable

   - Optional TLS

     Use optional TLVs to configure conference parameters.

     See Conferencing Features TLVs

     See Conferencing Options TLVs.

2. Add Conferees

   Add Conferees using the Resource Connect 0x0127 message.

   When adding the monitor channel, use the 0x0612 Connection Type TLV to indicate a one-way connection (0x01).

Multiple Sessions

To create multiple sessions, you can use the source ports on a DSP resource, in any combination.

For example, a DSP chip that is configured for the Monitor Conference function can provide the following:

- Three 9-port sessions and one 7-port session
- Six 5-port sessions and one 4-port session
- 17 two-port sessions

Deleting a Monitor Conference

You can delete a Monitor Conference by sending a Resource Delete 0x0126 message. Deletion can be either Forced or Graceful:

Forced Deletion

To release all channels in the conference and delete the conference using the Resource Delete 0x0126 message with the Forced Flag byte set to 0x01.

Graceful Deletion

To gracefully delete a Monitor Conference, release all monitor channels in the conference using the *Release Channel* message and then delete the conference using the Resource Delete 0x0126 message with the Forced Flag byte set to 0x00. All source channels are then automatically removed from the conference and the conference is gracefully deleted.

If you remove only the source channels from the Monitor Conference, the conference is not deleted. The conference retains its Conference ID and can be re-used.

Reconfiguring a Monitor Conference

To reconfigure an existing conference with new source channels, park the monitor channels using the Park Channel 0x00BF message. Removing the monitor channels from the conference (either by parking or releasing) automatically removes all of the source channels.

If a Resource Delete 0x0126 message that specifies graceful deletion has been sent, removing the monitor channels from the conference deletes the conference. When the source channels are removed, the monitor channels can then be reconnected to the conference, and new source channels connected using the same Conference ID number.

Configuring Conferencing Features

Feature Description

See Conferencing Features.

Overview

You set conferencing features using TLVs in the following messages:

- Resource Create 0x0124
- Resource Connect 0x0127
- Resource Modify 0x0125

Conferencing Features TLVs

Parameter	Notes	TLV(s)
Automatic Gain Control (AGC)	Automatic Gain Control (AGC) automatically detects variations in volume input and adjusts the input level to be within a decibel range that you can specify, in 1 dB increments. If a conferee's voice input (one conference leg) is either too quiet or too loud, AGC adjusts the input so that it falls within your specified acceptable range. Default = Disabled See Automatic Gain Control. **NOTE**: You can implement the Manual Volume Control feature to override the dB of a conferee using the 0x068B Input Gain Control TLV. See Manual Input Volume Control.	0x060C Automatic Gain Control Enable 0x060E Automatic Gain Control Parameters 0x060D Automatic Gain Control Input Level 0x068B Input Gain Control

Parameter	Notes	TLV(s)
Noise Gating	During natural breaks in conversation (when the speaker on a channel is not speaking) if significant ambient noise is coming into a conference from the speaker's line, that noise can be gated using the Noise Gating feature. As a conference grows in size, ambient noise is more likely to become a problem. You can customize the time span (time constant) over which noise is measured, in increments of 5 ms. You can also specify a maximum allowable estimated noise level in 1 dBm increments, from –54 dBm to –10 dBm. Defaults Disabled Time Constant = 35 Maximum Noise Level = -20 (1 dBm steps) Noise Gating Sensitivity = 3 See Noise Gating	Noise Gate Enable TLV (0x0608) Noise Gate Parameters TLV (0x0609)
Output Gain Control	You can adjust gain from –40dB to +10dB, in increments of 1dB. The CSP acts upon the most recent command. For example, if a change is made at the conference leg level and then at the conference level, the conference level setting overrides the leg level setting because that was the last change made. Default = 0db See Output Gain Control.	Output Gain Control (0x0607)
Echo Suppression	If a conference leg is generating echoed signal, you can mute that echoed signal using the Echo Suppression feature. Note how this differs from echo cancellation. Echo Suppression adaptively measures the reflected signal input, and adapts its echo estimate up to a maximum noise value, which you can select in 1 dBm increments, from –54 dBm to –10 dBm. Defaults Disabled Echo Return Loss (1 db steps) = -10 See Echo Suppression	Echo Suppression Enable (0x060A) Echo Suppression Parameters TLV (0x060B)

Parameter	Notes	TLV(s)
Conference Failure Behavior	You can configure a conference so that if the conference fails all the channels are parked. The application is responsible for rebuilding the conference on another DSP resource or for releasing the conference legs. Default = Purged	0x060F Conference Failure Behavior

Parameter	Notes	TLV(s)
Channel Selectable DTMF Tone Suppression in a Conference	*Resource Modify* (0x0125) message This TLV is used to modify parameters for channels connected to a conference or child conference. If TLV is set to DTMF Clamping enabled, the DTMF tones from that channel will be suppressed. If TLV is set to DTMF Clamping disabled, the DTMF tones from that channel will be heard by other channels in the conference. If TLV is not present in the message, the DTMF clamping behavior will not be modified. *Resource Connect* (0x0127) message This TLV is used to connect channels to a conference, or to move channels from a parent conference to a child conference. If used with Conference ID AIB, it connects channels to a conference. If used with Child Conference ID AIB, it moves channels from a parent conference to a child conference. If TLV is set to DTMF Clamping enabled, the tones from that channel will be suppressed. If TLV is set to DTMF Clamping disabled, the tones from that channel will be heard by other channels in the conference. If TLV is not present in message, the default behavior will be determined by the conference/child conference type, whether it is a DTMF-Clamped conference or not. *Resource Disconnect* (0x0128) message This TLV, with Child Conference ID AIB, is used to move a channel from a child conference back to the parent conference. If the TLV is set to DTMF Clamping enabled, the DTMF tones from that channel will be suppressed. If the TLV is set to DTMF Clamping disabled, the DTMF tones from that channel will be heard by other channels in the conference. If the TLV is not present in the *Resource Disconnect* message, the default behavior will be determined by the parent conference type, whether it is a DTMF-Clamped conference or not.	0x0604 DTMF Clamping/ Filtering Enable (Optional)

Parameter	Notes	TLV(s)
Configurable Conference Connection Mode	When the Configurable Conference Connection Mode feature is enabled, and a transmit tone request is received from the host for a channel that is connected to a conference, the following occurs: • The transmitter (host) temporarily suspends the output bus (SLD) while the conference receives any incoming voice traffic on the input bus (LSD) to be recorded. • After the call progress tone/file is complete, addition of the channel input and output buses to the conference is resumed. • Incoming voice traffic is enabled. *Generic Report* (0x0046) message The *Generic Report* (0x0046) message, generated by the CSP following the *System Configuration Query* (0x00B4) message with Query Type 0x14 (conferencing features config query), includes the two byte Data[30-31] Transit Connection Mode for the Conference ID queried. *Resource Create* (0x0124) message The optional TLV 0x068D Transit Connection Mode must be included in the Resource Create message while creating a conference to change 0x0000: Remove Leg (Default) to 0x0001: Connect Input Only. *Resource Connect* (0x0127) message The optional TLV 0x068D Transit Connection Mode must be included in the Resource Connect (0x0127) message while creating a conference to change 0x0000: Remove Leg (Default) to 0x0001: Connect Input Only.	Transit Connection Mode TLV (0x068D)

Configuring Conferencing Options

Feature Description See Conferencing Options.

Conferencing Options TLVs

TLV (Optional unless Noted)	**Notes/Values** * = default
0x05E5 Barge In	By default, Barge-In is enabled for conferencing. Include this TLV to disable it. 0x0000 - Disabled 0x0001 - Enabled *
0x0604 DTMF Clamping/Filtering Enable	Use this TLV to enable DTMF filtering. 0x0000 - Disabled* 0x0001 - Enabled
0x0605 EXS Conferencing Encoding Type	Use this TLV to specify the encoding type for a conference over the EXNET® ring. 0x0000 - u-law 0x0001 - A-law 0x0002 - Mixed *

Playing Files into a Conference

Overview

You can play a file into a conference using the *Play File Start* message.

Modifying a File Being Played

To modify a file that is playing to a conference, use the Conference AIB in the *Play File Modify* message and, optionally, the File ID TLV. If you use the File ID TLV, all instances of that File ID being played in the specified conference are modified. If the File ID TLV is absent, all the files playing into that conference are modified.

Stopping a File

To stop a file that is playing to a conference, use the Conference AIB and File ID TLV in the *Play File Stop* message. If the File ID TLV is absent, all the files playing into that conference are stopped.

Playing Multiple Files to a Conference

You can simultaneously play multiple files into a conference. The same File ID can be played multiple times. Any subsequent action on the playing file (such as Modify or Stop) identifies the file using a combination of the Conference ID and the Play File ID. Therefore, if you modify or stop a File ID, you cause the change to happen to all instances of that File ID being played in the specified conference.

File queues are not permitted when playing multiple files into a conference. Multiple file plays are not permitted when queued files are being played.

Resource Points

When playing multiple files into a conference simultaneously, standard Play File resource points are used for each Play File.

Tones

Contents

Digit Collection

DSP Service Request

Use this message to allocate an appropriate digit receiver and attache it to the specified channel. The attached digit receiver uses the Call Processing Event 0x002E message to report digits as they are decoded, one at a time. The receiver remains attached to the channel until the channel is released or until the host cancels digit collection, with the *DSP Service Cancel* (0x00BE) message.

Collect Digit String

Use this message to allocate and attache a digit receiver to a channel. Unlike the *DSP Service Request* message, this message causes the CSP to collect a group of digits in sequence until termination condition occurs. The termination condition could be the detection of a particular digit from a termination digit set, a fixed number of digits collected, or a timeout.

Digit Collection After Call Setup

After a call is set up, the CSP collects DTMF digits interactively. The host collects additional information, either one digit at a time with the *DSP Service Request* message, or as a sequenced group of digits with the *Collect Digit String* message.

Collecting Inpulse Data on Specified Channels

Use the *Inpulsing Parameters Configure* message. Inpulsing parameters define the address signaling type, the number of digit strings, and the collection method used during call setup.

The CSP supports four different inpulsing stages, each with one or two digit strings. When the host is instructing the CSP to collect address signaling information (that is typically presented with in-band dual frequency tones) it also specifies a preprogrammed inpulsing stage that describes how to perform the digit collection.

The inpulsing stage configuration options include the following:

- Address signaling type (DTMF, MFR1, MFR2)
- Number of strings (1 or 2)
- String collection method (fixed number digits, KP/ST framed, compelled)

Receiving Address Signaling Tones

Use the The *Inseize Control* message passes inseize instructions to the CSP for controlling incoming call setup in real time. The instruction, “Receive Stage N Address Data” allocates and attaches a DSP resource to a channel to collect an incoming digit stream.

Up to 100 digits can be collected within a given inpulsing stage. Use the *Inseize Instruction List Configure* (0x0029) message to preprogram instructions on a channel. See “Digit Collection During Call Setup” for more information. The instruction, Report Incoming Call With Address Digits, sends a *Request For Service With Data (*0x002D) message to the host, with single- or multiple-stage digit streams collected.

When the termination condition occurs, the CSP reports the entire group of digits collected, with the Call Processing Event 0x002E message. The receiver is then returned to the system receiver pool. If the configuration bit 4 is not set (or "by default") when the CSP detects the first valid digit, it automatically cancels any prompting tone or recorded announcement being played out to that channel.

Generating Call Progress Tones

Configuration

1. If required, customize tomes using the *Transmit Cadence Pattern Configure* message.
2. Configure a DSP with one of the following function types using the DSP SIMM Configure 0x00C0 message:

   0x30 - Universal Tone Generation µ-Law

   0x31 - Universal Tone Generation A-Law

Procedure

1. Connect the tone generator to the channel using the *Connect Tone Pattern* message.

   A call must be active (either incoming or outgoing) for a pattern to be generated.

   If you want to be notified with a Call Processing Event 0x002E message (0x21 - Tone Complete) when tone transmission has stopped, set the *Generate Event Flag*.

Performing Call Progress Analysis

Configuration

You must have a DSP configured with a Function Type for CPA:

0x07 - CPA A-Law

0x08 - CPA µ-Law

Initiate

You can initiate call progress in the following two ways:

- During call setup, with the outseize instruction of Do Call Progress Analysis
- Interactively, by assigning a CPA Receiver with the *DSP Service Request* message using the following values:

  Service Type

  0x03 - Call Progress Analysis Receiver

  CPA Class

  0x00 North American Default

  0x01 Dialtone

  0x02 CPC Detection

  0x03 Energy Detection

Host Indications

A *Call Progress Analysis Result* message is sent to host when a pattern is detected.

A Call Processing Event 0x002E message is sent to host when a CP pattern is detected

Off-hook Detection

When CPA is initiated with the *DSP Service Request* message, detection of off-hook is reported if the signaling interface supports it.

Outpulsing Tones

Outpulse Address Signaling Tones

For outbound calls, configure the CSP to generate DTMF, MFR1, or MFR2 address signaling to the downstream CSP or device. To initiate outpulsing, use the *Outseize Control* (0x002C) message with an action of “Outpulse Stage N Address Data.” Data ICBs in the *Outseize Control* (0x002C) message indicate the signaling tone type and the source of the digits. Digits can be specified in the *Outseize Control* (0x002C) message itself, or they can be taken from a list of previously-inpulsed digits. Digit durations are set by the PPL component on the associated line card.

Example

You set the following durations using PPL timers, and you modify them using the *PPL Timer Configure* message (0x00CF):

delay until the start of the first digit

- “on” time of the first digit
- “on” time of subsequent digits
- "off” time between all digits

For DTMF and MFR1 digits, outpulsing can also be initiated using the *Outpulse Digits* (0x0020) message. Within the message, you can specify the following durations:

- delay until the start of the first digit
- “on” time of the first digit
- “on” time of subsequent digits
- “off” time between all digits

Durations are specified in units of 10 milliseconds. Actual outpulsed durations, however, are truncated to 20 millisecond intervals. So a digit duration specified as a multiple of 20 milliseconds is outpulsed with that duration, while other digit durations are shortened.

For example, digits with a specified duration of 20, 40, or 60 milliseconds are outpulsed with those durations. But a digit with a specified duration of 30 milliseconds is truncated to have a 20 millisecond duration upon outpulsing.

Similarly, 50 milliseconds is outpulsed as 40 milliseconds, 70 milliseconds is outpulsed as 60 milliseconds, and so on. Because of the truncated durations, you should set a minimum outpulse duration for 20 milliseconds or greater.

The host dynamically configures the dBm levels of DTMF, MFR1, and MFR2 transmit tones, using the *Tone Configure* message (0x0031). Power levels are updated globally for each tone type. For DTMF tones, you must specify the dBm level for both low-band and high-band frequency components. For MFR1 and MFR2 tones, you must specify only one dBm level.

Off-hook Detection

When CPA is initiated with the *DSP Service Request* message, detection of off-hook is reported if the signaling interface supports it.

For example, you can configure the CPA receivers to detect a fax machine that the CSP has called. When the connection is made, the fax machine sends back a pattern containing Tone 0x0E (2100 Hz) continuously for approximately 2.3 seconds.

To configure the CPA receivers to detect this signal:

1. use the *CPA Pattern Configure* message to create a new pattern, consisting of Tone 0x0E continuously **ON** for a minimum of 2 seconds.
2. Add this pattern to a CPA Class using the *CPA Pattern Configure* message (described next). Whenever CPA is performed using that CPA Class, the CPA receiver scans for the fax signal.

Customizing Tones

Feature Description

See Tones in the DSP Series 2 Card Product Description chapter for more information.

Overview

Many telephone networks require call progress tones with different patterns of frequencies, at different intervals. For example, the CSP may need to provide a different cadence of the Ringback Pattern based on the called party address. To ensure network compatibility, you may need to adjust the attributes of the call progress tones.

Interval Durations

Interval durations are specified in units of 10 milliseconds. However, actual transmitted durations are truncated to 20 millisecond intervals. So an interval duration specified as a multiple of 20 milliseconds is transmitted with that duration, while other interval durations are shortened.

For example, intervals with a specified duration of 20, 40, or 60 milliseconds are transmitted with those durations. But an interval with a specified duration of 30 milliseconds is transmitted with a 20 milliseconds duration, an interval with a specified duration of 70 milliseconds is transmitted with a 60 milliseconds duration, and so on. Because durations are handled this way, your shortest interval duration should be 20 milliseconds.

Restoring Default Settings

To restore default settings for the transmit tone parameters, send the *Reset Configuration* (0x000B) message, or power down both Matrix Controller cards, and then power them up again.

Customizing Call Progress Tone Detection

Overview

The CSP allows the host to modify both the tones and the cadence of call progress tone patterns. When you modify a tone or a pattern, you affect all DSP chips that are configured to receive it.

You can modify the default settings for any Tone ID, using the *Tone Configure* message. Changing the specifications of a Tone ID affects all patterns in the CSP that are using that Tone ID.

The default parameter values for each call progress analysis tone pattern are shown in the Specifications section in the DSP Series 2 Card Product Description chapter. To modify these values, use the *CPA Pattern Configure* message.

Modifying a Call Progress Receive Tone:

You specify only the frequencies in the tone.

For example, to configure a CP Receive tone made up of the frequency 400 Hz, use the *Tone Configure* message to:

1. Set the number of frequencies making up the tone (Data 2) to 1.
2. Set the first frequency value to 400. You do no need to set any other frequency values

To delete a Call Progress Receive Tone, use the *Tone Configure* message and set the number of frequencies to 0. No frequency data needs to be sent.

Creating a New Pattern

You can modify patterns and create new patterns, using the *Call Progress Analysis Pattern Configure* message.

1. Modify the Tone IDs in the pattern if required.
2. Assign a unique Pattern ID, within the range of 0x01 – 0x1F. You must maintain an accurate list of current Pattern IDs and their specifications.
3. Add the pattern to a class with the *Call Progress Analysis Class Configure* message.

To minimize reconfiguration time, set the *Update All Flag* in only the last configuration message that you send.

Example

Assume the CSP is connecting to a fax machine. When the connection is made, the fax machine continuously sends back a pattern containing Tone 0x0E (2100 Hz) for approximately 2.3 seconds.

To configure the CPA receivers to detect this signal

1. Use the *CPA Pattern Configure* message to create a new pattern consisting of Tone 0x0E continuously **ON** for a minimum of 2 seconds.
2. Add this pattern to a CPA Class using the *CPA Pattern Configure* message.

Whenever CPA is performed using that CPA Class, the CPA receiver scans for the fax signal.

Restoring Default CPA Settings

To restore default settings for the transmit tone parameters, send the *Reset Configuration* message to the Matrix Controller, or power down both Matrix Controller cards, and then power them up again.

Example: Customizing a Pattern ID

Overview

The following example shows how to configure Pattern ID 0x17 to receive a three-second, 400 Hz tone. To perform this task, send three API messages as follows:

1. Send the *Tone Configure* message to replace the default tone (Tone ID 0x0F, 425 Hz) with the new, 400 Hz tone.

   Tone Type: 0x06

   Action: 0x01

   Data[0] Tone ID being changed: 0x0F

   Data[2] Number of frequencies in tone: 0x01

   Data[4] Frequency Value[0], Hz, MSB: 0x01

   Data[5] Frequency Value[0], Hz, LSB: 0x90

2. Send the *CPA Pattern Configure* message to create the new Pattern ID 0x17.

   Update All Flag: 0x01

   Pattern ID: 0x17

   Action: Add/Replace Pattern: 0x01

   Data[0] Pattern ID to report on detection: 0x17

   Data[1] Tone Group ID: 0x00

   Data[2] Pattern Configuration: 0x01

   Data[3] CPA Report on Pattern Loss: 0x17

   Data[4] Interval Cycles to Match: 0x01

   Data[5] Interval Cycles to Report: 0x01

   Data[7] Interval Descriptor Count: 0x01

   Data[8] Tone ID: 0x0F

   Data[10] Minimum filter (in 10 ms units) MSB: 0x00

   Data[11] Minimum filter (in 10 ms units) LSB: 0x50

   Data[12] Maximum filter (in 10 ms units) MSB: 0x00

   Data[13] Maximum filter (in 10 ms units) LSB: 0x00

3. Send the *CPA Class Configure* message to add Pattern 0x17 to the CPA class 0x00.

   Update All Flag: 0x01

   Class ID: 0x00

   Action: 0x01

   Data[0] Pattern ID: 0x17

Configuring Energy Detection

Feature Description

See Energy Detection.

Overview

Energy Detection allows the CSP to perform Call Progress Analysis for frequencies not supported by default. The Energy Detection DSP function matches cadences, based on the reported energy levels. CPA Class 3 is preconfigured for Energy Detection, using the standard CPA tones of ringback, double ringback, busy, and re-order.

To accommodate unique requirements for matching cadences, you can modify these patterns or add new patterns to the class and change the pattern cadence that energy detection scans for, using the *CPA Pattern Configure* message.

Procedure

Use the Call Progress Analysis Class Configure 0x00B3 message to configure the parameters for energy detection:

1. Set Sensitivity Level

   The Sensitivity Level is the amplitude above which the energy detector perceives a signal. Set the sensitivity level to detect and report energy that is greater than the prevailing background noise. When you expect a significant background noise, use the least sensitive setting (0 dBm). When you expect little background noise, use the most sensitive setting (-30 dBm). See Table 8-2 for sensitivity options.

   Table 8-2 shows the values you would enter in the Call Progress Analysis Class Configure 0x00B3 message.

Table 7-2 Set Sensitivity Level

Message Field	Values
Class ID	Class to be changed
Action	0x03 (Change Class Parameter)
Data[0] Parameter	0x0B (Mode Specific 1)
Data[1] Reserved	0x00
Data[2] Sensitivity Level	See message for values.

2. Set the Scan Duration

The Scan Duration is the repeating time interval over which the energy detector determines that energy is either Present or Not Present. Set the scan duration to be longer than expected energy bursts. You must use 20 millisecond intervals to set the scan duration, so energy is sampled for each 20 millisecond block within the specified duration.

Table 8-3 shows the values you would enter in the Call Progress Analysis Class Configure 0x00B3 message.

Table 7-3 Scan Duration Fields

Message Field	Values
Class ID	Class to be changed
Action	0x03 (Change Class Parameter)
Data[0] Parameter	0x0C (Mode Specific 2)
Data[1,2] Scan Duration	16-bit word defining the scan duration, in 10ms units

3. Set the Completion Timer

   The completion timer determines the maximum amount of time to scan for energy. Each scan cycle, as defined by the scan duration, is repeated until either energy is detected, or the completion timer expires. Dialogic recommends setting the completion timer for 3 to 4 times the scan duration, depending on the application. If the timer expires before energy is detected, the host receives a Call Processing Event 0x002E message of “No Energy Detected.”

Configure CPA Class for Energy Detection

You can configure the default CPA classes for Energy Detection by enabling it in the Call Progress Analysis Class Configure 0x00B3 message, as shown in Table 8-4.

Table 7-4 Changing mode parameters

Message Field	Values
CLASS ID	Class to be changed
ACTION	0x03 (Change Class Parameter)
DATA[0] Parameter	0x01 (Mode)
DATA[1] Reserved	0x00
DATA[2] Mode	0x02 (Energy Detection enabled)

Invoking Energy Detection

Overview

You can invoke Energy Detection in the following two ways:

- Interactively, with the *DSP Service Request* message
- As part of Call Progress Analysis

Interactive Energy Detection

You can invoke Energy Detection interactively, with the DSP Service Request 0x00BD message. Use the data bytes of the message to configure the sensitivity level, reporting mode, scan duration, and completion timer.

You also set how detected energy is reported to the host in a Call Processing Event 0x002E message. There are two modes:

- Report Initial Energy Detection Only
  The host receives a *Call Processing Event* of "Energy Result Report" with Data[0], indicating Energy Detected. Data[1] indicates the duration of the period of no energy. The DSP resource is then automatically released.
- Report All Energy Threshold Crossings

  All of the following are reported: the initial energy detection, all subsequent changes, and the duration of the previous state's ON or OFF interval.

  When energy (A) is first detected, the host receives a *Call Processing Event* of "Energy Result Report" with Data[0] indicating "Energy Detected." Data[1] indicates the duration of the preceding period of no energy (D1).

  When energy is no longer detected (B), the host receives a *Call Processing Event* message of "Energy Result Report" with Data[0] indicating "No Energy Detected." Data[1] indicates the duration of the preceding period of energy (D2).

Echo Cancellation

Contents

Overview

Introduction

The DSP Series 2 card Echo Canceller removes echoes caused by signal leakage at hybrid telephone line interfaces. You may want to implement an Echo Canceller for tandem calls on trunks with echo, or to clean an incoming signal before connecting to a media resource, such as a VRU or AMD.

Call Processing Event

The following event is sent in a Call Processing Event 0x002E message when a G.164 tone disabling event occurs on the near or far end.

Event: 0x49 - Echo Canceller Event

Alarm

A General Alarm (0x41) is sent in the Alarm 0x00B9 message when the Host requests to attach a receiver to a channel that already has an Echo Canceller attached but there is no resource available on the same slot as the Echo Canceller resource. This means that the receiver will not receive the output of the Echo Canceller, but will receive the original signal with any echo that may be present.

***Resource Modify* NACK**

The following response status values may be returned for the Resource Modify 0x0125 message:

0x1708 - No Active Echo Canceller

Returned when the switch receives a *Resource Modify* message with the Resource Type of Echo Canceller and there is no Echo Canceller attached to the Channel.

0x1709 - Invalid Echo Canceller State

Returned when the switch receives a *Resource Modify* message with the Resource Type of Echo Canceller and there is no Echo Canceller attached to the Channel.

Configuring Echo Cancel

Procedure

1. Configure DSP for Echo Cancel function using the DSP SIMM Configure 0x00C0 message.

   DSP Function Type

   Echo Cancel (0x33)

2. Set card-level Echo Cancel parameters using the Generic Card Configure 0x0122 message.

   Mandatory TLV

   0x05FA Card Object

   Object ID: Echo Cancel Parameters (0x0005)

   Optional TLVs

   0x0673 Echo Cancel Tap Length
   0x0674 Echo Cancel NLP Type
   0x0675 Echo Cancel ADAPT
   0x0676 Echo Cancel Bypass
   0x0677 Echo Cancel G.176 Modem Answer Detection
   0x0678 Echo Cancel NLP Threshold
   0x0679 Echo Cancel CNG Noise Threshold

Implementing Echo Cancel on a Tandem Call with Positive Voice Detection

Introduction

The following procedure details the steps for attaching an Echo Canceller to a conference leg, and then recording the output.

Prerequisites

You must have a DSP chip configured with the Echo Cancel function. See Configuring Echo Cancel.

Guidelines

- The Echo Cancel function must be on the same DSP chip as the channel(s) to which you are connecting it.
- You must attach the Echo Canceller to the channel before you connect the channel to any other DSP receiver.

Procedure

1. Connect Channels using the Connect 0x0000 message.
2. Attach Echo Cancel receiver using the Resource Connect 0x0127 message.

   AIBs

   A: Span/Channel (channel to attach Echo Cancel function)

   B: Slot (of DSP card).

   If using an additional DSP receiver, it must be on the same card that has Echo Cancel. Otherwise, enter a wild card and the best available card will be used based on resources available.

   Mandatory TLV

   0x0602 Resource Type

   Resource Type = Echo Cancel (0x0108)

   Optional TLVs (to override default card parameters)

   0x0673 Echo Cancel Tap Length
   0x0674 Echo Cancel NLP Type
   0x0675 Echo Cancel ADAPT
   0x0676 Echo Cancel Bypass
   0x0677 Echo Cancel G.176 Modem Answer Detection
   0x0678 Echo Cancel NLP Threshold
   0x0679 Echo Cancel CNG Noise Threshold
3. Attach Positive Voice Detection receiver to channel using the Resource Connect 0x0127 message.

AIBs

A: Span/Channel (channel to attach PVD function)

B: Slot (same DSP card as Echo Cancel receiver).

Mandatory TLV

0x0602 Resource Type

Resource Type = PVD/AMD (0x0109)

Optional TLVs (to override default card parameters)

4. When call is complete, disconnect Echo Canceller using the Resource Disconnect 0x0128 message.

   AIBs

   Source Channel: Span/Channel

   Mandatory TLV

   0x0602 Resource Type

   Resource Type = Echo Cancel (0x0108)

   Optional TLVs

   None

5. Disconnect PVD receiver using the Resource Disconnect 0x0128 message.

   AIBs

   Source Channel: Span/Channel

   Mandatory TLV

   0x0602 Resource Type

   Resource Type = PVD/AMD (0x0109)

   Optional TLVs

   None

Example Call Trace

```
X->H
    [00 0d 00 40 00 00 ff 00 01 0d 03 00 20 00 00]
H->X
    [00 0d 00 ba 00 01 ff 00 01 0d 03 00 20 00 01]
X->H
    [00 07 00 ba 00 01 ff 00 10]
Park Channel span/channel 20/00
H->X
    [00 12 00 bf 00 02 ff 00 02 0d 03 00 20 00 0d 03 00 20
     00 00]
X->H
    [00 07 00 bf 00 02 ff 00 10]
X->H
    [00 0d 00 40 00 01 ff 00 01 0d 03 00 20 01 00]
H->X
    [00 0d 00 ba 00 03 ff 00 01 0d 03 00 20 01 01]
X->H
    [00 07 00 ba 00 03 ff 00 10]
Park Channel span/channel 20/01
H->X
    [00 12 00 bf 00 04 ff 00 02 0d 03 00 20 01 0d 03 00 20
     01 00]
X->H
    [00 07 00 bf 00 04 ff 00 10]
Connect span/channel 20/00 and span/channel 20/01
H->X
    [00 11 00 00 00 05 ff 00 02 0d 03 00 20 00 0d 03 00 20
     01]
X->H
    [00 07 00 00 00 05 ff 00 10]
Resource Connect 20/00- Echo Canceller
H->X
    [00 17 01 27 00 00 ff 00 02 0d 03 00 20 00 01 01 ff 00
     01 06 02 00 02 01 08]
X->H
    [00 19 01 27 00 00 ff 00 10 00 02 0d 03 00 20 00 01 01
   04 00
     01 06 02 00 02 01 08]

Record File Start 20/00
H->X
    [00 46 01 1e 00 06 ff 00 02 0d 03 00 20 00 01 01 ff 00
     06 05 e0 00 04 00 dc 5b 17 05 e1 00 02 00 01 05 e2 00
     13 00 01 00 02 2f 50 61 74 45 63 68 6f 2f 6d 79 2e 72
     61 77 05 e9 00 02 02 26 05 fb 00 01 ca 05 e6 00 01 0c]
X->H
    [00 11 01 1e 00 06 ff 00 10 00 02 0d 03 00 20 00 01 01
   04]
H->X
    [00 0c 01 20 00 07 ff 00 01 0d 03 00 20 00]
```

```
X->H
    [00 0e 01 20 00 07 ff 00 10 00 01 0d 03 00 20 00]
X->H
    [00 19 00 2e 00 00 ff 00 01 0d 03 00 20 00 35 00 dc 5b
   17 00
     02 00 00 00 04 00 00]
H->X
    [00 05 00 2e 00 00 ff]
```

Implementing Echo Cancel on a Conference Leg

Introduction

The following procedure details the steps for attaching an Echo Canceller to a conference leg, and then recording the output.

Prerequisites

You must have a DSP chip configured with the Echo Cancel function. See Connecting to an ASR (Automatic Speech Recognition) Server.

Guidelines

The Echo Canceller must be attached to the channel before you connect it to the conference or to another DSP receiver.

Procedure

1. Park the channel using the Park Channel 0x00BF message.
2. Create the conference using the Resource Create 0x0124 message.
3. Attach Echo Canceller to channel using the Resource Connect 0x0127 message.

   AIBs

   Resource A: Channel (Input to Echo Canceller)

   Resource B: Slot (of card where conference resides)

   Mandatory TLV

   0x0602 Resource Type

   Resource Type = Echo Cancel (0x0108)

   Optional TLVs (to override default card parameters)

   0x0673 Echo Cancel Tap Length

   0x0674 Echo Cancel NLP Type

   0x0675 Echo Cancel ADAPT

   0x0676 Echo Cancel Bypass

   0x0677 Echo Cancel G.176 Modem Answer Detection

   0x0678 Echo Cancel NLP Threshold

   0x0679 Echo Cancel CNG Noise Threshold
4. Connect Channel to Conference using Resource Connect 0x0127 message.
5. Attach other DSP Resources as required, such as Record Conference Leg using Record File Start 0x011E message.
6. Disconnect Echo Canceller using the Resource Disconnect 0x0128 message.

   AIBs

   Source Channel: Span/Channel

Mandatory TLV

0x0602 Resource Type

Resource Type = Echo Cancel (0x0108)

Optional TLVs

None

Example Call Trace

```
Park Channel span/channel 20/00
H->X
    [00 12 00 bf 00 09 ff 00 02 0d 03 00 20 00 0d 03 00 20
     00 00]
X->H
    [00 07 00 bf 00 09 ff 00 10]
Resource Create - Conference
H->X
    [00 20 01 24 00 0c ff 00 01 22 03 00 ff ff 00 03 06 02
     00 02 01 00 06 03 00 02 00 00 06 13 00 02 00 06]
X->H
    [00 1e 01 24 00 0c ff 00 10 00 01 22 03 04 06 02 00 02
   06 10
     00 04 00 00 fd fe 06 02 00 02 01 00]
Resource Connect 20/00- Echo Canceller, use slot 4
H->X
    [00 17 01 27 00 00 ff 00 02 0d 03 00 20 00 01 01 04 00
     01 06 02 00 02 01 08]
X->H
    [00 19 01 27 00 00 ff 00 10 00 02 0d 03 00 20 00 01 01
   04 00
     01 06 02 00 02 01 08]
Resource Connect 20/00- Connect Leg to Conference
H->X
    [00 1e 01 27 00 0e ff 00 02 0d 03 00 20 00 55 02 fd fe
     00 02 06 02 00 02 01 00 06 12 00 02 00 00]
X->H
    [00 1a 01 27 00 0e ff 00 10 00 02 0d 03 00 20 00 55 02
   fd fe
     00 01 06 02 00 02 01 00]
Record File Start 20/00
H->X
    [00 46 01 1e 00 12 ff 00 02 0d 03 00 20 00 01 01 ff 00
     06 05 e0 00 04 00 dc 5b 17 05 e1 00 02 00 01 05 e2 00
     13 00 01 00 02 2f 50 61 74 45 63 68 6f 2f 6d 79 2e 72
     61 77 05 e9 00 02 02 26 05 fb 00 01 ca 05 e6 00 01 0c]
X->H
    [00 11 01 1e 00 12 ff 00 10 00 02 0d 03 00 20 00 01 01
   04]
```

Media Streaming over RTP

Contents

Overview

Introduction

RTP, the real-time transport protocol, provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality-of- service for real-time services.

Important! The DSP-2 card supports up to 15 RTP sessions.

TLVs Used

The following TLV is used to enable RTP:

0x0687 - Enable RTP for Play/Record File

The following TLVs are used when implementing Positive Voice Detection with RTP.

0x0688 - RTP Record Mode

0x0689 - Start/Stop Sending RTP Packets

0x068A - Calender Time Offset for Sending RTP Packets

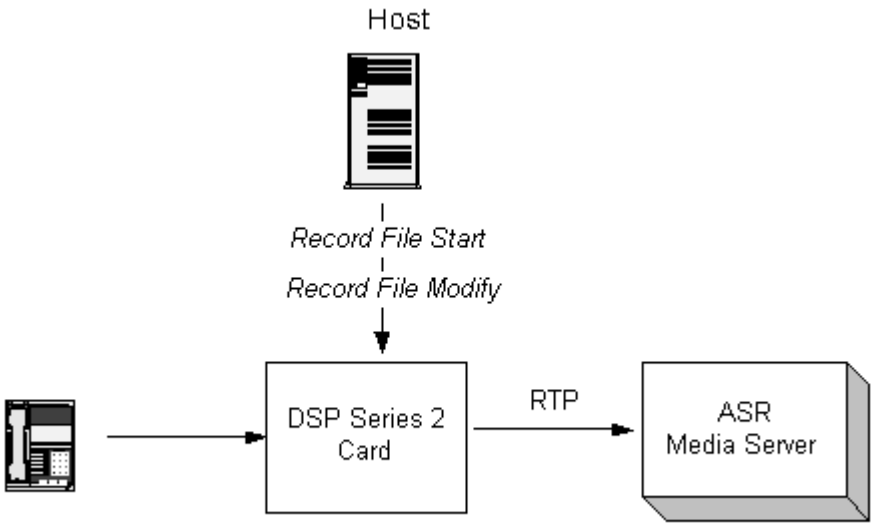
Resource Points

No additional DSP Resource Points are used for Play/Record with RTP beyond whatever are required for Play/Record.

Connecting to an ASR (Automatic Speech Recognition) Server

Introduction

You connect to an Automatic Speech Recognition server over RTP using the Record File Start 0x011E message. If you are also using Positive Voice Detection, you would use the Record File Modify 0x011F message.



Basic Implementation

You would use the Record File Start 0x011E message with the following TLVs:

Mandatory TLVs

- 0x0687 - Enable RTP for Play/Record File (0x01 - Enable)
- 0x29FF Media Local End Point Information
- 0x2A00 Media Remote End Point Information

Nested TLVs (used with 0x29FF and 0x2A00)

- 0x2A01 Media Per Stream Information
- 0x2A07 Media Port
- 0x2A0E Media Connection Address

See About Nested TLVs for Media Streaming

Optional TLVs

- 0x0688 - RTP Record Mode (to use RTP with Positive Voice Detection)

Example Call Trace

```
00 78 01 1e 00 00 ff 00 02 0d 03 00 00 00 01 01 04
00 0b 'Number of TLVs
05 e0 00 04 00 00 00 0a 'File ID Tag
05 e1 00 02 00 01 'File Format
05 e3 00 01 03 'Optional Gain Tag
05 e6 00 01 0c 'Optional File Event Descriptor
05 e8 00 05 00 03 e8 c8 fd 'Optional Beep Tone Parameters
05 e9 00 02 ff ff 'Disable Initial Silence Timer
05 ea 00 02 ff ff 'Optional Final Silence Timer
06 87 00 01 01 'Enable RTP
06 88 00 01 00 'Optional Record Mode Continous Recording
29 ff 00 14 'Local End Media Info
   2a 0e 00 04 87 77 2c 3c 'Local Media Connection Address
      2a 01 00 08 'Per Media Stream Information
         2a 07 00 04 00 00 17 70 'Local Media Port
2a 00 00 14 'Remote End Media Info
   2a 0e 00 04 87 77 2c 32 'Remote Media Connection
   Address
      2a 01 00 08 'Per Media Stream Information
         2a 07 00 04 00 00 17 72 'Remote Media Port
```

Connecting to an ASR with Positive Voice Detection

Introduction

NOTE: This feature is not supported in this release.

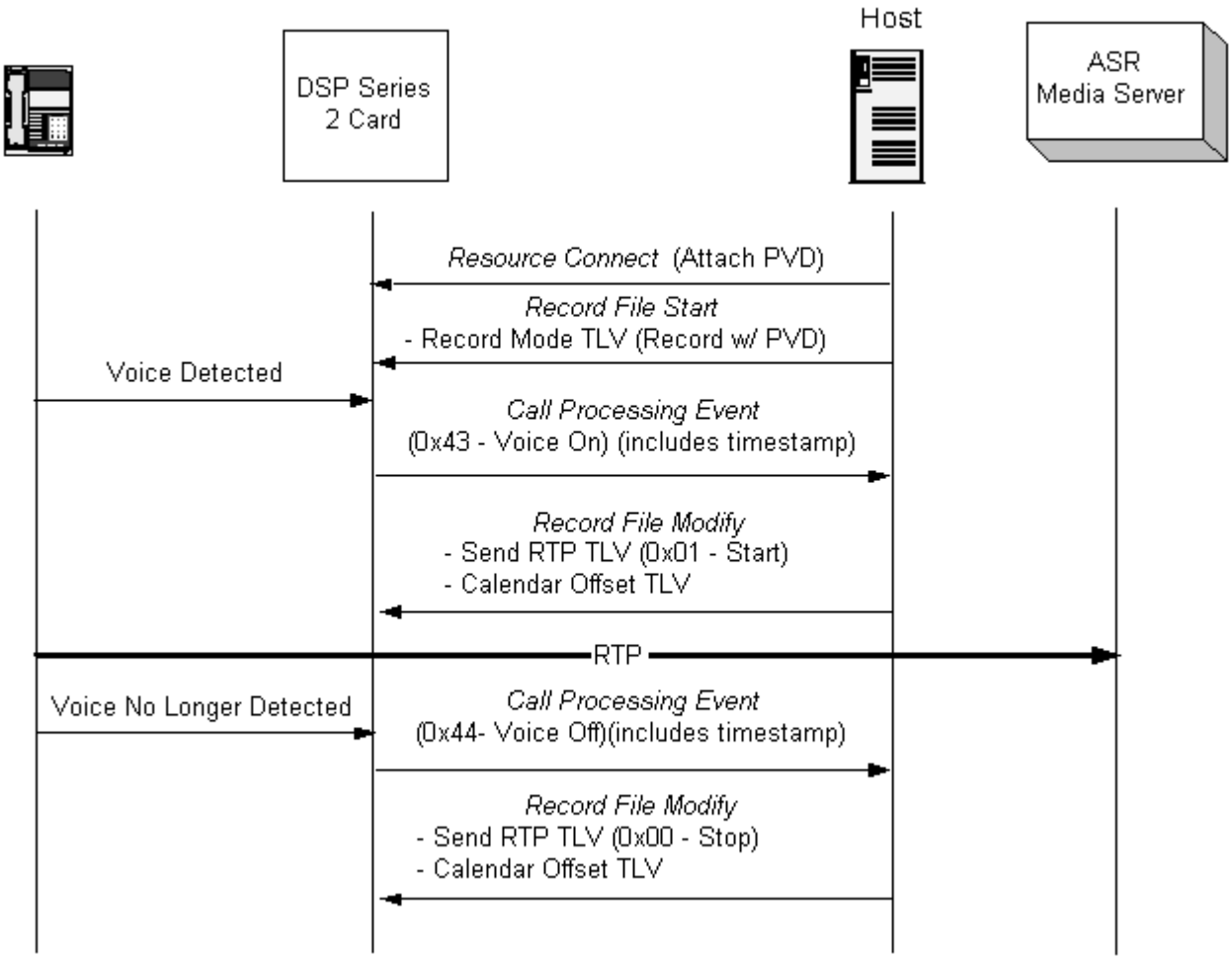
If you have enabled RTP with the Positive Voice Detection (in the Record File Start 0x011E message), you would start/stop sending RTP packets using the Record File Modify 0x011F message with the following TLVs:

- 0x0689 - Start/Stop Sending RTP Packets
- 0x068A - Calender Time Offset for Sending RTP Packets

Pre-requisite

You must have Positive Voice Detection enabled and implemented. See Implementing Positive Voice Detection/Answering Machine Detection.

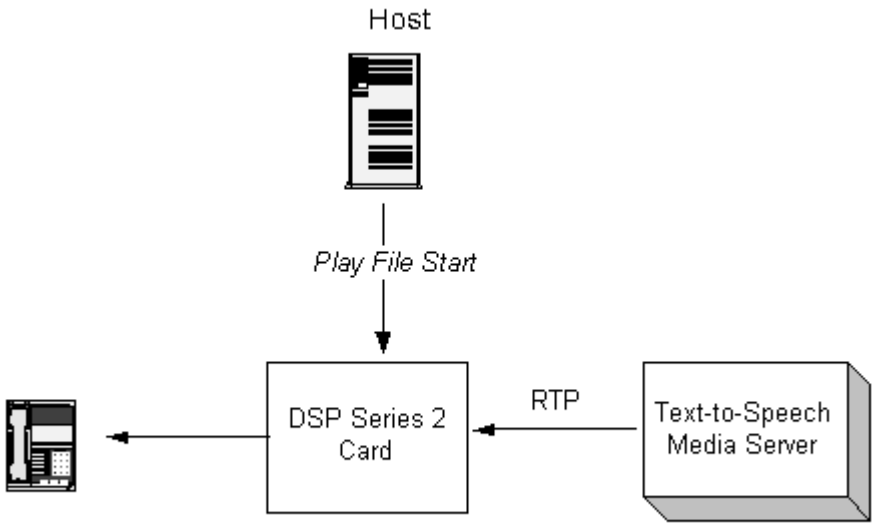
Message Sequence



Connecting to a TTS (Text-to-Speech) Server

Overview

You connect to a Text-to-Speech server over RTP using the *Play File Start* message.



Basic Implementation

You send the Play File Start 0x011B message with the following TLVs:

Mandatory TLVs

0x0687 - Enable RTP for Play/Record File (0x01 - Enable)

0x29FF Media Local End Point Information

0x2A00 Media Remote End Point Information

Nested TLVs (used with 0x29FF and 0x2A00)

0x2A01 Media Per Stream Information

0x2A07 Media Port

0x2A0E Media Connection Address

See About Nested TLVs for Media Streaming

Example Call Trace

```
00 6b 01 1b 00 0b ff 00 02 0d 03 00 00 00 01 01 ff 01
00 08 'Number of TLVs
05 e0 00 04 00 00 00 0a 'File ID Tag
05 e3 00 01 03 'Optional Gain Tag
05 e6 00 01 03 'Optional File Event Descriptor
06 14 00 08 00 00 00 00 00 04 E2 00 'Optional File Offset/
   Length
2a 0a 00 02 00 f0 'Optional Payload Size (3rd party tool
   compatability)
06 87 00 01 01 'Enable RTP
29 ff 00 14 'Local End Media Info
   2a 0e 00 04 87 77 2c 3c 'Local Media Connection Address
      2a 01 00 08 'Per Media Stream Information
         2a 07 00 04 00 00 17 70 'Local Media Port
2a 00 00 14 'Remote End Media Info
   2a 0e 00 04 87 77 2c 32 'Remote Media Connection
   Address
      2a 01 00 08 'Per Media Stream Information
2a 07 00 04 00 00 17 72 'Remote Media Port
```

About Nested TLVs for Media Streaming

The Media Streaming over RTP feature uses nested TLVs. The following provides an overview and example of nested TLVs. An API message can contain nested TLVs. These TLVs are nested into the value field of another TLV in order to capture multiple properties of a call within one TLV.

The following applies to these nested TLVs.

- The TLV Count fields in the API message accounts for parent TLVs but not the nested TLVs. The Length field in any parent TLV counts the bytes in all of its nested TLVs.
- The same TLV can be used multiple times within the same message depending on the context of the nested TLVs.

Example

Play File Start 00 4F 01 1B 00 0B FF 00 02 0D 03 00 00 00 01 01 FF 01 00 **04** (# TLVs)

TLV 1 05 0E 00 04 00 00 00 01 (File ID)

TLV 2 06 87 00 01 01 (Enable RTP for Play File)

TLV 3 29 FF 00 14 (Local End-Point Media Info) (DSP RTP Endpoint)
Nested TLV 1 2A 0E 00 04 87 77 2C 3C (Media Connection Address) DSP IP Address
Nested TLV 2 2A 01 00 08 (Per Media Stream Information)
Nested TLV 3 2A 07 00 04 00 00 17 70 (Media Port) DSP RTP Port

TLV 4 2A 00 00 14 (Remote End-Point Media Info) (Speech Box RTP End-Point)
Nested TLV 1 2A 0E 00 04 87 77 2C 24 (Media Connection Address) DSP IP Address
Nested TLV 2 2A 01 00 08 (Per Media Stream Information)
Nested TLV 3 2A 07 00 04 00 00 17 72 (Media Port) Speech Box RTP Port

Frequency Shift Keying

Contents

Overview

Introduction

The Frequency Shift Keying (FSK) feature, supported by the DSP Series 2 card, involves the transmitting and receiving of FSK modulated signals.

FSK is a modulation technique used by modems in which two different frequencies are used to represent the binary state of 1s and 0s as tones across telephone lines.

This feature supports both European Telecom (ETSI) and China Telecom (CTSI) SMS protocols and Caller ID.

FSK Functionality

The FSK feature supports the following functionality:

Caller ID/Call Naming Information

Transmitting information related to an incoming call. For example, Calling Line Identification Presentation (CLIP) and related services such as Message Waiting Indicator and Advice of Charge Information in an inband signal on a FXS (Foreign Exchange Station) line.

Short Message Service (SMS)

- ETSI SMS System Architecture
- CTSI SMS System Architecture

FSK provides the transmitting and receiving of Short Message Service (SMS) data and signaling information on the DSP Series 2 card. Customers will be able to build applications to use the CSP as a Short Message Service Center (SM-SC) for the wireline markets.

API Reference Data

- New and Modified API Messages
- New and Modified Tag Length Value (TLV) Blocks
- Modified Information Control Block

Glossary

CAS (Dual Tone-Alerting Signal)

DTMF (Dual Tone Multi-Frequency)

FSK (Frequency Shift Keying)

ISDN (Integrated Services Digital Network)

LE (Local Exchange)

PSTN (Public Switched Telephone Network

SAS (Subscriber Alerting Signal)

SM (Short Message)

SMS (Short Message Service)

Server (Short Message Server)

CPE (Customer Premesis Equipment)

TAS (TE Alerting Signal)

TE (Terminal Equipment)

Caller ID/Calling Name Information (In-band Signaling)

Overview

This feature enables the DSP Series 2 card to send the following FSK modulated signals for Caller ID/Calling Name information as an on-hook data transmission and Call Waiting/Caller ID information as an off-hook data transmission.

- On-Hook data transmission during ringing
- On-Hook data transmission prior to ringing
- On-Hook data transmission not associated with ringing
- Off-hook data transmission

On-Hook Data Transmission During Ringing

Data transmission occurs during the first long silent period between two ring patterns. The first long silent period provides sufficient duration for the data to be transmitted. The initial ringing provides an alert signal to the TE to expect a data transmission.

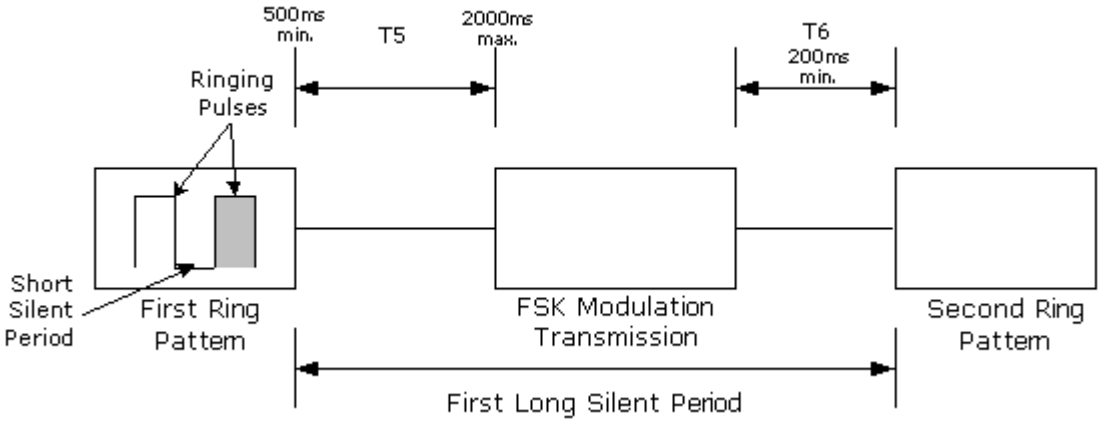
If the TE goes into a loop state before or during the data transmission, normal incoming call processing occurs and the data transmission is aborted.

Timing

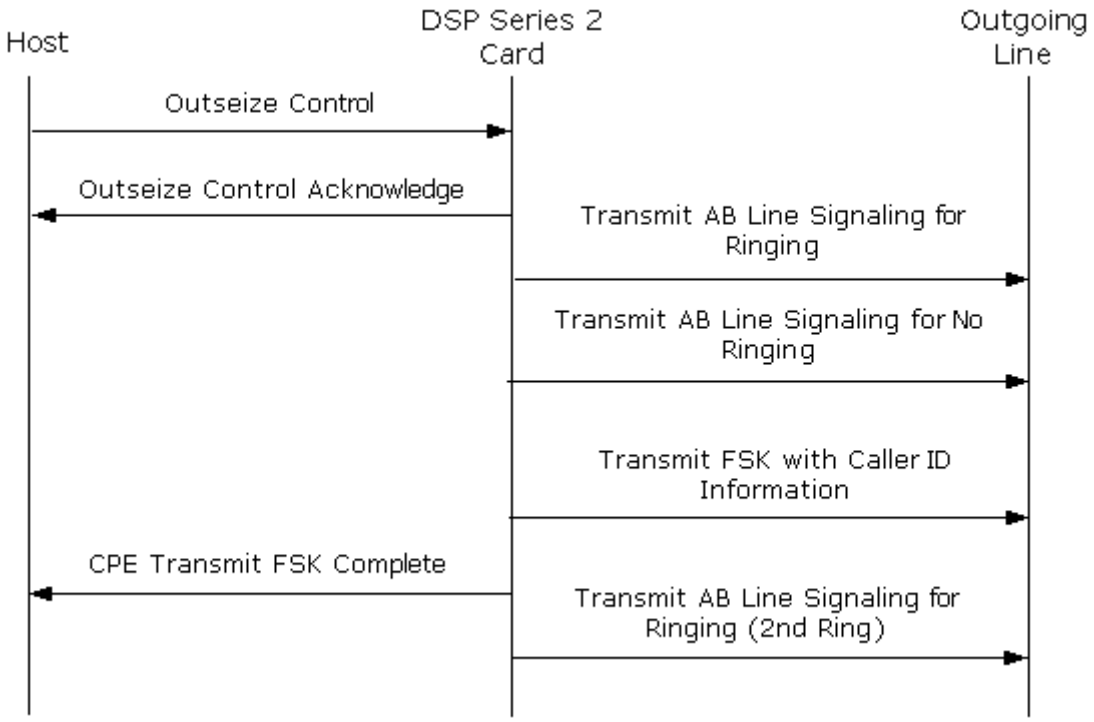
An example of the timing sequence is as follows. Refer to the timing diagram.

The FSK modulation transmission starts between a minimum of 500 ms and a maximum of 2000 ms (T5) after the end of the first ring pattern. The second ring pattern starts at a minimum of 200 ms (T6) after FSK modulation transmission has stopped. The lower limits are required to enable the TE to apply and remove the appropriate circuit for data reception.

Figure 7-1 Data Transmission During Ringing



FSK Data Transmission (without DT-AS) During Ringing



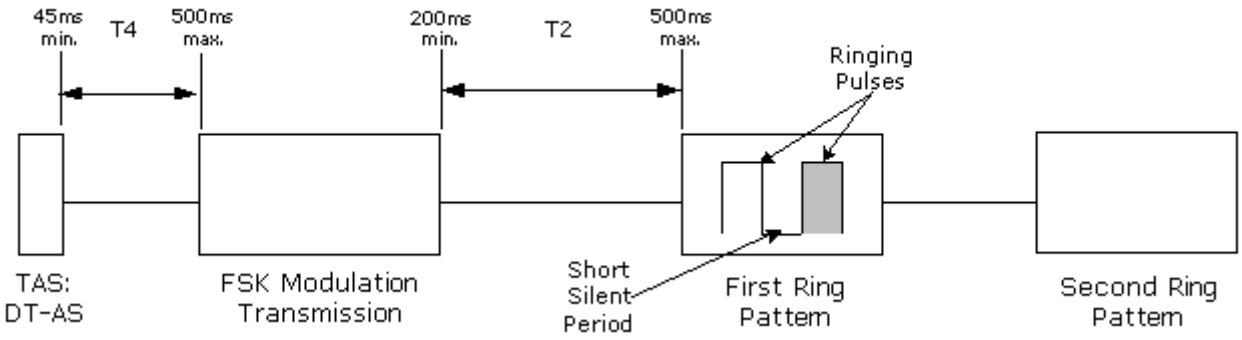
On-Hook Data Transmission Prior to Ringing

A TE Alerting Signal (TAS), Dual Tone-Alerting Signal (DT-AS), is used to signal the Terminal Equipment (TE) to expect a data transmission. The data transmission occurs prior to the normal ring pattern, after the DT-AS.

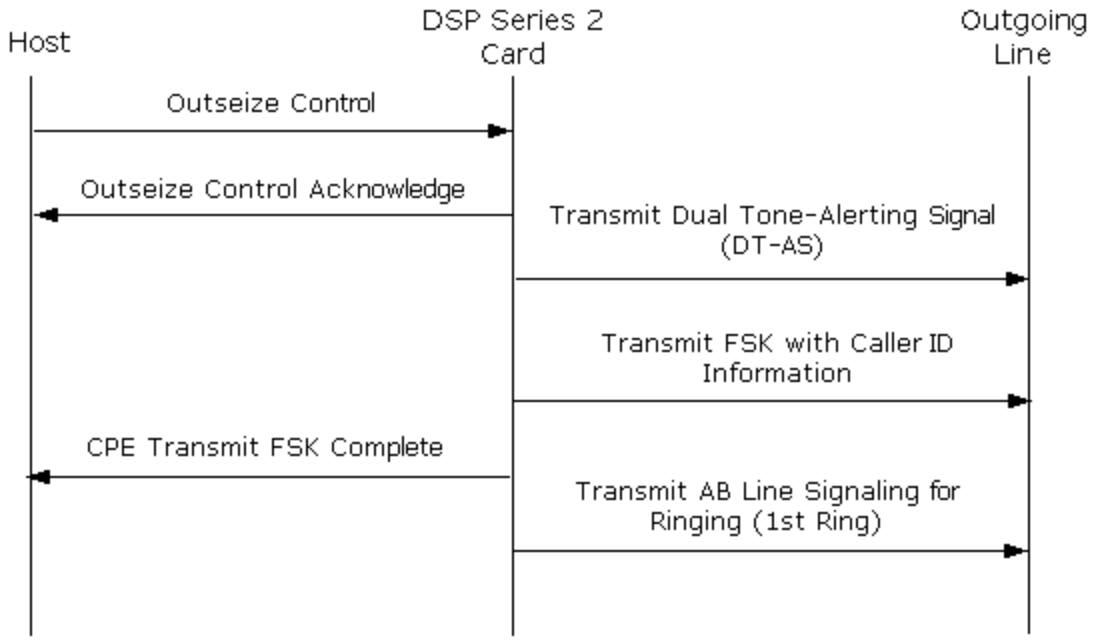
Timing

The DT-AS must precede the FSK modulation transmission between a minimum of 45 ms and a maximum of 500 ms (T4). The first ring pattern occurs between a minimum of 200 ms and a maximum of 500 ms (T2) after FSK modulation transmission has stopped. The lower limits are required to enable the TE to apply and remove the appropriate circuit for data reception.

Figure 7-2 Data Transmission Prior to Ringing



FSK Data Transmission with DT-AS Prior to Ringing



On-Hook Data Transmission Not Associated with Ringing

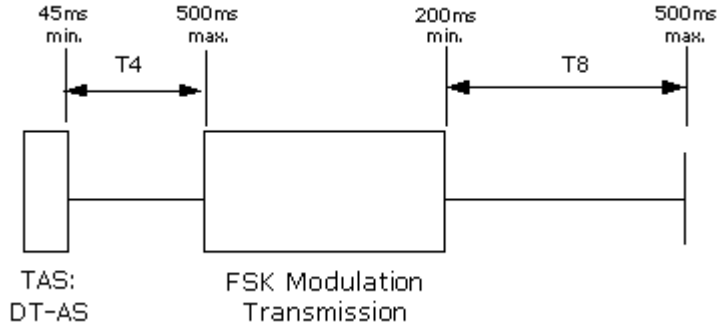
A TE Alerting Signal (TAS), Dual Tone-Alerting Signal (DT-AS), is used to signal the Terminal Equipment (TE) to expect a data transmission. Data transmission occurs after the DT-AS.

If the TE goes in loop state before or during the FSK modulation, the FSK modulation is aborted and normal outgoing call procedure occurs.

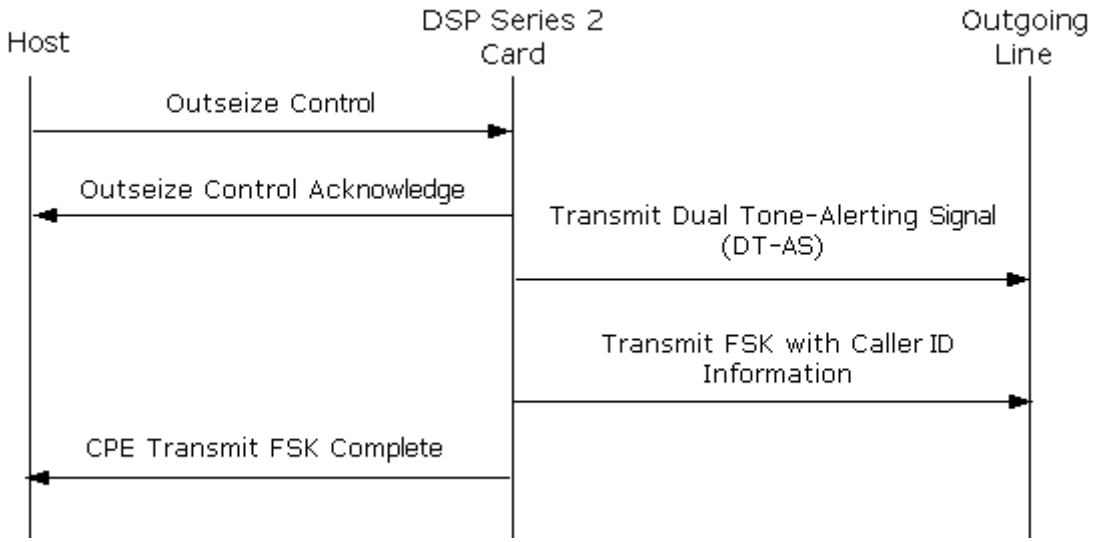
Timing

The DT-AS must precede the FSK modulation transmission between a minimum of 45 ms and a maximum of 500 ms (T4). The LE re-establishes the condition existing before the TAS is sent between a minimum of 200 ms and a maximum of 500 ms (T8) after FSK modulation transmission is stopped. The lower limit is required to enable the TE to apply and remove the appropriate circuit for data reception.

Figure 7-3 Data Transmission Not Associated with Ringing



FSK Data Transmission Not Associated with Ringing



Off-Hook Data Transmission

A TE alerting signal (TAS), DT-AS, is used to signal the TE to expect a data transmission.

The host application can also send a Subscriber Alerting Signal (SAS). For example, a call waiting tone is sent from the CSP to the subscriber before the FSK protocol signaling process. The sequence of events at the CSP are as follows:

1. The CSP blocks the voice path to and from the far-end party after receiving an indication from the host to transmit the DT-AS on a channel in the answered state. This minimizes interference with any alerting signal and the data transmission. It also prevents the far-end party from receiving these signals.

2. The DSP Series 2 transmits the DT-AS

3. The DSP Series 2 waits for the TE-Acknowledgement Signal (TE-ACK).

4. If the DSP Series 2 recognizes a valid TE-ACK within the time-out, the FSK modulation transmission will follow.

   If the DSP Series 2 does not recognize a valid TE-ACK within the time-out, the DSP Series 2 will not send any data transmission and the Matrix Series 3 will restore the voice path.

5. After the FSK modulation transmission, the voice transmission is restored by the Matrix Series 3.

Figure 7-4 Time Diagram Indicating a Successful Attempt

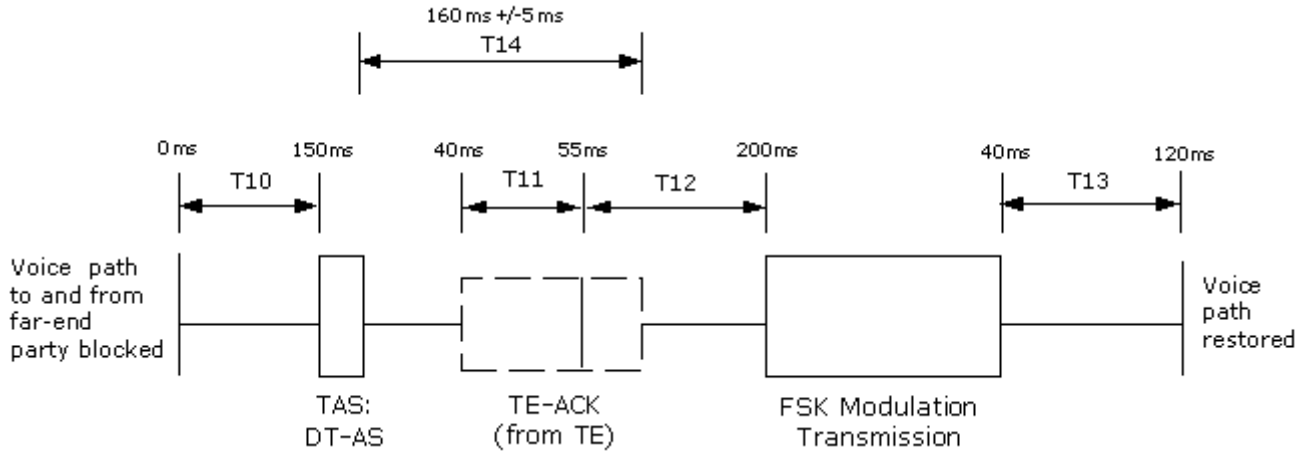


Figure 7-5 Time Diagram Indicating an Unsuccessful Attempt

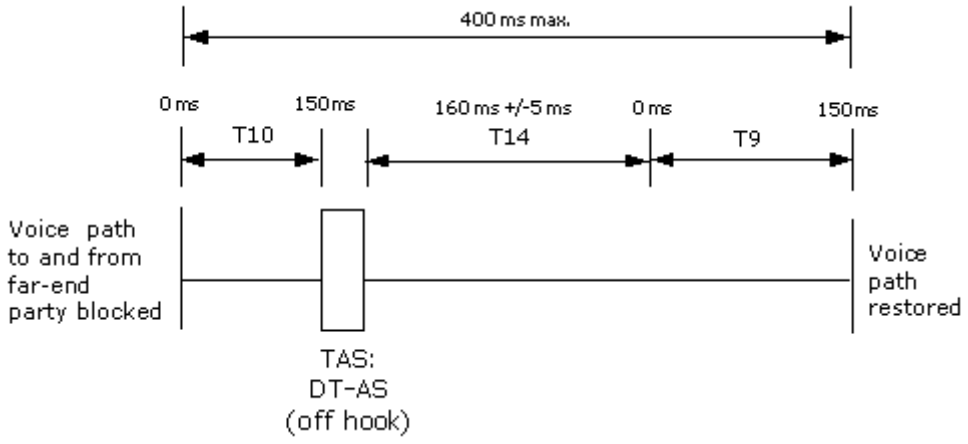


Table 7-5 Off-hook Timing Definitions and Values

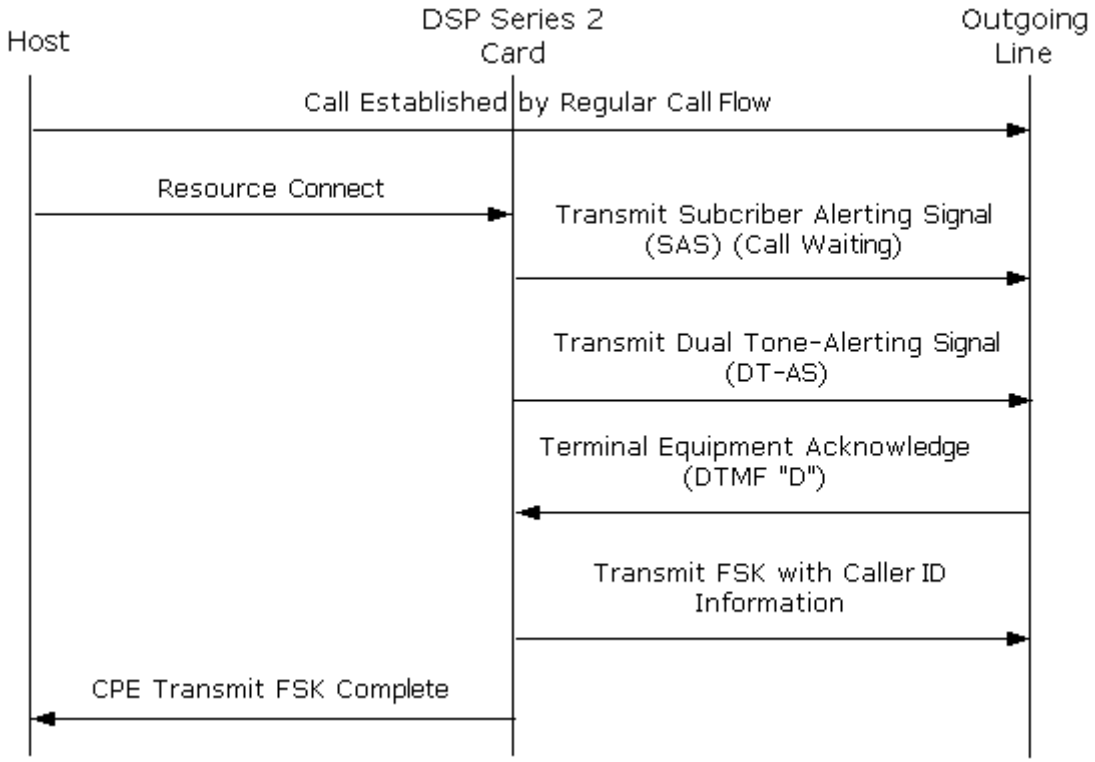
Time Interval	Value	Definition
T10	0 ms - 150 ms	The time between voice path blocking and the beginning of TAS sending (see Note)

Time Interval	Value	Definition
T11	40 ms - 200 ms	The time for the LE to recognize the TE-ACK
T12	55 ms - 200 ms	The time between TE-ACK recognition and start of FSK modulation transmission.
T13	40 ms - 120 ms	The time to restore the voice path after the end of FSK modulation transmission.
T14	160 +/-5 ms	The maximum time allowed within which a valid TE-ACK will be correctly detected. The time interval for which T14 is the maximum will begin at the end of TAS transmission.
T9	0 ms - 150 ms	The time to restore the voice path after the end of T14

Note: If a SAS is sent and the voice path has been blocked before the SAS is complete, either of following conditions would occur:

- If the voice path is restored between the SAS and TAS, then T10 is the time between the latter voice path blocking and the beginning of TAS being sent.
- If the voice path is not restored between the SAS and TAS, then T10 will start at the end of the SAS.

FSK Data Transmission with SAS and DT-AS



ETSI SMS System Architecture

Overview

The SMS system consists of a Short Message Terminal Equipment (SM-TE), a Short Message Service Center (SM-SC) and the PSTN.

SMS Protocol Stack

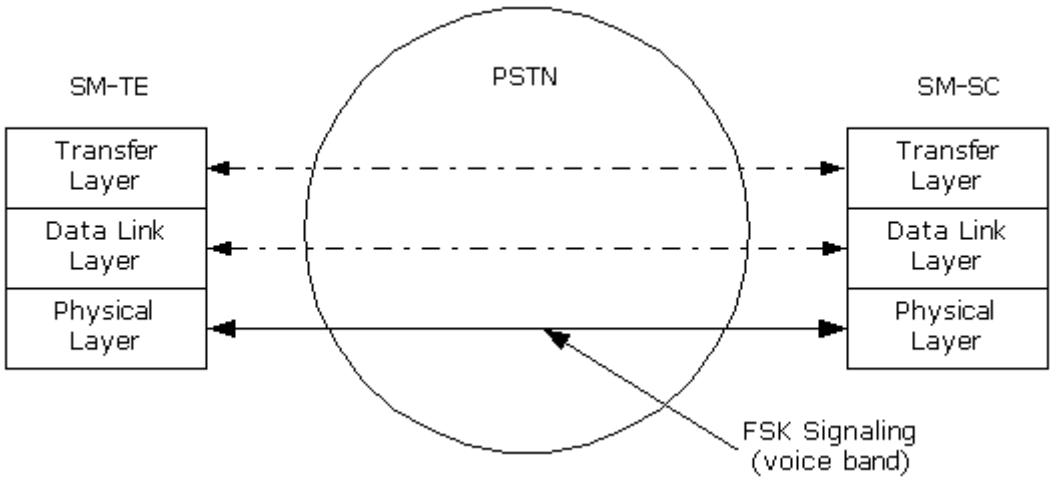
The SMS protocol stack, as shown in the figure below, provides an overview of the system architecture of SMS in relation to the PSTN. The protocol stack consists of three Short Message (SM) layers:

- Transfer Layer - provides the interface to the application.
- Data Link Layer - resides on the DSP Series 2 card
- Physical Layer - resides on the DSP chip

To receive and transmit a SM, the Transfer Layer uses the Data Link Layer to provide for a protected transmission of a SM between a SM-TE and SM-SC. The Physical Layer uses a 1200 Baud FSK modulation.

Both the SM-TE and the SM-SC must use the call control protocol required by the PSTN in order to establish and preserve a connection between them.

Figure 7-6 ETSI ISO/OSI Model of SMS Data Transmission



Short Message Transfer

Overview

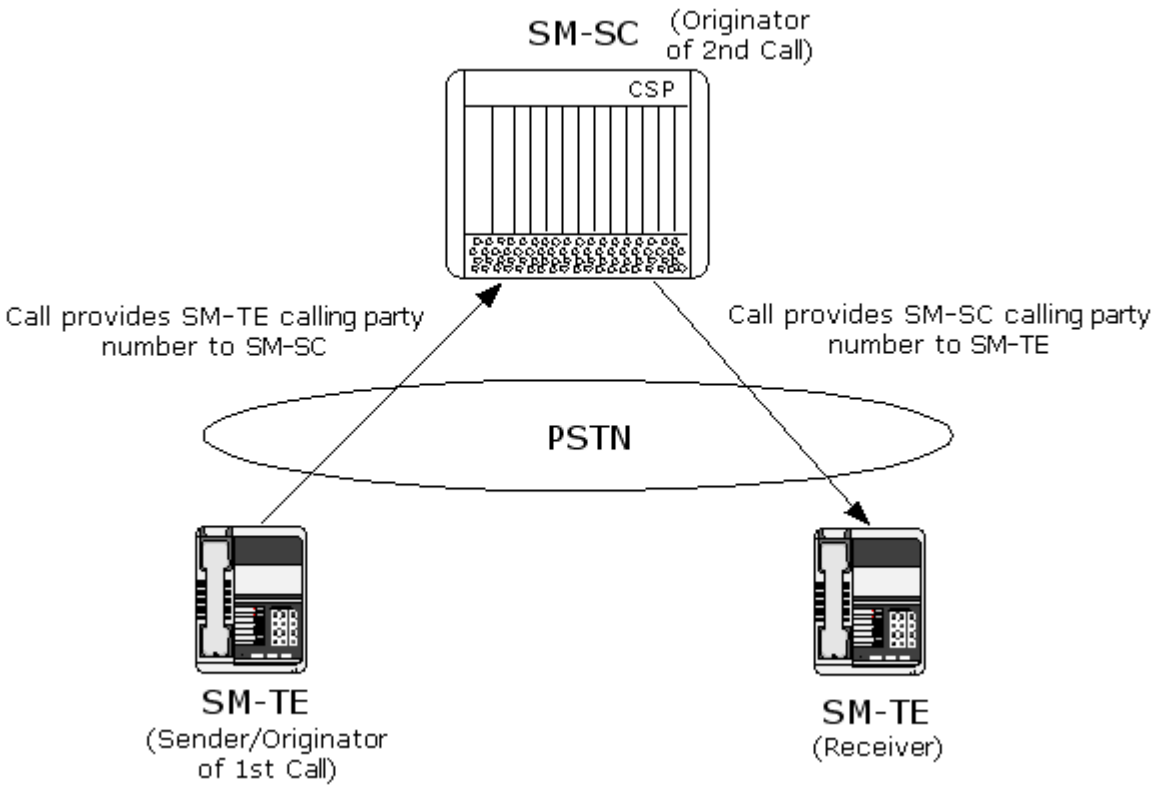
The SM-TE connects to the network using the PSTN as access. The SM-SC connects to the network via an ISDN primary rate interface, SS7 or other connection type.

To send and receive Short Message (SM) transfers, a voice-band communication path must be established in the PSTN between the SM-TEs and SM-SC using basic call control procedures.

The SM transfer is divided into two phases:

- SM submission - transfer of a SM from the SM-TE (sender) to the SM-SC.
- SM delivery - transfer of a SM from the SM-SC to the SM-TE (receiver).

Figure 7-7 Short Message Transfer



Short Message Submission

In the SM submission phase, the SM-TE establishes a a voice-band communication path to the SM-SC in order to submit the SM.

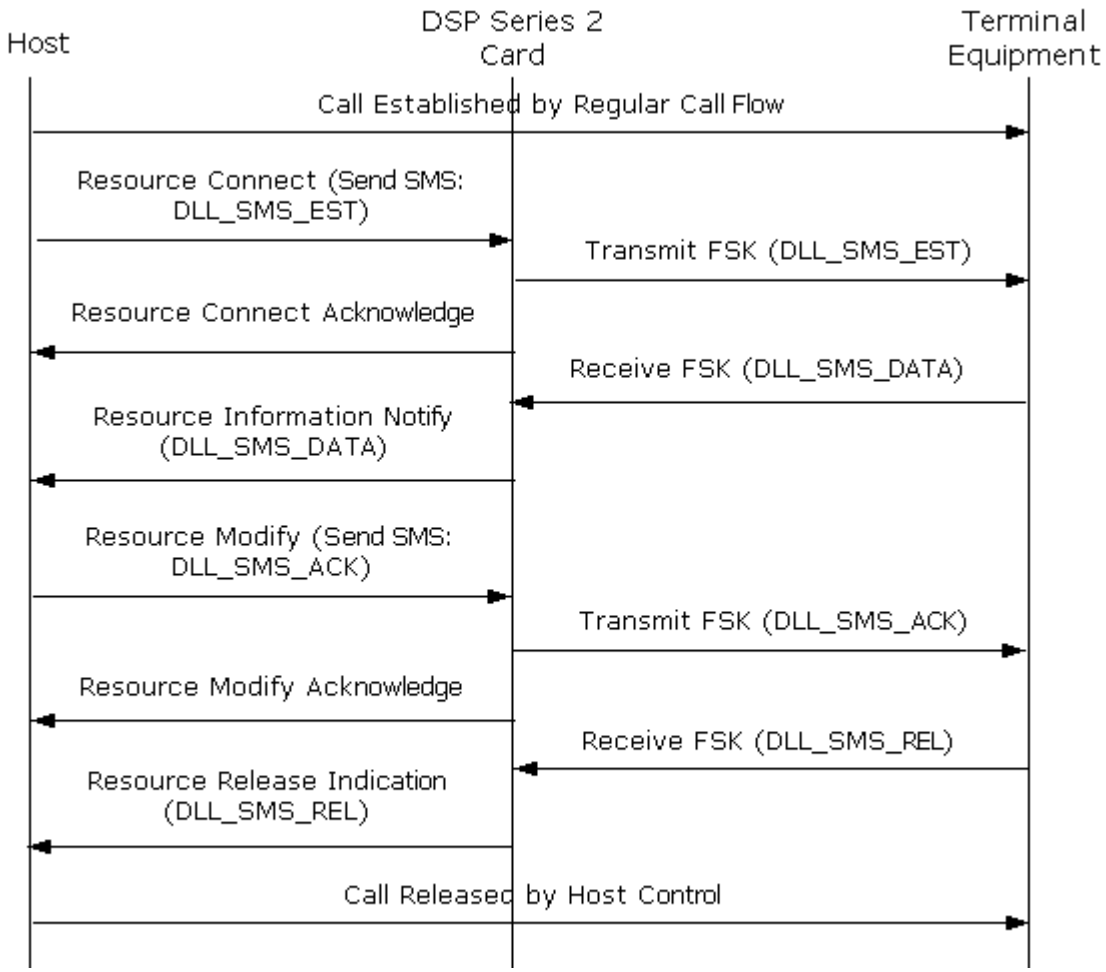
After the voice-band connection between SM-TE and SM-SC has been established, the end-to-end SM data transfer phase is entered for Short Message transfer from SM-TE to SM-SC.

When this occurs, the Caller ID (calling line identification) of the SM-TE is routed to the SM-SC. The SM-SC uses this information to identify the SM-TE and for billing purposes. After the SM has been transferred, the connection between SM-TE and SM-SC is released.

SMS Submission from SM-TE to SM-SC

After the SM-TE originates the call and the SM-SC answers it, the connection is ready for the SM transfer between SM-TE and SM-SC.

The SM-SC then initiates the data transfer by sending the appropriate Data Link Layer message DLL_SMS_EST. The Data Link Layer in SM-TE then sends the DLL_SMS_DATA message containing the SMS information. The SM-TE then releases the connection by sending the DLL_SMS_REL message.



Short Message Delivery

In the SM delivery phase, the SM-SC establishes a call to the SM-TE to deliver the SM to the SM-TE. In this case, the network provides the Caller ID (calling line identification) of the SM-SC to the SM-TE. The SM-TE uses this Caller ID information to identify and connect an incoming call from the SM-SC automatically.

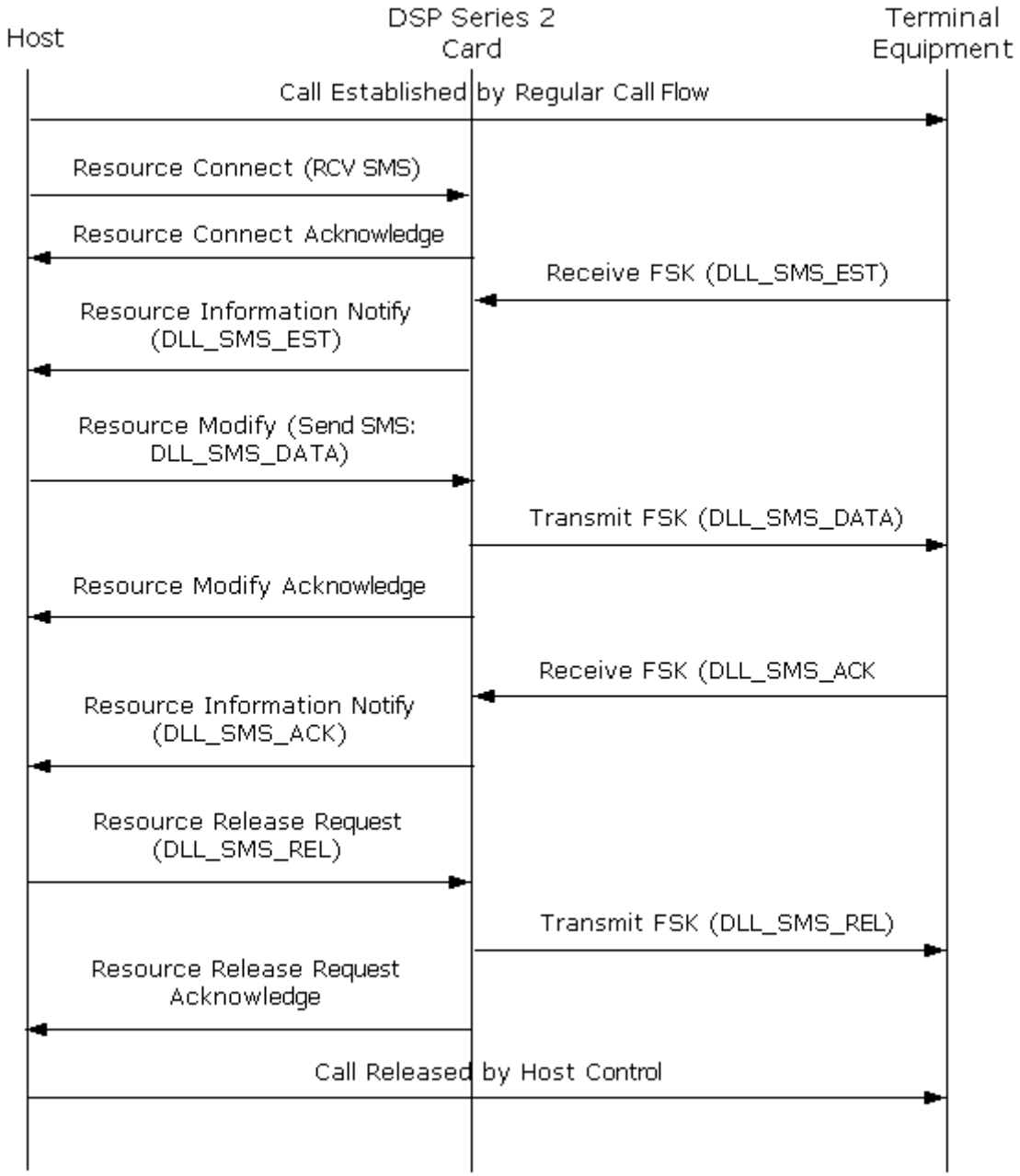
As in the SM submission phase, the SM is transmitted from the SM-SC to the SM-TE after the voice band connection between SM-SC and SM-TE has been established. After the SM has been transferred, the connection between SM-SC and SM-TE is released.

SMS Delivery from SM-SC to SM-TE

To deliver a SM to the designated SM-TE, the SM-SC calls the SM-TE's subscriber line.

Depending on the delivery mode identifier in the calling party number of the SM-SC when calling SM-TE’s subscriber line, the following two options are available:

- The SM-TE answers the call and receives the SM.
- The SM-SC terminates the call after a short time after transmitting the SM-SC's Caller ID to the SM-TE. The SM-TE then generates a call back to the SM-SC to receive the SM.


Host
DSP Series 2 Card
Terminal Equipment
Call Established by Regular Call Flow
Resource Connect (RCV SMS)
Resource Connect Acknowledge
Receive FSK (DLL_SMS_EST)
Resource Information Notify (DLL_SMS_EST)
Resource Modify (Send SMS: DLL_SMS_DATA)
Transmit FSK (DLL_SMS_DATA)
Resource Modify Acknowledge
Receive FSK (DLL_SMS_ACK
Resource Information Notify (DLL_SMS_ACK)
Resource Release Request (DLL_SMS_REL)
Transmit FSK (DLL_SMS_REL)
Resource Release Request Acknowledge
Call Released by Host Control

CTSI SMS System Architecture

Overview

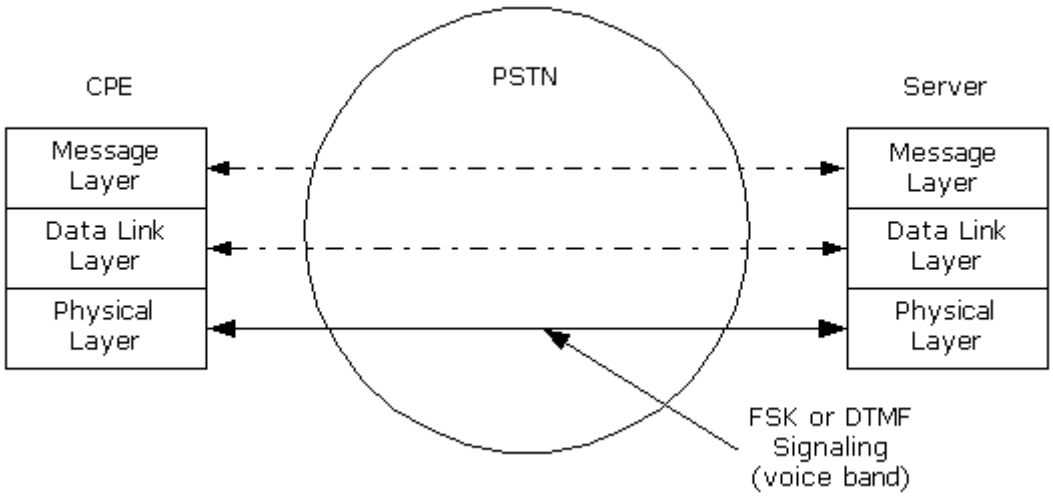
The China Telecom (CTSI) SMS system consists of a Customer Premises Equipment (CPE), a Short Message Server (Server) and the PSTN.

SMS Protocol Stack

The CTSI SMS protocol stack provides an overview of the system architecture of SMS in relation to the PSTN. The protocol stack consists of three Short Message (SM) layers:

- Message Layer - provides the interface to the application.
- Data Link Layer - resides on the DSP Series 2 card
- Physical Layer - resides on the DSP chip

Figure 7-8 CTSI Model of SMS Data Transmission



Physical Layer

CTSI requires that all servers must support both FSK and Dual Tone Multi-Frequency (DTMF).

DTMF Data Messages

CTSI SMS messages require using (DTMF) as the physical transport for SMS messages. DTMF provides tones to encode and transmit the binary SMS data. This not only changes the physical layer for message transmission, but also places a burden on the acknowledgement process to allow for choosing the appropriate uplink/downlink transport mechanism.

FSK Data Messages

Modulation: Bell 202, Continuous Phase Binary Frequency Shift Keying (FSK)

Signaling Messages

Customer Alerting Signal (CAS): The prompt signal from the server to the CPE 2150/2750 Hz for 85 ms.

CAS Response: The DTMF digits are described in the table below. In the DSP Series 2 architecture, these digits are detected on the DSP Chip, and reported to the PPL State machine for interpretation, and further state transitions.

Table 7-6 DTMF Digits

Name	Function	Composition	Meaning	Timing
Acknowledge of Caller Alerting Signal	Equipment Response	DTMF A	CTSI Equipment	On 150ms
		DTMF B	CTSI Equipment (FSK)	
Acknowledge of FSK Packet	Data Error Detection	DTMF D1	Data Transmission Correct	On 150 ms Off 100 ms (for each DTMF)
		DTMF D0	Data Transmission Error	

Uplink (CPE to Server)

The uplink from the CPE to the server can be either DTMF or FSK. FSK messages are as described for the Downlink, and DTMF is specified as below:

Timing: Short tone mode DTMF on 50-60 ms; off 50-60 ms

Table 7-7 DTMF Types

Name	Function	Timing	Note
Short tone mode	Uplink data transmission	On 50-60ms Off 50-60 ms	
Confirmation tone mode*	Message confirmation of CPE to the server; For auto-dialing	On 150 ms Off 100 ms	Use this mode to dial under normal situations

Name	Function	Timing	Note
Long tone mode	For auto-dialing	On 250ms Off 200ms	

* When sending DTMF using the confirmation tone mode, off (100ms) is limited for dialing. When sending the Acknowledge (ACK), after the DTMF for 150ms, CPE should get into the next state with no stop.

Data Link Layer

The data link layer provides the mechanism for reliable data transmission. The major functions of this layer are:

1. Packetizing and unpacketizing for the data link layer data packet
2. Set up and release a data link between the CPE and server
3. Timing control for the CPE and server
4. Error detection on data transmission
5. Retransmission when the error happens
6. Support function for the CPE with FSK uplink capability

On the DSP Series 2, the downlink Data Link Layer (DSP Series 2 to CPE) is always handled on the DSP chip.

In the uplink direction, when the uplink is FSK, the Data Link Layer is handled by the DSP chip.

When the uplink is DTMF, the Data Link Layer is handled by the PPL state machine and the FSK PPL.

Table 7-8 Uplink and Downlink Combinations

Direction	Type	Message Packet Coding	
		CPE without Uplink Capability	CPE with Uplink Capability
Server to CPE	User Data	Downlink FSK message packet (1)	Downlink FSK message packet (1)
	Control Signaling		Downlink FSK signaling packet (4)
CPE to Server	User Data	Uplink DTMF message packet (2)	Uplink FSK message packet (5)
	Control Signaling	Uplink DTMF signaling packet (3)	Uplink FSK signaling packet (6)

Downlink FSK Message Packet Format

This is the user data from the server to the CPE in FSK.

Table 7-9 User Data From Server to CPE (FSK)

Synchronization String	Message Type	Packet Number	Message Content	Checksum

Synchronization String

Eight (8) synchronization leading characters (0x55) plus one (1) synchronization ending character (0x00) (Supplied by the DSP Chip)

When the CPE gets at least five (5) synchronization leading characters and the synchronization ending character, the synchronization is considered as being set up.

Message Type

The type of the Message Content. To be consistent with other relevant protocols, this value is 0x84. This is supplied by the application developer in the API.

Message Length

The byte count of the Packet Number (1 byte) plus the byte count of the Message Content (supplied by API).

Packet Number

0x01

Message Content

Determined by the message layer. It may contain one or several CTSI operation commands. The longest length cannot be over 254 bytes (supplied by API).

Checksum

Used for the error detection during the data transmission. The calculation is: sum of all the bytes except for the synchronization string; modulus 256; complement (supplied by DSP Chip).

Uplink DTMF Message Packet/Uplink DTMF Signaling Packet

The message packet comes from the message layer; the signaling packet is described in the error control section.

Downlink FSK Signaling Packet/Uplink FSK Message Packet/Uplink FSK Signaling Packet

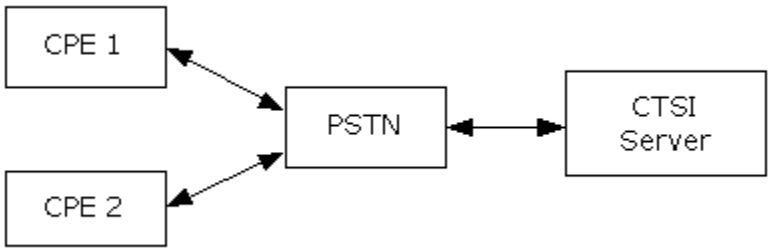
Described in the section of the management of CPE with uplink FSK capability.

Data Link Setup and Release Procedure

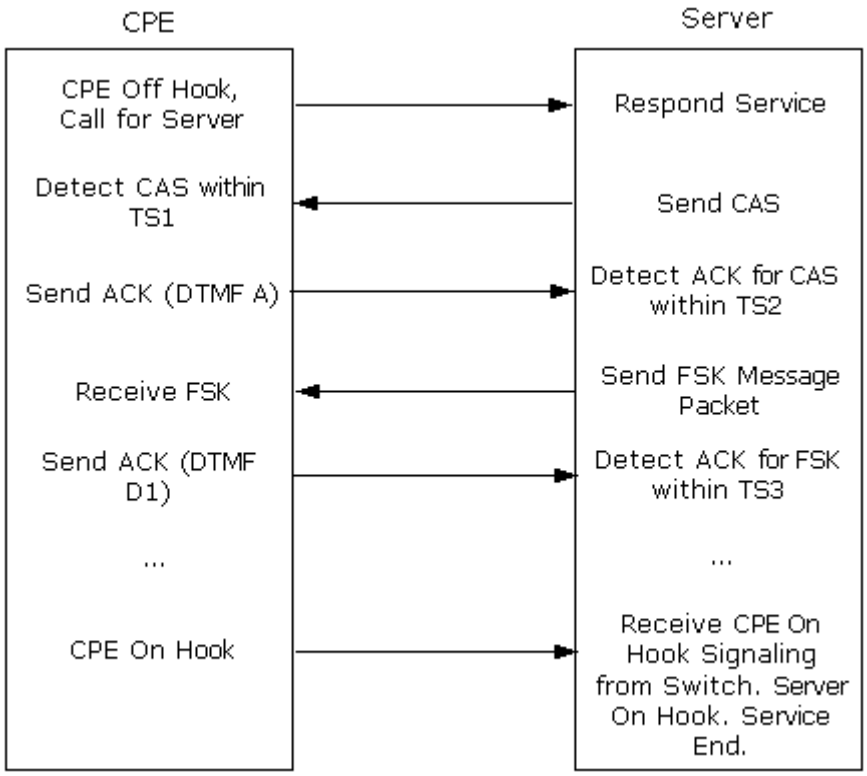
There are several scenarios for data link setup and release. The major problems in the data link layer are transmission error and service time-out. The mechanism to report these problems is the signaling. In this section, the transmission of signaling is in the form of DTMF. If user CPE supports FSK transmission, the signaling transmission should also use FSK. See the section of the management of CPE with uplink FSK capability.

The CPE is connected to the server through PSTN, as shown below:

Figure 7-9 CPE/Server Connection Diagram

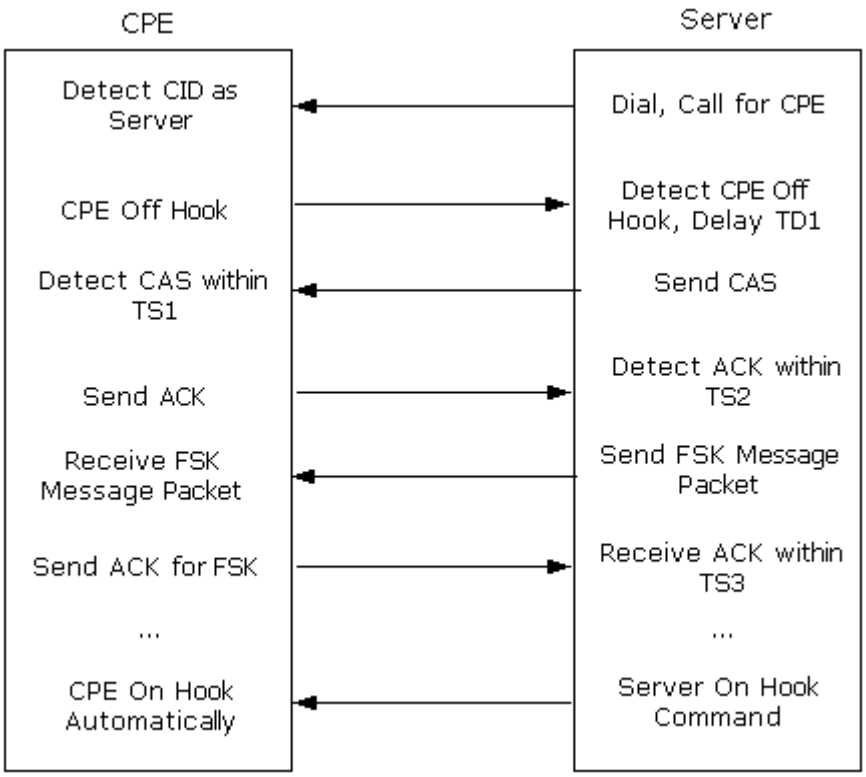


Call initiated by the CPE



Important! The CPE initiated call starts from the user dialing, or CPE automatic dialing. The CPE automatic dialing is suggested to use the confirmation tone mode (on 150ms, off 100ms).

Call initiated by the Server



Regular Timers In the protocol, the purpose of defining this type of timer is to assure the correct timing is received on the peer terminal, and to specify the service.

Table 7-10 Regular Timers

Timer	Location	Description	Usage	Length Txxx	Action at Timeout or Event
TD1	Server	After the server knows the CPE is off hook, TD1 is the delay before it sends CAS.	So the CPE has a delay after its off hook to stably receive the CAS. It is only used for server calling CPE.	TTD1= 90ms	Send CAS
CAS	Server	Duration of sending CAS		TCAS=75-85ms	Stop CAS
W_CAS_ASK (wait for ACK)	Server	Wait for ACK to CAS from CPE		TW_CAS_ACK= 155ms	

Timer	Location	Description	Usage	Length Txxx	Action at Timeout or Event
TD2	Server	Transferring time from waiting for CAS_ACK to sending FSK	Server system specification	Recommend 50-100 ms, Max. 500 ms	Send FSK
W_CAS_ASK (wait for ACK	Server	Wait for ACK to FSK from CPE		1000ms	
TD3	Server	Transferring time from waiting for FSK_ACK to sending FSK	Server system specification	Recommend 50-100 ms, May be affected by ICP	Send FSK

Overtime Timers

In the protocol, the purpose of defining this type of timer is to detect whether some error happens at the peer terminal, and to serve the corresponding re-transmission mechanism.

Error Control for the Data Link Layer

The error control in the protocol uses a stop-wait scheme. After the server sends a CTSI message packet to the CPE, it has to wait for the confirmation message from CPE. Then the server can continue its operation: send the next or re-send the last CTSI message packet.

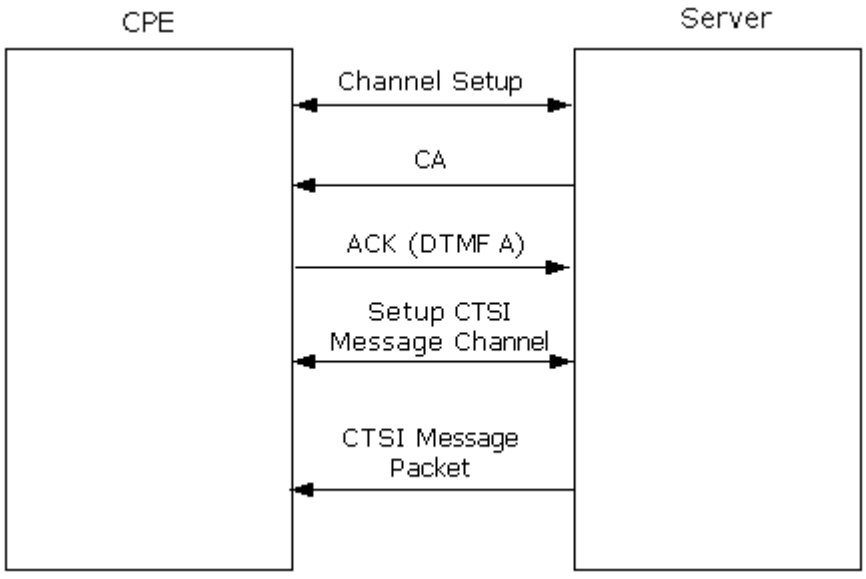


Table 7-11 Error Control

Server Message Type	CPE Confirmation Code	CPE Confirmation Message Meaning
CAS	DTMF A	Allow to set up CTSI service, uplink DTMF
	DTMF B	Allow to set up CTSI service, uplink FSK
	Other Code or Time-out with No Response	CPE rejection or problem in data link
Message Packet	DTMF D0	Message packet checksum error, please re-send
	DTMF D1	Message packet received, OK

Re-transmission Mechanism

The retransmission mechanism tries to keep the connection when an error condition occurs. If errors persist after the prescribe number of re-transmissions, the server disconnects with the CPE.

Table 7-12 Re-Transmission

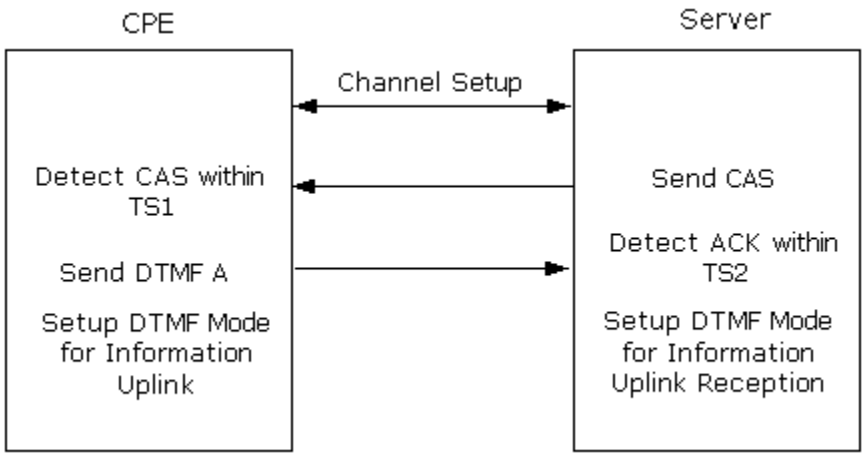
Re-transmission Contents	Re-transmission Confirmation Code	Re-transmission Count
CAS	TS2 is time-out, and TS1 is not time-out	3
FSK Message Packet	Receive CPE confirmation D0; or TS3 is time-out	3

CPE capabilities

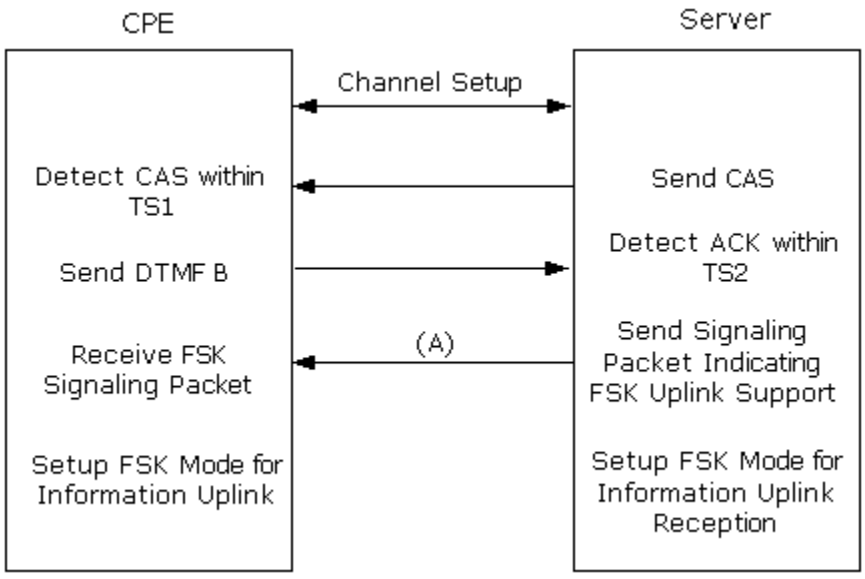
For the information uplink of CPE, both CPE and server need to support DTMF. Optionally, either the CPE or the server can support FSK uplink. At the start of the SMS interaction, the server and CPE exchange signaling messages to ensure common capabilities. If the CPE supports FSK uplink, it responds to the server's CAS with DTMF B. The server will send back a FSK signaling packet to indicate whether it supports FSK uplink from the CPE. If the server does, from then on the data uplink can use FSK.

If the server does not support FSK, the CPE will time out, and the CPE will request a DTMF uplink (using DTMF A). The server does not need to confirm the DTMF uplink capability with the CPE, because it is mandatory according to CTSI requirement. Refer to the following call flows that correspond to the various scenarios.

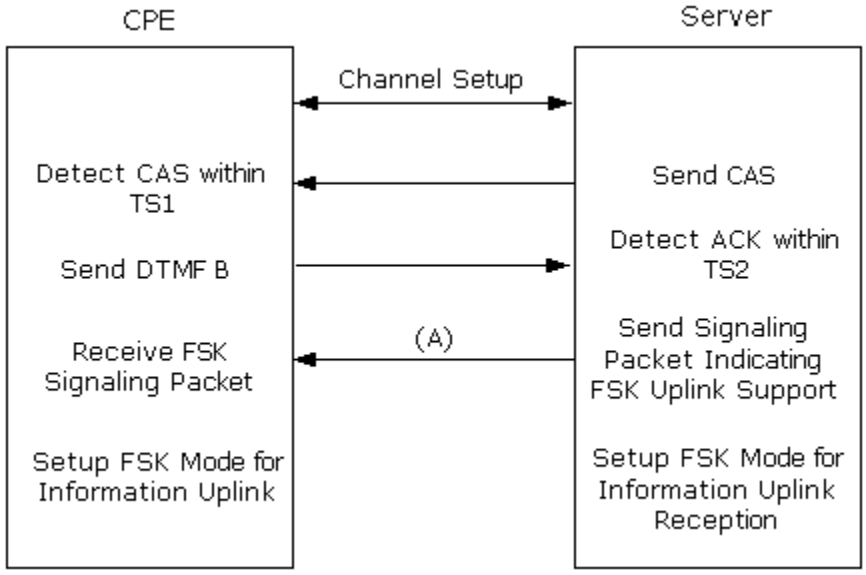
CPE Supports the DTMF Uplink Only



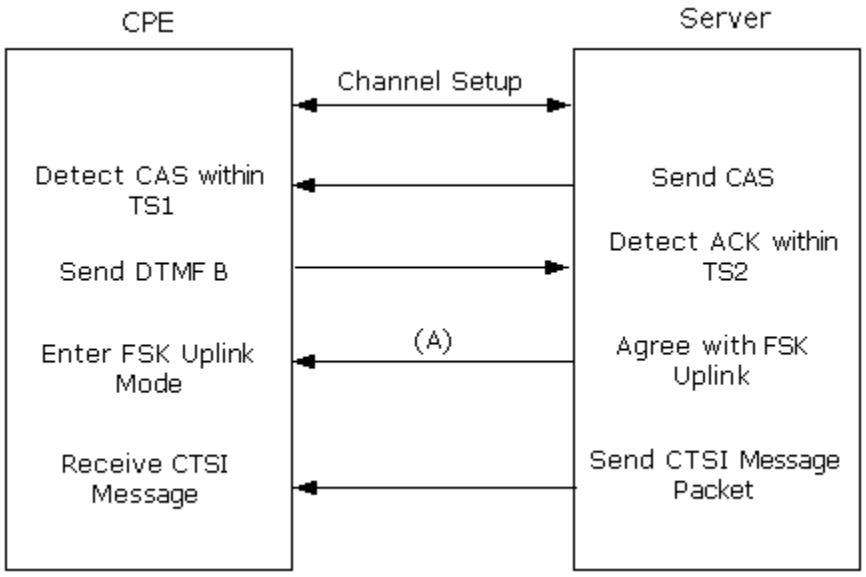
CPE Supports the FSK Uplink Only



Server Does Not Support FSK Uplink



Message Format using FSK at CTSI Data Link Layer



Downlink FSK Signaling Packet/Uplink FSK Message Packet/Uplink FSK Signaling Packet

The following is the user data required to manage the CPE with uplink FSK capability.

Synchronization String	Message Type	Message Length	Subtype Number	Message Content	Checksum

Synchronization String

Eight (8) synchronization leading characters (0x55) plus one (1) synchronization ending character (0x00) (Supplied by the DSP Chip)

When the CPE gets at least five (5) synchronization leading characters and the synchronization ending character, the synchronization is considered as being set up.

Message Type

The type of the Message Content. From CPE to server, this value is 0xFE. From server to CPE, this value is 0xFF.

Message Length

The byte count of the Sub-type Number (1 byte) plus the byte count of the Message Content.

Subtype Number

Depends on the different types of signaling.

Message Content

The longest length cannot be over 254 bytes.

Checksum

Used for the error detection during the data transmission. The calculation is: sum of all the bytes except for the synchronization string; modulus 256; complement.

Function	Message Type	Subtype Number	Parameter	Meaning	Location in Diagram
Server Capability Confirmation	0xFE	0x00	0x00	Support FSK Uplink	A
			0xFF	No Support for FSK Uplink	
Server Confirmation to CPE User Data Packet		0x01	0x00	User Data FSK Correct	E
			0xFF	User Data FSK Not Correct	

Function	Message Type	Subtype Number	Parameter	Meaning	Location in Diagram
Report on CPE Processing for Server Command	0xFF	A	0x00	Storage device does not exist	C
			0x01	Storage device full	
			0x02	Private account not exist	
			0x03	User terminates download due to exception	
			0x04	Others, user refusal	
			0x0A	Device download succeed	
			0x0B	CTSI command can not be parsed	
CPE User Data Packet		B	User Data	User Uplink Data	D
Report on CPE Receiving CTSI commands	0xFF	D	0x00	CTSI Command Packet Receive Error	B
			0x01	CTSI Command Packet Received Correctly	

Message Layer (Handled at the Application)

On the server side, the function of the message layer is to prepare the message packet content and parse the CPE feedback message. On the CPE side, the function of the message layer is to parse CTSI message packet and prepare the CPE feedback message. The message layer provides the message exchange mechanism between the CPE and server.

The message layer defines the format of message packet, the message (command) format, and the feedback message format.

The format of message packet: the message packet is a group of commands (message) sent from the server to the CPE. It may contain one or several CTSI commands (message).

CTSI Command 1	CTSI Command 2	.....	CTSI Command N

CTSI Command

Composed of command code, length and command argument.

Command Argument

Describes the actual content for command execution (variable length).

Length is the byte count of the command argument. Since the command is encapsulated in the message packet, and the total length of one message packet is less than 254, the length of the command argument is less than 252. When there is no command argument, the length field is 0.

CTSI Command

There are two forms of the CTSI command.

Command Code	0x00

Command Code	Length	Command Argument

Command Code

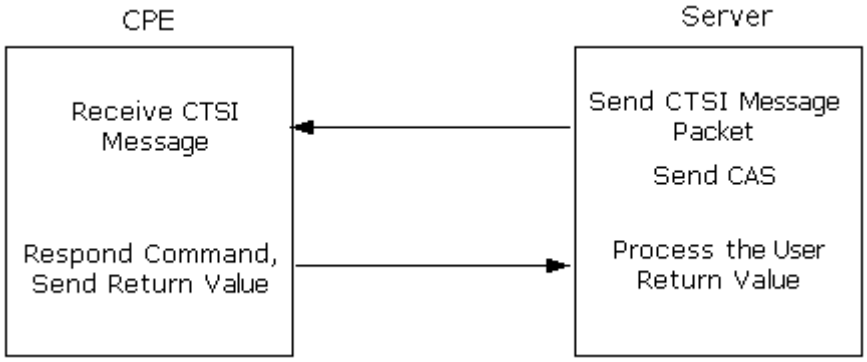
Identify the command code as listed below.

Command Code	Subroutine of Command Process Program	Function
0xB0	Query_Server_ID	Service Query
0xB1	Information_Download	Information Download
0xB2	Information_Upload	Information Upload
0xB3	Information_Input	Screen Query
0xB4	Information_Output	Screen Display
0xB5	Security_Set	Security Management
0xB6	Graph_Made	Graphics Production Management

Command Code	Subroutine of Command Process Program	Function
0xB7	Change_Receive_State	Change CPE Receiving State
0xB8	Query_CPE_Config	CPE Configuration Parameter Query

Message Exchange in the Message Layer:

The message exchange between the CPE and server always starts with the server sending a CTSI command. The CPE parses and executes the CTSI command, then it sends the feedback. The CPE does not initially send content to the server. In another words, the service application from CPE is submitted through the server sending a service query command.



Command Return Value

According to the CTSI command, the return value contains the result of the command execution (e.g. download command), or the feedback or user input (e.g. input command), or the CPE configuration parameters (e.g. CPE Configuration Parameter Query command). Not all CTSI commands need a return value (e.g. change CPE receiving state command). The return value is one ASCII code string.

Command Return Value Coding

The command return value coding method is decided according to the CPE type, FSK or DTMF.

FSK Coding

ASCII code of the return value using the regular FSK coding (0x00 ~ 0xFF).

DTMF Coding

There are two DTMF methods, non-encoded and encoded.

Important! DTMF C is reserved to avoid interfering the switch function.

- Non-encoded Method

The return value string is proceeded and inspected. If the character belongs to the sent DTMF character set, it will be transmitted using DTMF tone. Otherwise, this character is discarded. The sent DTMF character set is: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, #, *, A, B, D}

- Encoded Method

The return value string is proceeded. Each ASCII character is broken into two hexadecimal values (4 bits each), and sending these two hexadecimal values using the DTMF tone defined in the following table. The higher 4 bits are sent first and the lower 4 bits are sent second. For example, 0xCF generates three DTMF tones: * 1 #.

Hexadecimal Value	DTMF Sequence
0x00	DTMF 0
0x01	DTMF 1
0x02	DTMF 2
0x03	DTMF 3
0x04	DTMF 4
0x05	DTMF 5
0x06	DTMF 6
0x07	DTMF 7
0x08	DTMF 8
0x09	DTMF 9
0x0A	DTMF A
0x0B	DTMF B
0x0C	DTMF *, then DTMF 1
0x0D	DTMF D
0x0E	DTMF *, then DTMF 2

Hexadecimal Value	DTMF Sequence
0x0F	DTMF #

CPE Feedback Message

The feedback message is the message reported to the server during the CPE execution of the CTSI command. There are two types to message: Command Parsing and Execution Feedback and CPE User Input.

- Command Parsing and Execution Feedback

Not all CTSI commands require the CPE to return the command parsing and execution feedback message. However, some commands (e.g. information download command) need the return of parsing status. If the command code cannot be parsed, the server needs to be notified. Refer to the format below:

Length	1B	Fixed Length
Content	Command Parsing and Execution Feedback Message Flag	Argument
Coding	A	User Data
DTMF Coding	Non-encoded Method	
FSK Coding	Direct Coding	

Command Parsing and Execution Feedback Message Flag	Argument	Description
A	0	Storage Component Does Not Exist
	1	Storage Component Full
	2	Personal Account Does Not Exist
	3	User Stops Download
	4	Others, User Reject
	A	Component Download Successful
	B	CTSI Command Cannot Be Parsed

CPE User Input Message

According to the command, sometimes the CPE user related information needs to be sent back to the server, e.g., the information upload command, screen query command (refer to the CTSI command description). The return value can be fixed length or variable length. For the return message with variable-length, usually it contains a length field. Refer to the format below:

Return Message Packet with Fixed Length

Length	1B	Fixed Length
Content	CPE User Input Message Flag	Argument
Coding	B	User Data
DTMF Coding	Non-encoded Method	Encoded Method
FSK Coding*	Direct Coding	

Return Message Packet with Variable Length

Length	1B	1B	Fixed Length
Content	CPE User Input Message Flag	Length	Argument
Coding	B	0xYY**	User Data
DTMF Coding	Non-encoded Method	Encoded Method	
FSK Coding	Direct Coding		

* FSK coding: This is for the CPE with uplink FSK capability. For the CPE without uplink FSK capability, it used DTMF.

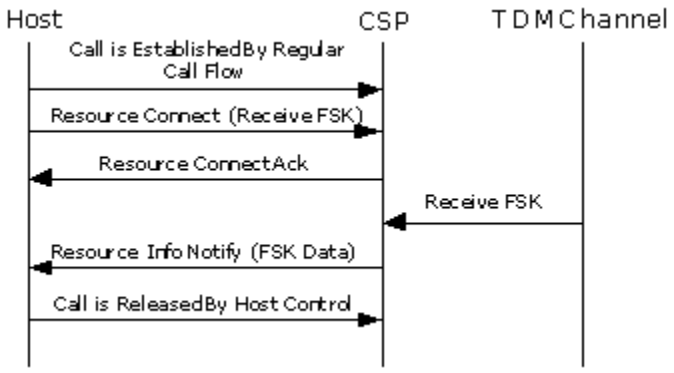
**YY is the binary form of the user data length.

The DSP SIMM Configure message supports the DSP Series 2 function types 0x35 (Transmit FSK Only) and 0x36 (Transmit and Receive FSK).

Receiving FSK

Overview

The Receiving FSK feature uses the *Resource Connect* (0x0127) and *Resource Information Notify* (0x0141) messages that allow the host to receive FSK data from an off-hook channel. Refer to the call flow below.



Currently, the CSP supports sending Frequency Shift Keying (FSK) data containing On-hook Caller ID in the *Outseize Control* (0x002C) message and Off-hook Caller ID in the *Resource Connect* (0x0127) message. It also supports sending and receiving FSK data containing SMS DLL messages that comply with ETSI and CTSI (China Telecom) call flows.

The following API messages and TLVs have been modified. Refer to the detailed information in the API Reference for the modified APIs and TLVs.

Modified APIs

- The *Resource Connect* (0x0127) message supports the new resource type Receive FSK (0x010D) under the Mandatory TLV Resource Type (0x0602).
- The *Resource Information Notify* (0x0141) message supports the new resource type Receive FSK (0x010D) under the Mandatory TLV Resource Type (0x0602).
- The *Call Processing Event* (0x002E) message, field 0x50 DSP Series 2 FSK Events, supports new event types: FSK Receive Failed (0x0007) and FSK Receive Complete (0x0008).

Modified TLVs

- A new resource type Receive FSK (0x010D) has been added to the Resource Type (0x0602) TLV.

- A new data type Raw Data (0x0002) has been added to the FSK Data (0x0690) TLV. This TLV is used to send FSK data received from the DSP chip, without any modifications or formatting, to the host.

T.30 FAX

Contents

Configuring T.30 Fax

Feature Description

See T.30 Fax.

Configuration

1. Use the DSP SIMM Configure 0x00C0 message to assign T.30 Fax Function Type to the DSP.
2. Configure T.30 FAX parameters using FAX TLVs in the Generic Card Configure 0x0122 message.
3. Use the Resource Connect 0x0127 message to initiate a fax.

   Resource Type TLV: Send FAX (0x0105) or Receive FAX (0x106)

Page Range in TIFF File

This feature allows the user of the DSP2's Fax transmission capability to select a range of pages for transmission out of a single TIFF file. This allows you to incorporate selective page transmissions into your application for such reasons as removal of cover pages for fax store and forward applications or fax back services which send selective sections of repair manuals.

To use this feature, send the Resource Connect 0x0127 message with the 0x068C FAX Page Range TLV.

TIFF Tag List

The list below indicates the active TIFF Tags. Any TIFF Tags other than the ones listed below will be ignored.

- TIFFTAG_NEWSUBFILETYPE
- TIFFTAG_IMAGE WIDTH one of 864, 1216, 1728, 2048, 2432, 2592, 3072, 3456, 3648, 4096, or 4864
- TIFFTAG_IMAGE LENGTH any value
- TIFFTAG_BITS PER SAMPLE must be 1
- TIFFTAG_COMPRESSION must be 1, 3, or 4
- TIFFTAG_FILL ORDER any value
- TIFFTAG_STRIP OFFSETS must be 1
- TIFFTAG_SAMPLES PER PIXEL
- TIFFTAG_ROWS PER STRIP any value
- TIFFTAG_STRIP BYTE COUNTS must only be one count
- TIFF TAG_T4 OPTIONS 0 (MH), 1 (MR)
- TIFFTAG_T6 OPTIONS if present, indicates MMR
- TIFFTAG_RESOLUTION UNIT, ignored

- TIFFTAG_PAGE NUMBER, used if present
- TIFFTAG_SOFTWARE, ignored
- TIFFTAG_DATETIME, ignored
- TIFFTAG_X RESOLUTION
- TIFFTAG_Y RESOLUTION 98, 100, 196, 200, 391, 400

Resolution types allowed

- 204, 98
- 200, 100
- 204, 196
- 200, 200
- 204, 391
- 408, 391
- 300, 300
- 400, 400

Query To Query T.30 FAX settings, send the Generic Card Query 0x0123 message with one or more of the T.30 FAX TLVs.

Positive Voice Detection/Answering Machine Detection (PVD/AMD)

Contents

PVD/AMD Overview

Introduction

The Positive Voice Detection/Answering Machine Detection (PVD/AMD) functions provide analysis of the incoming Pulse Coded Modulation (PCM) data input to determine if the PCM is one of the detectable signals.

The host has the ability to configure and query the PVD and AMD parameters on the DSP Series 2 card. The host can also attach a PVD or AMD receiver to a particular channel, so that it can detect the signal on that particular channel. While attaching the receiver, the host has the option to set the PVD/AMD parameters for a particular channel that are different from the board level default values. When the signal is detected on the particular channel, the CSP acts like the host using the Call Processing Events.

PVD/AMD Functionality

The following functions comprise the PVD/AMD feature. Refer to *Figure 7-10*, *Figure 7-11*, and *Figure 7-12* to support PVD/AMD functionality.

User Defined State Machine

Using the Call Processing Events (CPEs) provided by the PVD and/or AMD, the developer can create a state machine that allows:

- Solutions to be deployed that match a culture with a region. For example, differences between greetings can determined.
- Maximum flexibility and control to address field problems in the most effective and timely manner.

Energy Detection

The energy of the signal is approximated by taking the sum of the absolute value of the speech samples over a five millisecond (ms) window.

Positive Voice Detection (PVD)

The PVD feature, under host control, provides energy detection on a selected channel. The following parameter such as time constant, dB level, and sensitivity settings can be configured and are defined by the energy detection receiver. The energy detection receiver makes no distinction between a recorded voice, a live voice or any noise.

The PVD feature informs the host application when it detects the following:

- Energy/noise (rising edge)
- Energy/noise is removed (falling edge) with the duration of the noise (in the data of the falling edge)
- Silence

The PVD takes the signal output of the energy block and computes a long term estimate of the background noise based on the statistics of the signal. Signals are reported as positive voice when the short term energy measurement exceeds the long term noise by the specified API programmable noise margin. The PVD brackets individual words with detection events.

Answering Machine Detection (AMD)

The AMD uses the output of the PVD to provide indications of long term voice activity, and the presence of tonal signals. Long term voice requires that the positive voice be present for a user-programmable turn-on time to indicate voice present, and be absent for a hangover time to indicate a long-term voice is not present.

In addition, the AMD provides a report of any single tone present on the voice channel.

- The AMD adds a tone receiver looking in the frequency ranges defined in the PVD/AMD configuration.
- The AMD falling edge event also indicates the frequency of the detected tone.
- The AMD only detects single frequency tones.

Example: If you perform PVD/AMD on a fax machine the application will receive PVD events and the AMD tone events. This is because the PVD does not discriminate on tones; it is looking for any energy that falls within the defined Time Constant, dB level, and sensitivity settings. If you did not have the AMD portion applied in the *Resource Connect* message during a fax test, you would have no idea that the energy being detected is actually a tone at a given frequency.

Figure 7-10 PVD/AMD Functional Diagram

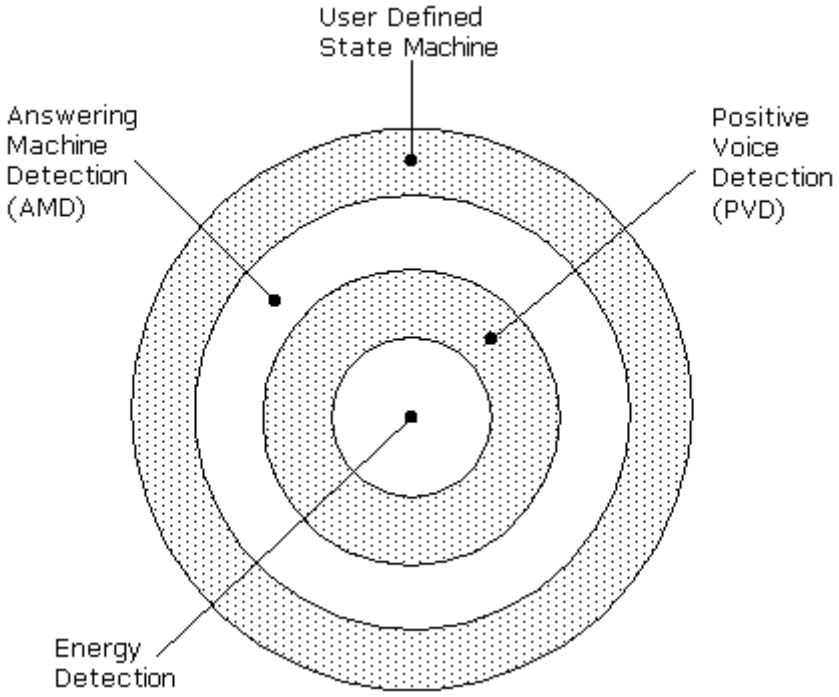


Figure 7-11 PVD/AMD Block Diagram

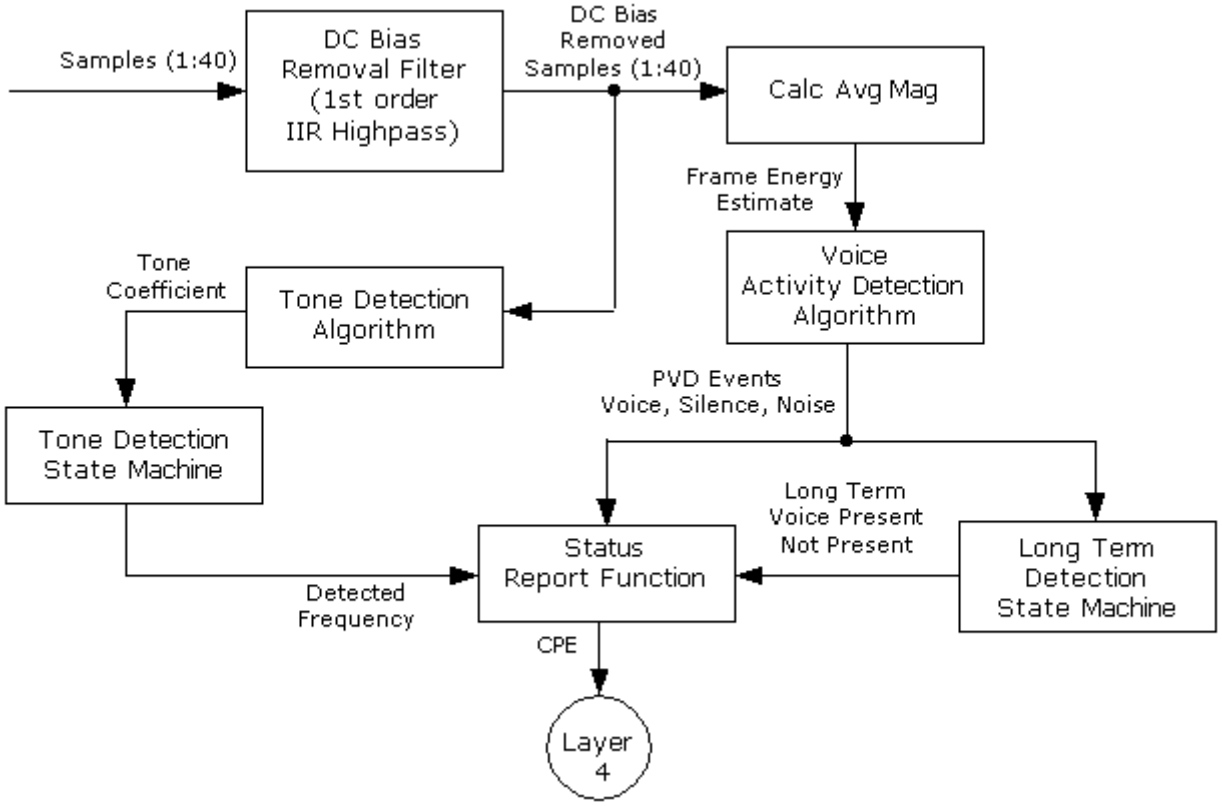
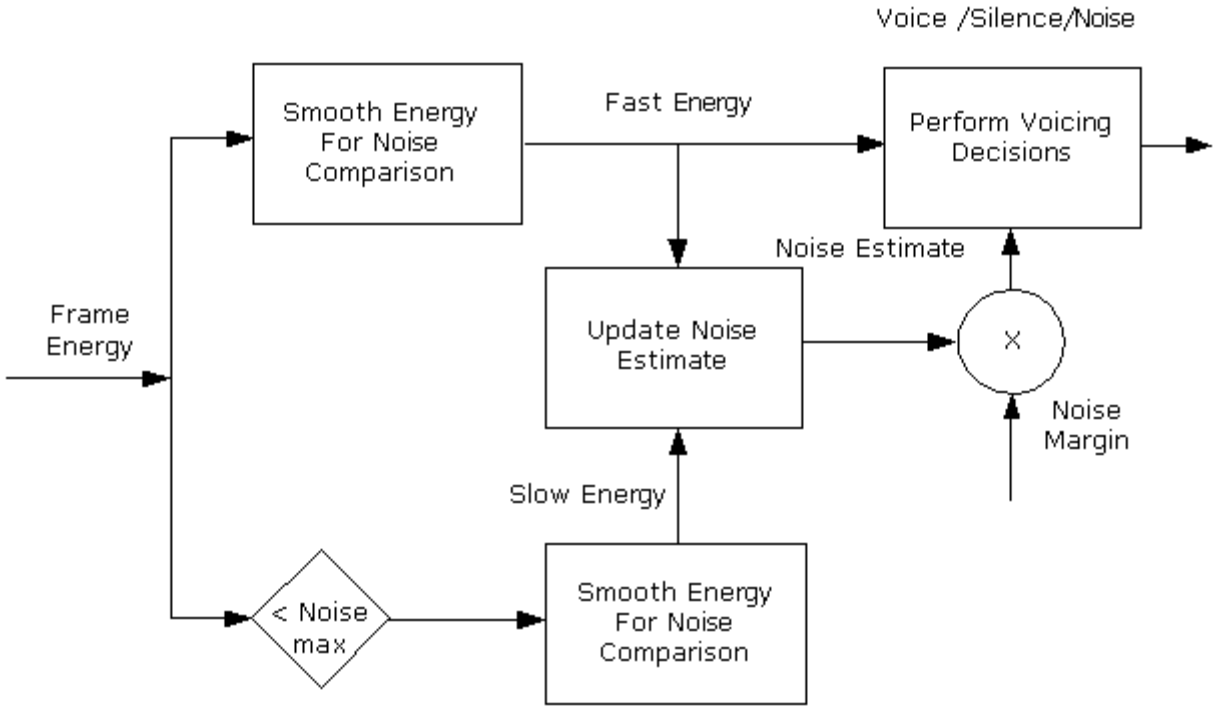


Figure 7-12 Voice Activity Detector



Implementing PVD/AMD

Introduction

You can configure the CSP to analyze incoming PCM data and detect voice or an answering machine. You can set PVD/AMD parameters for a DSP Series 2 card and for an individual channel. When a signal is detected, the host is notified with a *Call Processing Event* (0x002E) message.

The optimal settings used to configure PVD and AMD tone and wav files parameters are described below.

Configuration

1. Configure DSP for PVD/AMD function using the *DSP SIMM Configure* (0x00C0) message.

   DSP Function Type

   PVD/AMD (0x34)

2. Use the *Generic Card Configure* (0x0122) message to set the PVD and AMD parameters using the PVD Parameters TLV (0x0619) and AMD Parameters TLV (0x0620). This message allows you to set the PVD/AMD parameters at the DSP Series 2 card level.

   Set card-level PVD/AMD parameters using the *Generic Card Configure* (0x0122) message.

   Mandatory TLV

   0x05FA Card Object

   Object Type: PVD/AMD Parameters (0x0006)

Implementation

Use the *Resource Connect* (0x0127) message to attach or connect a system resource to a span channel, in this case PVD/AMD. During a call if the PVD and AMD Parameters TLVs are set the parameters in this message override any card level settings that are configured with the *Generic Card Configure* (0x0122) message.

1. Attach PVD/AMD receiver to channel using the Resource Connect 0x0127 message.

   AIBs

   Resource A: Span/Channel

   Resource B: Slot

   Mandatory TLV

   0x0602 Resource Type

   Resource Type = PVD/AMD (0x0109)

Optional TLVs (to override default card parameters)

Below, as an example, is the *Resource Connect* (0x0127) message with the PVD and AMD Parameters TLVs. Sending the information below will enable CPE Events 0x42 to 0x48 in the *Call Processing Event* (0x2E) message (refer to the *API Reference*).

0x0619 PVD Parameters TLV

00 06 Length
00 23 Time Constant
FF EC Maximum Noise Level
00 03 Noise Gating Sensitivity

0x0620 AMD Parameters TLV

00 16 Length
01 2C Frequency Band 1: Min. Detection Freq. 1 Hz set to 300
0B B8 Frequency Band 1: Max. Detection Freq. 1 Hz set to 3000
02 EF Frequency Band 2 :Min. Detection Freq. 2 Hz set to 51
06 40 Frequency Band 2: Max. Detection Freq. 2 Hz set to 1600
06 41 Frequency Band 3: Min.Detection Freq. 3 Hz set to 1601
0B B8 Frequency Band 3: Max. Detection Freq. 3 Hz set to 3000
00 02 Min. Tone on time for declaration of tone present 10 ms to 2
00 0F Min. Tone on time for declaration of tone not present 10 ms set to15
FF CA Min. Level of Tone to declare present dBm set to -35
00 1E Min.Voice On Time for declaration of AMD_VOICE present 10 ms set to 30
00 0F Min. Voice Off Time for declaration of AMD_VOICE not present 10 ms set to 15

0x061B PVD/AMD Reports TLV

00 02 Length
00 0F Set to report all events Silence, PVD, Tones, and AMD

2. To disconnect the channel from the DSP resource, use the *Resource Disconnect* (0x0128) message.

   AIBs

   Resource A: Span/Channel

   Mandatory TLV

   0x0602 Resource Type

   Resource Type = PVD (0x0108) or AMD (0x0109)

   Optional TLVs

   None

Query Use the *Generic Card Query* (0x0123) message to query the card-level defaults for PVD/AMD.

PVD/AMD Examples of Call Processing Events

Call Processing Event Details

The following examples describe the call processing event details that you would find in the socket.log.

Call Processing Event (0x002E) message and the event is Truly Silent Channel (0x42) (PVD/AMD)

00 15 Message Length

00 2E Message type Call Processing event

00 25 00 00 01 0D 03 00 00 01

42 Truly Silent Channel (PVD/AMD)

38 6D 4A 15 Calendar Time (in seconds)

00 14 Milliseconds Remainder (Calendar Time in milliseconds)

32 BE Milliseconds Since Last Event

Call Processing Event (0x002E) message and the event is Short Term Voice Detect Rising Edge (0x43) (PVD/AMD)

00 15 00 2E 00 26 00 00 01 0D 03 00 00 01

43 Event of Short Term Voice Detect Rising Edge (PVD/AMD)

38 6D 4A 1B Calendar Time (in seconds)

00 55 Milliseconds Remainder (Calendar Time in milliseconds)

17 B1 Milliseconds Since Last Event

Call Processing Event (0x002E) message and the event is Short Term Voice Detect Falling Edge (0x44) (PVD/AMD)

00 15 00 2E 00 28 00 00 01 0D 03 00 00 01

44 Event of Short term Voice Detect Falling Edge (PVD/AMD)

38 6D 4A 1C Calendar Time (in seconds)

03 43 Milliseconds Remainder (Calendar Time in milliseconds)

05 A5 Milliseconds Since Last Event

Call Processing Event (0x002E) message and the event is Long Term Voice Detect Rising Edge (0x45) (PVD/AMD)

00 15 00 2E 00 27 00 00 01 0D 03 00 00 01

45 Event of Long Term Voice Detect Rising Edge (PVD/AMD)

38 6D 4A 1B Calendar Time (in seconds)

01 86 Milliseconds Remainder (Calendar Time in milliseconds)

4B 0A Milliseconds Since Last Event

Call Processing Event (0x002E) message and the event is Long Term Voice Detect Falling Edge (0x46) (PVD/AMD)

00 15 00 2E 00 29 00 00 01 0D 03 00 00 01

46 Event of Long Term Voice Detect Falling Edge (PVD/AMD)

38 6D 4A 1C Calendar Time (in seconds)

03 DE Milliseconds Remainder (Calendar Time in milliseconds)

06 D6 Milliseconds Since Last Event

Call Processing Event (0x002E) message and the event is Tone Detect Rising Edge (0x47) (PVD/AMD)

00 17 00 2E 00 FD 00 00 01 0D 03 00 00 01

47 Event of Tone Detect Rising Edge (PVD/AMD)

38 6D 49 BD Calendar Time (in seconds)

03 7A Milliseconds Remainder (Calendar Time in milliseconds)

56 EE Milliseconds Since Last Event

04 B0 Frequency Band Detected (Events 0x47 and 0x48 only)

Call Processing Event (0x002E) message and the event is Tone Detect Falling Edge (0x48) (PVD/AMD)

00 17 00 2E 00 FE 00 00 01 0D 03 00 00 01

48 Event of Tone Detect Falling Edge (PVD/AMD)

38 6D 49 BD Calendar Time (in seconds)

03 C5 Milliseconds Remainder (Calendar Time in milliseconds)

00 4B Milliseconds Since Last Event

00 00 Frequency Band Detected (Events 0x47 and 0x48 only)

Detailed Information

Use **Milliseconds Since Last Event** to derive the signal duration. The PVD/AMD application needs to differentiate between PVD and AMD based on the signal durations.

For example the event of Rising Edge for event 0x43 short term voice and 0x44 Falling Edge short term voice have the data block of **Milliseconds Since Last Event**. If you take the difference between the 0x43 and 0x44 events, that will give you the duration of the signal detected.

When you are just talking into the phone, the blocks of 0x43 and 0x44 (short term) and 0x45 and 0x46 (long term) voice durations will be very similar. In other words, the difference between the on and off signal edges of the **Since Last Event** parameter gives you the duration of the entire signal.

If you are playing on-hold voice or music, the values of the signal durations should be longer at some point. You should see a difference in those measurements.

Examples for Using PVD/AMD

Overview

Prior to using the PVD/AMD feature, you first must have the PVD/AMD function configured on the DSP Series 2 card. Refer to the following examples on how the PVD/AMD feature can be used.

CSP/DSP Series 2 Card

As the events are routed to the host application, the application must make a determination as to whether a person, fax, answering machine, etc. is on the line.

The PVD/AMD events can be received at anytime, so if the application is using these events as triggers to send an API message to the CSP, you must wait for an Acknowledgement to any outstanding API message before issuing a new command for a given span/channel.

Examples of PVD/AMD Feature

Deliver an entire message regardless if it is a person or answering machine

In this example, the application is looking for some initial PVD noise event. On the subsequent falling edge and silence detection, the application starts the message with a *Play File Start* (0x011B) message. If another rising edge or tone is detected the application issues a *Play File Stop* (0x0120) message and waits for another silence before playing the message again. The application continues in this loop until the entire message is delivered without noise interruption. This is done by marking the initial noise, waiting for silence, and then issuing the *Play File Start* message. The application then enters a loop of waiting for the rising edge noise or a play file completed event to release the call.

Deliver two different messages: one for a person and the other for a answering machine

In this example, the application is required to decide on some pre-defined length of noise (for example, <1.5 second but > .25 second) on the first PVD energy detected, followed by some pre-defined length of silence (for example, 1 second) to indicate a person has been detected and issue a *Play File Start* of the "person" message.

If more noise is detected, the application will stop the *Play File Start* message, but then have to make another decision on the party (either answering machine or person based on the duration/type of the noise). Any tone detection is considered an answering machine or fax (if 2100 Hz).

DSP Series 2 to Matrix Controller Series 3 Over Ethernet

Overview

Important! This feature is not supported by the DSP Series 2 Plus card.

The default communications between the DSP Series 2 and the CSP Matrix Series 3 Cards use the mid-plane HDLC bus. Communication over the HDLC bus slows the maximum number of messages that can be transmitted to be less than 100 per second, not allowing the user to achieve high call rates that require DSP services.

Using Ethernet communication for messaging between the DSP Series 2 and CSP Matrix Series 3 Cards allows increased performance by providing more message transactions per second.

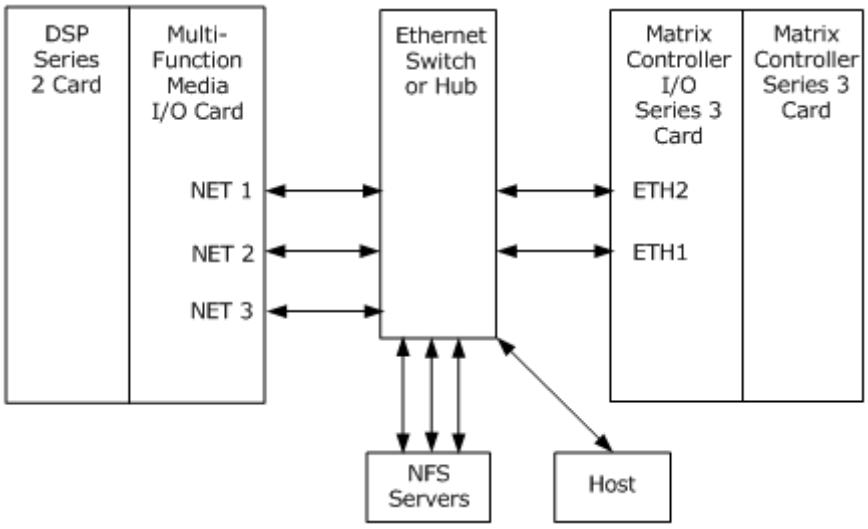
This feature supports a new RCOMM state machine on the DSP Series 2 card that monitors the status of the link between this card and CSP Matrix Series 3 Card. Communication over the HDLC bus will continue to be supported. Licensing of this feature is not required.

Configuration

This feature is configured by using the DIP switch S1, position 6 on the DSP series 2 card. By default, HDLC communication is enabled with DSP switch, position 6 set to ON. Ethernet communication is enabled with DIP switch, position 6 set to OFF. Refer to the CSP Hardware Product Descriptions for more details.

When using the Ethernet communication mode, all the I/O ports are connected to the same physical Ethernet switch/hub. Redundancy at the port level is provided by the RCOMM state machine that monitors the status of the link between the DSP Series 2 and Matrix Controller Series 3 I/O ports. The message packets are sent out of the port which indicates a link-up state. Additional logic provides a fall-back to the HDLC option in the event of a failure of all the Ethernet ports. See the figure below.

Figure 7-13 DSP Series 2 to Matrix Controller Series 3 Ethernet Interconnections



Monitoring

The *Alarm* (0x00B9) message supports the new 0x02 Card Alarm, 0x80 DSP Ethernet Unavailable:

- If at any point the message communication mode is changed from Ethernet to the HDLC bus, an alarm is sent to the host. Refer to the API Reference for more details.
- An *Alarm Cleared* (0x00C1) message is sent if the communication mode is changed back to the Ethernet mode.

8 DSP Series 2 Cards Product Description

DSP Series 2 Card Configurations

The CSP supports the following DSP Series 2 cards and associated I/O cards:

- DSP Series 2 and associated Multi-Function Media I/O
- DSP Series 2 Plus and associated Multi-Function Media I/O Plus

The Overview section in this chapter provides detailed information on the DSP Series 2 Plus card and a comparison to the DSP Series 2 card.

> **Important!** In this document, unless otherwise stated, **DSP Series 2** will be used for both the DSP Series 2 and DSP Series 2 Plus cards. Any software or hardware differences are indicated.

OVERVIEW

- DSP Series 2 Card
- DSP Series 2 Plus Card
- Features
- Architecture Overview
- Hardware
- DSP Resource Points
- DSP Resource Specifications
- Mixing DSP Cards in a CSP
- Default DSP Function Type Configuration
- Redundancy
- Administration
- Media Streaming over RTP

DSP Series 2 Card

Introduction

The DSP Series 2 card is a high-performance media processing resource that is fully integrated into the developer's switching platform, providing a consistent and easy-to-manage integrated telecommunications and media services environment.

The DSP Series 2 eliminates the need for separate voice response units (VRU) by providing powerful media processing services and resources within the DSP Series 2 environment. This also reduces T1/E1 communications, saving service providers both capital expense costs and on-going operating costs.

The DSP Series 2 enables the CSP switching platform to operate as a service node or intelligent peripheral. Configured with DSP Series 2 resources, the CSP can be programmed to inter-operate with speech recognition and/or bulk storage to provide a comprehensive media server solution. The CSP supports the Network File System (NFS) with on-board cache for voice file storage. You can have up to eight NFS servers.

> **Important!** The CSP supports Network File System (NFS) Version 2.

For a comparison to the DSP-ONE card, see *DSP Series 2 and DSP-ONE Cards Compared (8-81)*.

Benefits

The DSP Series 2 was designed to meet the demand for an integrated solution for media-intensive services with the CSP architecture. Design criteria for media-rich resources, streamlined development, cost reduction and operational ease of use results in important benefits for developers, including:

- Ability to offer Media-Rich Revenue Generating Services: the DSP Series 2 offers a rich, programmable environment in which to create innovative services.
- Faster Time-to-Market: since media processing is integrated into the CSP platform, there is no need to develop interfaces between external VRU systems and the service platform.
- Reduced Costs: since media resources are built-in, costs for external VRU systems, plus the cost of T1/E1 service are eliminated
- Ability to provide a single, integrated solution offering.

DSP Series 2 Plus Card

Introduction

The DSP Series 2 Plus card and Multi-Function Media I/O Plus card and the existing DSP Series 2 card and Multi-Function Media I/O card are supported for the 8.4.1 CI software release. This section describes any differences between these CSP cards and includes any additional hardware and software functionality.

The DSP Series 2 Plus card is functionally equivalent to the DSP Series 2 card and is also backwards compatible with the DSP Series 2 card. Both cards can be used in the same CSP chassis.

This section describes any differences between these CSP cards and includes any new hardware and software functionality.

Benefits to Customer

The DSP Series 2 Plus card provides the customer with the following:

- An upgraded platform for future DSP enhancements
- Meets Restriction of Hazardous Substances (ROHS) standard required for shipments to the European Union (EU)

Licensing Configuring and Querying

The licensing requirements for the DSP Series 2 Plus card are the same as required for the DSP Series 2 card.

The DSP Series 2 Plus card is configured and queried in the same way as the DSP Series 2 card.

Users must update their code so that the host will recognize the new DSP Series 2 Plus and Multi-Function Media Plus I/O board IDs.

Ethernet Architecture and Functionality

The DSP Series 2 Plus main board has the capability of providing the following six 10/100 Base-T auto-negotiating signals to the Ethernet ports on the Multi-Function Media I/O Plus card. For the this release of the DSP Series 2 Plus card, only the top two control Ethernet ports CTR 0 and CTR 1 will be used. These two ports are connected to the CPU processor and have their own MAC addresses. These ports provide for communication on the control plane (for example, matrix to host communication, NFS, RTP and redundancy).

> **Important!** The four data ports DATA 0 - DATA 3 which are connected to the internal Ethernet switch will be disabled and reserved for future use.

Ethernet Link Redundancy

The DSP Series 2 Plus card supports Ethernet Link Redundancy over control ports CTR 0 and CTR 1. It does not support physically separated networks connected to each control port. Both ports perform as a logical interface with one port being active at a time. If the active link goes down, the hot standby port automatically takes over.

If the host does not configure the DSP Series 2 Plus for Ethernet Link Redundancy, then only control port CTR 0 should be used.

Hardware Compatibility

- The DSP Series 2 Plus card may reside in a CSP that also contains DSP Series 2 and DSP-ONE cards, and the platform can pool identical resources from these cards and use them together.
- The 2090 chassis can accept a maximum of five DSP Series 2 Plus cards, while the 2040 chassis can accept a maximum of two DSP Series 2 Plus cards.
- The DSP Series 2 Plus card may **not** reside in an CSP that also contains MFDSP cards.

Modules and DSPs

The DSP Series 2 Plus card comes with either one or two DSP 2 Modules with four DSPs each, for a maximum of eight DSPs per card.

RAM, Flash Memory and Processor Speeds

The DSP Series 2 Plus card main board has 128 Mbyte of external synchronous SDRAM and 2 MBytes of Flash memory. Both the CPU and I/O processors are faster. The CPU supports a 450 Mhz processor and the I/O supports a 100 Mhz processor.

I/O Card

The DSP Series 2 Plus card requires the Multi-Function Media I/O Plus card for I/O functionality.

API Changes

The DSP Series 2 Plus functionality does not require any new API messages. The following two API messages have been modified:

- *Card Status Query* (0x0083)
- *Card Status Report* (0x00A6)

The Card Type field has been modified with new card types and card IDs as follows:

- 0x90 DSP Series 2 Plus
- 0x92 Multi-Function Media I/O Plus

DSP Series 2 Cards Comparison

Comparison	DSP Series 2 Plus	DSP Series 2
Hardware		
RAM and Flash Memory	128 Mbyte of external synchronous SDRAM and 2 MBytes of Flash memory	128 Mbyte of external synchronous SDRAM and 2 MBytes of Flash memory
Processor Speeds	CPU processor 450 Mhz I/O processor 100 Mhz	CPU processor 300 Mhz I/O processor 83 Mhz
Maximum number of DSP chips	Eight (8)	Eight (8)
I/O Card	Uses the Multi-Function Media I/O Plus card for File Record/Playback	Uses the Multi-Function Media I/O card for File Record/ Playback
Maximum number of Ethernet ports	Six (6) ports connected as follows: Two (2) control ports connected to main board CPU Four (4) data ports connected to an internal Ethernet switch	Three (3) ports connected to an internal Ethernet switch
Software	Same functionality	Same functionality
Licensing	Same Licensing requirements	Same Licensing requirements
API Messages	*Card Status Query* (0x0083) *Card Status Report* (0x00A6) New card types: 0x90 DSP Series 2 Plus 0x92 Multi-Function Media I/O Plus	No change

Features

Introduction

The DSP Series 2 offers a comprehensive set of features to support media processing and management for CSP-based solutions. Highlights include:

Announcement Processing

The system supports NFS based announcements, with on-board cache for voice file storage. Multiple NFS servers can be supported; the system manages the location of files through an ASCII file.

Dynamic Voice Recording and Playback

DSP Series 2 supports on-board temporary recording storage as well as external disk-based long term storage.

High Density Tone Receivers

Depending on the receiver type, the system supports up to 512 tone receivers per DSP.

Large Conferencing Capability

The system supports up to 256 conferees per DSP, and 128 conferees per conference bridge.

- The user can adjust gain and speed, and skip forward and backward. The DSP Series 2 supports configurable beep tones, configurable silence parameters, and two-way call recording without using conference ports
- The direct interface between the DSP card and the file server uses a 100 mbps I/O interface, off loading CPU processing, which in turn reduces overhead and improves performance.

Echo Cancellation

The DSP Series 2 card Echo Canceller removes echoes caused by signal leakage at hybrid telephone line interfaces. You may want to implement an Echo Canceller for tandem calls on trunks with echo, or to clean an incoming signal before connecting to a media resource, such as a Voice Response Unit or Answering Machine Detection.

Positive Voice Detection/ Answering Machine Detection

You can configure the CSP to analyze incoming PCM data and detect voice or an answering machine. You can set PVD/AMD parameters for a DSP Series 2 card and for an individual channel.

Media Streaming over RTP

RTP, the real-time transport protocol, provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality-of- service for real-time services.

Chip Monitoring

This feature provides the Chip Monitoring function that monitors the state of the DSP chips on the DSP Series 2 card.

At startup, the Chip Monitoring function constantly polls all the DSP chips to make sure they are operating. If a DSP chip does not respond to three consecutive poll messages, the Chip Monitoring function tries to reset the chip and sends a DSP Alarm (DSP Chip Reset 0x08) to the host. If the chip activates after being reset, the DSP Chip Reset alarm is cleared. If the chip does not activate after three attempts to reset it, then the chip is designated as out of service and the DSP Alarm (DSP Out of Service 0x01) is sent to the host. The Stop Monitoring Event is sent to the chip whenever the host takes that chip out of service.

The *Alarm* (0x00B9) message, 0x06 DSP Alarms, enables the DSP Chip Reset (0x08) alarm. This alarm indicates that the DSP chip has been reset internally after an error occurred. This alarm is cleared when the DSP chip resumes operation after reset.

Frequency Shift Keying

The Frequency Shift Keying (FSK) feature, supported by the DSP Series 2 card, involves the transmitting and receiving of FSK modulated signals. FSK is a modulation technique used by modems in which two different frequencies are used to represent the binary state of 1s and 0s as tones across telephone lines. This feature supports both European Telecom (ETSI) and China Telecom (CTSI) SMS protocols and Caller ID.

Receiving FSK

The Receiving FSK feature uses the *Resource Connect* (0x0127) and *Resource Information Notify* (0x0141) messages that allow the host to receive FSK data from an off-hook channel.

Currently, the CSP supports sending Frequency Shift Keying (FSK) data containing On-hook Caller ID in the *Outseize Control* (0x002C) message and Off-hook Caller ID in the *Resource Connect* (0x0127) message. It also supports sending and receiving FSK data containing SMS DLL messages that comply with ETSI and CTSI (China Telecom) call flows.

Refer to *Frequency Shift Keying (7-69)*, for detailed information modified API messages and TLVs. Also refer to the detailed information in the API Reference for the modified APIs and TLVs.

Architecture Overview

In keeping with Dialogic's commitment to open, programmable services, the DSP Series 2 delivers the following key architectural advantages:

- Modular Design
- Scalability
- High Performance and Reliability

Modular Design

The DSP Series 2 takes advantage of Dialogic's commitment to technology re-use by utilizing the motherboard from our IP Network Interface Series 2 card, based on Motorola Power QUICC 8260 technology with 128 MB of SDRAM.

The DSP supports on-board Ethernet switch, with eight internal and three external ports. External ports can be aggregated supporting up to three 100 mbps streams.

The system supports up to two modules per board; each module includes four DSPs. Each DSP supports up to four separate functions, two for transmit and two for receive, and each stream supports up to 256 DS0s.

The product is supported by a variety of message and query statistics, including cache statistics and function usage statistics, so that developers and service providers can monitor their services and optimize their DSP configurations.

Scalability

The DSP Series 2 card is offered in two configurations:

- One module, with 2048 resource points total
- Two modules, with 4096 resource points total

Additional resource points can be licensed on a per system basis in 2048 resource point increments.

The NFS service is optional and is licensed separately.

High Performance and Reliability

The DSP Series 2 card is supported by a pooling scheme for dynamic resource allocation. Resource points are allocated only when they are used. This provides flexibility in the way that resources are allocated at any given point in time, and results in better utilization of resources and improved performance.

To achieve high reliability, the DSP Series 2 card supports card-level hardware redundancy coupled with dynamic resource allocation. If one DSP Series 2 card fails, the system CPU will remove the resource points that are included in the base modules from the resource pool, but will reallocate licensed capacities to the remaining in-service hardware in order to preserve the maximum resources possible.

Every port in the CSP has direct access to every DSP resource (except for channels in Gateway Mode. See the *Developer's Guide: IP* for more information on Gateway Mode). A bi-directional PCM bus allows simultaneous incoming and outgoing operations, such as tone reception and tone generation.

You can upgrade your system with additional DSP cards as your needs grow, and multiple cards can load-share. The DSP cards are hot-swappable, which simplifies maintenance and upgrades, and increases overall reliability.

Hardware

Compatibility

The DSP Series 2 card may reside in an CSP that also contains DSP-ONE cards, and the platform can pool identical resources from these cards and use them together.

The 2090 chassis can accept a maximum of five DSP Series 2 cards, while the 2040 chassis can accept a maximum of two DSP Series 2 cards.

The DSP Series 2 card may **not** reside in an CSP that also contains MFDSP cards.

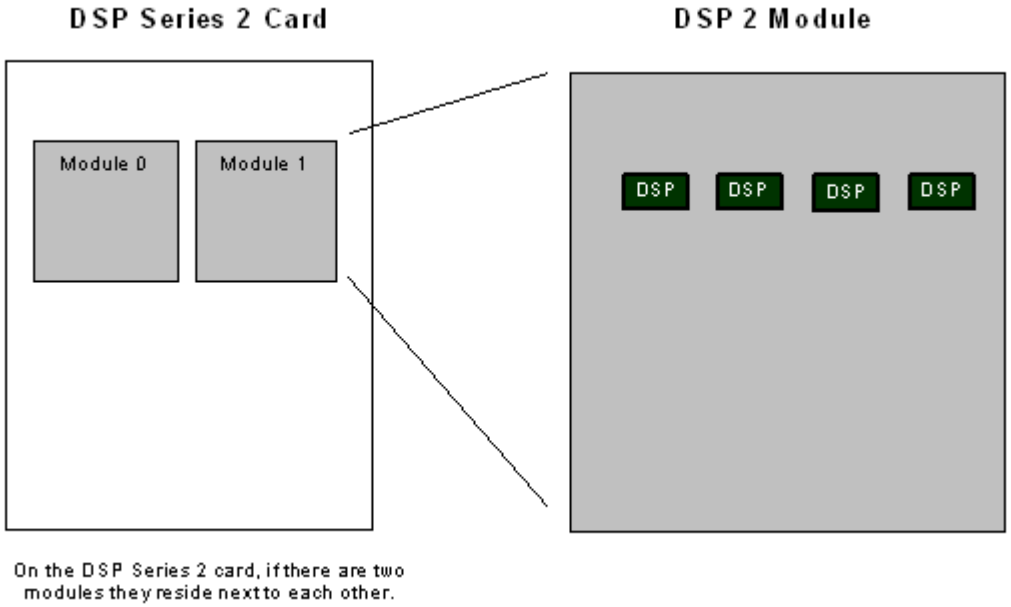
I/O

The DSP Series 2 card requires a Multi-Function Media I/O card for File Record/Playback and Fax functionality.

Modules and DSPs

The DSP Series 2 card comes with either one or two DSP 2 Modules with four DSPs each, for a maximum of eight DSPs per card.

Figure 8-1 Modules on the DSP Series 2 Card



DSP Resource Points

Overview

The DSP Series 2 card uses a pooling scheme for dynamically allocating resources through Resource Points. Resource Point licensing provides the DSP Series 2 with flexibility, scalability, and redundancy. Each function type has associated licensing per channel, measured in Resource Points. The total available Resource Points used for Tone Reception, conferencing, and File Playback/Record are managed as a resource pool.

Standard Resource Points

The DSP Card comes standard with 2,048 for each module. You can purchase additional Resource Points in increments of 2,048. When you insert the card in the chassis, the Resource Points are reported to the Matrix Controller card and added to the pool.

Table 8-1 Resource Points Model Numbers

Model Number	Description
CSP-DSP-1310	One module with 2,048 default Resource Points. Equivalent to a DSP-ONE card with one module.
CSP-DSP-1320	Two modules with 2,048 default points each, for a total of 4,096 Resource Points. Equivalent to a DSP-ONE with 2 modules.
CSP-BIO-1000	The Multifunction Media IO card. This card must be purchased for every DSP Series 2 card for which play or record functionality is to be used (not including temporary playback/record functionality).
CSP-LDP-1000	An additional 2,048 resource points

Benefits

- Flexibility

  Resource Points are allocated only when they are used. This results in better utilization of resources and improved performance.

- Scalability

  Licensing DSP resources through Resource Points, you can buy only as much functionality and capacity as you need, and upgrade incrementally as your needs increase.

- Redundancy

  Licensed Resource Points are made available to the other DSP Series 2 cards in the CSP. If a DSP Series 2 card fails, its licensed Resource Points (but not the standard Resource Points that come with the card) remain in the pool to be used by the remaining cards.

Resource Point Checking

When a DSP function is requested, the Matrix Controller card checks that there are enough total Resource Points available to provide the function and that a DSP chip with sufficient channels is available for that function type. If there are no more Resource Points available, or if an appropriate DSP chip is not available, the Matrix Controller card returns a NACK.

DSP Streams

The concept of DSP ***streams*** is integral to DSP licensing on the DSP Series 2 card. Each DSP chip on the DSP Series 2 card supports four one-way (simplex) paths: two for transmitting and two for receiving. Each of these paths is called a DSP stream.

Based on these four streams, each DSP chip can be assigned up to four function types. Each stream can support a maximum of 256 channels, depending on available licensed resources and on the function type assigned. Note that Conferencing and File Playback/Record requires two streams (one for transmit and one for receive).

Resource Points on Card Failure

If a DSP Series 2 card fails, the standard Resource Points are removed from the resource pool. However, all Resource Points purchased through additional licenses remain available in the pool, and may be used by the remaining cards in the CSP.

Configuration

See Downloading License Keys to the CSP.

Determining your DSP Resource Point Needs

Introduction

Use the following steps to determine how may Resource Points you need to license, based on your resource and hardware needs.

1. Determine how many resources you need
2. Determine how much hardware you need
3. Determine how many Resource Points you need

Determine how many DSP Resources you Need

How many resources for each of the following do you want to use:

- Tone receivers
- Tone transmitters
- Conferences
- File Playback/Record

Typically, you determine the amount of resources you need by calculating the Busy Hour Call Attempts (BHCA) and percentage of each call that a particular resource type is used.

Because Resource Points are pooled and dynamically allocated, the DSP Series 2 card provides a new flexibility in distributing resources at any given moment.

Resource Points may be used for any configured function type, and are removed from the pool for only the length of time that the given function type is in use.

Your hardware may have more bandwidth available than you calculated for per stream, for each function type. So the actual allocation of functions is dictated by demand from the host application. Dialogic recommends that you license enough resources for the highest likely demand.

To help you determine how many resources you need, Dialogic has added statistics functionality to the system software. You can receive statistics on many aspects of DSP resource usage and allocation, including the following:

- Fixed memory for all individual DSP chips (total blocks used, average blocks used, and maximum blocks used)
- Cache for all individual DSP chips (accesses, misses, hits, and average free blocks)

- DSP Functions for individual DSP chips (total requests for a function, average and minimum free channels)
- NFS statistics on all individual DSP chips, including NFS for all individual DSP chips (total number and size of NFS reads, writes, and records) as well as read and write time delays (minimum, maximum, and average)

Determine how Much Hardware you Need

When you have determined how many resources of each type that you need, you must determine how many DSP Series 2 cards and the number of modules you need to supply the above resources, by doing the following:

1. Determine the total number of streams you need by dividing the number of resources you need by the maximum number of channels per stream, then rounding up.
2. Determine the number of DSP chips required by dividing the total number of streams by two. You must perform separate calculations for functions that require two streams (File Playback/Record and Conferencing) and functions that require only one stream (transmit or receive).
3. Determine the number of modules needed by dividing the number of chips by four.

You now have the number of two-module cards you need. If you are using one-module cards, multiply by two.

Other Hardware Considerations

Other things you must consider when determining the amount of hardware you need:

- You have a maximum of 512 simultaneous NFS reads per card
- Conferencing must reside on its own DSP chip (no other functions are allowed on a DSP chip with the Conferencing function type)
- File Playback/Record must reside on its own DSP chip (no other functions are allowed on a DSP chip with File Playback/Record function type)
- You have a maximum of five DSP Series 2 cards per CSP 2000 chassis or a maximum of two DSP Series 2 cards per CSP 1000 chassis.
- File Playback/Record functionality requires the Multi-Function Media I/O card.

Determine how many Resource Points you Need to Purchase

To determine the Resource Points per channel required for each function refer to Table 7-2. For example, if you want the maximum MFR1 receivers for a stream, you would need to license 256 x 5 Resource Points (1,280).

When you know the total Resource Points you need, subtract the Resource Points that come with the card. You are then left with the number of Resource Points that you need to license.

The Resource Points and maximum channels per DSP stream are shown in the table below (each module with four DSP chips, each DSP chip with four streams, and assuming a two-module card).

DSP Resource Specifications

Table 7-2 shows the number of DSP resources available per DSP chip, channels per stream, and resource points used for each function on the DSP Series 2 cards.

Table 8-2 DSP Resource Specifications

Number	Function	Resources per DSP Chip (assuming sufficient Resource Points)	Maximum Channels per DSP Stream	Resource Points Used per Channel
Tone Reception				
0x01	DTMF (µ-law)	384	192	5
0x02	MFR1 (µ-law)	512	256	5
0x03	DTMF (A-law)	384	192	5
0x04	MFR1 (A-law)	512	256	5
0x05	MFR2 (A-law)	512	256	8
0x06	MFR2 (µ-law)	512	256	8
0x07	CPA (A-law)	384	192	10
0x08	CPA (µ-law)	384	192	10
0x09	Dial Pulse	N/A		
0x0A	Energy Detection	384	192	10
0x0C	DTMF HPF (µ-law)	N/A		
0x0D	DTMF HPF (A-law)	N/A		
Tone Generation*				
0x30	Universal µ-law		256	0
0x31	Universal A-law		256	0
Conferencing				
0x21	Monitor	256 Resources (128 + broadcast per conference)	128	8

Number	Function	Resources per DSP Chip (assuming sufficient Resource Points)	Maximum Channels per DSP Stream	Resource Points Used per Channel
0x22	Unified	256 Resources (128 + broadcast per conference)	128	8
0x23	DTMF Clamped	256 Resources (128 + broadcast per conference)	128	8
File Record/Playback (requires 2 streams)				
0x1D	File Record/Playback	128**	64	12
Miscellaneous				
0x32	T.30 FAX	12	6	12
0x33	Echo Cancellation	64	32	12
0x34	Positive Voice Detection/ Answering Machine Detection	384	192	5

Mixing DSP Cards in a CSP

Compatibility

IMPORTANT: An MFDSP card may not reside in the same chassis as a DSP Series 2 card.

Card Combinations	Compatible?
MFDSP and DSP-ONE	Yes
DSP-ONE and DSP Series 2	Yes
MFDSP and DSP Series 2	No

Function Allocation

In CSPs that use both the DSP-ONE and DSP Series 2 card, DTMF, MFR1, MFR2, and CPT transmitters all use resources from the DSP-ONE first if it is configured for them. The *RAN Connect* message always uses resources from the DSP-ONE card first. Receivers and Bong Tone are assigned in a round-robin fashion. The chip with the most resources available gets Dynamic Conferencing and Clamped Conferencing first. The chip with the least resources available gets Static Conferencing and Monitor Conferencing first.

DSP-ONE and DSP 2 Sample Configuration 1

The following sample configuration illustrates an CSP with one DSP Series 2 card (with one module) and one DSP-ONE card (with 2 modules)

DSP-ONE card with two DSP Modules

- Seven DSPs DTMF Detection = 154 Ports
- Seven DSPs CPA Detection = 140 Ports
- One DSP DTMF Generation = 2,048 Ports
- One DSP CPA Generation = 2,048 Ports

DSP Series 2 card with one DSP Module

- Two DSPs DTMF Detection and Tone Generation = 1,024 Ports of DTMF Detection, and 1,024 Ports of any tone generation
- Two DSPs CPA Detection and Tone Generation = 1,024 Ports of CPA detection, and 1,024 ports of any tone generation

DSP-ONE and DSP 2 Sample Configuration 2

The following sample configuration illustrates an CSP with one DSP Series 2 card with one module, and one DSP-ONE card with two VRAS modules.

DSP-ONE card with two VRAS Modules

- Three DSPs DTMF Detection = 66 Ports
- Three DSPs CPA Detection = 60 Ports
- One DSP DTMF Generation = 2,048 Ports
- One DSP CPA Generation = 2,048 Ports
- Two VRAS = 104 minutes of announcements and 110 simultaneous channels

DSP Series 2 card with one DSP Module

- Three/two DSP DTMF Detection = 768 Ports
- Three/two DSP CPA Detection = 768 Ports
- One DSP Record/Playback = Virtually unlimited announcement storage, 128 simultaneous channels.

Default DSP Function Type Configuration

Table 8-3 Default Function Configuration

Module Number	DSP Number	Rcv 0	Txmt 0	Rcv 1	Txmt 1
0	0	DTMF µ-law (0x01)	Universal µ-law (0x30)	DTMF µ-law (0x01)	Universal µ-law (0x30)
0	1	DTMF µ-law (0x01)	Universal µ-law (0x30)	DTMF µ-law (0x01)	Universal µ-law (0x30)
0	2	CPA µ-law (0x08)	Universal µ-law (0x30)	CPA µ-law (0x08)	Universal µ-law (0x30)
0	3	File Record/ Playback (0x1D)	Not allowed (0x00)	File Record/ Playback (0x1D)	Not allowed (0x00)
1	0	DTMF µ-law (0x01)	Universal µ-law (0x30)	DTMF µ-law (0x01)	Universal µ-law (0x30)
1	1	DTMF µ-law (0x01)	Universal µ-law (0x30)	DTMF µ-law (0x01)	Universal µ-law (0x30)
1	2	MFR1 µ-law (0x02)	Universal µ-law (0x30)	MFR1 µ-law (0x02)	Universal µ-law (0x30)
1	3	MFR1 µ-law (0x02)	Universal µ-law (0x30)	MFR1 µ-law (0x02)	Universal µ-law (0x30)

Redundancy

Introduction

To achieve high reliability, DSP Series 2 cards supports card-level hardware redundancy coupled with dynamic resource allocation.

The DSP Series 2 card offers redundancy in several forms, on several levels:

- DSP resources are pooled, so you can have extra licensed Resource Points available for backup.
- You can store voice files on remote Network File System servers, so the files are not lost if a card fails.
- You can distribute functions types across two or more cards.
- You can use redundant servers with RAID to mirror the content among multiple drives on a server.

Redundancy Through Pooling

The DSP resources on DSP Series 2 cards are pooled. That is, all resources licensed through Resource Points are available to the entire system and are shared over all DSP Series 2 cards. The 4096 DSP Resource Points that come with a two-module card by default (2,048 for a one-module card) are also pooled. However, if a DSP Series 2 card fails, its default Resource Points are subtracted from the pool. The resources attached to a call must be reconfigured by the host application to another DSP Series 2 card.

Because licensed Resource Points are made available to the other DSP Series 2 cards in the CSP, having a surplus of available licensed Resource Points provides another method of redundancy.

Redundancy through Distributed Function Types

Dialogic recommends that you assign the same functions across two or more DSP Series 2 cards, so that if one card fails, the function is still available on another card.

Redundancy through Network File System servers

If a DSP Series 2 card fails, the voice files stored on it are lost from volatile memory. But you can restore those files from a Network File System server. For redundancy and bandwidth considerations, Dialogic recommends using a machine for NFS that is separate from your application host computer. The I/O card is required to use the NFS features.

Redundancy through Multiple Network Files Servers

You can have multiple NFS servers. This is not a 1:1 redundancy scheme, but you can put the same files in two different places, and there is load sharing between the servers.

Redundancy Between Server Drives

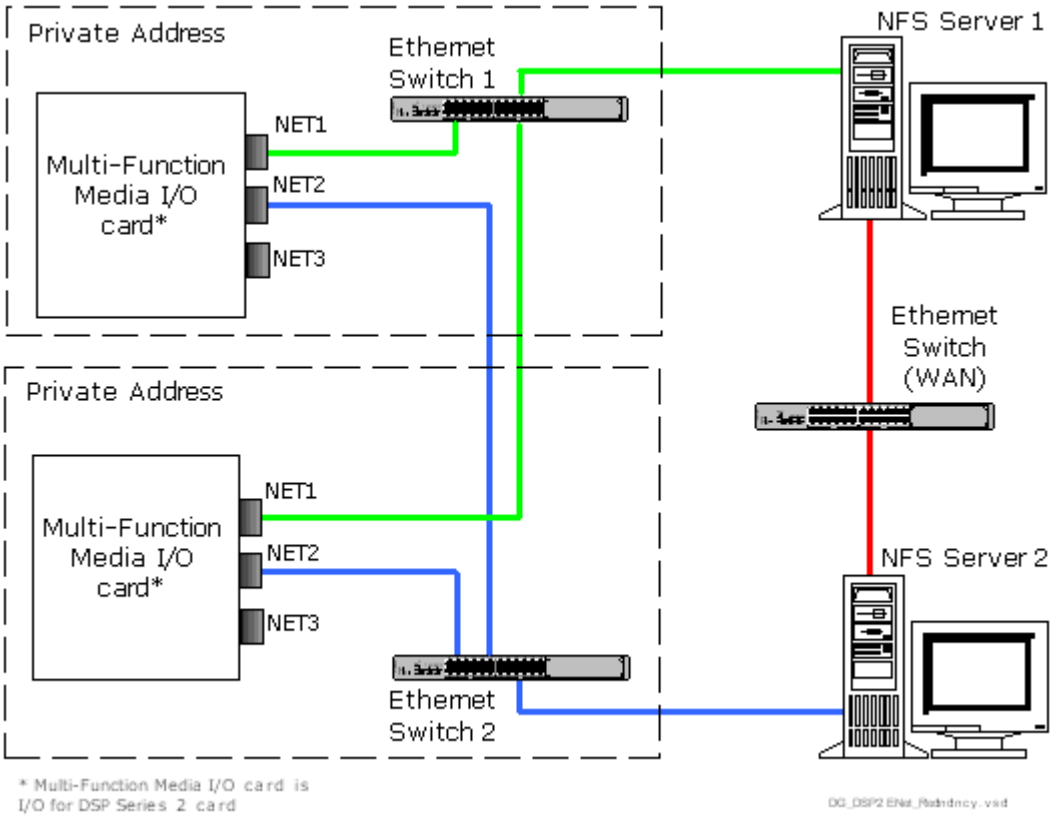
There is internal redundancy on each server, because data is mirrored among multiple drives using Raid 5.

Redundancy through the Ethernet ports on the I/O card

The Multi-Function Media I/O card has three Ethernet ports, NET1, NET2 and NET3, that provide several levels of redundancy. When connecting the I/O card Ethernet ports, the following conditions apply:

- You can connect only one Ethernet port per card to a specific Ethernet switch and associated NFS server.
- If used, each of the other ports on that card must also be connected to a specific Ethernet switch and associated NFS server.
- For each I/O card, if you wanted to use all three Ethernet ports, you would need three Ethernet switches and NFS servers.

The diagram below shows Ethernet ports NET1 and NET2 in use. NET3, although functional, is not being used.



Ethernet Link Redundancy

Overview

The DSP Series 2 card functionality support the *IP Address Configure* (0x00E7) message in order to enable Ethernet Link Redundancy.

To enable this feature, the *IP Address Configure* message includes the following TLVs:

- Ethernet Link Redundancy (0x09) TLV to enable and disable Ethernet Link Redundancy
- Activate Ethernet Port (0x0A) TLV to select the active port with the values being 1 (NET1/upper port), 2 (NET2/middle port), or 3 (NET3/lower port).

While the DSP Series 2 card is in Ethernet Link Redundancy mode, all three external ports NET1, NET2 and NET3, located on the Multi-Function Media I/O card, can be used.

Important! Only one port can be the active redundant port.

Important! By default, Ethernet Link Redundancy will be disabled to provide complete backwards compatibility.

Host Support

To support Ethernet Link Redundancy, the host can do the following:

- Enable/disable Ethernet port redundancy using the Ethernet Link Redundancy (0x09) TLV within the *IP Address Configure* (0x00E7) message.
- Force an Ethernet Link Switchover using the Activate Ethernet Port TLV (0x0A) within the *IP Address Configure* (0x00E7) message.
- Query the Ethernet Mode and Active Link via *IP Address Query* (0x00E6) message.

Alarms

When a Link is detected as being down on the Active port, the DSP Series 2 card automatically detects the condition and switches over to an available Standby port without host intervention. The switchover response time is less than two seconds, so that NFS file read and write functions are not impacted.

The DSP Series 2 card generates the following alarms when relating to the status of the ports. Refer to the *Alarm* (0x00B9) message, 0x02 Card Alarms. When resolved, these alarms are cleared by sending the *Alarm Clear* (0x00C1) message.

- An Ethernet Link Failure (Major, 0x22) alarm when the link is detected as being down on any of the three Ethernet ports.
- An Ethernet Link-Up (Informative, 0x39) alarm, when the link is detected as being up on any of the three Ethernet ports.

Important! The Ethernet Link-Up (0x39) alarm is not cleared, and is sent only when an Ethernet port link has been detected for the first time.

- An Ethernet Port Switchover (Informative, 0x4F) alarm is sent if the active Ethernet port configuration changes while in Redundant Port mode. This alarm can not be cleared.

The Ethernet mode and Active port configured via *IP Address Configure* (0x00E7) message is retained after a soft reset.

Administration

Statistics Monitoring

Use the Statistics Query 0x0121 message to retrieve the following information:

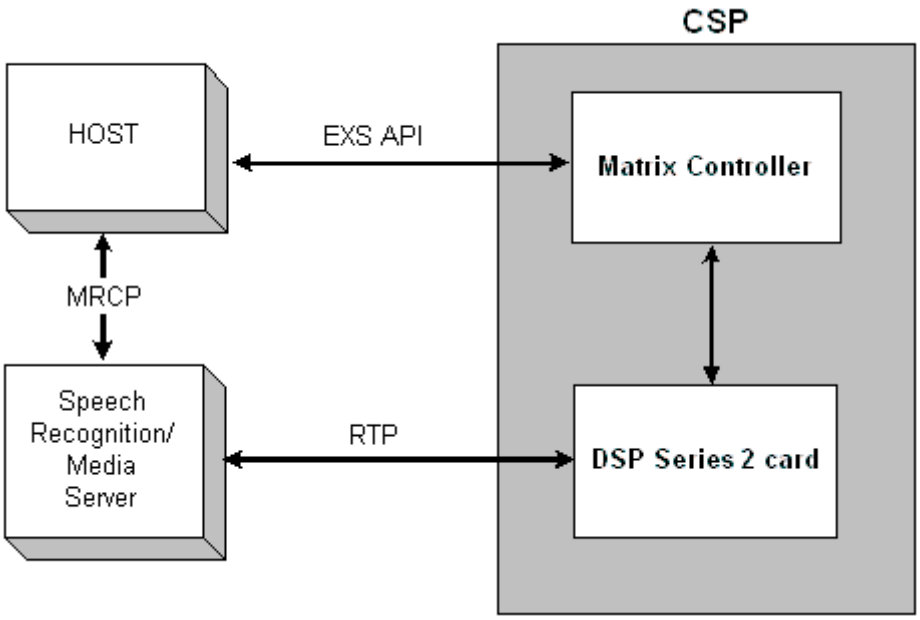
- Network File System (NFS) statistics per card, including total number and size of NFS reads, writes, and records, as well as read and write time delays (minimum, maximum, and average)
- Function statistics on individual DSP chips, including total requests for a function, average free channels, and minimum free channels
- Cache statistics on all individual DSP chips, including accesses, misses, hits, and average free blocks
- Fixed memory statistics on all individual DSP chips, including total, average, and maximum blocks used.

Media Streaming over RTP

Introduction

RTP, the real-time transport protocol, provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality-of- service for real-time services.

The DSP Series 2 card supports the direct termination of G.711 u-law RTP streams (payload size: 20 milliseconds) from any third party media server or speech server, required for Automatic Speech Recognition (ASR), Text To Speech (TTS), or any CTI applications using voice recognition.



Configuration

See *Media Streaming over RTP* in Chapter 8 to configure this feature.

Definitions

The following definitions are taken from *RFC 1889, RTP: A Transport Protocol for Real-Time Applications*.

RTP Payload

The data transported by RTP in a packet, for example audio samples or compressed video data.

RTP Packet

A data packet consisting of the fixed RTP header, a possibly empty list of contributing sources, and the payload data. Some underlying protocols may require an encapsulation of the RTP packet to be defined. Typically one packet of the underlying protocol contains a single RTP packet, but several RTP packets may be contained if permitted by the encapsulation method.

Port

The abstraction that transport protocols use to distinguish among multiple destinations within a given host computer. TCP/IP protocols identify ports using small positive integers. RTP depends upon the lower-layer protocol to provide some mechanism such as ports to multiplex the RTP and RTCP packets of a session.

Transport Address

The combination of a network address and port that identifies a transport-level endpoint, for example an IP address and a UDP port. Packets are transmitted from a source transport address to a destination transport address.

RTP session

The association among a set of participants communicating with RTP. For each participant, the session is defined by a particular pair of destination transport addresses (one network address plus a port pair for RTP and RTCP).

FILE RECORD AND PLAYBACK

Overview

Features

The DSP Series 2 card has the following File Record/Playback functionality:

- Large Vocabulary/Disk Based Files
- Permanent File Recording (stored on an NFS server)
- Temporary Recording (stored on the card)
- Configurable Beep Tone (notifies parties that they can begin recording their voice)
- Gain/Speed Modification
- Skip Forward/Backward (in 100 ms increments)
- Silence Threshold Modification
- Conference Recording
- Pause and Resume
- Dual channel recording
- Moving temporary recordings from the card's cache into permanent storage on a server

Supported Encoding Formats

See *Encoding Formats (8-36)* for more information.

Configuration

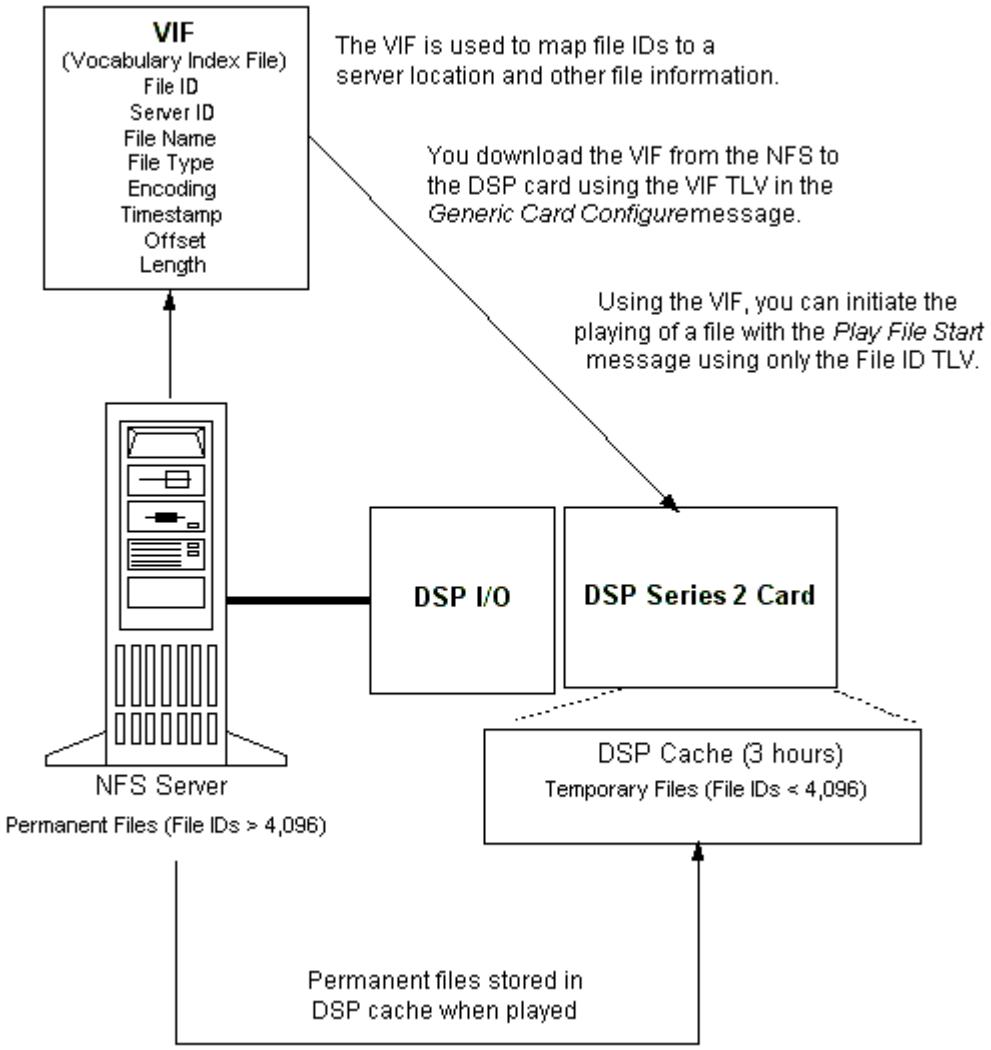
See Record/Play Files.

File Storage

Introduction

Recorded voice files are stored remotely on one or more separate Network File System servers and in volatile memory. Each DSP Series 2 card can store up to three hours of files in a local cache. This cache holds files to be played as well as temporary recordings. Files are stored in cache blocks of one second, and longer messages link multiple blocks together.

The File Record/Playback function supports the playing of pre-recorded voice files referenced by a universal, 32-bit file ID. Although it is not necessary to use a unique file ID for each file, having a unique ID allows you to track the status of individual files in alarms and status messages.



Vocabulary Index File

You must use an ASCII Vocabulary Index File (VIF) to track File IDs below 0x00100000.

If the File ID is 0x00100000 or higher, you need to include the TLV 0x05E1 (File Format) and TLV 0x05E2 (File Location), in addition to the TLV 0x05E0(File ID) in order to access them.

You can modify the VIF using any ASCII text editor. Make sure that the File IDs you use for temporary, internal recordings are always different from the File IDs you use for Playback Files. Consider designating one range for the first type and another range for the second type.

Make sure that the File IDs you use for temporary, internal recordings are always different from the File IDs you use for Playback Files. Consider designating one range for the first type and another range for the second type.

Each line of the file corresponds to a valid File ID that can be played. The information elements on each line must be separated by white space, and each line must end in a carriage return and a line feed.

The size of the VIF is 2 MB. The size allows the DSP Series 2 card to handle more announcements, an important capability when adding multiple language support for global customers.

A VIF supports up to 40,000 lines without noticeable pre-start delay. If a VIF has greater than 40,000 lines, it may take approximately one second before the file starts to play.

When the VIF is updated, all caches are cleared.

See VIF Format.

Vocabulary Index File (VIF) Bypass

You can indicate a cached file’s location, format, and encoding type in the *Play File Start* message (0x011B) instead of using the VIF.

NFS File Failure

If an NFS file fails and is not covered by redundancy, the CSP retries the file a number of times (you can configure this number). If the file fails to play after the configured number of retries, the file is designated as "bad" and there will be no more attempts to play the file. This way, other files from other servers are not delayed.

Backward Compatibility

You can use the *Recorded Announcement Connect* (0x0055) message with the DSP Series 2 card, but only for Recorded Announcement IDs below 4,096 stored on the DSP Card. To store files on an NFS and to take advantage of all the voice file functionality of the DSP Series 2 card, you must use the *Resource Connect* message.

Permanent Recordings

You should use higher File IDs for longer files and for files you want to keep for a significant period. File IDs of 0x00100000 or higher are not stored on the card. They are stored on a separate NFS server, accessed through the I/O card. These files are not stored in cache, but as they are being played, they temporarily use a small amount of cache, so you should consider this when designing your cache management.

If the file ID is 0x00100000 or higher and the DSP Series 2 card is configured for Network File System (NFS) the DSP Series 2 card automatically starts retrieving this file from the NFS server. The location of the file and any other information that is needed about this file such as encoding, offset, and length, is sent in either the *Play File Start* or *Record File Start* message. Files with these high IDs cannot be chained.

Temporary Recordings

Temporary recordings, such as those used for directory assistance requests, should use File IDs below 0x00100000. The host application must play back the temporary file on the same DSP Series 2 card that did the recording.

These temporary files are erased after being played to clear space for newer files. However, because you can't replicate temporary recordings such as directory assistance requests once they are erased, they are always erased last.

Files below 0x00100000 originate on the NFS server but they are cached on the DSP Series 2 card. Make sure that the File IDs you use for temporary recordings are always different from the File IDs you use for permanent files. Consider designating one range for the first type and another range for the second type.

If the DSP Series 2 card is configured to use NFS and the File ID is below 0x00100000, the DSP Series 2 card first checks its cache. If the File ID is not found in cache, the DSP Series 2 card checks the NFS server.

File Chaining

Using the *Play File Start* message, you can transmit up to 32 chained voice files stored on the card. Only file IDs listed in the Vocabulary File Index can be chained. File IDs below 4096 support existing RAN messages, and up to 64 of these can be chained.

Summary of File IDs

	0x00100000 and higher	4096-FFFFF	File IDs 0-4095
Storage	Stored on an external network server, accessed via I/O card	Cached on the DSP 2 card after being downloaded once	Cached on the DSP 2 card
Chaining	Cannot be chained	Up to 32 can be chained, using the *Play File Start* message	Up to 32 can be chained using the *Play File Start* message Up to 64 can be chained using the *Recorded Announcement Connect* message
Tracking	Use the File Location TLV or VIF Bypass feature	Use the Vocabulary Index File or VIF Bypass feature	Use the Vocabulary Index File or VIF Bypass feature
Compatible with RAN messages	No	No	Yes
Queueing	Yes	Yes	Yes, using the *Play File Start* message

Encoding Formats

Supported Formats

The DSP Series 2 card supports both the Raw and .wav file formats for the following encoding formats:

- A-law (0x00)
- µ-law (0x01)
- G.726 32 Kbps (0x02)
- 32 Kbps OKI ADPCM (0x03)
- 24 Kbps OKI ADPCM (0x04)
- 16-bit linear, 11025 Khz (0x05)
- µ-law, 11025 Khz (0x06)
- A-law, 11025 Khz (0x07)
- 8-bit linear, 11025 Khz (0x08)
- 8-bit linear, 8000 Khz (0x09)
- 16-bit linear, 8000 Khz (0x0A)
- G.726 40 Kbps (0x0B)
- G.726 24 Kbps (0x0C)
- G.726 16 Kbps (0x0D)
- * Note that 16 bit uncompressed data requires twice the processor and DSP bandwidth, and will adversely affect channel densities accordingly.

ADPCM

Dialogic supports Adaptive Differential Pulse Code Modulation (ADPCM) for Playback files. ADPCM is a compression algorithm for voice that records only the difference between samples, and dynamically adjusts the coding scale to accommodate large and small differences.

Because OKI requires fewer resources than G.726, OKI is more suitable for resource-intensive applications, such as speeding up or slowing down voice files. And OKI provides voice quality comparable to G.726.

When you choose either OKI or G.726, the caches on both the DSP and the 8260 chip store the data in that encoding format only. Both the Skip Ahead / Skip Back feature and the Speed Up / Slow Down feature take

this file format choice into consideration. The Play File Offset and Length feature does not take this file format choice into consideration. Instead, the host application is responsible for considering that choice.

Compression Benefits

Compared to standard PCM, ADPCM effectively doubles your cache capacity. For two seconds of voice with standard PCM, 16 KB are required. For two seconds of voice with ADPCM, only 8KB are required. Because only half the data is read or written compared to standard PCM, ADPCM also increases performance over the Network File System (NFS) server link.

Compression is performed as the voice is initially recorded, and decompression is performed just before the file is played out. ADPCM does not degrade the performance of the playback or record functionality.

VIF Format

Table 8-4 Vocabulary Index File Format

Element Number	Element Name	Description
1	File ID	A unique number that is used when playing a file out. Valid Range: 0 <= file ID < 0x00100000
2	Primary Server ID	The Primary server where the file is stored
3	Secondary Server ID	The Secondary server where the file is stored
4	File Name	The File Name including the full path (up to 128 Chars)
5	File Type	See the 0x05E1 File Format TLV for values.
6	File Encoding	
7	Timestamp	Date/Time: mm/dd/yyyy hr:min
8	Offset	The offset in bytes where the file is to play from
9	Length	The number of bytes to be played

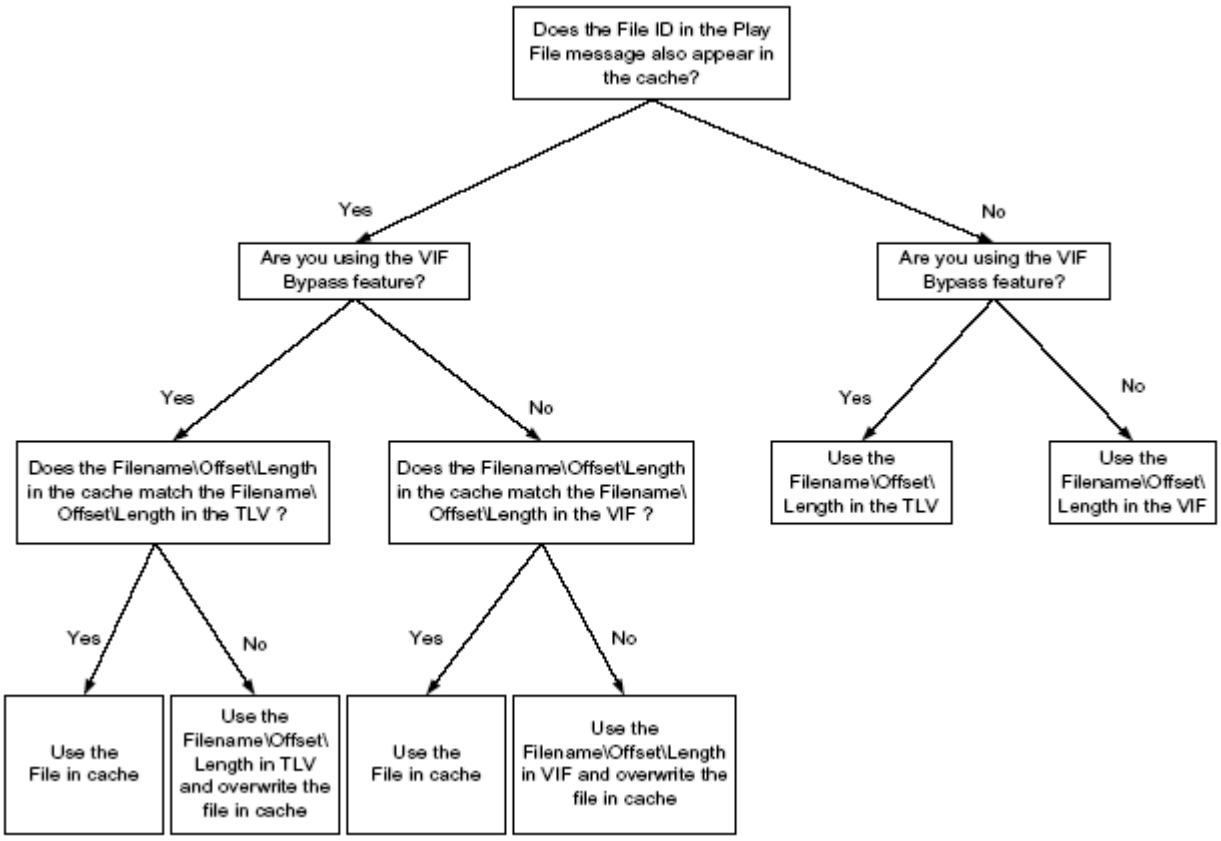
Example The following is an example of a Vocabulary Index File. Note that some, but not all, of the files include the last two fields (Offset and Length)

```
1001 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52 1473  9330
1011 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52 10803 7856
1021 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52 18659 7856
1031 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52 26515 8348
1041 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52
1051 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52 44192 9330
1061 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52
1071 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52 61869 9329
1081 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52 71198 8348
1091 2 /F-0289_Playback_by_File_Offset/count1-30.raw.ulaw 0 1 10/09/2003 15:52 79546 7365
```

File Retrieval Process

The decision tree below shows the process for retrieving a file.

Figure 8-2 File Retrieval Process



File ID The File ID represents not only the file name and location, but also the Offset and Length.

Files IDs can overlap. For example, if one file has an Offset of 8000 and a Length of 16000, another file can have an Offset of 12000 and a Length of 8000.

When the File ID is requested to be played, the DSP Series 2 card checks to see if the File ID is already in cache. DSP Series 2 has one set of options for files already in cache, and another set for files that are not found in cache.

File IDs Already in Cache

If the File ID is already in cache, the DSP compares the Offset and Length in cache either to the VIF or, if using the VIF Bypass feature, to the Offset and Length TLV (see *VIF Bypass*). If the Offset and Length are different in cache, then the file is cleared from the cache and a new cache entry is added from either the VIF or, if using the VIF Bypass feature, from the TLV.

File IDs Not in Cache

If a file is not already in cache, the Offset and Length are specified as follows:

- If you are using a VIF, and the Offset and Length are specified there, then that Offset and Length are used.
- If you are using a VIF, but the Offset and Length are not specified there, then the File Offset and Length TLV is used instead.
- If you are using the VIF Bypass feature, the File Offset and Length TLV is the only way to specify the Offset and Length.

Dynamic File Management

Using NFS, recorded voice files are handled dynamically through a Least Recently Used algorithm. Voice files are downloaded to the card's memory one at a time, and as the memory on the card is consumed, a Least Recently Used (LRU) algorithm keeps track of the order in which the files are played. When the memory is completely consumed, the following takes place:

1. The LRU algorithm identifies the Least Recently Used File ID.
2. This File ID is then deleted.
3. The memory blocks are freed for further use.

The DSP Series 2 card also has one hour of cache per DSP chip. The purpose of this cache is to increase performance for frequently used messages, not to increase storage capacity.

When the host application sends an instruction to play a specific file on a specific DSP chip, the DSP chip then:

1. Checks its one hour cache for the announcement file.
2. Checks the main board's three hour cache for the announcement file.

Checks the Network File System server (if used) for the announcement file.

Playback Features

Playback by File Offset and Length

This feature provides two benefits:

- an application can play from a particular location within a file by providing an offset, in bytes, from the beginning of the file
- a user can specify how long the file should play, so it doesn't have to play to the end

When only part of a file is requested to be played, only the requested bytes are cached (not the entire file) on both the 8260 chip and the DSP. Because the Offset and Length are in bytes, and because they represent bytes in a file, the application is responsible for managing the different encoding formats, and for how bytes should represent time.

Handling Inconsistent File Sizes

If the Length plus the Offset is greater then the actual file size, the file plays until the end. If the Offset is greater then the actual file size, a File Open error occurs.

Dynamic File Management

Using NFS, recorded voice files are handled dynamically through a Least Recently Used algorithm. Voice files are downloaded to the card's memory one at a time, and as the memory on the card is consumed, a Least Recently Used (LRU) algorithm keeps track of the order in which the files are played. When the memory is completely consumed, the following takes place:

1. The LRU algorithm identifies the Least Recently Used File ID.
2. This File ID is then deleted.
3. The memory blocks are freed for further use.

The DSP Series 2 card also has one hour of cache per DSP chip. The purpose of this cache is to increase performance for frequently used messages, not to increase storage capacity.

When the host application sends an instruction to play a specific file on a specific DSP chip, the DSP chip then:

1. Checks its one hour cache for the announcement file.
2. Checks the main board's three hour cache for the announcement file.

Checks the Network File System server (if used) for the announcement file.

CONFERENCING

Contents

Overview

Conferencing Capabilities

The DSP Series 2 card provides the following conferencing capabilities:

- 128 conferences available per DSP stream
- 128 legs/conferees available per conference (all conference types)
- 256 legs/conferees available per DSP chip (you can use half of a chip's legs in any one conference)
- Direct Media Service Access to Conferences
- Manual Volume Control

Features

The robust suite of Conferencing Features includes:

- Automatic Gain Control
- Output Gain Control
- Noise Gating
- Echo Suppression
- Conference Tracking
- Conference Failure Handling

Conferencing Options

You can configure a conference with the following options:

Barge In

DTMF Clamping/Filtering

EXS Conferencing Encoding Type

Conference Failure Behavior

0x05E5 Barge In

0x0604 DTMF Clamping/Filtering Enable

0x0605 EXS Conferencing Encoding Type

0x0606 Monitor Conference Enable

0x060F Conference Failure Behavior

Resource Location

When recording a conference or playing a file to a conference, the conference resources and the record/playback resources must be on the same DSP Series 2 card.

Supported Conference Types

The DSP Series 2 card supports the following conference types:

- Unified (µ-Law, A-Law, mixed, DTMF Clamped)
- Monitor

Manual Input Volume Control

This feature enables a manual volume control on a parties inbound leg to a conference. The Automatic Gain Control for that party is automatically disabled, and the requested gain is added to the conferee's speech prior to being added to the conference.

An example application of this feature would be where an individual wants to ensure that his contributions to the conference are heard by the other participants so adds a few dB of gain to their voice, in effect, raising the volume of their voice without additional effort.

Configuration

See Conferencing.

Unified Conference

The Unified Conference type provides the following functionality:

- Conferees can be µ-Law, A-Law, or mixed. There is no need to specify the encoding when creating the conference.
- Unlimited Broadcast or “listen only” capability (µ-Law and A-Law)
- Dynamically increase conference size

Dynamic Sizing

You can connect channels to a conference even if the number of conferees exceeds the configured conference size, as long as the DSP Chip hosting the conference has available resources.

When the CSP creates a Unified conference, it places the conference on whichever DSP chip has the largest number of free channels at that moment, maximizing the opportunity for that conference to grow dynamically.

Child Conference

You can create a private sub-conference (child conference) within a larger conference (parent conference). When a smaller group of participants within a larger conference want to talk privately, they can do so while still hearing the audio from the larger conference. The sub-conference parties participate with one another in full duplex, while becoming listen-only parties in the larger conference.

The feature allows users to create a child conference, move parties from the parent conference to the child conference and back, and delete the child conference.

No Resource Points are used for a Child Conference unless a leg is added to the child conference. If a leg is added to the child conference, Resource Points are consumed the same way they are consumed in parent conference.

DTMF Clamping

You can enable DTMF Clamping on a Unified conference to prevent DTMF echo. DTMF echo causes false DTMF indications on conference legs. This feature enables users to provide DTMF digit receivers on each conference leg without one conference participant inadvertently controlling another participant’s conference parameters.

Conferencing over an EXNET® Ring

For a Unified conference, two local timeslots per conference are allocated to broadcast over the ring in both µ-law and A-law. In contrast, conference types 0xN7, 0xN8, 0xN9, and 0xNA, use only one local timeslot per conference to broadcast over the ring, in either µ-law or A-law, but not both. Users who use only µ-law do not need to allocate midplane ports for A-law, and vice versa.

Supervisor Conference

This feature provides support for a new conferee type for unified conferences on both the DSP Series 2 and DSP Series 2 Plus cards This feature will handle a typical call center scenario that involves a caller, operator and supervisor.

The caller will be connected to a conference in an input-only mode. The operator and the supervisor will be connected to a conference in two-way mode. The caller and the operator will be connected using a one-way connect message to enable the caller to hear to the operator.

- The operator can talk and hear both the caller and supervisor
- The supervisor can hear the caller, but the caller cannot hear the supervisor.

Configuring this Feature

To configure this feature, the *Resource Connect* (0x0127) message shall be used to connect the caller, agent and supervisor to a unified conference. The caller will be connected in a new talk-only mode. The agent and the supervisor will be connected in the two-way connect mode.

The optional Connection Type TLV (0x0612) will be used to specify the type of the connection to the conference. A party connected in talk-only mode will require the same number of resource points (for example, 8 resource points) as a party connected in two-way mode.The host should be able to connect a one-way (talk only) conferee in a conference to another channel using one-way connect.

The caller channel has to be connected to the conference in one-way talk only mode and the agent channel is connected to the conference in two-way mode. The *Connect One-Way Forced* (0x0050) message, used to specify the agent as source channel and caller as destination channel, shall be used to connect them so that the caller will be able to hear the agent.

The host should be able to query the connection type of a one-way (talk only) conferee. If the host sends the *System Configuration Query* (0x00B4) message with Report Type as Detailed Conference Information 0x10, the *Generic Report* (0x0046) message returns the connection type of a one-way connected (talk-only) conferee as 0x03.

Functional Description

The following functional description provides an example of this feature. Refer to the figure below.

Channel A is the agent, C is the caller and S is the supervisor.

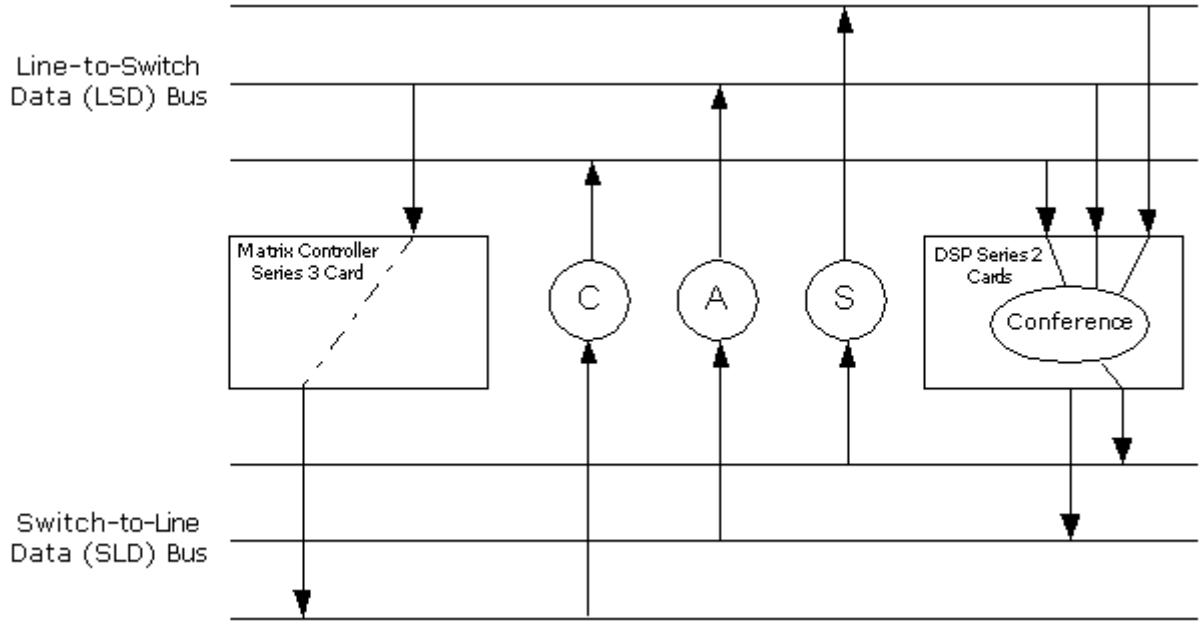
Channels A and S are connected two-way to the conference. So the DSP Series 2 card reads the LSD bus for these two timeslots and transmits the conference output on the SLD bus for these.

Channel C is connected in input-only mode to the conference. So the DSP Series 2 card reads the LSD data for this timeslot and adds it to the conference.

Channel C is connected to channel A using one-way connect. So the CSP Matrix Series 3 Card reads the LSD bus for timeslot A and switches this data on the SLD bus for channel C.

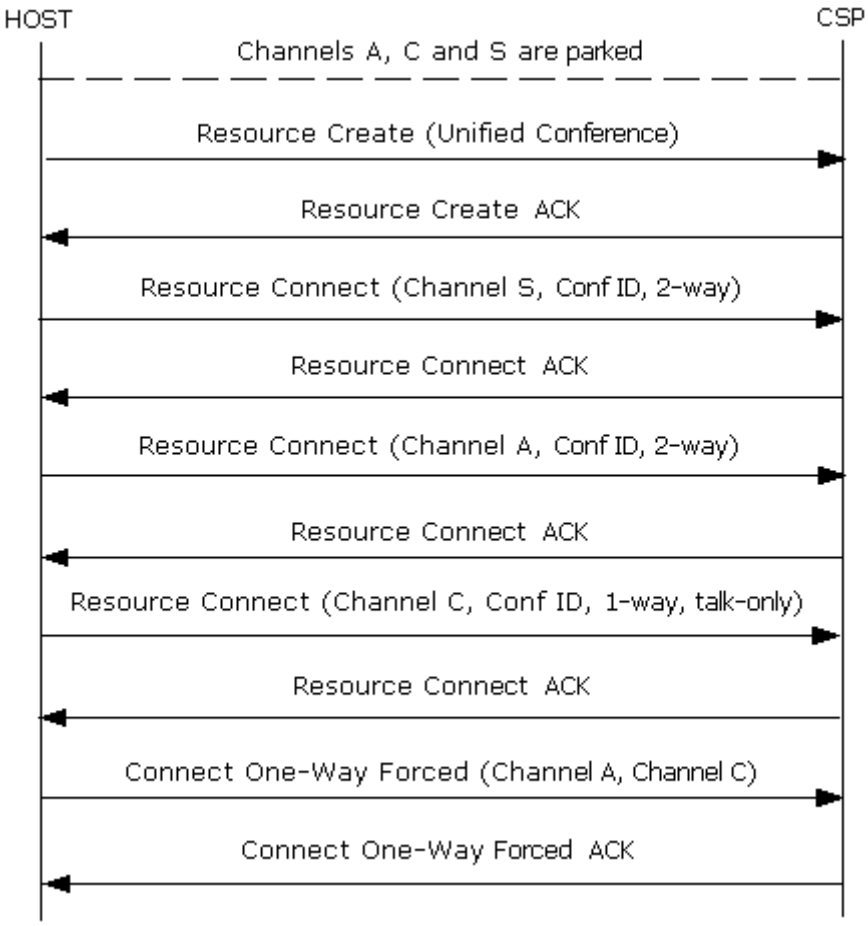
Therefore A and S are able to converse and listen to C. C will be able to hear only A.

Figure 8-3 Conference with Caller, Agent and Supervisor



Call Flow

The following call-flow shows the sequence of messages between the host and the CSP when setting up a unified conference between a caller, agent and supervisor.



Configuration

See Creating a Conference.

Monitor Conference

Overview

A Monitor conference allows one or more monitor channels to listen to as many as nine conversations in a single monitor conference. The source channels do not need to be associated with each other in any way. This feature is useful for applications requiring supervisory monitoring, such as in Automatic Call Distributor (ACD) systems.

Each DSP chip that is configured for the Monitor Conference function type can support source channels of A-law and µ-law encoding, and output either A-law or µ-law according to the configuration of the monitor and source channel(s). At least one DSP chip must be configured this way to enable the Monitor Conference feature.

Monitor conferences are not maintained upon switchover of a Matrix Controller card.

Configuration

See Creating a Monitor Conference.

Conferencing Features

Overview

The robust suite of Conferencing Features includes:

- Automatic Gain Control
- Output Gain Control
- Noise Gating
- Echo Suppression
- Conference Tracking
- Conference Failure Handling
- Channel Selectable Tone Suppression in a Conference
- Increased Number of Conferences
- Configurable Conference Connection Mode

You can configure these features to handle such things as network and signal problems (background noise, echo, and speakers who are too loud or too quiet), conference failure, and conference tracking.

These parameters are especially important for large conferences because as conferences grow, so does the likelihood of impaired signals and networks, and so does the number of users affected. These settings apply to up to two streams of 128 participants.

There are default settings for these features at the card level, and these should work well in most environments. However, you can change the defaults for an entire card, for an entire conference when the conference is created, and for individual conferees (conference legs).

Automatic Gain Control

Automatic Gain Control (AGC) automatically detects variations in volume input and adjusts the input level to be within a decibel range that you can specify, in 1 dB increments. So if a conferee's voice input (one conference leg) is either too quiet or too loud, AGC adjusts the input so that it falls within your specified acceptable range. You can specify the duration (time constant) of the sample used to determine the volume of the conferee's voice, in increments of 5 ms.

Output Gain Control

When output gain (volume) is too low or too high, you can control it using the Output Gain Control feature. You can adjust gain from –40dB to +10dB, in increments of 1dB. The CSP acts upon the most recent

command. For example, if a change is made at the conference leg level and then at the conference level, the conference level setting overrides the leg level setting because that was the last change made.

Noise Gating

During natural breaks in conversation (when the speaker on a channel is not speaking) if significant ambient noise is coming into a conference from the speaker's line, that noise can be gated using the Noise Gating feature. As a conference grows in size, ambient noise is more likely to become a problem.

You can customize the time span (time constant) over which noise is measured, in increments of 5 ms. You can also specify a maximum allowable estimated noise level in 1 dBm increments, from –54 dBm to –10 dBm.

Echo Suppression

If a conference leg is generating echoed signal, you can mute that echoed signal using the Echo Suppression feature. Note how this differs from echo cancellation. Echo Suppression adoptively measures the reflected signal input, and adapts its echo estimate up to a maximum noise value, which you can select in 1 dBm increments, from –54 dBm to –10 dBm.

Conference Tracking

You can query the channel IDs of the three active (loudest) parties that are being summed together and the time they have been active. The report is returned in the *Generic Report* message (0x0046).

Conference Failure Handling

If a DSP card, module, or chip that is hosting a conference fails, all the channels in the failed conferences can now be parked instead of purged, although the default is still to purge. You can override this default for all conferences created, individual conferences, or individual conference legs. The application is responsible for rebuilding the conference on another DSP resource or for releasing the conference legs.

Channel Selectable Tone Suppression in a Conference

This feature provides channel selectable DTMF tone suppression on a per-conferee basis. While connecting a channel to a conference, the host indicates whether it wants DTMF tones coming from the conferenced channel to be suppressed (not heard by other channels) or allowed (heard by other channels). The host is also able to modify the DTMF clamping behavior of a channel that is already connected to a conference. The DTMF Clamping/Filtering Enable TLV (0x0604) supports this feature in the following API messages:

- *Resource Modify* (0x0125) message

- *Resource Connect* (0x0127) message
- *Resource Disconnect* (0x0128) message

Increased Number of Conferences

Currently, the CSP supports 511 non-extended conferences. The 9-bit Conference ID limits the number of conferences per CSP to 511. At this time, only 256 conferences per CSP are allowed to have active broadcasts at any one time. A conference has an active broadcast if it has a one-way (listen only) leg attached to it. This number will be increased to 1023 so that all conferences on a CSP may have active broadcasts at the same time.

This feature allows the application administrator to place all E1 and T1 channels in a two-party conference, the minimum conference size allowed. The Extended Conference IDs are supported on both the DSP-ONE and the DSP Series 2 cards.

The application administrator, such as CSA, is required to configure the CSP to handle 1023 conferences (extended conferencing). The extended conference information is stored with the system information and shared with the standby CSP Matrix Series 3 Card.

This feature is implemented by adding a new configuration type, Extended Conferences Enable (0x017) in the *System Configuration* (0x00AF) and *System Configuration Query* (0x00B4) messages.

Configurable Conference Connection Mode

When the Configurable Conference Connection Mode feature is enabled, and a transmit tone request is received from the host for a channel that is connected to a conference, the voice traffic on the channel input bus will be connected to the conference and remain while the tone/file is in progress.

Default Mode Enabled

When a transmit tone request is received from the host for a channel that is connected to a conference, the following occurs:

- The channel input (LSD) and output (SLD) buses connected to the conference are temporarily suspended until the transmit tone/file is complete.
- After the call progress tone/file is complete, addition of the channel input and output buses to the conference is resumed.
- Incoming voice traffic is enabled.

Configurable Conference Connection Mode Enabled

When the Configurable Conference Connection Mode feature is enabled, and a transmit tone request is received from the host for a channel that is connected to a conference, the following occurs:

- The transmitter (host) temporarily suspends the output bus (SLD) while the conference receives any incoming voice traffic on the input bus (LSD) to be recorded.
- After the call progress tone/file is complete, addition of the channel input and output buses to the conference is resumed.
- Incoming voice traffic is enabled.

Configuration See Configuring Conferencing Features.

TONES

Contents

Overview

Tone Reception

The DSP Series 2 card supports the following tone reception functions:

- DTMF Digit Reception (µ-law and A-law)
- Call Progress Generation and Analysis (µ-law and A-law)
- MFR1/R2 Tone Reception

Tone Generation

The DSP Series 2 card uses a Unified Tone Transmitter. The Unified Tone Transmitter supports up to 512 channels per DSP chip of mixed DTMF/MFR1/R2, Bong Tone, and Call Progress tone generation for either µ-law or A-law.

Configuration

See Tones.

Terms

Call Progress Tone Patterns

Call progress tone patterns are audible signals that indicate the progress or disposition of a telephone call. Examples of call progress tone patterns include the busy signal, ringback, and dial tone.

Call Progress Analysis (CPA)

The act of receiving and interpreting call progress tone patterns is called Call Progress Analysis (CPA).

For call progress analysis, the CSP groups patterns into classes to facilitate management of CPA receivers. These CPA classes help assign call progress tone patterns to CPA receivers. When the host assigns a DSP resource to scan for call progress tones, the host specifies a class of patterns. The DSP resource then scans for the patterns defined in that class.

For example, you could place all progress tones used in a single country into one class. You could then assign this class to a DSP chip for call progress analysis for that country. The host can modify and create classes, using the *CPA Pattern Configure* message.

When a Call Processing receiver activates, it detects call progress patterns that are part of a Call Progress Analysis (CPA) Class. The receiver ignores any tones that are not part of a pattern in this CPA Class. This scenario allows the receiver to skip over tones that are not of interest, reducing the chance of receiving false *Call Progress Analysis Result* API messages.

By default, the CSP uses the following four CPA classes:

- 0x00 Standard North America
- 0x01 Dial Tone
- 0x02 CPC Detection
- 0x03 Energy Detection

Frequency

A **frequency** is the most basic building block of a call progress tone pattern.

Tone

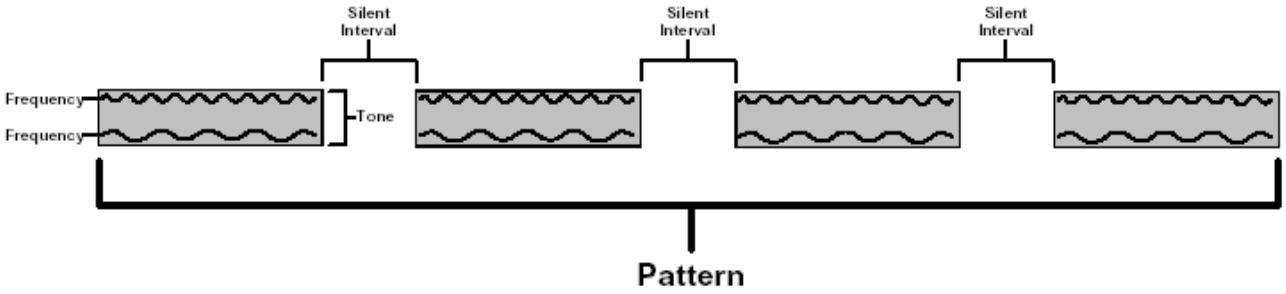
A **tone** is a combination of frequencies, or it is a single frequency that has been designated as a tone. The frequency or frequencies in a tone have specified dBm levels.

Tone Pattern

A **tone pattern** is a single tone or a sequence of tones, divided at precise intervals by silence. A tone pattern is formed when tones are combined, with specific intervals of silence between them.

For example, the industry-standard “call waiting tone” is actually a pattern, in which Tone 1 is on for 300 milliseconds, off for 9700 milliseconds, and on again for 300 milliseconds. You can think of a call progress tone pattern as being at the top of a hierarchy, with tone in the middle and frequency at the bottom. That is, one or more frequencies make up a tone, and one or more tones make up a pattern.

Figure 8-4 The Composition of a Call Progress Tone Pattern



Note: One or more frequencies can be used in a tone, and one or more tones can be used in a pattern

Tone Generation

Types of Tones

The CSP lets you transmit address signaling and call progress tones. Each tone type is available for both µ-law and A-law. You can generate the following tones:

- Dual Tone Multi-frequency (DTMF)
- Multi-frequency R1 (MFR1)
- Multi-frequency R2 (MFR2 – backward and forward) *
- Call Progress Tone Patterns (CPT4)
- Calling Card Prompt (Bong Tone)

* Every card that has an MFR2 receiver must also have a corresponding forward and backward Universal transmitter. To duplicate the generation of a tone for redundancy, configure the tone function on separate cards.

DTMF Receiver with Termination of Tone Announcement

This feature supports the following functionality:

- Cancel tone generation or playback
- Application defined termination digit for play files

The DSP Service Request (0x00BD) and DSP Service Cancel (0x00BE) API messages have been modified to enable this feature.

This feature also permits the host to specify which DTMF digit to look for in order to terminate the file or tone that is currently playing.

Call Setup

Digit Collection During Call Setup

During call setup, the CSP can collect DTMF, MFR1, and MFR2 (Forward and Backward) digits.

DNIS and ANI Identification

You can program the CSP to automatically collect the inpulsing address data that typically accompany an incoming call. This information identifies the Dialed Number Identification Service (DNIS) and/or the Automatic Number Identification (ANI). A service application can use this data to validate and authenticate subscribers, and to identify service.

International Call Progress Analysis

The CSP can transmit and receive international tones, including a 400 Hz tone used outside of North America.

Reporting Call Progress Tones

Detected call progress tones can be reported to the host in the following two ways:

- During call setup, with the outseize instruction of Do Call Progress Analysis
- Interactively, by assigning a CPA Receiver with the *DSP Service Request* message.

Table 7-5 lists the maximum quantities for call progress tone detection in the CSP.

Table 8-5 System Maximum for call progress tone detection

Tone Specification	System Maximum
Patterns	30
Classes	15
Patterns per class	15
Frequencies per tone	2

Receiving Call Progress Tone Patterns

The algorithm for detecting call progress tones has the following two phases:

1. Pattern Detection – The first phase recognizes the pattern, based on the tones and cadences received.
2. Pattern Match – The second phase counts tone intervals, to confirm the presence of a pattern. The number of intervals to match is specified by the pattern's Interval Cycles to Match parameter.

The algorithm does not move to the second phase until the pattern is identified. Configurable timers determine the maximum time to remain in each phase.

The CSP scans for call progress tones in the specified class, and reports them in the *CPA Result* message. CPA terminates upon reporting a result.

Energy Detection

Overview

The Energy Detection feature detects energy on a channel. You can use this feature for CPA when there is not a pattern to match available, or to report energy detection to the host application. To compensate for background noise and to filter out short energy bursts, you can set the sensitivity level and the scan duration (defined below).

You can determine these settings based upon the type of energy to be detected, and to some extent, by trial and error. After setting both levels, test to see if the desired energy is being detected, then adjust the settings as necessary.

For CPA, the Energy Detection feature determines only that a pattern has occurred. It does not distinguish tones. The Energy Detection class (Class 3) is composed of the most common call setup tones: Ringback, Double Ringback, Busy, and Reorder. You can add any pattern to this class, or use any other class for Energy Detection, with the following restrictions:

- The pattern must consist of alternating periods of tones and silence. Energy Detection recognizes a pattern by periods with energy (tone) and no energy (silence). Because Energy Detection must detect both tone and silence, it cannot detect a “pattern” that has tone but no silent interval.
- The Energy Detection DSP function can detect only tones from the default patterns defined for the Energy Detection class (Tone IDs 0, 1, 2, and 5). You can still use a pattern for Energy Detection that does not use one of these tones. To do so, you must change the Tone IDs in the pattern to one of those allowed, using the *CPA Pattern Configure* message.

Parameters The following parameters configure energy detection:

- Sensitivity Level
- Scan Duration
- Completion Timer

Sensitivity Level

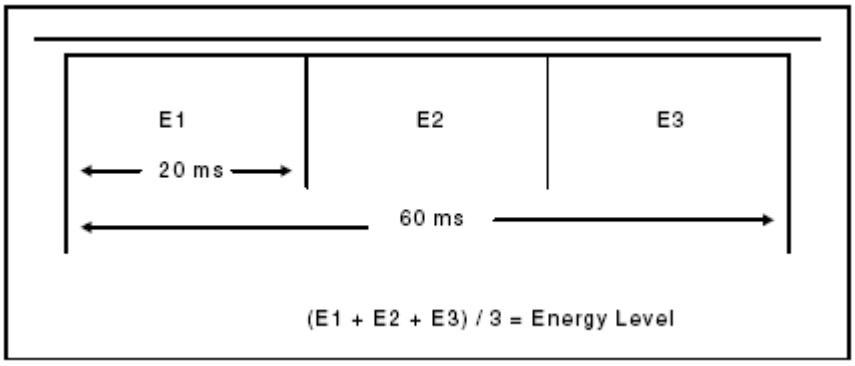
The Sensitivity Level is the amplitude above which the energy detector perceives a signal. Set the sensitivity level to detect and report energy that is greater than the prevailing background noise. When you expect a significant background noise, use the least sensitive setting (0 dBm). When you expect little background noise, use the most sensitive setting (-30 dBm).

Scan Duration

The Scan Duration is the repeating time interval over which the energy detector determines that energy is either Present or Not Present. Set the scan duration to be longer than expected energy bursts. You must use 20 millisecond intervals to set the scan duration, so energy is sampled for each 20 millisecond block within the specified duration.

The scan cycles repeat until the completion timer expires. The calculated energy level is the average over all blocks, as shown in Table 7-4. If the calculated energy level exceeds the configured sensitivity level, energy is reported.

Figure 8-5 Scan Cycle



Completion Timer

The completion timer determines the maximum amount of time to scan for energy. Each scan cycle, as defined by the scan duration, is repeated until either energy is detected, or the completion timer expires. We recommend setting the completion timer for 3 to 4 times the scan duration, depending on the application. If the timer expires before energy is detected, the host receives a *Call Processing Event* message of "No Energy Detected."

Detection Methods

You can invoke Energy Detection in the following two ways:

- Interactively, with the *DSP Service Request* message
- As part of Call Progress Analysis

Interactive Energy Detection

You can invoke Energy Detection interactively, with the *DSP Service Request* message. Use the data bytes of the message to configure the sensitivity level, reporting mode, scan duration, and completion timer. The detected energy is reported to the host in a *Call Processing Event* message in one of two modes:

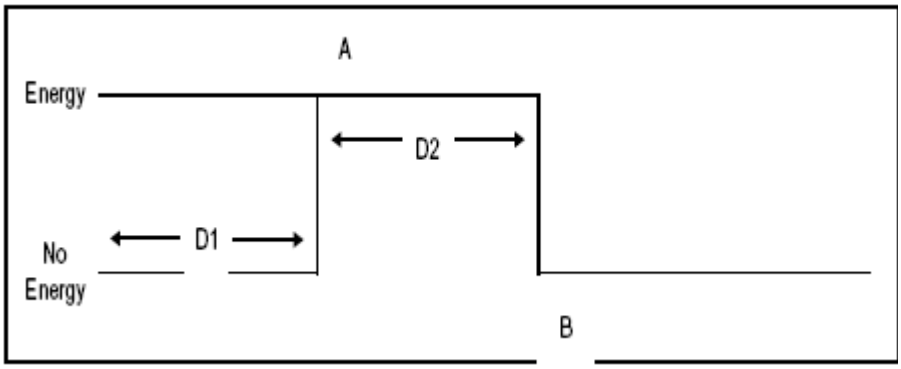
- Report Initial Energy Detection Only
  The host receives a *Call Processing Event* of "Energy Result Report" with Data[0], indicating Energy Detected. Data[1] indicates the duration of the period of no energy. The DSP resource is then automatically released.
- Report All Energy Threshold Crossings

  All of the following are reported: the initial energy detection, all subsequent changes, and the duration of the previous state's ON or OFF interval.

  When energy (A) is first detected, the host receives a *Call Processing Event* of "Energy Result Report" with Data[0] indicating "Energy Detected." Data[1] indicates the duration of the preceding period of no energy (D1).

  When energy is no longer detected (B), the host receives a *Call Processing Event* message of "Energy Result Report" with Data[0] indicating "No Energy Detected." Data[1] indicates the duration of the preceding period of energy (D2).

Figure 8-6 Energy Result Report



The DSP resource remains attached and reports energy changes until the completion timer expires, or until the host sends a *DSP Service Cancel 0x00BE* message, or until the channel returns to an idle state.

Energy Detection as Call Progress Analysis

Energy Detection allows the CSP to perform Call Progress Analysis for frequencies not supported by the default CPA DSP load. The Energy Detection DSP function matches cadences, based on the reported energy levels. CPA Class 3 is pre-configured for Energy Detection, using the standard CPA tones of ringback, double ringback, busy, and re-order.

To accommodate unique requirements for matching cadences, you can modify these patterns or add new patterns to the class. The host can change the pattern cadence that energy detection scans for, using the *CPA Pattern Configure* message.

Configuration See Configuring Energy Detection.

T.30 FAX

Contents

Overview

Overview

T.30 is a fax handshake protocol that describes the overall procedure for establishing and managing communication between two fax devices.

T.30 covers these five phases of a faxing:

1. Setup the Call
2. Select the Communication Mode
3. Transmit the Message
4. Post Message Processing (Including End-of-Message and Confirmation)
5. Release Call / Disconnect

Note: T.30 does not allow for data transmission or processing. T.30 allows only for the transfer of fax image standards T.4 and T.6.

For more information about the Group 3 fax standard, see ITU T30 Fax standard.

Requirements

- Either DSP Series 2 card with associated Multi-Function Media I/O card
- One DSP Series 2 License
- 12 Resource Points per fax session
- Network File System (NFS)

Supported Fax Standards

- V.21 (300 bps) for T.30 fax negotiation
- V.27ter (2,400 / 4,800 bps, required by Group 3)
- V.29 (9,600, 7,200 bps)
- V.17 (14.4, 12, 9.6, 7.2 kbps) transmit/receive

Supported Image Format Conversion

- Image formats conversion, either off-line or while the fax is being sent or received, for MH, MR, (ITUT.4), and MMR (ITUT.6)
- Document files must be in TIFF-F or TIFF-S Format

Configuration

See Record/Play Files.

Supported Features

Features

The T.30 Fax suite supports the following enhanced Group 3 fax features:

- Resolutions: normal, fine, very fine
- Standard page width formats
- Conversion of document formats into different formats, resolutions, or encodings online (on-the-fly) or offline
- Encoding: 1D, 2D, and MMR
- Fax transmission or reception at rates up to 14,400 bps
- 12 simultaneous faxes per DSP chip (6 per stream)
- ITU Group 3 compliance for worldwide fax machine compatibility
- Document fax conversion (shrinking to fit A4, for example) to be supported “on the fly” via TLV
- Page format conversions: A3, A4, and B4
- Fax session and document status monitoring
- Fax queues
- FoIP (on the IP Signaling Series 3 card)
- Error Correction Mode (ECM)
- Permanent fax records (stored on the NFS server)
- Configurable per-card, per-chip, and per-call
- Digital signal processor (DSP)-based flexibility
- Transmit Page Range in TIFF File.

NOTE: This release does not support T.30 Polling Mode.

Fax Applications

Introduction

The DSP Series 2 card supports a suite of intelligent fax capabilities that are compliant with the ITU T.30 fax protocol, recognized worldwide. You can use the T.30 Fax suite to develop applications that transmit and receive Group 3 facsimiles concurrently on all channels, at rates up to 14,400 bps.

Fax Messaging Applications

Supported Fax Applications include:

- LAN Fax
- Fax Broadcast
- Fax-on-demand
- Fax Messaging

Fax Server Products

Fax server products can be key components of large-scale, media server-based fax applications for the enterprise (for example, fax servers) and for large-scale Fax Service Provider systems. Dedicated fax servers serve more users while reducing hardware cost. Fax servers let companies distribute information to large audiences for pennies a page, and to eliminate bottlenecks associated with the limited capacities of traditional fax machines.

T.38 Support

The T.30 Fax suite can also be used to produce fax server products for IP telephony gateways when used with T.38. You must use the IP Signaling Series 3 card for T.38. The DSP Series 2 card does not itself support or terminate T.38.

Store and Forward

A host application running on the CSP with the T.30 Fax suite can use ITU T.37 fax store and forward. Although the T.30 Fax suite does not explicitly support ITU T.37 fax store and forward, the suite does support the T.37 page format, which is specified as 1D encoding, LOW resolution, and A4 page width.

ADMINISTRATION

Contents

Overload Management

Overview

The Overload Management feature allows the CSP to monitor and balance DSP resources among all the DSP cards, based upon how busy each DSP Series 2 card is.

- CPU Overload
- File Overload

The DSP Series 2 overload levels are reported to the Matrix Controller card:

- when any level changes
- upon Matrix Controller card switchover (reported to the newly active Matrix Controller card)

Overload Levels

There are two overload levels to the Matrix Controller card:

- Overload 1

  A DSP resource at Overload Level 1 will be used only if no other resources in the CSP are available.
- Overload 2

  A DSP resource at Overload Level 2 will not be used for some functions, and queued for others.

Alarms

When the configured process overload thresholds are exceeded, the DSP Series 2 card sends an *Alarm* message to the Host application. When the thresholds are no longer exceeded, the DSP Series 2 card sends an *Alarm Cleared* message to the Host application.

CPU Overload

CPU Overload Level 1

When an available DSP Series 2 resource is at CPU Overload Level 1, the resource will only be used when no other DSP resource is available in the CSP. This applies to all DSP function types.

CPU Overload Level 2

A request for any of the following DSP resources is NACK'd with "No Resources Due to DSP Overload" (0x1707)

- Tone Transmitter
- Conferencing
- Call Progress Analysis

- Energy Detection

If a request for a DTMF, MFR1, or MFR2 receiver has been queued, the Matrix Controller card sends an *Alarm* message of DSP Request Queued Due to DSP Series 2 CPU Overload Level 2 (0x40). This is a General alarm with a severity of Major.

File Overload

The File Overload Levels are set when any of the following conditions occur:

- the CPU idle time is below its threshold setting
- the VRA Process time is greater than its threshold setting
- the VRA IO Queue process time is greater than its threshold setting

The File Overload Levels are cleared when all of the following conditions are met:

- the CPU idle time is no longer below its threshold setting
- the VRA Process time is no longer greater than its threshold setting
- the VRA IO Queue process time is no longer greater than its threshold setting

File Overload Level 2

Any request for a DSP File Transmitter or Recorder is NACK'd with "No Resources Due to DSP Overload" (0x1707).

Configuration

See Configuring Overload Management.

Overload Alarms

The following *Alarm* messages are sent to the host when an overload condition occurs. When the condition clears, an *Alarm Cleared* message is sent.

Condition	Alarm
CPU Overload 1	None
CPU Overload 2	Alarm Type: Card Severity: Major Alarm: NFS Major Traffic (0x45) Overload Alarm Type: CPU Idle (0x0A)
	Alarm Type: General Severity: Major Overload Alarm Type: DSP Request Queued Due to DSP Series 2 CPU Overload Level 2 (0x40)
File Overload Level 1: VRA Process Time	Alarm Type: Card Severity: Minor Alarm Number: NFS Minor Traffic (0x44) Overload Alarm Type: “VRA Process” (0x0B)
File Overload Level 1: VRA IO Queue Process Time	Alarm Type: Card Severity: Minor Alarm: NFS Minor Traffic (0x44) Overload Alarm Type: VRA IO Queue Process (0x0C)
File Overload Level 2	None
File Overload Level 2 for VRA Process Time	Alarm Type: Card Severity: Major Alarm: NFS Major Traffic (0x45) Overload Alarm Type: VRA Process (0x0B)

Condition	Alarm
File Overload Level 2 for VRA IO Queue Process Time	Alarm Type: Card Severity: Major Alarm: NFS Major Traffic (0x45) Overload Alarm Type: VRA IO Queue Process (0x0C)

Statistics Query

Overview

Use the Statistics Query 0x0121 message to retrieve the following information:

- Network File System (NFS) statistics per card:
  - total number and size of NFS reads, writes, and records
  - read and write time delays (minimum, maximum, and average)
- Function statistics on individual DSP chips:
  - total requests for a function
  - average free channels
  - minimum free channels
- Cache statistics on all individual DSP chips:
  - accesses
  - misses
  - hits
  - average free blocks
- Fixed memory statistics on all individual DSP chips, including total, average, and maximum blocks used.

Cache Query

Overview

The DSP Series 2 Cache Query permits the host to check whether the specified file is present in the DSP Series 2 main board and DSP chip cache memory.

This feature is implemented by using query type, Cache File Query 0x18 in the *System Configuration Query* (0x00B4) message.

The File ID will be passed in the data field of the *System Configuration Query* message. The result of this query is sent back to the host using the *Generic Report* (0x0046) message with report type, Cache File Query 0x18.

DSP Alarms

Table 7-6 shows the alarms reported in the Alarm 0x00B9 message that relate to DSP functionality. See the message for details.

Table 8-6 DSP Related Alarms

Alarm Type	Severity	Alarm Number/Name
General (0x01)	Major	0x06 - DSP Resource Blocked
		0x07 - DSP Resource Function Type Not Configured
		0x08 - DSP Resource Management Inconsistent
		0x3F - Insufficient Resource Points for DSP Function
		00x40 - DSP Request Queued Due to DSP Series 2 CPU Overload Level 2
	Informational	0x34 - DSP CPA Configuration Conflict
Card (0x02)	Major	0x41 - NFS Server Mounting Error
		0x43 - DSP Series 2 File Space Out of Memory
		0x45 - NFS Traffic Threshold Major Alarm
		0x46 - DSP Series 2 TFTP Communications Error
		0x47 - NFS Read Error
		0x48 - NFS Write Error
		0x4B - NFS Server Unmount Error
		0x4C - NFS Open File Error
	Minor	0x44 - NFS Traffic Minor Threshold Alarm
		0x42 - DSP Series 2 File Space Low

Alarm Type	Severity	Alarm Number/Name
Card (cont.)	Informational	0x49 - Vocabulary Index File Read
		0x4A - NFS Server Unmounted
		0x40 - NFS Server Mounted Successfully
Channel (0x04)	Major	0x0A - DSP Resource Wait Timeout
DSP SIMM (0x05)	Major	0x04 - Recorded Announcement Download Inconsistency
		0x05 - Recorded Announcement ID Inconsistency (on same card)
		0x08 - Recorded Announcement File System Conversion Failure
		0x09 - Recorded Announcement File System Timeout
		0x0C - Recorded Announcement Defragmentation Failure
	Informational	0x01 - DSP SIMM Taken Out of Service
		0x03 - Recorded Announcement Erase Complete
		0x06 - Recorded Announcement Download Ready
		0x07 - Recorded Announcement File System Conversion Success
		0x0B - Recorded Announcement Defragmentation Success
		0x0D - Recorded Announcement Single Deletion Complete
		0x0E - Recorded Announcement Need Defragmentation

Alarm Type	Severity	Alarm Number/Name
DSP (0x06)	Major	0x01 - DSP Out of Service
		0x03 - DSP Playback Underrun
		0x04 - DSP Record Overrun
		0x05 - DSP Temporary Storage Minor Alarm
		0x06 - DSP Temporary Storage Full
		0x08 - DSP Chip Reset
	Informational	0x02 - Resources Exceed Limit
		0x07 - DSP Points Exceed Limit

SPECIFICATIONS

DSP Series 2 and DSP-ONE Cards Compared

Table 8-7 DSP Features by Card

Feature	DSP Series 2	DSP-ONE
I/O Card	uses the Multi-Function Media I/O card for File Record/Playback	No I/O Card
Maximum number of DSP chips	8	16
Configuration API Message	*DSP SIMM Configure* (0x00C0) *Generic Card Configure* (0x0122)	*DSP SIMM Configure* (0x00C0)
Tone Generation and Reception	a DSP is assigned to play a universal tone, and that can be any tone type with the same encoding format.	a DSP is assigned a particular tone type, and can play that for 2,048 channels
Record/ Play files	Record/Play features	Voice Recorded Announcements provided on the VRA SIMM
Announcement Tracking	the host application uses an ASCII Vocabulary Index File (VIF) to track playback files stored on the servers.	the Matrix Controller card keeps track of announcement IDs and where announcements are stored on the cards
Conference Types	- Monitor - Unified - Advanced	- Standard - Mixed - Monitor - Unified
Conference API Messages	- *Resource Create* - *Resource Connect* - *Resource Delete*	- *Conference Create* - *Connect-to-Conference* - *Connect One-Way to Conference* - *Conference Delete*
Statistics	Supported	Not Supported
NFS server file storage	Supported	Not Supported
Overload Management	Supported	Not Supported
Coin Tone Detection	Not Supported	Supported
E1 Dial Pulse Address Signaling	Not Supported	Supported

DTMF Digit Specifications

Table 8-8 DTMF digit specifications

BCD	DTMF Digit	Frequencies (Hz)
0	0	941 Hz + 1336 Hz
1	1	697 Hz + 1209 Hz
2	2	697 Hz + 1336 Hz
3	3	697 Hz + 1477 Hz
4	4	770 Hz + 1209 Hz
5	5	770 Hz + 1336 Hz
6	6	770 Hz + 1477 Hz
7	7	852 Hz + 1209 Hz
8	8	852 Hz + 1336 Hz
9	9	852 Hz + 1477 Hz
A	A	697 Hz + 1633 Hz
B	B	770 Hz + 1633 Hz
C	C	852 Hz + 1633 Hz
D	D	941 Hz + 1633 Hz
E	*	941 Hz + 1209 Hz
F	#	941 Hz + 1477 Hz

MF Digit Specifications

Table 8-9 MFR1, MFR2 digit specifications

BCD	MFR1 Digit	MFR1 Frequency (Hz)	MFR2 Digit	MFR2 Forward Frequency (Hz)	MFR2 Backward Frequency (Hz)
0	0	1300 Hz + 1500 Hz	1	1380+1500	1140+1020
1	1	700 Hz + 900 Hz	2	1380+1620	1140+900
2	2	700 Hz + 1100 Hz	3	1500+1620	1020+900
3	3	900 Hz + 1100 Hz	4	1380+1740	1140+780
4	4	700 Hz + 1300 Hz	5	1500+1740	1020+780
5	5	900 Hz + 1300 Hz	6	1620+1740	900+780
6	6	1100 Hz + 1300 Hz	7	1380+1860	1140+660
7	7	700 Hz + 1500 Hz	8	1500+1860	1020+660
8	8	900 Hz + 1500 Hz	9	1620+1860	900+660
9	9	1100 Hz + 1500 Hz	A	1740+1860	780+660
A	KP	1100 Hz + 1700 Hz	B	1380+1980	1140+540
B	ST	1500 Hz + 1700 Hz	C	1500+1980	1020+540
C	STI	900 Hz + 1700 Hz	D	1620+1980	900+540
D	STII	1300 Hz + 1700 Hz	E	1740+1980	780+540
E	STIII	700 Hz + 1700 Hz	F	1860+1980	660+540

Default Transmit Call Progress Tones

Table 7-10 shows the default tones that the CSP uses to generate call progress tone patterns. Each call progress tone is distinguished by an ID. Note that some of the tones are single frequencies, while others are combinations of frequencies, at specific dBm levels.

You can modify any of the default tones shown in the table above, but you cannot create a new Tone ID. The tone that was originally associated with that Tone ID becomes unavailable and the change affects all patterns that use that tone.

Table 8-10 Default transmit call progress tones

		Tone ID											
Specification	**00**	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**0A**	**0B**	**0C**
Frequency Count	0	1	1	2	2	2	4	1	1	1	1	1	1
Frequency 0 (Hz)		440	480	350	440	480	1,400	914	985	1,371	1,429	1,777	400
dBm 0		-15	-15	-15	-20	-20	-8	-13	-13	-13	-13	-13	-18
Frequency 1 (Hz)				440	480	620	2,060						
dBm 1				-15	-20	-20	-8						
Frequency 2 (Hz)							2,450						
dBm 2							-8						
Frequency 3 (Hz)							2,600						
dBm 3							-8						

Default Transmit Call Progress Tone Patterns

To form call progress tone patterns, the call progress tones are combined and separated by intervals of silence. Table 7-11 shows the default transmit call progress tone patterns and their parameters.

Each pattern is composed of one to three cycle blocks, which include the following components:

- Tone ID
- ON duration
- OFF duration
- Number of times to repeat the cycle block

0xFFFF = continuous

Table 8-11 Default Transmit Call Progress Tone Patterns

Transmit Cadence Pattern	ID	Cycle Blocks	Cycle Block Number	Tone ID	ON Duration	OFF Duration	Cycles
Silence	0x00	1	1	0x00	0xFFFF	0	1
Dial Tone	0x01	1	1	0x03	0xFFFF	0	1
Ringback	0x02	1	1	0x04	2000	4000	1
Line Busy	0x03	1	1	0x05	500	500	1
Reorder	0x04	1	1	0x05	250	250	1
Warning	0x05	1	1	0x06	100	100	1
Call Waiting	0x06	2	1	0x01	300	9700	1
			2	0x01	300	10	1
ONI Call (Zip Tone)	0x07	2	1	0x02	100	100	1
			2	0x02	100	10	1
ANI Failure (Zip Tone)	0x08	1	1	0x02	80	10	1
Confirmation	0x09	1	1	0x03	100	100	3
Recall Dial Tone	0x0A	2	1	0x03	100	100	3
			2	0x03	0xFFFF	10	1
Class of Service Tone 2	0x0B	1	1	0x05	800	10	1

Transmit Cadence Pattern	ID	Cycle Blocks	Cycle Block Number	Tone ID	ON Duration	OFF Duration	Cycles
Class of Service Tone 3	0x0C	1	1	0x02	800	10	1
Intercept	0x0D	3	1	0x07	280	10	1
			2	0x09	280	10	1
			3	0x0B	380	10	1
Vacant Code	0x0E	3	1	0x08	380	10	1
			2	0x09	280	10	1
			3	0x0B	380	10	1
Reorder - LEC	0x0F	3	1	0x07	280	10	1
			2	0x0A	380	10	1
			3	0x0B	380	10	1
No Circuit - LEC	0x10	3	1	0x08	380	10	1
			2	0x0A	380	10	1
			3	0x0B	380	10	1
Reorder - Carrier	0x11	3	1	0x08	280	10	1
			2	0x09	380	10	1
			3	0x0B	380	10	1
No Circuit - Carrier	0x12	3	1	0x07	380	10	1
			2	0x09	380	10	1
			3	0x0B	380	10	1
Continuous 400 Hz	0x13	1	1	0x0C	0xFFFF	10	1
Specialized Tone	0x14	2	1	0x00	500	10	1
			2	0x02	800	10	1
Bong Tone	0x15	Reserved, cannot be modified by host					
	0x16	1	1	0x0D	0xFFFF	0x0000	1
	0x17	1	1	0x0B	0xFFF	0x000	1
	0x18-0x3F	User-defined					

Default Tone Patterns for CPA Detection

All of the tones supported by the CSP for are shown in Table 7-12. The group of tone IDs may use up to 16 frequencies. A pattern is formed when the tones below are played at specific intervals.

Table 8-12 Tones supported by the CSP

Tone ID	Frequency Count	Frequency 0 (Hz)	Frequency 1 (Hz)
0x00	1	0	
0x01	2	350	440
0x02	2	440	480
0x03	1	440	
0x04	1	480	
0x05	2	480	620
0x06	1	620	
0x07	1	914	
0x08	1	985	
0x09	1	1371	
0x0A	1	1429	
0x0B	1	1777	
0x0C	1	2000	
0x0D	1	1700	
0x0E	1	2100	
0x0F	1	425	
0x10	1	500	
0x11	1	1100	

CPA Tone Patterns

The default tone patterns for call progress analysis appear in Table 7-13. To modify or create new patterns, use the *CPA Pattern Configure* message. Click on the pattern name to see the default parameters (web-based documentation only).

Table 8-13 Default tone patterns for call progress analysis

Value	CPA Tone Pattern
0x01	Ringback
0x02	Double Ringback
0x03	Busy
0x04	Reorder
0x05	PBX intercept
0x06	SIT Intercept
0x07	Vacant Code
0x08	Reorder LEC
0x09	No Circuit LED
0x0A	Reorder Carrier
0x0B	No Circuit Carrier
0x0C	PBX Dialtone
0x0D	Standard Dial Tone
0x0E	CPC Detection

Default CPA Tone Pattern Parameters

Ringback

Ringback	
Pattern ID	0x01
Tone Group ID	0x00
Configuration Bits	0x00
CPA Result on Pattern Loss	0x80
Interval Cycles to Match	1
Interval Cycles to Report	3
Interval Descriptor Count	2
Interval Descriptors	
Tone Interval 0: Tone ID	0x02
Reserved	0x00
Minimum Filter (ms)	600
Maximum Filter (ms)	2200
Tone Interval 1: Tone ID	0x00
Reserved	0x00
Minimum Filter (ms)	2800
Maximum Filter (ms)	5000

Double Ringback

Double Ringback	
Pattern ID	0x02
Tone Group ID	0x00
Configuration Bits	0x00
CPA Result on Pattern Loss	0x80
Interval Cycles to Match	1
Interval Cycles to Report	3
Interval Descriptor Count	4
Tone Interval 0: Tone ID	0x02
Reserved	0x00
Minimum Filter (ms)	420
Maximum Filter (ms)	580
Tone Interval 1: Tone ID	0x00
Reserved	0x00
Minimum Filter (ms)	200
Maximum Filter (ms)	400
Tone Interval 2: Tone ID	0x02
Reserved	0x00
Minimum Filter (ms)	420
Maximum Filter (ms)	580
Tone Interval 3: Tone ID	0x00
Reserved	0
Minimum Filter (ms)	2000
Maximum Filter (ms)	2500

Busy

Busy	
Pattern ID	0x03
Tone Group ID	0x00
Configuration Bits	0x00
CPA Result on Pattern Loss	0x03
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	2
Tone Interval 0: Tone ID	0x05
Reserved	0x00
Minimum Filter (ms)	420
Maximum Filter (ms)	580
Tone Interval 1: Tone ID	0x00
Reserved	0x00
Minimum Filter (ms)	420
Maximum Filter (ms)	580
Tone Interval 2: Tone ID	
Reserved	
Minimum Filter (ms)	
Maximum Filter (ms)	
Tone Interval 3: Tone ID	
Reserved	
Minimum Filter (ms)	
Maximum Filter (ms)	

Reorder

Reorder	
Pattern ID	0x04
Tone Group ID	0x00
Configuration Bits	0x00
CPA Result on Pattern Loss	0x04
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	2
Tone Interval 0: Tone ID	0x05
Reserved	0x00
Minimum Filter (ms)	200
Maximum Filter (ms)	300
Tone Interval 1: Tone ID	0x00
Reserved	0x00
Minimum Filter (ms)	200
Maximum Filter (ms)	300
Tone Interval 2: Tone ID	
Reserved	
Minimum Filter (ms)	
Maximum Filter (ms)	
Tone Interval 3: Tone ID	
Reserved	
Minimum Filter (ms)	
Maximum Filter (ms)	

PBX intercept

PBX Intercept	
Pattern ID	0x05
Tone Group ID	0x00
Configuration Bits	0x02
CPA Result on Pattern Loss	0x05
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	2
Tone Interval 0: Tone ID	0x03
Reserved	0x00
Minimum Filter (ms)	100
Maximum Filter (ms)	300
Tone Interval 1: Tone ID	0x06
Reserved	0x00
Minimum Filter (ms)	100
Maximum Filter (ms)	300
Tone Interval 2: Tone ID	
Reserved	
Minimum Filter (ms)	
Maximum Filter (ms)	
Tone Interval 3: Tone ID	
Reserved	
Minimum Filter (ms)	
Maximum Filter (ms)	

SIT Intercept

SIT Intercept	
Pattern ID	0x06
Tone Group ID	0x00
Configuration Bits	0x02
CPA Result on Pattern Loss	0x06
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	3
Tone Interval 0: Tone ID	0x07
Reserved	0x00
Minimum Filter (ms)	200
Maximum Filter (ms)	350
Tone Interval 1: Tone ID	0x09
Reserved	0x00
Minimum Filter (ms)	200
Maximum Filter (ms)	350
Tone Interval 2: Tone ID	0x0B
Reserved	0x00
Minimum Filter (ms)	300
Maximum Filter (ms)	460
Tone Interval 3: Tone ID	
Reserved	
Minimum Filter (ms)	
Maximum Filter (ms)	

Vacant Code

Vacant Code	
Pattern ID	0x07
Tone Group ID	0x00
Configuration Bits	0x02
CPA Result on Pattern Loss	0x07
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	3
Tone Interval 0: Tone ID	0x08
Reserved	0x00
Minimum Filter (ms)	300
Maximum Filter (ms)	460
Tone Interval 1: Tone ID	0x09
Reserved	0x00
Minimum Filter (ms)	200
Maximum Filter (ms)	350
Tone Interval 2: Tone ID	0x0B
Reserved	0x00
Minimum Filter (ms)	300
Maximum Filter (ms)	460
Tone Interval 3: Tone ID	
Reserved	
Minimum Filter (ms)	
Maximum Filter (ms)	

Reorder LEC

Reorder LEC	
Pattern ID	0x08
Tone Group ID	0x00
Configuration Bits	0x02
CPA Result on Pattern Loss	0x08
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	3
Tone Interval 0: Tone ID	0x07
Reserved	0x00
Minimum Filter (ms)	200
Maximum Filter (ms)	350
Tone Interval 1: Tone ID	0x00
Reserved	0x00
Minimum Filter (ms)	300
Maximum Filter (ms)	460
Tone Interval 2: Tone ID	0x0B
Reserved	0
Minimum Filter (ms)	300
Maximum Filter (ms)	460

No Circuit LED

No Circuit LED	
Pattern ID	0x09
Tone Group ID	0x00
Configuration Bits	0x02
CPA Result on Pattern Loss	0x09
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	3
Tone Interval 0: Tone ID	0x08
Reserved	0x00
Minimum Filter (ms)	300
Maximum Filter (ms)	460
Tone Interval 1: Tone ID	0x0A
Reserved	0x00
Minimum Filter (ms)	300
Maximum Filter (ms)	460
Tone Interval 2: Tone ID	0x0B
Reserved	0
Minimum Filter (ms)	300
Maximum Filter (ms)	460

Reorder Carrier

Reorder Carrier	
Pattern ID	0x0A
Tone Group ID	0x00
Configuration Bits	0x02
CPA Result on Pattern Loss	0x0A
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	3
Tone Interval 0: Tone ID	0x08
Reserved	0x00
Minimum Filter (ms)	200
Maximum Filter (ms)	350
Tone Interval 1: Tone ID	0x09
Reserved	0x00
Minimum Filter (ms)	300
Maximum Filter (ms)	460
Tone Interval 2: Tone ID	0x0B
Reserved	0
Minimum Filter (ms)	300
Maximum Filter (ms)	460

No Circuit Carrier

No Circuit Carrier	
Pattern ID	0x0B
Tone Group ID	0x00
Configuration Bits	0x02
CPA Result on Pattern Loss	0x0B
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	3
Tone Interval 0: Tone ID	0x07
Reserved	0x00
Minimum Filter (ms)	300
Maximum Filter (ms)	460
Tone Interval 1: Tone ID	0x09
Reserved	0x00
Minimum Filter (ms)	300
Maximum Filter (ms)	460
Tone Interval 2: Tone ID	0x0B
Reserved	0
Minimum Filter (ms)	300
Maximum Filter (ms)	460

PBX Dialtone

PBX Dialtone	
Pattern ID	0x0C
Tone Group ID	0x00
Configuration Bits	0x01
CPA Result on Pattern Loss	0x0C
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	7
Tone Interval 0: Tone ID	0x01
Reserved	0x00
Minimum Filter (ms)	80
Maximum Filter (ms)	120
Tone Interval 1: Tone ID	0x00
Reserved	0x00
Minimum Filter (ms)	80
Maximum Filter (ms)	120
Tone Interval 2: Tone ID	0x01
Reserved	0x00
Minimum Filter (ms)	80
Maximum Filter (ms)	120
Tone Interval 3: Tone ID	0x00
Reserved	0x00
Minimum Filter (ms)	80
Maximum Filter (ms)	120
Tone Interval 4: Tone ID	0x01
Reserved	0x00
Minimum Filter (ms)	80
Maximum Filter (ms)	120

PBX Dialtone	
Tone Interval 5: Tone ID	0x00
Reserved	0x00
Minimum Filter (ms)	80
Maximum Filter (ms)	120
Tone Interval 6: Tone ID	0x01
Reserved	0x00
Minimum Filter (ms)	500
Maximum Filter (ms)	0

Standard Dial Tone

Standard Dial Tone	
Pattern ID	0x0D
Tone Group ID	0x00
Configuration Bits	0x05
CPA Result on Pattern Loss	0x0D
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	1
Tone Interval 0: Tone ID	0x01
Reserved	0x00
Minimum Filter (ms)	500
Maximum Filter (ms)	0

CPC Detection

CPC Detection	
Pattern ID	0x0E
Tone Group ID	0x00
Configuration Bits	0x01
CPA Result on Pattern Loss	0x0E
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	1
Tone Interval 0: Tone ID	0x01
Reserved	0x00
Minimum Filter (ms)	1500
Maximum Filter (ms)	0

Fax

Fax	
Pattern ID	0x13
Tone Group ID	0x00
Configuration Bits	0x00
CPA Result on Pattern Loss	0x13
Interval Cycles to Match	1
Interval Cycles to Report	1
Interval Descriptor Count	2
Tone Interval 0: Tone ID	0x11
Reserved	0x00
Minimum Filter (ms)	425
Maximum Filter (ms)	575
Tone Interval 1: Tone ID	0x00
Reserved	0x00
Minimum Filter (ms)	2550
Maximum Filter (ms)	3450

CPA Pattern Parameter Definitions

Table 8-14 CPA Pattern Parameter definitions

Pattern ID	Description
Configuration Bits	Pattern-specific configuration (see *CPA Pattern Configure* message) 0x01 = Last Interval Continuous (for pattern ending with continuous tone) 0x02 = Detect Tone After Determination (for reporting Intercept Tones) 0x04 = Use As Internal Dial Tone (to detect dial tone without invoking Call Progress Analysis)
CPA Result on Pattern Loss	Call Progress Analysis Result to report when a pattern has been matched but discontinues before “interval cycles to report”. (*CPA Result* message).
Interval Cycles to Match	The number of times an Interval Sequence must be repeated before declaring the pattern valid
Interval Cycles to Report	The number of times an Interval Sequence must be repeated before reporting a Call Progress Result to the host
Interval Descriptor Count	The number of Interval Descriptors in the pattern
Interval Descriptor	An Interval Descriptor is a list of the tones and intervals that constitute a pattern. The CSP uses the Interval Descriptor to confirm that a specific pattern has been detected. An Interval Descriptor consists of the Tone ID, Min. Filter, and Max. Filter, where: Tone ID – A valid Tone ID Minimum/Maximum Filter (MSB, LSB) – Minimum and maximum duration for this tone interval - A tone is determined as valid if the on and off cycles fall within the minimum and maximum times.

CPA Classes

The parameters associated with a CPA class are defined in Table 7-16. To modify a class or to add a new class, use the *CPA Class Configure* message. The CSP supports up to 15 classes. A class can contain up to 15 pattern members, and 30 patterns are allowed for each CSP. Class elements are given in Table 7-15.

Table 8-15 CPA Classes

Class	**Standard N.A. 0x00**	**Dial Tone 0x01**	**CPC Det. 0x02**	**Energy Det. 0x03**
Mode	0x01 (Adv. Ans. Det.)			Energy Detection
Tone Burst Answer Threshold	4/250 ms			4/500 ms
Start Delay	0x00	0x00	0x00	0x00
Continuous Silence Timeout (ms)	25,000	25,000	0xFFFF	25,000
Confirmation (ms)	13,000	25,000	0xFFFF	13,000
Confirmation/No Report (ms)	22,000	15,000	0xFFFF	22,000
Minimum Frequency Glitch Time (ms)	40	20	20	20
Minimum Silence Glitch Time (ms)	60	20	20	60
Maximum On Timer (ms)	2,800	25,000	0xFFFF	2,800
Maximum Off Timer (ms)	5,000	25,000	0xFFFF	5,000
Mode Specific Value 1	0x0000	0x0000	0x0000	0x0005
Mode Specific Value 2	0x0000	0x0000	0x0000	0x0002

Class	**Standard N.A. 0x00**	**Dial Tone 0x01**	**CPC Det. 0x02**	**Energy Det. 0x03**
Patterns in Class	0x01 Ringback 0x02 Double Ringback 0x03 Busy 0x04 Reorder 0x05 PBX Intercept 0x06 SIT Intercept 0x07 Vacant Code 0x08 Reorder-LEC 0x09 No Circuit LEC 0x0A Reorder Carrier 0x0B No Circuit Carrier 0x0C PBX Dial Tone 0x0D Standard Dial Tone	0x0C PBX Dial Tone 0x0D Standard Dial Tone	0x0E CPC Detection	0x01 Ringback 0x02 Double Ringback 0x03 Busy 0x04 Reorder

CPA Class Parameter Definitions

Table 8-16 CPA Class Parameters and Definitions

Class Parameter	Definition
Mode	Call Progress Analysis Mode (for example, Advanced Answer Detect, Energy Detection).
Tone Burst Answer Threshold	The number of frequency bursts in 500ms before answer is declared.
Start Delay	Amount of time to wait before analysis begins.
Continuous Silence Timeout	Amount of time after CPA is initiated to wait to hear any tone.
Confirmation	Amount of time to wait to detect a pattern.
Confirmation/No Report	Amount of time after detection of a tone to wait for pattern confirmation before terminating CPA.
Minimum Frequency Glitch Time	Amount of time to validate a tone detection.
Minimum Silence Glitch Time	Amount of time to validate a silence detection.
Maximum On Timer	The maximum amount of time a tone must be “on” without a pattern match before the “continuous on” result is reported.
Maximum Off Timer	The maximum amount of time silence must be detected before the “not determined” result is reported; amount of time to validate a silence detection before declaring it as silence.
Mode Specific Value	Values to configure for specific CPA mode.

Pulse Dialing Detector

Overview

This section describes the addition of a pulse dialing detector into the DSP Series 2 card of the CSP.

Both DTMF and Dial Pulse detection are supported. Dial Pulse detection is co-resident with the DTMF detector and supports all of the features associated with the DTMF signal detection, such as record/playback cancellation on detection. The DTMF and Dial Pulse detectors will have the same densities as the DTMF detector alone (192 bits per stream).

It is common in some areas to have legacy switches and telephones that only support pulse (rotary) dialing. By incorporating a pulse dialing detector, the CSP will enable the customer to develop applications which accept user input from callers who do not have DTMF-capable telephones.

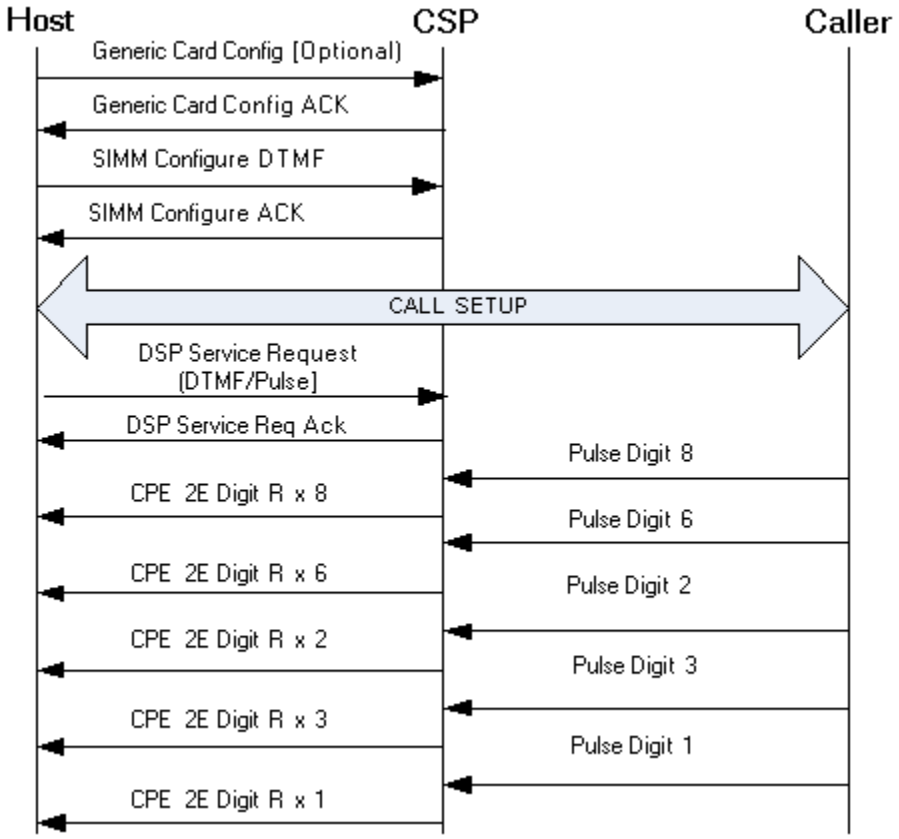
APIs and TLVs Used to Configure this Feature

To configure this feature, the host may optionally send a *Generic Card Configure* (0x0122) message to the DSP Series 2 card to change the default parameters to enable pulse dialing. These configurable parameters are provided in the modified Card Object (0x05FA) TLV and the new Dial Pulse Detection Parameters (0x0750) TLV. Default settings are provided by the customer. All configurable parameters can be queried using the *Generic Card Query* (0x0123) message.

Following this optional configuration, the *DSP SIMM Configure* (0x00C0) message will enable DTMF/Pulse dialing on the CSP. When the call is setup, a *DSP Service Request* (0x00BD) message or *Collect Digit String* (0x00BC) message enables the DSP Series 2 card to begin receiving pulse or DTMF digit indications. In the case of a *DSP Service Request*, these digits will show up individually, and in the case of collect digit strings, the digits will show up when the requested digit string arrives. Pulse dialing can be canceled using the *DSP Service Cancel* (0x00BE) message.

Call Flow

The following call flow shows the basic configuration for setting up pulse dialing.



Dial Pulse Detection Parameters (0x0750) TLV

The Figures 7-7 and 7-8 aid in describing the Dial Pulse Detection Parameters (0x0750) TLV. Refer to this TLV in the API Reference.

Figure 8-7 Representative Pulse Digit

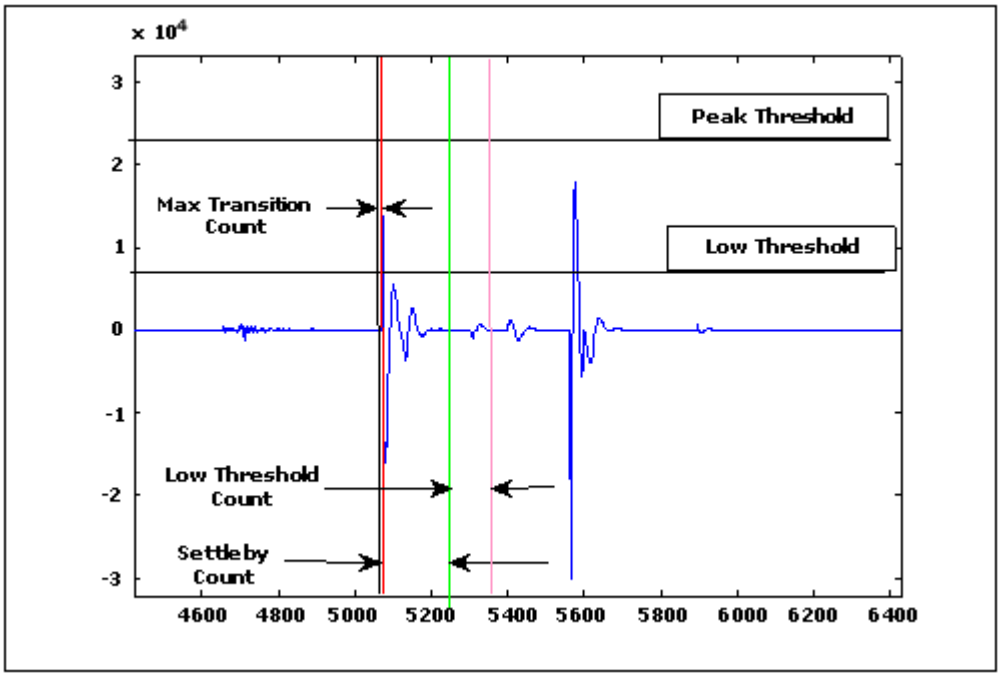
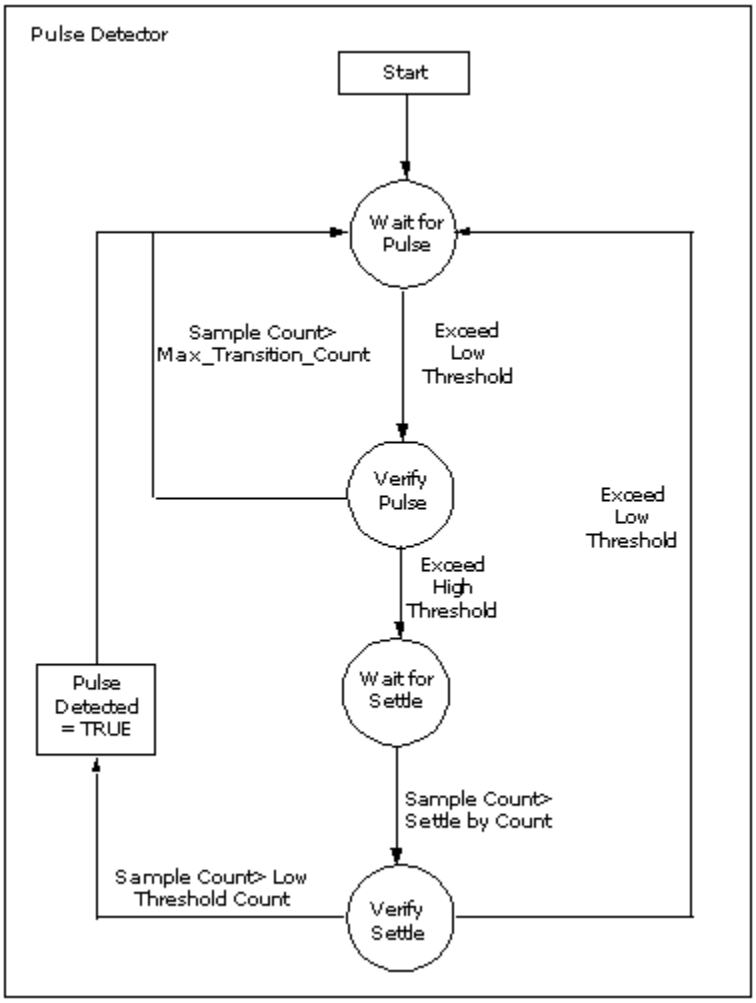


Figure 8-8 State Machine Detection of Dial Pulses



9 Configuring the DSP-ONE Card

Contents

OVERVIEW

Introduction

The DSP-ONE card provides Digital Signal Processing functionality in the CSP, including Voice Recorded Announcements, Tones, and Conferencing.

Every port in the CSP has direct access to every DSP resource (except for channels in Gateway Mode. See the *Developer's Guide: IP* for more information on Gateway Mode). A bi-directional PCM bus allows simultaneous incoming and outgoing operations, such as tone reception and tone generation.

You can upgrade your system with additional DSP cards as your needs grow, and multiple cards can load-share. The DSP cards are hot-swappable, which simplifies maintenance and upgrades, and increases overall reliability.

Hardware

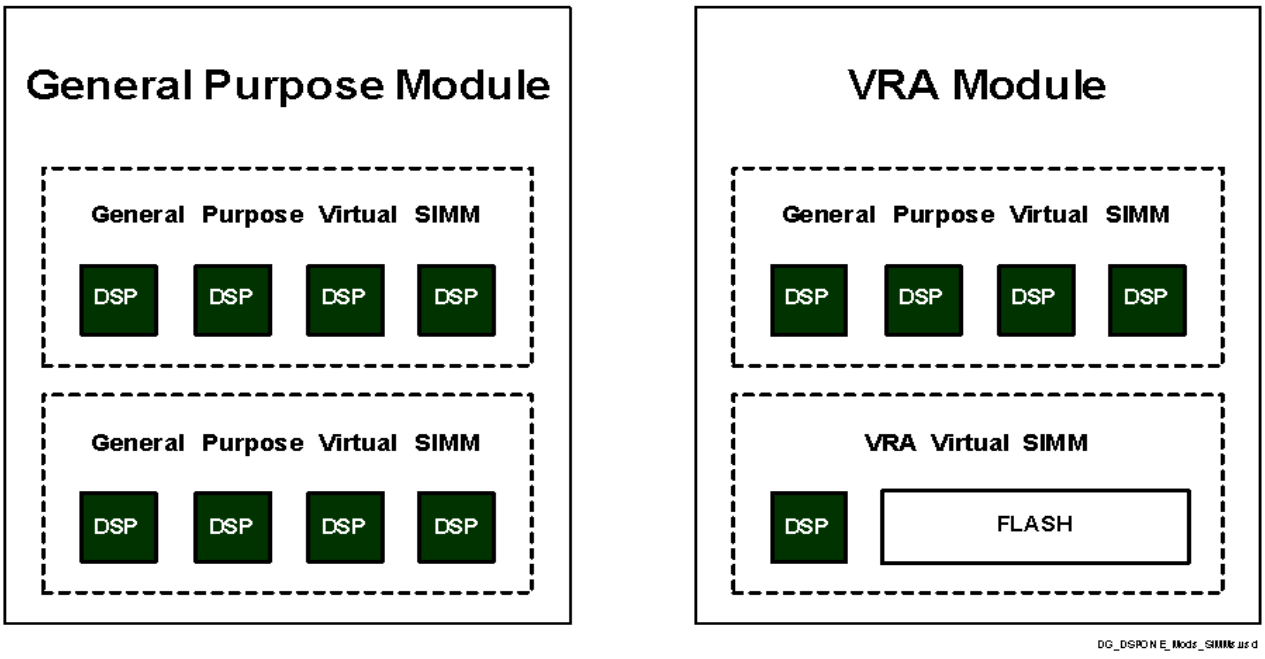
The DSP-ONE card has daughter modules that are not field-replaceable. The eight DSPs on each module are grouped logically into two "Virtual SIMMs."

A DSP-ONE card with two daughter modules, each of which has two virtual SIMMs, has a total of 16 DSP chips.

The DSP-ONE card can be populated with any of the following combinations of modules:

- One General Purpose Module (contains two logical General Purpose SIMMs)
- Two General Purpose Modules
- One VRA Module (contains one logical General Purpose SIMM and one logical VRA SIMM)
- Two VRA Modules
- One General Purpose Module and one VRA Module

Figure 9-1 SIMMs and Modules on the DSP-ONE Card



DG_DSPONE_Mods_SIMMs.ai d

DSP Features

Table 9-1 DSP ONE Features

Feature	NOTES
Maximum number of DSP chips	16
Configuration API Message	*DSP SIMM Configure* (0x00C0)
Tone Generation and Reception	a DSP is assigned a particular tone type, and can play that for 2,048 channels
Call Progress Analysis	
Energy Detection	
Voice Recorded Announcements	provided on the VRA SIMM
Announcement Tracking	the Matrix Controller card keeps track of announcement IDs and where announcements are stored on the cards
Conferencing	- Standard - Mixed - Monitor - Unified
Coin Tone Detection	
E1 Dial Pulse Address Signaling	

DSP Resources

Table 9-2 shows the number of DSP resources available per DSP chip for each function on the DSP-ONE card.

Table 9-2 Resources per DSP SIMM

Number	Function	Resources
Tone Reception		
0x01	DTMF (µ-law)	22
0x02	MFR1 (µ-law)	45
0x03	DTMF (A-law)	22
0x04	MFR1 (A-law)	45
0x05	MFR2 (A-law)	29
0x06	MFR2 (µ-law)	29
0x07	CPA (A-law)	20
0x08	CPA (µ-law)	20
0x09	Dial Pulse	20
0x0A	Energy Detection	17
0x0C	DTMF HPF (µ-law)	N/A
0x0D	DTMF HPF (A-law)	N/A
Tone Generation*		
0x10	DTMF (µ-law)	2048
0x11	DTMF (A-law)	2048
0x12	MFR1 (µ-law)	2048
0x13	MFR1 (A-law)	2048
0x17	MFR2 Forward (µ-law)	2048
0x16	MFR2 Forward (A-law)	2048
0x19	MFR2 Backward (µ-law)	2048
0x18	MFR2 Backward (A-law)	2048
0x14	CPT4 (µ-law)	2048
0x15	CPT4 (A-law)	2048

Number	Function	Resources
0x1A	Bong Tone (µ-law)	16
0x1B	Bong Tone (A-law)	16
0x30	Universal Gen. (µ-law)	N/A
0x31	Universal Gen. (A-law)	N/A
Conferencing		
0x1E	µ-law Standard	14 (7+ Broadcast per conference)
0x1F	A-law Standard	14 (7+ Broadcast per conference)
0x20	Mixed	34 (9 per conference)
0x21	Monitor	34 (9 + Broadcast per conference)
0x22	Unified Dynamic	31 (25 + Broadcast per conference)
0x23	Unified Dynamic DTMF Clamped	24 (24 + Broadcast per conference)
Announcement Generation		
0x1C	VRA SIMM/Module	55
0x1D	File Record/Playback	N/A

*The transmit tones for DTMF, MFR1, MFR2, and CPT4 are non-blocking, so they can be connected to up to 2,048 channels per node. Therefore, these tone generator types are not listed here.

Mixing DSP Cards in a CSP

Compatibility

- an MFDSP card may **NOT** reside in the same chassis as a DSP Series 2 card.

Card Combinations	Compatible?
MFDSP and DSP-ONE	Yes
DSP-ONE and DSP Series 2	Yes
MFDSP and DSP Series 2	No

Function Allocation

In CSPs that use both the DSP-ONE and DSP Series 2 card, DTMF, MFR1, MFR2, and CPT transmitters all use resources from the DSP-ONE first if it is configured for them. The *RAN Connect* message always uses resources from the DSP-ONE card first. Receivers and Bong Tone are assigned in a round-robin fashion. The chip with the most resources available gets Dynamic Conferencing and Clamped Conferencing first. The chip with the least resources available gets Static Conferencing and Monitor Conferencing first.

DSP-ONE and DSP 2 Sample Configuration 1

The following sample configuration illustrates an CSP with one DSP Series 2 card (with one module) and one DSP-ONE card (with 2 modules)

DSP-ONE card with two DSP Modules

- Seven DSPs DTMF Detection = 154 Ports
- Seven DSPs CPA Detection = 140 Ports
- One DSP DTMF Generation = 2,048 Ports
- One DSP CPA Generation = 2,048 Ports

DSP Series 2 card with one DSP Module

- Two DSPs DTMF Detection and Tone Generation = 1,024 Ports of DTMF Detection, and 1,024 Ports of any tone generation
- Two DSPs CPA Detection and Tone Generation = 1,024 Ports of CPA detection, and 1,024 ports of any tone generation

DSP-ONE and DSP 2 Sample Configuration 2

The following sample configuration illustrates an CSP with one DSP Series 2 card with one module, and one DSP-ONE card with two VRAS modules.

DSP-ONE card with two VRAS Modules

- Three DSPs DTMF Detection = 66 Ports
- Three DSPs CPA Detection = 60 Ports
- One DSP DTMF Generation = 2,048 Ports
- One DSP CPA Generation = 2,048 Ports
- Two VRAS = 104 minutes of announcements and 110 simultaneous channels

DSP Series 2 card with one DSP Module

- Three/two DSP DTMF Detection = 768 Ports
- Three/two DSP CPA Detection = 768 Ports
- One DSP Record/Playback = Virtually unlimited announcement storage, 128 simultaneous channels.

DSP Function Types

Table 9-3 Default Function Configuration

Module Number	DSP Number	Rcv 0	Txmt 0	Rcv 1	Txmt 1
0	0	DTMF µ-law (0x01)	Universal µ-law (0x30)	DTMF µ-law (0x01)	Universal µ-law (0x30)
0	1	DTMF µ-law (0x01)	Universal µ-law (0x30)	DTMF µ-law (0x01)	Universal µ-law (0x30)
0	2	CPA µ-law (0x08)	Universal µ-law (0x30)	CPA µ-law (0x08)	Universal µ-law (0x30)
0	3	File Record/ Playback (0x1D)	Not allowed (0x00)	File Record/ Playback (0x1D)	Not allowed (0x00)
1	0	DTMF µ-law (0x01)	Universal µ-law (0x30)	DTMF µ-law (0x01)	Universal µ-law (0x30)
1	1	DTMF µ-law (0x01)	Universal µ-law (0x30)	DTMF µ-law (0x01)	Universal µ-law (0x30)
1	2	MFR1 µ-law (0x02)	Universal µ-law (0x30)	MFR1 µ-law (0x02)	Universal µ-law (0x30)
1	3	MFR1 µ-law (0x02)	Universal µ-law (0x30)	MFR1 µ-law (0x02)	Universal µ-law (0x30)

Resources Available per DSP Chip

Table 9-4 shows the number of DSP resources available per DSP chip for each function on the DSP-ONE cards.

Table 9-4 Resources per DSP Chip

Number	Function	Resources per DSP chip
Tone Reception		
0x01	DTMF (µ-law)	22
0x02	MFR1 (µ-law)	45
0x03	DTMF (A-law)	22
0x04	MFR1 (A-law)	45
0x05	MFR2 (A-law)	29
0x06	MFR2 (µ-law)	29
0x07	CPA (A-law)	20
0x08	CPA (µ-law)	20
0x09	Dial Pulse	20
0x0A	Energy Detection	17
Tone Generation*		
0x10	DTMF (µ-law)	2048
0x11	DTMF (A-law)	2048
0x12	MFR1 (µ-law)	2048
0x13	MFR1 (A-law)	2048
0x17	MFR2 Forward (µ-law)	2048
0x16	MFR2 Forward (A-law)	2048
0x19	MFR2 Backward (µ-law)	2048
0x18	MFR2 Backward (A-law)	2048
0x14	CPT4 (µ-law)	2048
0x15	CPT4 (A-law)	2048
0x1A	Bong Tone (µ-law)	16

Number	Function	Resources per DSP chip
0x1B	Bong Tone (A-law)	16
Conferencing		
0x1E	µ-law Standard	14 (7+ Broadcast per conference)
0x1F	A-law Standard	14 (7+ Broadcast per conference)
0x20	Mixed	34 (9 per conference)
0x21	Monitor	34 (9 + Broadcast per conference)
0x22	Unified Dynamic	31 (25 + Broadcast per conference)
0x23	Unified Dynamic DTMF Clamped	24 (24 + Broadcast per conference)
Announcement Generation		
0x1C	VRA SIMM/Module	55

*The transmit tones for DTMF, MFR1, MFR2, and CPT4 are non-blocking, so they can be connected to up to 2,048 channels per node. Therefore, these tone generator types are not listed here.

Configuration the DSP Card

Purpose

This section describes how to configure the DSP chips. The maximum number of DSP chips for the DSP-ONE cards is 16.

Configuring Functions

Use the *DSP SIMM Configure* (0x00C0) message to configure individual DSP resource functions on one SIMM. In the message, you can assign different functions to each of the DSP chips on the SIMM. You can assign only one function type per DSP chip, and these function types are listed in the message. To configure SIMMs, follow the steps below:

1. Take each SIMM out of service using the *Service State Configure* (0x000A) message.
2. Configure each SIMM separately, with the *DSP SIMM Configure* (0x00C0) message. The host can specify the function to be loaded to each DSP chip.
3. Bring the SIMM back into service, with the *Service State Configure* (0x000A) message.

NOTE: VRA SIMMs on the DSP-ONE card are automatically configured for the VRA function.

TONES

Contents

Tone Generation

Types of Tones

The CSP lets you transmit address signaling and call progress tones. Each tone type is available for both µ-law and A-law. You can generate the following tones:

- Dual Tone Multi-frequency (DTMF)
- Multi-frequency R1 (MFR1)
- Multi-frequency R2 (MFR2 – backward and forward)
- Call Progress Tone Patterns (CPT4)
- Calling Card Prompt (Bong Tone)

The transmit tones for DTMF, MFR1, MFR2, and CPT4 are non-blocking. That is, when you generate one of these tones, it can connect to any number of channels. For example, a single DSP chip configured to generate MFR1 tones can provide MFR1 tones to all 2,048 ports in the CSP.

In contrast, the Calling Card Prompt (or Bong Tone) is a blocking tone that can connect to only one channel at a time. The Bong Tone is a blocking tone because it varies over time, so it must be played from start to end. A DSP chip can generate up to sixteen instances of the Bong Tone. When more than one channel requires a Bong Tone, the channels connect simultaneously to an instance of the Bong Tone, and remain connected from start to finish.

On the DSP-ONE card you can configure more than one DSP chip to generate Bong Tone on a single DSP card. But do not configure more than one transmitter on a DSP card to generate MFR1/MFR2, DTMF, or call progress tones of the same encoding format (µ-law /A-law). If you do, the configuration message receives a negative acknowledgment (NACK). Before you add a second CPT4 to a DSP chip, you must remove CPT4 from the first DSP chip, even if the DSP chip is out of service.

Table 9-5 summarizes the attributes of the tone generators.

Table 9-5 Tone Generator Attributes

Tone Type	Blocking/Non-Blocking	Number of Generators Allowed per DSP Card
DSP-ONE Card		
DTMF	Non-Blocking	One A-law / One µ-law
MFR1		
MFR2		
CPT4		
BONG	Blocking	No restriction
DSP Series 2 Card		
Universal	Blocking	No restriction

For MFR2 transmission and reception, every card that has an MFR2 receiver must also have a corresponding forward and backward MFR2 transmitter (or Universal transmitter for the DSP Series 2 card). To duplicate the generation of a tone for redundancy, configure the tone function on separate cards.

Address Signaling Tones

For outbound calls, configure the CSP to generate DTMF, MFR1, or MFR2 address signaling to the downstream CSP or device. To initiate outpulsing, use the *Outseize Control* (0x002C) message with an action of “Outpulse Stage N Address Data.” Data ICBs in the *Outseize Control* (0x002C) message indicate the signaling tone type and the source of the digits. Digits can be specified in the *Outseize Control* (0x002C) message itself, or they can be taken from a list of previously-inpulsed digits. Digit durations are set by the PPL component on the associated line card.

Example: You set the following durations using PPL timers, and you modify them using the *PPL Timer Configure* message (0x00CF):

- delay until the start of the first digit
- “on” time of the first digit
- “on” time of subsequent digits
- "off” time between all digits

For DTMF and MFR1 digits, outpulsing can also be initiated using the *Outpulse Digits* (0x0020) message. Within that message, the host can specify the following durations:

- delay until the start of the first digit
- “on” time of the first digit
- “on” time of subsequent digits
- “off” time between all digits

Durations are specified in units of 10 milliseconds. Actual outpulsed durations, however, are truncated to 20 millisecond intervals. So a digit duration specified as a multiple of 20 milliseconds is outpulsed with that duration, while other digit durations are shortened. For example, digits with a specified duration of 20, 40, or 60 milliseconds are outpulsed with those durations. But a digit with a specified duration of 30 milliseconds is truncated to have a 20 millisecond duration upon outpulsing. Similarly, 50 milliseconds is outpulsed as 40 milliseconds, 70 milliseconds is outpulsed as 60 milliseconds, and so on. Because of the truncated durations, you should set a minimum outpulse duration for 20 milliseconds or greater.

The host dynamically configures the dBm levels of DTMF, MFR1, and MFR2 transmit tones, using the *Tone Configure* message (0x0031). Power levels are updated globally for each tone type. For DTMF tones, you must specify the dBm level for both low-band and high-band frequency components. For MFR1 and MFR2 tones, you must specify only one dBm level.

Table 9-6 DTMF digit specifications

BCD	DTMF Digit	Frequencies (Hz)
0	0	941 Hz + 1336 Hz
1	1	697 Hz + 1209 Hz
2	2	697 Hz + 1336 Hz
3	3	697 Hz + 1477 Hz
4	4	770 Hz + 1209 Hz
5	5	770 Hz + 1336 Hz
6	6	770 Hz + 1477 Hz
7	7	852 Hz + 1209 Hz
8	8	852 Hz + 1336 Hz
9	9	852 Hz + 1477 Hz
A	A	697 Hz + 1633 Hz
B	B	770 Hz + 1633 Hz
C	C	852 Hz + 1633 Hz
D	D	941 Hz + 1633 Hz
E	*	941 Hz + 1209 Hz
F	#	941 Hz + 1477 Hz

Table 9-7 MFR1, MFR2 digit specifications

BCD	MFR1 Digit	MFR1 Frequency (Hz)	MFR2 Digit	MFR2 Forward Frequency (Hz)	MFR2 Backward Frequency (Hz)
0	0	1300 Hz + 1500 Hz	1	1380+1500	1140+1020
1	1	700 Hz + 900 Hz	2	1380+1620	1140+900
2	2	700 Hz + 1100 Hz	3	1500+1620	1020+900
3	3	900 Hz + 1100 Hz	4	1380+1740	1140+780
4	4	700 Hz + 1300 Hz	5	1500+1740	1020+780
5	5	900 Hz + 1300 Hz	6	1620+1740	900+780
6	6	1100 Hz + 1300 Hz	7	1380+1860	1140+660
7	7	700 Hz + 1500 Hz	8	1500+1860	1020+660
8	8	900 Hz + 1500 Hz	9	1620+1860	900+660
9	9	1100 Hz + 1500 Hz	A	1740+1860	780+660
A	KP	1100 Hz + 1700 Hz	B	1380+1980	1140+540
B	ST	1500 Hz + 1700 Hz	C	1500+1980	1020+540
C	STI	900 Hz + 1700 Hz	D	1620+1980	900+540
D	STII	1300 Hz + 1700 Hz	E	1740+1980	780+540
E	STIII	700 Hz + 1700 Hz	F	1860+1980	660+540

Call Progress Tone Pattern Transmission

Call progress tone patterns are audible signals that indicate the progress or disposition of a telephone call. Examples of call progress tone patterns include the busy signal, ringback, and dial tone.

About Call Progress Tone Patterns

A **frequency** is the most basic building block of a call progress tone pattern. A **tone** is a combination of frequencies, or it is a single frequency that has been designated as a tone. The frequency or frequencies in a tone have specified dBm levels. A **tone pattern** is a single tone or a sequence of tones, divided at precise intervals by silence.

A tone pattern is formed when tones are combined, with specific intervals of silence between them. For example, the industry-standard “call waiting tone” is actually a pattern, in which Tone 1 is on for 300 milliseconds, off for 9700 milliseconds, and on again for 300 milliseconds. You can think of a call progress tone pattern as being at the top of a hierarchy, with tone in the middle and frequency at the bottom. That is, one or more frequencies make up a tone, and one or more tones make up a pattern.

You can create and modify and create patterns with the message, *CPA Pattern Config*ure.

Figure 9-2 Call Progress Tone Pattern Hierarchy

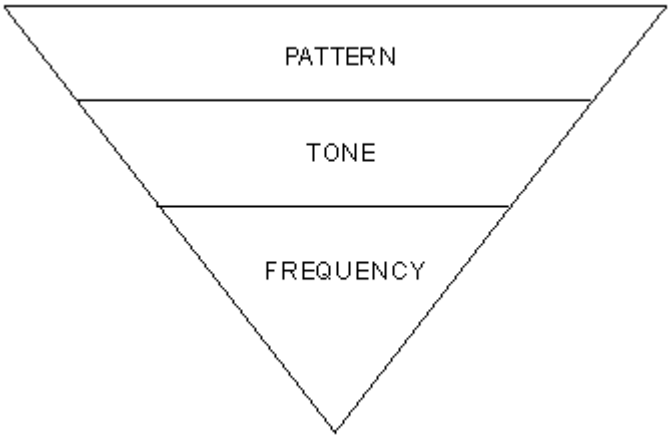
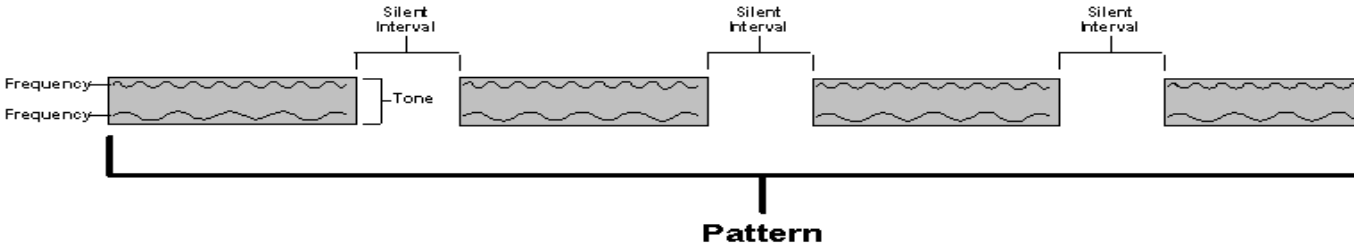


Figure 9-3 The Composition of a Call Progress Tone Pattern



Note: One or more frequencies can be used in a tone, and one or more tones can be used in a pattern

Default Transmit Call Progress Tones

Table 9-8 shows the default tones that the CSP uses to generate call progress tone patterns. Each call progress tone is distinguished by an ID. Note that some of the tones are single frequencies, while others are combinations of frequencies, at specific dBm levels.

Table 9-8 Default transmit call progress tones

	Tone ID											
Specification	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**0A**	**0B**	**0C**
Frequency Count	1	1	2	2	2	4	1	1	1	1	1	1
Frequency 0 (Hz)	440	480	350	440	480	1,400	914	985	1,371	1,429	1,777	400
dBm 0	-15	-15	-15	-20	-20	-8	-13	-13	-13	-13	-13	-18
Frequency 1 (Hz)			440	480	620	2,060						
dBm 1			-15	-20	-20	-8						
Frequency 2 (Hz)						2,450						
dBm 2						-8						
Frequency 3 (Hz)						2,600						
dBm 3						-8						

CAUTION

Although you can create a new tone or modify any of the default tones shown in the table above, you cannot create a new Tone ID. You must use an existing Tone ID for your new or modified tone. The tone that was originally associated with that Tone ID becomes unavailable. When you change the specifications of a tone, the change affects all patterns that use that tone.

Default Call Progress Tone Patterns

To form call progress tone patterns, the call progress tones are combined and separated by intervals of silence. Table 9-9 shows the default transmit call progress tone patterns. The parameters for each pattern are shown in Table 9-9.

Table 9-9 Default call progress tone patterns

Value	Call Progress Tone Pattern
0x01	Dial Tone
0x02	Ringback Tone
0x03	Line Busy Tone
0x04	Reorder Tone
0x05	Warning Tone
0x06	Call Waiting Tone
0x07	ONI Call (Zip Tone)
0x08	ANI Failure (Zip Tone)
0x09	Confirmation Tone
0x0A	Recall Dial Tone
0x0B	Class of Service Tone 2
0x0C	Class of Service Tone 3
0x0D	Intercept
0x0E	Vacant Code
0x0F	**Reorder – LEC**
0x10	No Circuit – Carrier
0x11	Reorder – Carrier
0x12	No Circuit – Carrier
0x13	Continuous 400 Hz Tone
0x14	Specialized tone to wait 500 ms and then play 480 Hz for 800 ms
0x15	Reserved for Bong Tone, cannot be modified

Parameters for Default Tone Patterns

Each pattern is composed of one to three cycle blocks, which include the following components:

- Tone ID
- ON duration
- OFF duration
- Number of times to repeat the cycle block

Table 9-10 shows the parameters for the default tone patterns (0xFFFF = continuous).

Table 9-10 Parameters for default tone patterns

Transmit Cadence Pattern	ID	Cycle Blocks	Cycle Block Number	Tone ID	ON Duration	OFF Duration	Cycles
Dial Tone	0x01	1	1	0x03	0xFFFF	0	1
Ringback	0x02	1	1	0x04	2000	4000	1
Line Busy	0x03	1	1	0x05	500	500	1
Reorder	0x04	1	1	0x05	250	250	1
Warning	0x05	1	1	0x06	100	100	1
Call Waiting	0x06	2	1	0x01	300	9700	1
			2	0x01	300	10	1
ONI Call (Zip Tone)	0x07	2	1	0x02	100	100	1
			2	0x02	100	10	1
ANI Failure (Zip Tone)	0x08	1	1	0x02	80	10	1
Confirmation	0x09	1	1	0x03	100	100	3
Recall Dial Tone	0x0A	2	1	0x03	100	100	3
			2	0x03	0xFFFF	10	1
Class of Service Tone 2	0x0B	1	1	0x05	800	10	1
Class of Service Tone 3	0x0C	1	1	0x02	800	10	1
Intercept	0x0D	3	1	0x07	280	10	1
			2	0x09	280	10	1
			3	0x0B	380	10	1
Vacant Code	0x0E	3	1	0x08	380	10	1
			2	0x09	280	10	1
			3	0x0B	380	10	1

Transmit Cadence Pattern	ID	Cycle Blocks	Cycle Block Number	Tone ID	ON Duration	OFF Duration	Cycles
Reorder - LEC	0x0F	3	1	0x07	280	10	1
			2	0x0A	380	10	1
			3	0x0B	380	10	1
No Circuit - LEC	0x10	3	1	0x08	380	10	1
			2	0x0A	380	10	1
			3	0x0B	380	10	1
Reorder - Carrier	0x11	3	1	0x08	280	10	1
			2	0x09	380	10	1
			3	0x0B	380	10	1
No Circuit - Carrier	0x12	3	1	0x07	380	10	1
			2	0x09	380	10	1
			3	0x0B	380	10	1
Continuous 400 Hz	0x13	1	1	0x0C	0xFFFF	10	1
Specialized Tone	0x14	2	1	0x00	500	10	1
			2	0x02	800	10	1
Bong Tone	0x15	Reserved, cannot be modified by host					
	0x16	1	1	0x0D	0xFFFF	0x0000	1
	0x17	1	1	0x0B	0xFFF	0x000	1
	0x18-0x3F	User-defined					

Generating Call Progress Tone Patterns

To generate call progress tone patterns, use the *Connect Tone Pattern* (0x002F) message. A call must be active (either incoming or outgoing) for a pattern to be generated. Then, use the *Connect Tone Pattern* message to connect the channel to the pattern that is specified by the Pattern ID.

Adjusting Attributes of Call Progress Tones

Overview

Many telephone networks require call progress tones with different patterns of frequencies, at different intervals. For example, the CSP may need to provide a different cadence of the Ringback Pattern based on the called party address. To ensure network compatibility, you may need to adjust the attributes of the call progress tones.

Modifying an Existing Tone or Pattern

The CSP allows the host to modify both the tones and the cadence of call progress tone patterns. When you modify a tone or a pattern, you affect all DSP chips that are configured to generate it. Although there is no difference in tone or pattern specifications with regard to µ-law and A-law encoding, you must configure the DSP chip appropriately (µ-law or A-law) with the *DSP SIMM Configure* (0x00C0) message.

The host can modify the default settings for any Tone ID, with the *Tone Configure Transmit* (0x0031) message. Changing the specifications of a Tone ID affects all patterns in the CSP using that Tone ID.

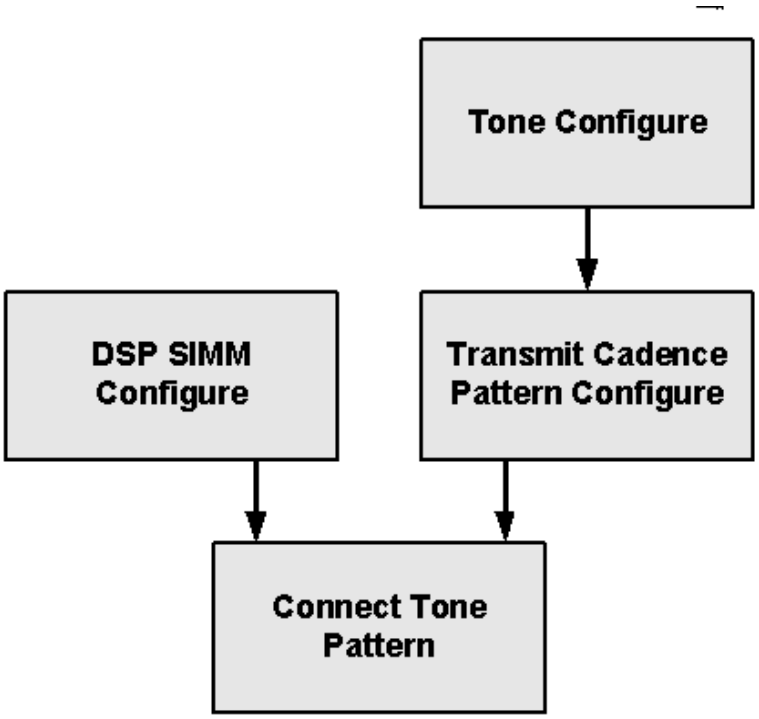
Creating a New Tone or Pattern

The host can also modify or create new patterns, with the *Transmit Cadence Pattern Configure* (0x0030) message. When you create a new pattern, you must assign a unique Pattern ID, within the range of 0x16 to 0x3F. The host must maintain an accurate list of current Pattern IDs and their specifications.

Order of Modifications

You must modify Tone IDs before you modify the affected patterns. You must modify patterns before you use the *Connect Tone Pattern* (0x002F) message. This is accomplished by sending a sequence of API messages, as depicted in Figure 9-4 below. To minimize reconfiguration time, set the Update All Flag in only the last configuration message that you send

Figure 9-4 API Message Sequence



Interval Durations

Interval durations are specified in units of 10 milliseconds. However, actual transmitted durations are truncated to 20 millisecond intervals. So an interval duration specified as a multiple of 20 milliseconds is transmitted with that duration, while other interval durations are shortened.

For example, intervals with a specified duration of 20, 40, or 60 milliseconds are transmitted with those durations. But an interval with a specified duration of 30 milliseconds is transmitted with a 20 milliseconds duration, an interval with a specified duration of 70 milliseconds is transmitted with a 60 milliseconds duration, and so on. Because durations are handled this way, your shortest interval duration should be 20 milliseconds.

Restoring Default Settings

To restore default settings for the transmit tone parameters, send the *Reset Configuration* (0x000B) message, or power down both Matrix Controller cards, and then power them up again.

Tone Reception

Collecting Digits and Analyzing Call Progress

The CSP provides tone reception for collecting digits and for analyzing call progress. The software maintains separate tone receiver pools for DTMF, MFR1, MFR2 (Backward and Forward), and call progress reception, dynamically configured and verified upon start-up. These pools are shared resources, for all channels to use.

Address Signaling Tones

This section describes the different types of Address Signaling tones.

Digit Collection During Call Setup

During call setup, the CSP can collect DTMF, MFR1, and MFR2 (Forward and Backward) digits.

CAUTION

For MFR2 transmission and reception, any card with an MFR2 receiver must also have an MFR2 forward and backward transmitter. If only MFR2 receivers are present, an incoming call causes a DSP Function Not Configured alarm and the MFR2 receivers are not used.

DNIS and ANI Identification

You can program the CSP to automatically collect the inpulsing address data that typically accompany an incoming call. This information identifies the Dialed Number Identification Service (DNIS) and/or the Automatic Number Identification (ANI). A service application can use this data to validate and authenticate subscribers, and to identify service.

Programming Digit Collection

To program digit collection during call setup, the host uses a combination of the *DSP SIMM Configure, Inpulsing Parameters,* and *Inseize Control* messages.

DSP SIMM Configure

This API message configures the function of an individual DSP chip. Tone reception functions include DTMF, MFR1, MFR2, and E1 Dial Pulse reception (E1 Dial Pulse is not available on the DSP Series 2 card).

Inpulsing Parameters Configure

This API message configures parameters for collecting inpulse data on specified channels. Inpulsing parameters define the address signaling type, the number of digit strings, and the collection method used during call setup.

The CSP supports four different inpulsing stages, each with one or two digit strings. When the host is instructing the CSP to collect address signaling information (that is typically presented with in-band dual frequency tones) it also specifies a preprogrammed inpulsing stage that describes how to perform the digit collection. The inpulsing stage configuration options include the following:

- Address signaling type (DTMF, MFR1, MFR2)
- Number of strings (1 or 2)
- String collection method (fixed number digits, KP/ST framed, compelled)

Inseize Control

This API message passes inseize instructions to the CSP for controlling incoming call setup in real time. The instruction, "Receive Stage N Address Data" allocates and attaches a DSP resource to a channel to collect an incoming digit stream. Up to 100 digits can be collected within a given inpulsing stage. Use the *Inseize Instruction List Configure* (0x0029) message to preprogram instructions on a channel. See "Digit Collection During Call Setup" for more information.

> **Important!** The instruction, Report Incoming Call With Address Digits, sends a *Request For Service With Data (*0x002D) message to the host, with single- or multiple-stage digit streams collected.

Digit Collection After Call Setup

After a call is set up, the CSP collects DTMF digits interactively. The host collects additional information, either one digit at a time with the *DSP Service Request* message, or as a sequenced group of digits with the *Collect Digit String* message.

DSP Service Request

This API message allocates an appropriate digit receiver and attaches it to the specified channel. The attached digit receiver uses the *Call Processing Event* message to report digits as they are decoded, one at a time. The receiver remains attached to the channel until the channel is released or until the host cancels digit collection, with the *DSP Service Cancel* (0x00BE) message.

Collect Digit String

This API message allocates and attaches a digit receiver to a channel, but unlike the *DSP Service Request* message, it causes the CSP to collect a group of digits in sequence until termination condition occurs.

The termination condition could be the detection of a particular digit from a termination digit set, a fixed number of digits collected, or a timeout.

When the termination condition occurs, the CSP reports the entire group of digits collected, with the *Call Processing Event* message. The receiver is then returned to the system receiver pool. If the configuration bit 4 is not set (or "by default") when the CSP detects the first valid digit, it automatically cancels any prompting tone or recorded announcement being played out to that channel.

Call Progress Tone Pattern Reception

Overview

Call progress tone patterns are audible signals that indicate the progress or disposition of a telephone call. The busy signal, ringback, and dial tone are all examples of call progress tone patterns. The act of receiving and interpreting call progress tone patterns is called Call Progress Analysis (CPA).

International Call Progress Analysis

The CSP can generate and receive Call Progress (CP) Tones, such as dial tone and line busy tone. The system software has long handled the frequencies used in North American phone systems. The CSP can now transmit and receive other tones, including a 400 Hz tone used outside of North America.

When a CP-receiver activates, it detects call progress patterns that are part of a Call Progress Analysis (CPA) Class. The CP-receiver ignores any CP-receive tones that are not part of a pattern in this CPA Class. This scenario allows the receiver to skip over tones that are not of interest, reducing the chance of receiving false *Call Progress Analysis Result* API messages.

Call Progress Tone Patterns

For information about how call progress tone patterns are formed, refer to the section, Call Progress Tone Pattern Transmission.

For call progress analysis, the CSP groups patterns into classes to facilitate management of CPA receivers. These CPA classes help assign call progress tone patterns to CPA receivers. When the host assigns a DSP resource to scan for call progress tones, the host specifies a class of patterns. The DSP resource then scans for the patterns defined in that class.

> **Example:** You could place all progress tones used in a single country into one class. You could then assign this class to a DSP chip for call progress analysis for that country. The host can modify and create classes, using the *CPA Pattern Configure* message.

Default Call Progress Analysis Tone Patterns and Classes

All of the tones supported by the CSP are shown in Table 9-11. The group of tone IDs may use up to 16 frequencies. A pattern is formed when the tones below are played at specific intervals.

Table 9-11 Tones supported by the CSP

Tone ID	Frequency Count	Frequency 0 (Hz)	Frequency 1 (Hz)
0x00	1	0	
0x01	2	350	440
0x02	2	440	480
0x03	1	440	
0x04	1	480	
0x05	2	480	620
0x06	1	620	
0x07	1	914	
0x08	1	985	
0x09	1	1371	
0x0A	1	1429	
0x0B	1	1777	
0x0C	1	2000	
0x0D	1	1700	
0x0E	1	2100	
0x0F	1	425	
0x10	1	500	
0x11	1	1100	
0x12	1	1398	
0x13	1	1820	

Default Tone Patterns for CPA

The default tone patterns for call progress analysis appear in Table 9-12. To modify or create new patterns, use the *CPA Pattern Configure* message.

Table 9-12 Default tone patterns for call progress analysis

Value	CPA Tone Pattern
0x01	Ringback
0x02	Double Ringback
0x03	Busy
0x04	Reorder
0x05	PBX Intercept
0x06	SIT Intercept A
0x07	Vacant Code
0x08	Reorder-LEC
0x09	No Circuit-LEC
0x0A	Reorder-Carrier
0x0B	No Circuit-Carrier
0x0C	PBX Dial Tone
0x0D	Standard Dial Tone
0x0E	CPC Detection
0x14	SIT Intercept B

CPA Pattern Parameters The parameters associated with a CPA pattern appear in Table 9-13.

Table 9-13 Parameters associated with CPA patterns

Pattern ID	Description
Configuration Bits	Pattern-specific configuration (see *CPA Pattern Configure* message) 0x01 = Last Interval Continuous (for pattern ending with continuous tone) 0x02 = Detect Tone After Determination (for reporting Intercept Tones) 0x04 = Use As Internal Dial Tone (to detect dial tone without invoking Call Progress Analysis)
CPA Result on Pattern Loss	Call Progress Analysis Result to report when a pattern has been matched but discontinues before “interval cycles to report”. (*CPA Result* message).
Interval Cycles to Match	The number of times an Interval Sequence must be repeated before declaring the pattern valid
Interval Cycles to Report	The number of times an Interval Sequence must be repeated before reporting a Call Progress Result to the host
Interval Descriptor Count	The number of Interval Descriptors in the pattern
Interval Descriptor	An Interval Descriptor consists of the Tone ID, Min. Filter, and Max. Filter, where: Tone ID – A valid Tone ID Minimum/Maximum Filter (MSB, LSB) – Minimum and maximum duration for this tone interval - A tone is determined as valid if the on and off cycles fall within the minimum and maximum times.

An Interval Descriptor is a list of the tones and intervals that constitute a pattern. The CSP uses the Interval Descriptor to confirm that a specific pattern has been detected.

Reporting Call Progress Tones

Detected call progress tones can be reported to the host in the following two ways:

- During call setup, with the outseize instruction of Do Call Progress Analysis
- Interactively, by assigning a CPA Receiver with the *DSP Service Request* message.

Table 9-14 lists the maximum quantities for call progress tone detection in the CSP.

Table 9-14 Maximum number for call progress tone detection

Tone Specification	System Maximum
Patterns	30
Classes	15
Patterns per class	15
Frequencies per tone	2

Receiving Call Progress Tone Patterns

The algorithm for detecting call progress tones has the following two phases:

1. Pattern Detection – The first phase recognizes the pattern, based on the tones and cadences received.
2. Pattern Match – The second phase counts tone intervals, to confirm the presence of a pattern. The number of intervals to match is specified by the pattern's Interval Cycles to Match parameter.

The algorithm does not move to the second phase until the pattern is identified. Configurable timers determine the maximum time to remain in each phase.

The CSP scans for call progress tones in the specified class, and reports them in the *CPA Result* message. CPA terminates upon reporting a result.

Off-hook Detection

When CPA is initiated with the *DSP Service Request* message, detection of off-hook is reported if the signaling interface supports it.

The parameter values for each call progress analysis tone pattern are shown in Table 9-15, Table 9-16, Table 9-17, and Table 9-18. To modify these values, use the *CPA Pattern Configure* message.

Example:

- You can configure the CPA receivers to detect a fax machine that the CSP has called. When the connection is made, the fax machine sends back a pattern containing Tone 0x0E (2100 Hz) continuously for approximately 2.3 seconds.
- To configure the CPA receivers to detect this signal, use the *CPA Pattern Configure* message to create a new pattern, consisting of Tone 0x0E continuously **ON** for a minimum of 2 seconds.
- You would then add this pattern to a CPA Class using the *CPA Pattern Configure* message (described next). Whenever CPA is performed using that CPA Class, the CPA receiver scans for the fax signal.

Table 9-15 Default Parameters for 0x00 to 0x07 - Part 1

Elements	Pattern			
	Ringback	Double Ringback	Busy	Reorder
Pattern ID	0x01	0x02	0x03	0x04
Tone Group ID	0x00	0x00	0x00	0x00
Configuration Bits	0x00	0x00	0x00	0x00
CPA Result on Pattern Loss	0x80	0x80	0x03	0x04
Interval Cycles To Match	1	1	1	1
Interval Cycles to Report	3	3	1	1
Interval Descriptor Count	2	4	2	2
Tone Interval 0 Tone ID	0x02	0x02	0x05	0x05
Reserved	0x00	0x00	0x00	0x00
Minimum Filter (ms)	600	420	420	200
Maximum Filter (ms)	2200	580	580	300

Elements	Pattern			
	Ringback	**Double Ringback**	**Busy**	**Reorder**
Tone Interval 1 Tone ID	0x00	0x00	0x00	0x00
Reserved	0x00	0x00	0x00	0x00
Minimum Filter (ms)	2800	200	420	200
Maximum Filter (ms)	5000	400	580	300
Tone Interval 2 Tone ID		0x02		
Reserved		0x00		
Minimum Filter (ms)		420		
Maximum Filter (ms)		580		
Tone Interval 3 Tone ID		0x00		
Reserved		0		
Minimum Filter (ms)		2000		
Maximum Filter (ms)		2500		

Table 9-16 Default Parameters for 0x00 to 0x07 - Part 2

Elements	Pattern		
	PBX Interception	**SIT Interception**	**Vacant Code**
Pattern ID	0x05	0x06	0x07
Tone Group ID	0x00	0x00	0x00
Configuration Bits	0x02	0x02	0x02
CPA Result on Pattern Loss	0x05	0x06	0x07
Interval Cycles To Match	1	1	1
Interval Cycles to Report	1	1	1
Interval Descriptor Count	2	3	3
Tone Interval 0 Tone ID	0x03	0x07	0x08
Reserved	0x00	0x00	0x00
Minimum Filter (ms)	100	200	300
Maximum Filter (ms)	300	350	460
Tone Interval 1 Tone ID	0x06	0x09	0x09
Reserved	0x00	0x00	0x00
Minimum Filter (ms)	100	200	200
Maximum Filter (ms)	300	350	350
Tone Interval 2 Tone ID		0x0B	0x0B
Reserved		0x00	0x00
Minimum Filter (ms)		300	300
Maximum Filter (ms)		460	460

Table 9-17 Default Parameters for 0x08 to 0x13 - Part 1

Elements	Pattern			
	Reorder LEC	No Circuit LED	Reorder Carrier	No Circuit Carrier
Pattern ID	0x08	0x09	0x0A	0x0B
Tone Group ID	0x00	0x00	0x00	0x00
Configuration Bits	0x02	0x02	0x02	0x02
CPA Result on Pattern Loss	0x08	0x09	0x0A	0x0B
Interval Cycle To Match	1	1	1	1
Interval Cycle to Report	1	1	1	1
Interval Descriptor Count	3	3	3	3
Tone Interval 0 Tone ID	0x07	0x08	0x08	0x07
Reserved	0x00	0x00	0x00	0x00
Minimum Filter (ms)	200	300	200	300
Maximum Filter (ms)	350	460	350	460
Tone Interval 1 Tone ID	0x00	0x0A	0x09	0x09
Reserved	0x00	0x00	0x00	0x00
Minimum Filter (ms)	300	300	300	300
Maximum Filter (ms)	460	460	460	460
Tone Interval 2 Tone ID	0x0B	0x0B	0x0B	0x0B
Reserved	0	0	0	0
Minimum Filter (ms)	300	300	300	300
Maximum Filter (ms)	460	460	460	460

Table 9-18 Default Parameters for 0x08 to 0x13 - Part 2

Elements	Pattern			
	PBX dial tone	**Standard dial tone**	**CPC Detection**	**Fax**
Pattern ID	0x0C	0x0D	0x0E	0x13
Tone Group ID	0x00	0x00	0x00	0x00
Configuration Bits	0x01	0x05	0x01	0x00
CPA Result on Pattern Loss	0x0C	0x0D	0x0E	0x13
Interval Cycle To Match	1	1	1	1
Interval Cycle to Report	1	1	1	1
Interval Descriptor Count	7	1	1	2
Tone Interval 0 Tone ID	0x01	0x01	0x01	0x11
Reserved	0x00	0x00	0x00	0x00
Minimum Filter (ms)	80	500	1500	425
Maximum Filter (ms)	120	0	0	575
Tone Interval 1 Tone ID	0x00			0x00
Reserved	0x00			0x00
Minimum Filter (ms)	80			2550
Maximum Filter (ms)	120			3450
Tone Interval 2 Tone ID	0x01			
Reserved	0x00			

Minimum Filter (ms)	80			
Maximum Filter (ms)	120			
Tone Interval 3 Tone ID	0x00			
Reserved	0x00			
Minimum Filter (ms)	80			
Maximum Filter (ms)	120			
Tone Interval 4 Tone ID	0x01			
Reserved	0x00			
Minimum Filter (ms)	80			
Maximum Filter (ms)	120			
Tone Interval 5 Tone ID	0x00			
Reserved	0x00			
Minimum Filter (ms)	80			
Maximum Filter (ms)	120			
Tone Interval 6 Tone ID	0x01			
Reserved	0x00			
Minimum Filter (ms)	500			
Maximum Filter (ms)	0			

CPA Class Parameters

By default, the CSP uses the following four classes:

- 0x00 Standard North America
- 0x01 Dial Tone
- 0x02 CPC Detection
- 0x03 Energy Detection

The parameters associated with a CPA class are defined in Table 9-19. To modify a class or to add a new class, use the *CPA Class Configure* message. The CSP supports up to 15 classes. A class can contain up to 15 pattern members, and 30 patterns are allowed for each CSP. Class elements are given in Table 9-20.

Table 9-19 CPA Class Parameters and Definitions

Class Parameter	Definition
Class Members	Patterns that are included in the class
Confirmation	Amount of time to wait to detect a pattern
Confirmation/No Report	Amount of time after detection of a tone to wait for pattern confirmation before terminating CPA
Continuous Silence Timeout	Amount of time after CPA is initiated to wait to hear any tone
Maximum On Timer	The maximum amount of time a tone must be “on” without a pattern match before the “continuous on” result is reported
Maximum Off Timer	The maximum amount of time silence must be detected before the “not determined” result is reported; amount of time to validate a silence detection before declaring it as silence
Minimum Frequency Glitch Time	Amount of time to validate a tone detection
Minimum Silence Glitch Time	Amount of time to validate a silence detection
Mode	Call Progress Analysis Mode (for example, Advanced Answer Detect, Energy Detection)
Mode Specific Values	Values to configure for specific CPA mode
Start Delay	Amount of time to wait before analysis begins
Tone Burst Answer Threshold	The number of frequency bursts in 500ms before answer is declared

Table 9-20 Class Elements

Class	Standard N. American 0x00	Dial Tone 0x01	CPC Detection 0x02	Energy Detection 0x03
Mode	0x01 (Adv. Answer Detect)			Energy Detection
Tone Burst Answer Threshold	4/250 ms			4/500 ms
Start Delay	0x00	0x00	0x00	0x00
Continuous Silence Timeout (ms)	25,000	25,000	0xFFFF	25,000
Confirmation (ms)	13,000	25,000	0xFFFF	13,000
Confirmation /No Report (ms)	22,000	15,000	0xFFFF	22,000
Min. Frequency Glitch Time (ms)	40	20	20	20
Min. Silence Glitch Time (ms)	60	20	20	60
Maximum On Timer (ms)	2,800	25,000	0xFFFF	2,800
Maximum Off Timer (ms)	5,000	25,000	0xFFFF	5,000
Mode Specific 1	0x0000	0x0000	0x0000	0x0005
Mode Specific 2	0x0000	0x0000	0x0000	0x0002
Pattern Members	0x01 Ringback 0x02 Double Ringback 0x03 Busy 0x04 Reorder 0x05 PBX Intercept 0x06 SIT Intercept A 0x07 Vacant Code 0x08 Reorder-LEC 0x09 No Circuit LEC 0x0A Reorder Carrier 0x0B No Circuit Carrier 0x0C PBX Dial Tone 0x0D Standard Dial Tone 0x14 SIT Intercept B	0x0C PBX Dial Tone 0x0D Standard Dial Tone	0x0E CPC Detection	0x01 Ringback 0x02 Double Ringback 0x03 Busy 0x04 Reorder

Customizing Call Progress Tone Patterns for Reception

Overview

Follow the instructions below to set up your Call Progress Tone Patterns for reception.

Configuring a Call Progress Receive Tone:

You specify only the frequencies in the tone.

For example, to configure a CP Receive tone made up of the frequency 400 Hz, use the *Tone Configure* message to:

1. Set the number of frequencies making up the tone (Data 2) to 1.
2. Set the first frequency value to 400. You do no need to set any other frequency values

Deleting a Call Progress Receive Tone:

To delete a Call Progress Receive Tone, use the *Tone Configure* message to set the number of frequencies to 0. No frequency data needs to be sent.

1. Set the number of frequencies making up this tone (Data 2) to 0
2. Do not specify any frequency values and you are finished.

Effect of CPA Tone Configure on a DSP

The CSP allows the host to modify both the tones and the cadence of call progress tone patterns. When you modify a tone or a pattern, you affect all DSP chips that are configured to receive it.

Effect of Tone Configure message on the CSP

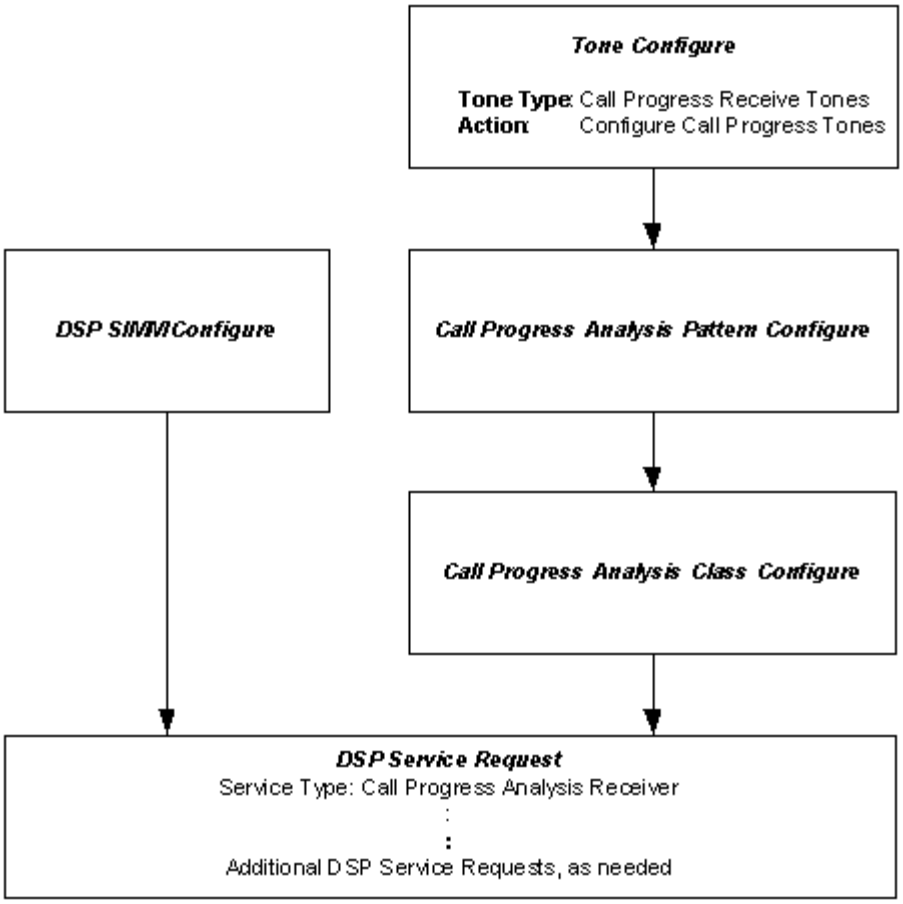
The host can modify the default settings for any Tone ID, using the *Tone Configure* message. Changing the specifications of a Tone ID affects all patterns in the CSP that are using that Tone ID.

Using the *Call Progress Analysis Pattern Configure* message

The host can also modify patterns and create new patterns, using the *Call Progress Analysis Pattern Configure* message. When you create a new pattern, you must assign a unique Pattern ID, within the range of 0x01 – 0x1F. The host must maintain an accurate list of current Pattern IDs and their specifications.

You must modify Tone IDs before you modify the affected patterns. The new pattern must be added to a class with the *Call Progress Analysis Class Configure* message. This sequence is depicted in *Figure 9-5*. To minimize reconfiguration time, set the *Update All Flag* in only the last configuration message that you send.

Figure 9-5 Adding New Pattern to Class



Restoring Default CPA Settings

To restore default settings for the transmit tone parameters, send the *Reset Configuration* message to the Matrix Controller, or power down both Matrix Controller cards, and then power them up again.

Configuring a Sample Pattern ID

Overview The following example shows how to configure Pattern ID 0x17 to receive a three-second, 400 Hz tone. To perform this task, the host sends three API messages as follows:

1. *Tone Configure* message to replace the default tone (Tone ID 0x0F, 425 Hz) with the new, 400 Hz tone.
2. *CPA Pattern Configure* message - create the new Pattern ID 0x17.
3. *CPA Class Configure* message - add Pattern 0x17 to class 0x00.

Table 9-21 First Message: *Tone Configure*

Byte	Field Description
0	Message Frame 0xFE
1, 2	Length (2 bytes) 0x0011
3, 4	Message Type (2 bytes) 0x0031
5	Reserved 0x00
6	Sequence Number
7	Logical Node ID 0xFF
8	Update All Flag 0x01
9	Reserved 0x00
10	Tone Type 0x06
11	Action 0x01
12	Data[0] Tone ID being changed 0x0F
13	Data[1] Reserved 0x00
14	Data[2] Number of frequencies in tone 0x01
15	Data[3] Reserved 0x00
16	Data[4] Frequency Value[0], Hz, MSB 0x01
17	Data[5] Frequency Value[0], Hz, LSB 0x90
18	Data[6] Reserved 0x00
19	Data[7] Reserved 0x00
20	Checksum

Table 9-22 Second Message: *Call Progress Analysis Pattern Configure*

Byte	Field Description
0	Message Frame 0xFE
1, 2	Length (2 bytes) 0x0016
3, 4	Message Type (2 bytes) 0x00B2
5	Reserved 0x00
6	Sequence Number
7	Logical Node ID 0xFF
8	Update All Flag 0x01
9	Pattern ID 0x17
10	Action: Add/Replace Pattern 0x01
11	Data[0] Pattern ID to report on detection 0x17
12	Data[1] Tone Group ID 0x00
13	Data[2] Pattern Configuration 0x01
14	Data[3] CPA Report on Pattern Loss 0x17
15	Data[4] Interval Cycles to Match 0x01
16	Data[5] Interval Cycles to Report 0x01
17	Data[6] Reserved 0x00
18	Data[7] Interval Descriptor Count 0x01
19	Data[8] Tone ID 0x0F
20	Data[9] Reserved 0x00
21	Data[10] Minimum filter (in 10 ms units) MSB 0x00
22	Data[11] Minimum filter (in 10 ms units) LSB 0x50
23	Data[12] Maximum filter (in 10 ms units) MSB 0x00
24	Data[13] Maximum filter (in 10 ms units) LSB 0x00
25	Checksum

Table 9-23 Third Message: *Call Progress Analysis Class Configure*

Byte	Field Description
0	Message Frame 0xFE
1, 2	Length (2 bytes) 0x0009
3, 4	Message Type (2 bytes) 0x00B3
5	Reserved 0x00
6	Sequence Number
7	Logical Node ID 0xFF
8	Update All Flag 0x01
9	Class ID 0x00
10	Action 0x01
11	Data[0] Pattern ID 0x17
12	Checksum

Energy Detection

Energy On Channel

The Energy Detection feature detects energy on a channel. You can use this feature for CPA or to report energy detection to the host application. To compensate for background noise and to filter out short energy bursts, the host application can set the sensitivity level and the scan duration (defined below). You can determine these settings based upon the type of energy to be detected, and to some extent, by trial and error. After setting both levels, test to see if the desired energy is being detected, then adjust the settings as necessary.

For CPA, the Energy Detection feature determines only that a pattern has occurred. It does not distinguish tones. The Energy Detection class (Class 3) is composed of the most common call setup tones: Ringback, Double Ringback, Busy, and Reorder. You can add any pattern to this class, or use any other class for Energy Detection, with the following restrictions:

- The pattern must consist of alternating periods of tones and silence. Energy Detection recognizes a pattern by periods with energy (tone) and no energy (silence). Because Energy Detection must detect both tone and silence, it cannot detect a “pattern” that has tone but no silent interval.
- The Energy Detection DSP function can detect only tones from the default patterns defined for the Energy Detection class (Tone IDs 0, 1, 2, and 5). You can still use a pattern for Energy Detection that does not use one of these tones. To do so, you must change the Tone IDs in the pattern to one of those allowed, using the *CPA Pattern Configure* message.

To use the Energy Detection feature, you must configure at least one DSP chip for Energy Detection, with the *DSP SIMM Configure* message.

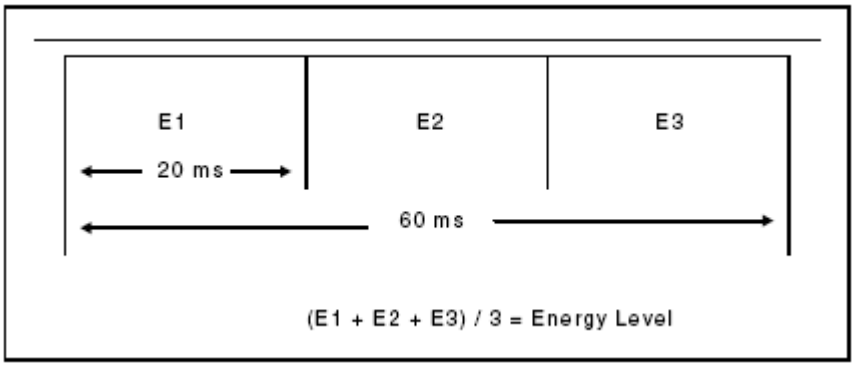
Configurable Parameters

Use the following parameters to modify energy detection:

- Sensitivity Level: the amplitude above which the energy detector perceives a signal. Set the sensitivity level to detect and report energy that is greater than the prevailing background noise. When you expect a significant background noise, use the least sensitive setting (0 dBm). When you expect little background noise, use the most sensitive setting (-30 dBm).
- Scan Duration: the repeating time interval over which the energy detector determines that energy is either Present or Not Present. Set the scan duration to be longer than expected energy bursts. You must use 20 millisecond intervals to set the scan duration, so energy is sampled for each 20 millisecond block within the specified duration.

  The scan cycles repeat until the completion timer expires. The calculated energy level is the average over all blocks, as shown in *Figure 9-6*. If the calculated energy level exceeds the configured sensitivity level, energy is reported.

Figure 9-6 Scan Cycle



- Completion Timer
  The completion timer determines the maximum amount of time to scan for energy. Each scan cycle, as defined by the scan duration, is repeated until either energy is detected, or the completion timer expires. We recommend setting the completion timer for 3 to 4 times the scan duration, depending on the application. If the timer expires before energy is detected, the host receives a *Call Processing Event* message of “No Energy Detected.”

Detection Methods

You can invoke Energy Detection in the following two ways:

- Interactively, with the *DSP Service Request* message
- As a part of Call Progress Analysis

Interactive Energy Detection

You can invoke Energy Detection interactively, with the *DSP Service Request* message. Use the data bytes of the message to configure the sensitivity level, reporting mode, scan duration, and completion timer. The detected energy is reported to the host in a *Call Processing Event* message in one of two modes: Report Initial Energy Detection Only, or Report All Energy Threshold Crossings.

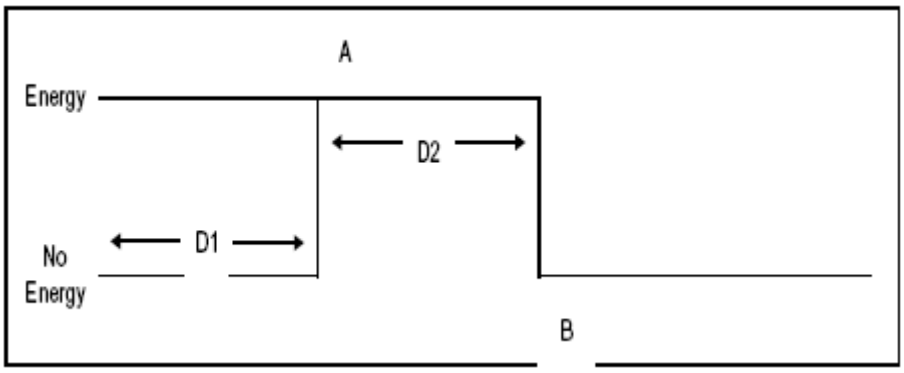
- Report Initial Energy Detection Only
  The host receives a *Call Processing Event* of "Energy Result Report" with Data[0], indicating Energy Detected. Data[1] indicates the duration of the period of no energy. The DSP resource is then automatically released.
- Report All Energy Threshold Crossings

  All of the following are reported: the initial energy detection, all subsequent changes, and the duration of the previous state's ON or OFF interval.

  When energy (A) is first detected, the host receives a *Call Processing Event* of "Energy Result Report" with Data[0] indicating "Energy Detected." Data[1] indicates the duration of the preceding period of no energy (D1).

  When energy is no longer detected (B), the host receives a *Call Processing Event* message of "Energy Result Report" with Data[0] indicating "No Energy Detected." Data[1] indicates the duration of the preceding period of energy (D2).

Figure 9-7 Energy Result Report



The DSP resource remains attached and reports energy changes until the completion timer expires, or until the host sends a *DSP Service Cancel 0x00BE* message, or until the channel returns to an idle state.

Energy Detection as Call Progress Analysis

Energy Detection allows the CSP to perform Call Progress Analysis for frequencies not supported by the default CPA DSP load. The Energy Detection DSP function matches cadences, based on the reported energy levels. CPA Class 3 is preconfigured for Energy Detection, using the standard CPA tones of ringback, double ringback, busy, and re-order.

To accommodate unique requirements for matching cadences, you can modify these patterns or add new patterns to the class. The host can change the pattern cadence that energy detection scans for, using the *CPA Pattern Configure* message.

Changing Mode Parameters

You can also configure the other CPA classes for Energy Detection by changing the mode parameter in the *CPA Class Configure* message, as shown in Table 9-24.

Table 9-24 Changing mode parameters

Message Field	Values
CLASS ID	Class to be changed
ACTION	0x03 (Change Class Parameter)
DATA[0] Parameter	0x01 (Mode)
DATA[1] Reserved	0x00
DATA[2] Mode	0x02 (Energy Detection enabled)

Specifying Sensitivity Level

The sensitivity level and scan duration are specified in the "Mode Specific Data" of the *CPA Class Configure* message, as shown in Table 9-25 and Table 9-26.

Table 9-25 Sensitivity Level Fields

Message Field	Values
Class ID	Class to be changed
Action	0x03 (Change Class Parameter)
Data[0] Parameter	0x0B (Mode Specific 1)
Data[1] Reserved	0x00
Data[2] Sensitivity Level	0x00 = 0 dBm 0x01 = -5 dBm 0x02 = -10 dBm 0x03 = -15 dBm 0x04 = -20 dBm 0x05 = -25 dBm 0x06 = -30 dBm

Specifying Scan Duration

Table 9-26 Scan Duration Fields

Message Field	Values
Class ID	Class to be changed
Action	0x03 (Change Class Parameter)
Data[0] Parameter	0x0C (Mode Specific 2)
Data[1,2] Scan Duration	16-bit word defining the scan duration, in 10ms units

Coin Tone

Overview

The Coin Tone feature is supported by the DSP-ONE card. It detects Coin Tones from pay telephones for the nickel, dime, quarter, and dollar. Coin Tones, including their timing and bandwidth specifications, are detailed in TR-TSY000456.

Avoids False Coin Tone Detections

The Coin Tone feature is capable of avoiding false Coin Tone detections caused by speech input, by detecting no more than a cumulative $1.00 when exposed to the DTMF talkoff test suite specified in TR-TSY000763.

Resides on the DSP Chip

All detection, cadencing, and discrimination of Coin Tones is performed on the DSP chip. Each DSP chip can service 20 simultaneous channels of coin detection.

Enabling the Coin Tone Receiver

Coin Tone detection is enabled by sending the *DSP SIMM Configure* message with function type Coin Detection (0x0F for µ-law, 0x0E for A-Law).

Requesting the Coin Tone Receiver

The Coin Tone receiver must be requested at least 200 ms prior to the first coin being detected for initialization. When a Coin Tone receiver is requested and there are no Coin Tone receivers currently available, the request is queued to a waiting list.

Attaching the Coin Tone Receiver

The host application may attach a Coin Tone receiver to a channel by sending the *DSP Service Request* API message with a Service Type of Coin Tone Receiver. The CSP then starts reporting Coin Tone detections. When a Coin Tone is detected, the CSP sends a Call Processing Event to the host application that indicates the value of the coin dropped, in cents.

Disconnecting the Coin Tone Receiver

The host application is responsible for disconnecting the Coin Tone receiver, by sending a *DSP Service Cancel* message. If the call is disconnected, the CSP releases the Coin Tone receiver.

Canceling the Coin Tone Receiver

The host application may cancel a Coin Tone receiver on a channel by sending the *DSP Service Cancel* message with a Service Type of Coin Tone Receiver.

Timing Out the Coin Tone Receiver

In the *DSP Service Request* message, you can specify the maximum time for the Coin Tone receiver to be attached. The Completion Timer specifies the maximum amount of time to scan for Coin Tones. If the completion timer expires, the CSP releases the Coin Tone receiver. To disable the timer, set this field to 0xFFFF.

VOICE RECORDED ANNOUNCEMENTS

Contents

Overview

The DSP-ONE cards can provide Voice Recorded Announcements (VRAs) on the VRA SIMM.

On the DSP-ONE card, the VRA feature physically resides on the VRA Module. Each VRA Module has two “logical SIMMs.” One of these logical SIMMs performs the VRA function, and the other logical SIMM provides four general purpose DSP chips.

Managing Voice Recorded Announcements

A voice recorded announcement is managed using a numeric Recorded Announcement ID (RAN ID) which must be a number between 0 and 4,095. The host assigns a RAN ID to each announcement as the announcement is downloaded. This RAN ID is then used whenever the host, using the *Recorded Announcement Connect* message (0x0055), commands the CSP to play out the announcement.

VRA Hardware Configuration

Both cards carry non-volatile flash memory, so they do not lose announcement information when they are removed from the CSP or when power is lost. You can store a particular announcement on multiple SIMMs simultaneously, but only once on each SIMM. If the host attempts to download the same announcement twice to the same SIMM, an alarm is returned to the host. Using these DSP cards, the maximum announcements per CSP is 4,096.

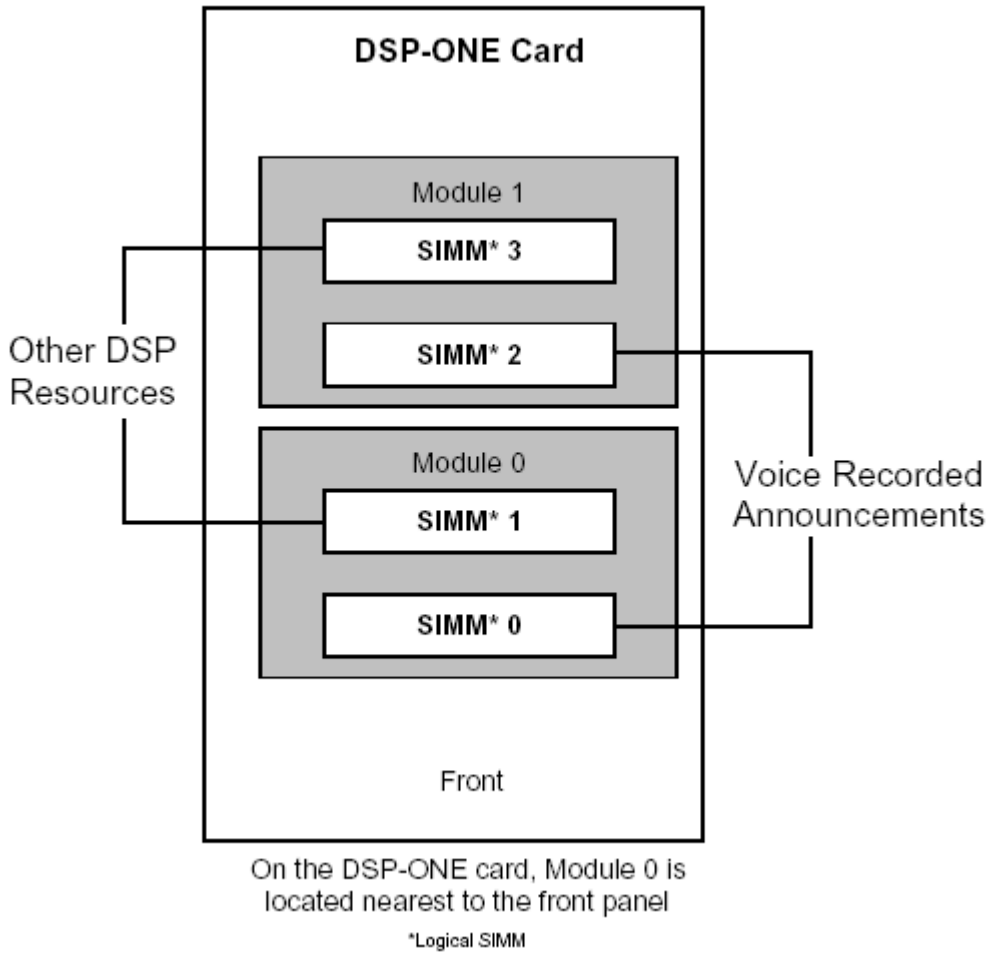
VRA Modules on the DSP-ONE Card

Each DSP-ONE card can be configured with either one or two VRA modules. Each VRA module contains two logical SIMMs. One of these logical SIMMs (number 0 or 2) is dedicated to voice recorded announcements, and you cannot change this default. The other logical SIMM (number 1 or 3) is available for other DSP functions, and it must be configured by the host.

Each logical VRA SIMM (one DSP chip per SIMM) can store up to 52 minutes of recorded voice, or up to 2,048 individual announcements, whichever comes first. Any single announcement can be up to 34.95 minutes long.

Figure 9-8 shows two VRA modules. The logical SIMM's number depends on the location of the VRA module.

Figure 9-8 VRA Module on the DSP-ONE Card



VRA SIMMs on the DSP-ONE Card

The VRA SIMM performs voice recorded announcements only. Each card can have up to four VRA SIMMs. Each VRA SIMM can store 1,000 seconds of recorded voice or up to 300 individual announcements, whichever comes first.

Downloading Voice Recorded Announcements

To download a voice recorded announcement, the host first sends a *Recorded Announcement Download Initiate* message (0x0052). The data in this message consists of the following:

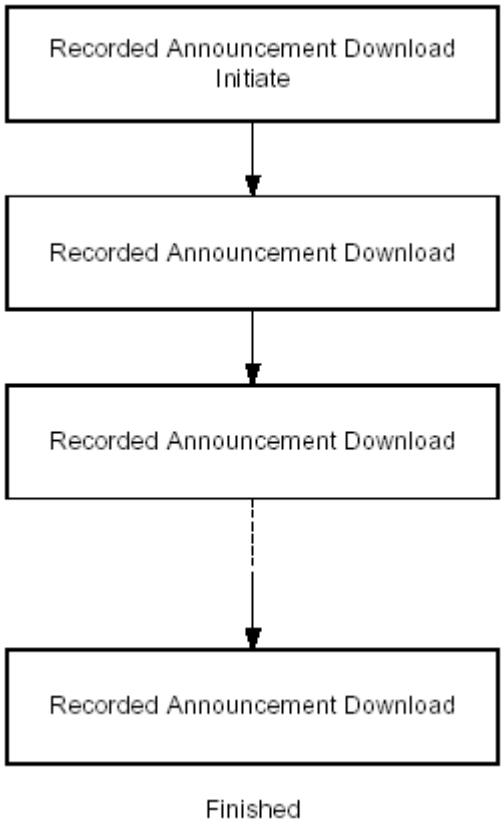
- Slot number of the DSP card
- SIMM number on that card
- Recorded Announcement ID
- Recorded Announcement Size
- Recorded Announcement Checksum
- Recorded Announcement Format
  (the only format currently supported is 8-bit PCM sampled at 8,000 times/second)
- Encoding Format (A-law, µ-law). The DSP-ONE card can play out announcements in any encoding format.

The DSP card stores the information above, and the CSP returns a positive acknowledgment. The host then sends the *Recorded Announcement Download* message (0x0053) containing the slot number of the DSP card, the number of the SIMM, the Announcement ID, and the actual PCM recorded voice data. On the DSP-ONE card, it takes less than three seconds to download one second of recorded voice.

If you try to download an announcement before you send the *Recorded Announcement Download Initiate* message (0x0052) you receive a Response Status of No Recorded Announcement Download Initiate Message Received.

Each *Recorded Announcement Download* message (0x0053) can send a maximum of 230 bytes, so you may need to send more than one of these messages for a single announcement. The download process is depicted in Figure 9-9.

Figure 9-9 Recorded Announcement Download



SIMM Checks Format

When the host has sent the entire announcement, the VRA SIMM checks the announcement format, encoding format, and checksum, to ensure that they match the values that were sent in the *Recorded Announcement Download* message (0x0052). The DSP card sends the information to the Matrix Controller, and the announcement is ready to be used. If the values do not match, the host is informed with an *Alarm* message.

One Announcement at a Time

You can download only one announcement at a time. You must wait for the first download to complete before you begin the second download. If you try to download more than one announcement at a time, you receive a Response Status of “Announcement Download Unsuccessful - Another Announcement Download is in Progress.”

Important! Send only **one** *Recorded Announcement Download* message (0x0052) for each announcement. Wait for one announcement to finish before you send another *Recorded Announcement Download* message for another announcement. The Recorded Announcement Download is finished when you receive the Alarm message (0x00B9) indicating Recorded Announcement Download Ready.

Unique Recorded Announcement ID

On an individual SIMM, each announcement must have a unique Recorded Announcement ID. You cannot download two announcements that have the same ID to the same VRA SIMM.

However, if there is more than one SIMM in the CSP, you can download the same announcement with the same ID to a different SIMM.

Invalid RAN Size and DSP Chip Dead Responses

If the RAN size is larger than the permitted size of 16,777,215 bytes (0xFFFFFF) the *Recorded Announcement Download Initiate* message (0x0052) receives a NACK of Invalid RAN Size (0x000A).

If a DSP chip fails to respond to the system monitor, the *Recorded Announcement Download* message (0x0053) receives a NACK of DSP Dead (0x0009).

Playing Announcements

Multiple VRA SIMMs

If you have multiple VRA SIMMs, you can play more announcements simultaneously. Using DSP-ONE cards, each logical VRA SIMM can play announcements simultaneously to 55 channels. Two logical VRA SIMMs can play announcements simultaneously to 110 channels.

If a particular announcement is played frequently, you should download that announcement to multiple VRA SIMMs so that the announcement can be played to more channels simultaneously. When the host commands the CSP to play a particular announcement, the CSP searches for the announcement on all VRA SIMMs. As soon as the CSP finds the announcement, it plays the announcement from that VRA SIMM.

Erasing Voice Recorded Announcements

You can erase individual announcements using the procedure described in the Single Message Deletion section of this chapter.

You can also erase all announcements from a SIMM using the *Recorded Announcement Delete* message (0x0054*)*. It takes approximately 30 seconds to erase a VRA SIMM. Before you replace or download announcements with this message, you must wait for an *Alarm* message indicating VRA Erase Complete.

Replacing Voice Recorded Announcements

To replace an announcement with a new announcement, we recommend that you download the new announcement with a new, unused ID. You can then use the new ID to play out the new announcement.

If you cannot use a new ID, you must re-use an existing ID, so you must erase all SIMMs that contain the existing ID. You must then download announcements to any erased SIMMs before those SIMMs can play out announcements again.

If your CSP contains more than one SIMM, you can maximize the number of available voice recorded announcements by erasing and downloading one SIMM at a time. While one SIMM is being erased and downloaded, other SIMMs can continue to provide voice recorded announcements.

Connecting To Voice Recorded Announcements

After an announcement is stored on a VRA SIMM, the host issues the *Recorded Announcement Connect* message (0x0055*)* and specifies the announcement's ID, as well as the span and channel to receive the announcement.

On the DSP-ONE card, each logical VRA SIMM can output to 55 channels simultaneously, which could mean 55 instances of the same announcement, 55 different announcements, or any combination of these. If the same announcement is stored on two logical SIMMs, for example, it can play to 110 channels simultaneously.

If the CSP receives requests for VRA resources that exceed the system's configuration, the requests are queued, then serviced when resources become available. When a request is queued to hear a specific message, the message is played as soon as the resource is available.

Barge-In Feature

The Barge-in feature lets an announcement repeat continually, as long as at least one channel is connected to it. Other channels requesting the announcement are allowed to connect ("barge in") to the playing announcement at any time. You enable the Barge-In feature with the Configuration Flag field in the Recorded Announcement Connect message (0x0055)message. If you are using Play File messages with the DSP2 card, you enable the Barge-In feature with the Barge in TLV.

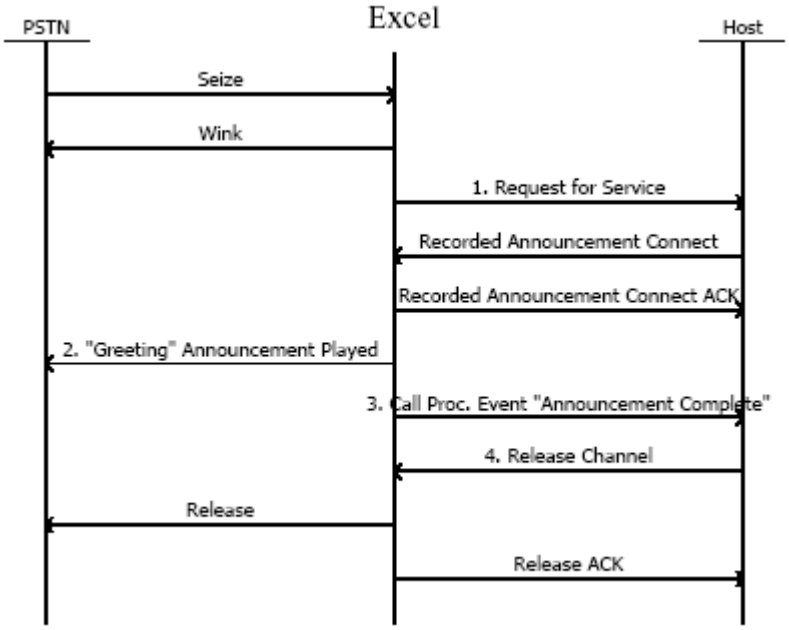
Recorded Announcement Call Flows

The diagram below shows a simple example of how the VRA features play out a greeting.

Trunk Type:E&M Wink Start

Inseize Control Instructions:

1. Generate Inseize Acknowledgment (Wink 1)
2. Report Incoming Call
3. Wait for Host Control



Description

1. Receive incoming call
2. Play out greeting
3. Receive *Call Processing Event* of “Recorded Announcement Complete”
4. Send *Release Channel 0x0008* message

Recorded Announcement Alarms

The Matrix Controller card maintains information from the *Recorded Announcement Download Initiate* message (0x0052) on all downloaded announcements in the CSP. Each DSP card maintains information on the announcements that it contains. When an announcement is downloaded, both cards verify that the information is correct and that it does not conflict with other announcements downloaded previously. For example, the cards verify that the same Announcement ID is not used for different announcements.

For alarms that indicate recorded announcement conflicts, the host must respond to quickly to ensure that the intended announcements are being stored and played-out.

See the *Alarm* message (0x00B9) in the *API Reference* for detailed information.

Single Message Deletion

You can delete a single Recorded Announcement (RAN) without erasing all the RANs on a logical VRA SIMM.

RANs are stored in flash memory on logical VRA SIMMs that reside on the DSP-ONE card. When a RAN is downloaded to a logical VRA SIMM, the host assigns an ID to it.

Limitation

VRA SIMM flash memory file system Revision 0 does not allow Single Message Deletion. VRA SIMM flash memory file system Revision 1 is required. This revision is included in software release 8.20 (CI) or higher. However, even if you are using software release 8.20 (CI) or higher, you must still use the *Recorded Announcement File System Convert* (0x0118) message to change the file system.

DSP SIMM Alarms

The following alarms are documented in Chapter 1 of the *API Reference*, within the *Alarm* (0x00B9) message:

- *Recorded Announcement File System Conversion Success* (0x07)
- *Recorded Announcement File System Conversion Failure* (0x08)
- *Recorded Announcement File System Timeout* (0x09)
- *Recorded Announcement Defragmentation Success* (0x0B)
- *Recorded Announcement Defragmentation Failure* (0x0C)
- *Recorded Announcement Single Deletion Complete* (0x0D)
- *Recorded Announcements Need Defragmentation* (0x0E)

Invoking Single Message Deletion

To enable the Single Message Deletion feature, you must first convert the flash memory file system on each logical VRA SIMM to Revision 1. You must perform the conversion once for each logical VRA SIMM.

You must take the logical VRA SIMM out of service before converting the file system. Any RANs that are on the logical VRA SIMM at the time of conversion are saved to the new file system format. The conversion normally takes about three to five minutes, but will be faster if no RANs are on the logical VRA SIMM. An alarm is sent to indicate when the conversion is complete.

If the VRA SIMM flash file system has not been converted to the new Revision 1, the *Recorded Announcement Single Delete* (0x0117) message is NACKed (unless the RAN ID field is 0xFFFF). You can

mix the new and the old flash file systems on a single DSP-ONE card, but only the VRA SIMMs with the new system will allow single message deletion.

Sequence

The order of the API messages to delete single messages is as follows. You need to perform this procedure only once for each SIMM:

Determine and change the file system:

1. Determine the flash file system Revision on a VRA SIMM by using *Recorded Announcement File System Query* (0x0102) to generate a *Recorded Announcement File System Report* (0x0119).
2. Take the VRA SIMM out of service with *Service State Configure* (0x0A).
3. Convert the flash to file system Revision 1 by using *Recorded Announcement File System Convert* (0x0118).
4. Wait three to five minutes for the VRA SIMM alarm *Recorded Announcement File System Conversion Success* (0x07)**.**
5. If necessary, download more RANs using *Recorded Announcement Download Initiate* (0x53) and *Recorded Announcement Download* (0x52).
6. Bring the VRA SIMM in service with *Service State Configure* (0x0A).
7. Play the RANs as needed.

Edit RAN ID 0x0002:

1. Delete RAN ID 0x0002 using *Recorded Announcement Single Delete* (0x0117). Wait for the alarm, *Recorded Announcement Single Delete Complete* (0x0D).
2. Download the new RAN ID 0x0002 using *Recorded Announcement Download Initiate* (0x53) and *Recorded Announcement Download* (0x52).
3. When enough RANs have been deleted to make it worthwhile to recover the unused memory, take the VRA SIMM out of service with *Service State Configure* (0x0A). You can query the amount of dirty memory using *Recorded Announcement File System Query* (0x0102).
4. Recover the memory used by the deleted RANs with *Recorded Announcement File System Defragment* (0x0103).
5. Wait for the alarm, *Recorded Announcement Defragmentation Success* (0x0B).
6. Bring the VRA SIMM in service with *Service State Configure* (0x0A).

Example

Edit the RAN ID

Next, you must delete the RAN ID. Suppose RAN ID 0x0002 is the recorded announcement, “You have won one hundred dollars,” and the host wants to change the announcement to “You have won ten dollars.” RAN ID 0x0002 is currently playing to several channels from a number of DSP-ONE cards. The host wants to stop these plays in progress, so it sends the *Recorded Announcement Single Delete* message with the forced flag set, and with the slot and VRA SIMM field set to the wild card (0xFF). This action cancels any current plays of RAN ID 0x0002 and removes all instances of RAN ID 0x0002 stored on cards that support single message deletion.

> **Important!** In this example, you cannot delete instances of RAN ID 0x0002 if they are stored cards with file system Revision 0.

Each VRA SIMM that deletes RAN ID 0x0002 then generates the alarm *Single Message Deletion Complete*. As soon as the host receives this alarm, it may download the new RAN ID 0x0002 to each VRA SIMM. As soon as the first download completes, the new message is available to play.

Recover the Flash Memory

The final step in using the Single Message Deletion feature is to recover the flash memory used by the deleted RANs. The single message deletion created a new RAN ID 0x0002 and marked the old RAN ID 0x0002 as not to be used. But it did not overwrite the flash memory used by the old message with the new message because flash memory is a write-once mechanism.

You can recover the flash memory using the *Recorded Announcement File System Defragment* (0x0103) message. You can defragment whenever it is convenient or when more memory is needed for RAN downloads. The *Recorded Announcement File System Report* (0x0119) message provides defragmentation levels and memory use information. When a large proportion of the memory could be recovered by defragmentation, an alarm is sent.

The host must first take each VRA SIMM Out of Service, defragment the flash memory, then bring the VRA SIMM In Service, one SIMM at a time.

E1 Dial Pulse Address Signaling

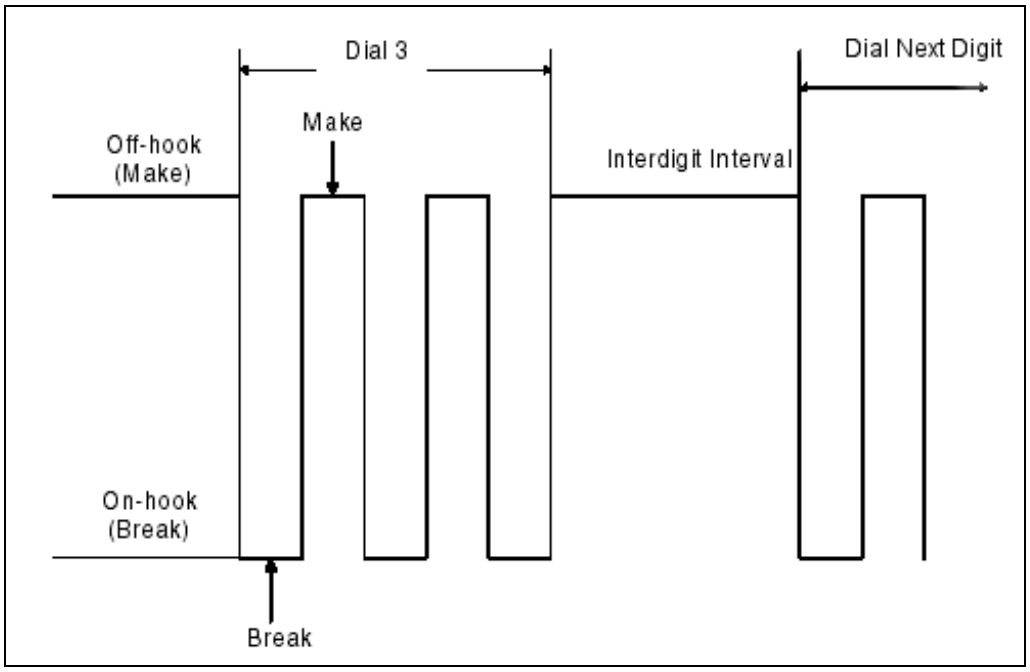
Overview

Dial Pulses are defined as momentary open loop signals, generated by sending a break (on-hook pulses) separated by a make (short off-hook pulses), representing the digits of the address.

For example, the Digit 3 is represented by three pulses; the Digit 0 is represented by ten pulses. Each string of pulses, representing a single digit, is separated by a longer off-hook pulse. The duration of this longer off-hook pulse is called the interdigit interval (Figure 9-10).

The CSP can generate and receive Dial Pulse address signaling during call setup only, not when a connection is already established.

Figure 9-10 Dial Pulse Signaling



Dial Pulse Generation

The E-ONE cards generates Dial Pulse, unlike the other address signaling, which is generated by the DSP card. The make and break durations are managed by a set of PPL Dial Pulse Transmission Timers, which you can configure using the *PPL Timer Configure 0x00CF* message. You can modify the make and break signaling bit values with the *PPL Transmit Signal Configure 0x00D2* message.

Dial Pulse Reception

For the DSP card to receive Dial Pulses, you must download the function type of “Dial Pulse Reception” to a SIMM, using the *DSP SIMM Configure* message. A set of PPL Dial Pulse Receive Timers manages the make and break durations. You can configure these timers with the *PPL Timer Configure 0x00CF* message. You can modify the make and break signaling bit values with the *PPL Transmit Signal Configure PPL Transmit Signal Config 0x00D2* message.

CONFERENCING

Contents

Overview

The conferencing feature lets the CSP make multiple full-duplex and half-duplex connections. You assign conferencing using the *DSP SIMM Configure* message (0x00C0). When one or more DSP chips are configured for the conferencing function, the CSP dynamically manages all conferencing DSP resources.

A single DSP chip can generate several independent conferences, in any combination, up to the maximum number allowed for the conference type and the maximum number of channels serviced by the DSP chip. For example, on the DSP-ONE card, a DSP chip configured for a standard conference can support two 7-party conferences, four 3-party conferences, or any combination of fourteen channels with a conference size of seven or smaller.

Use the *Conference Create* message (0x004B) to create a multi-channel conference. A DSP chip configured for conferencing may have more than one conference created on it, in any combination of the maximum allowed channels.

You designate the Conference Type in byte 9 of the *Conference Create* message (0x004B).

The available conference types are as follows:

0xN1 Standard, µ-law encoded

0xN2 Standard, A-law encoded

0xN3 Mixed (µ-law and A-law)

0xN4 Monitor

0xN5 Unified Dynamic

0xN6 Unified Dynamic with DTMF Clamping

0xN7 Unified Dynamic, µ-law Broadcast

0xN8 Unified Dynamic, µ-law Broadcast with DTMF Clamping

0xN9 Unified Dynamic, A-law Broadcast

0xNA Unified Dynamic, A-law Broadcast with DTMF Clamping

Standard and Mixed Conferences

Overview

The information in this section applies to both standard and mixed conferences, except for the following distinctions:

- A standard conference can support channels of either A-law or µ-law, but not both. But a mixed conference can support both A-law and µ-law encoded channels.
- A standard conference can be broadcast, but a mixed conference cannot be broadcast.
- A standard conference supports up to seven full-duplex parties, but a mixed conference supports nine.

The broadcast resource is a one-way output from a conference that allows a party to listen to the conference, but not to participate in it. Any number of parties can be connected to the broadcast resource. Connections to broadcast resources are made with the *Connect One-Way to Conference* (0x004F) message. If you do not need to broadcast a conference, use the mixed conference, because it supports more parties than a standard conference.

Creating a Standard or Mixed Conference

You create conferences with the *Conference Create* (0x004B) message, either during initial configuration (if the conference sizes can be predetermined) or dynamically, during real-time call processing. The number of full-duplex conference connections depends on the resources for each type. One-way conferencing lets you broadcast a standard conference to any number of channels.

When a conference is created, a Conference ID is returned by the CSP in the response to the *Conference Create* (0x004B) message. The CSP assigns Conference IDs in descending order as they are created. The Conference ID must then be used for subsequent references to the conference.

Connecting to a Standard or Mixed Conference

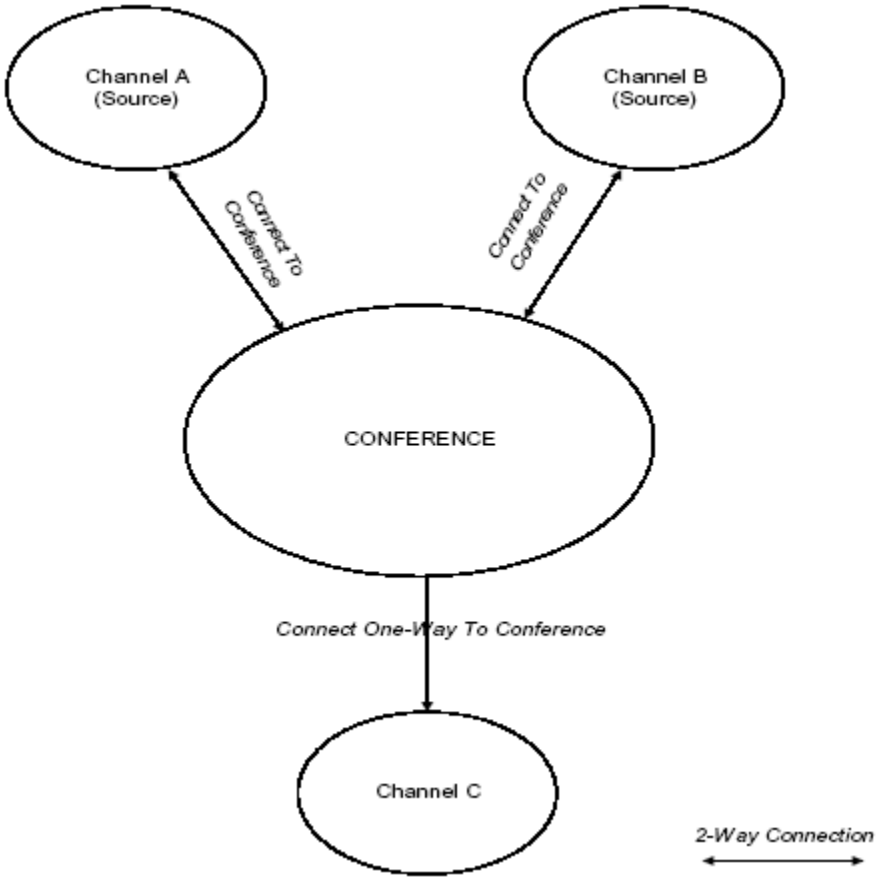
You can selectively add full-duplex connections to an existing conference, with the *Connect to Conference* (0x004E) message. You can also add half-duplex connections, with the *Connect One-Way to Conference* (0x004F) message. Connections are removed from a conference with the *Release Channel* (0x0008) message. When an on-hook is detected, the CSP automatically deletes that connection from the conference, and reports to the host with a Channel Released (0x0049) message.

Deleting a Standard or Mixed Conference

A conference remains in existence until a *Conference Delete Request* (0x004C) message is sent, which can be either forced or graceful.

- Forced - Set the Forced Flag byte to 1 to release all channels in the conference and delete the conference.
- Graceful - Set the Forced Flag byte to 0. The conference is deleted when all channels in the conference are released or removed.

Figure 9-11 Simple Conference



Monitor Conference

Overview

The Monitor Conference feature allows one or more monitor channels to listen to as many as nine conversations in a single monitor conference. The source channels do not need to be associated with each other in any way. This feature is useful for applications requiring supervisory monitoring, such as in Automatic Call Distributor (ACD) systems.

Unlike the other conference types, Monitor conferences are not maintained upon switchover of a a Matrix Controller card.

A monitor conference consists of source channels, which are grouped on a DSP conference resource. A monitor session consists of the monitor conference and the monitor channels that are connected to it. You can think of a monitor conference as a mixed conference with listen-only channels. However, the monitor conference is more flexible because:

- It lets channels connect to the monitor conference after call setup. It is not necessary to know at setup time all of the callers that will eventually connect to the conference.
- The call processing states of the source channels added to a monitor session are irrelevant. A source channel can be added to a monitor session with no effect on its current connection state. Subsequent state transitions of the channel have no effect on its association with the monitor conference.
- The source channels connected to a monitor conference do not have to be associated with one another in any way.

Creating a Monitor Conference

Each DSP chip that is configured for the Monitor Conference function type can support source channels of A-law and µ-law encoding, and output either A-law or µ-law according to the configuration of the monitor and source channel(s). At least one DSP chip must be configured this way to enable the Monitor Conference feature.

A monitor conference is created using the *Conference Create* (0x004B) message. The host application manages a monitor conference in virtually the same way that it manages a standard conference. To create multiple sessions, you can use the source ports on a DSP resource, in any combination.

For example, a DSP chip on the DSP-ONE card that is configured for the Monitor Conference function can provide the following:

- Three 9-port sessions and one 7-port session
- Six 5-port sessions and one 4-port session
- 17 two-port sessions

To create a Monitor Session, use the message sequence in Table 9-27.

Table 9-27 Message sequence for creating a Monitor Session

Step	Message	Description
1	DSP SIMM Configure	Configure a DSP chip for the Monitor Conference function (0x21).
2	Park Channel	Park the channel that will be the monitor channel.
3	Conference Create	Specify the conference type as “Monitor” (0x04). If accepted, the CSP returns the Conference ID in the response
4	Connect One-Way to Conference	Specify the Monitor Channel and the Conference ID.
5	Connect to Conference	Send one message for each source channel to be monitored. The channels connected to the monitor conference do not necessarily have to be connected to each other.

Reconfiguring a Monitor Conference

To reconfigure an existing conference with new source channels, park the monitor channels using the *Park Channel* (0x00BF) message. Removing the monitor channels from the conference (either by parking or releasing) automatically removes all of the source channels.

If a *Conference Delete Request* (0x004C) message that specifies graceful deletion has been sent, removing the monitor channels from the conference deletes the conference. When the source channels are removed, the monitor channels can then be reconnected to the conference (*Connect One-Way to Conference*), and new source channels connected (*Connect to Conference*) using the same Conference ID number.

Deleting a Monitor Conference

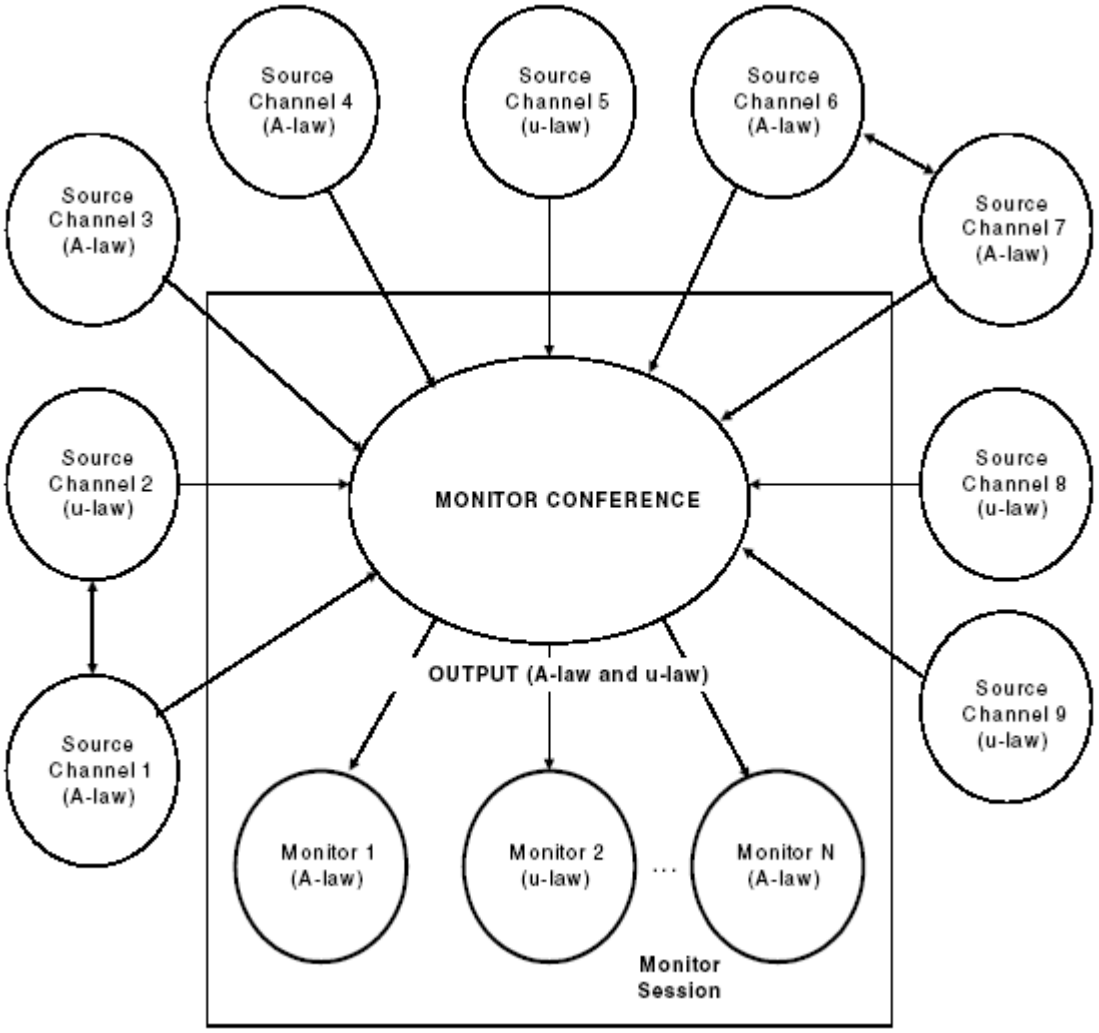
You can delete a Monitor Conference by sending a *Conference Delete Request* (0x004C) *Conference Delete Request* (0x004C) message. Deletion can be either Forced or Graceful:

- Forced: Set the Forced Flag byte to 1. This releases all channels in the conference and deletes the conference.
- Graceful - Set the Forced Flag byte to 0. To delete a Monitor Conference, all monitor channels in the conference must be released or removed, and a *Conference Delete Request* (0x004C) message must be sent. To releases all monitor channels, send the *Release Channel* (0x0008) message. All source channels are then automatically removed from the conference, a *Conference Delete Request* (0x004C) is sent, and the conference is gracefully deleted.

  If you remove only the source channels from the Monitor Conference, the conference is not deleted. If a *Conference Delete Request* (0x004C) message has not been sent for the Monitor Conference, then removing the monitor channels from the conference releases all parties from the Monitor Conference, but does not delete the conference itself.

  The conference retains its Conference ID, and the conference can be re-used. Monitor channels can then be reconnected to the conference (*Connect One-Way To Conference*) and new source channels can be connected (*Connect To Conference*) using the same Conference ID. Figure 9-12 shows a complex monitor session.

Figure 9-12 Complex Monitor Session

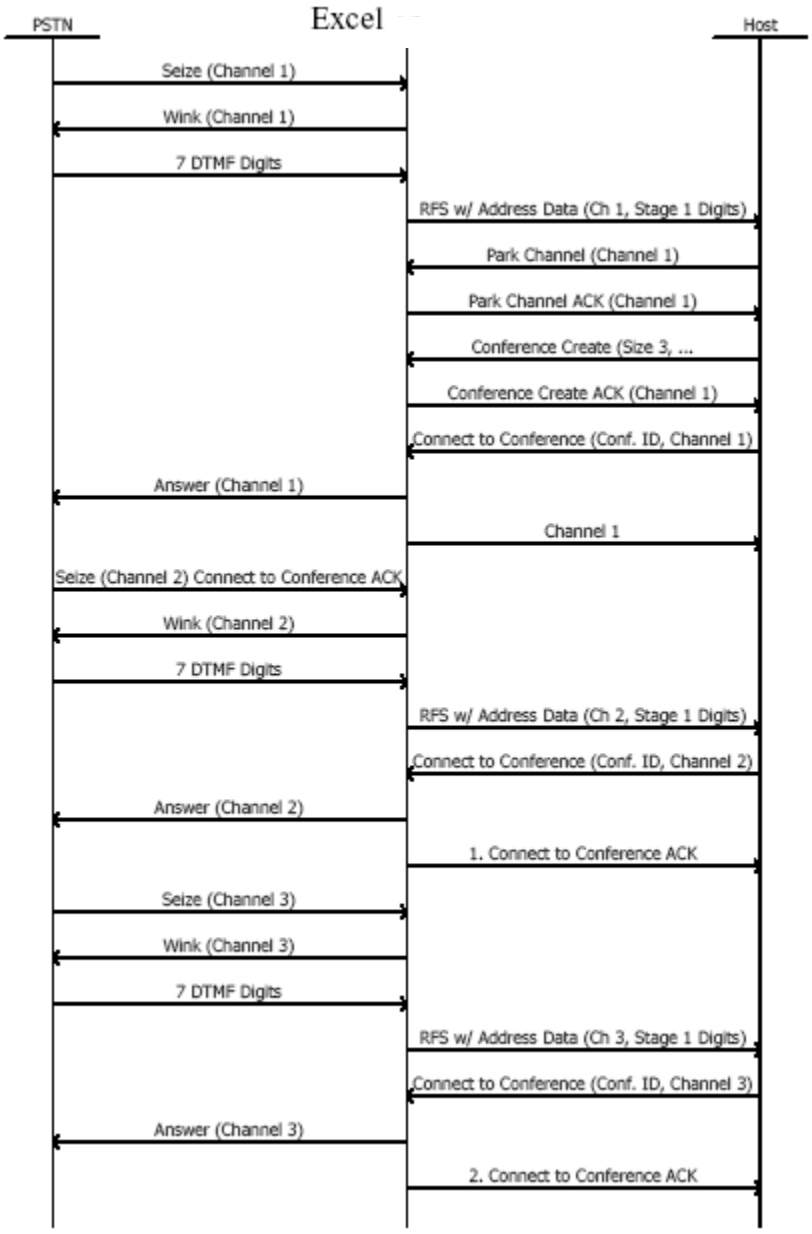

Source Channel 4 (A-law)
Source Channel 5 (u-law)
Source Channel 6 (A-law)
Source Channel 3 (A-law)
Source Channel 7 (A-law)
Source Channel 2 (u-law)
MONITOR CONFERENCE
Source Channel 8 (u-law)
OUTPUT (A-law and u-law)
Source Channel 9 (u-law)
Source Channel 1 (A-law)
Monitor 1 (A-law)
Monitor 2 (u-law)
...
Monitor N (A-law)
Monitor Session

Call Flows

This section contains call flows pertaining to the DSP card.

Conferencing Incoming Calls

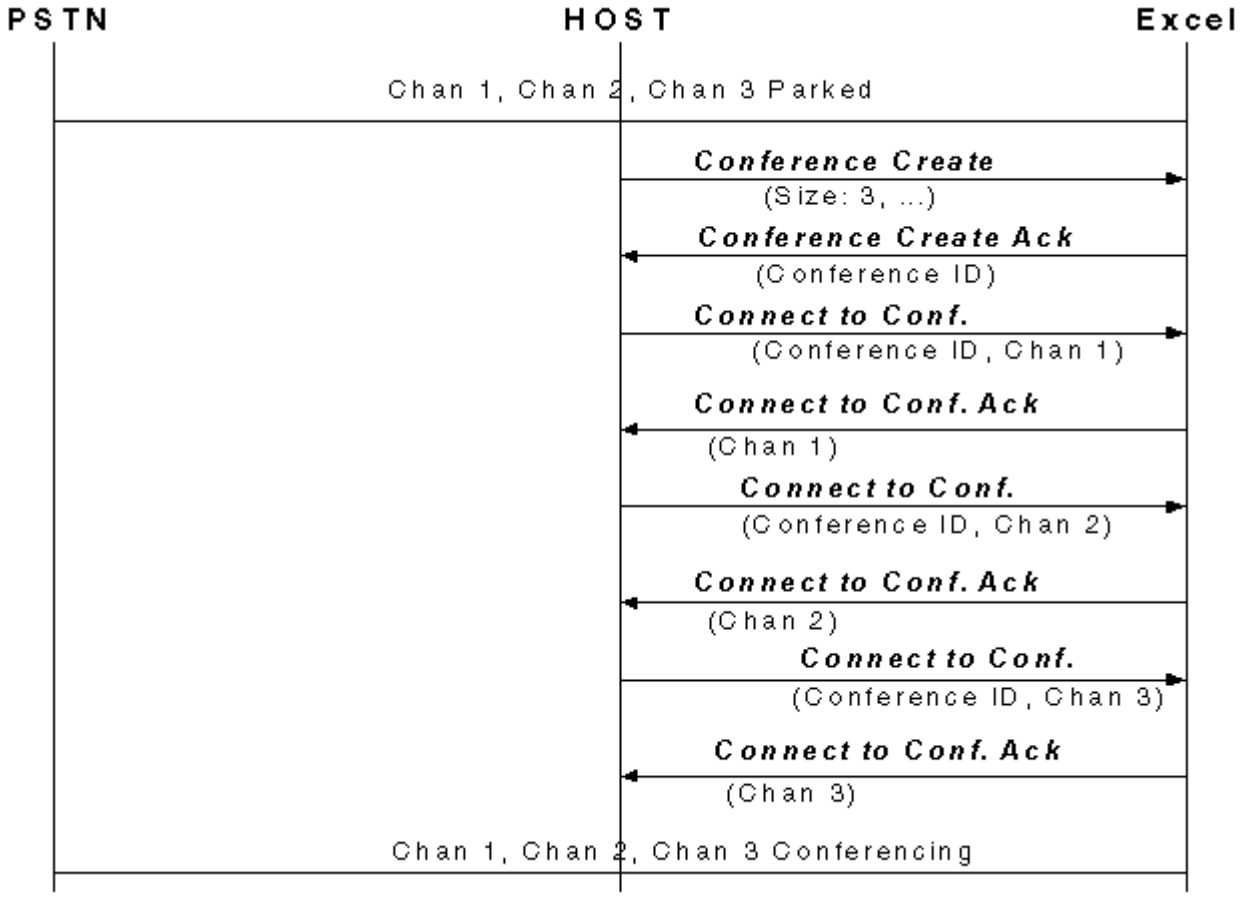
The example below shows conferencing of incoming calls.



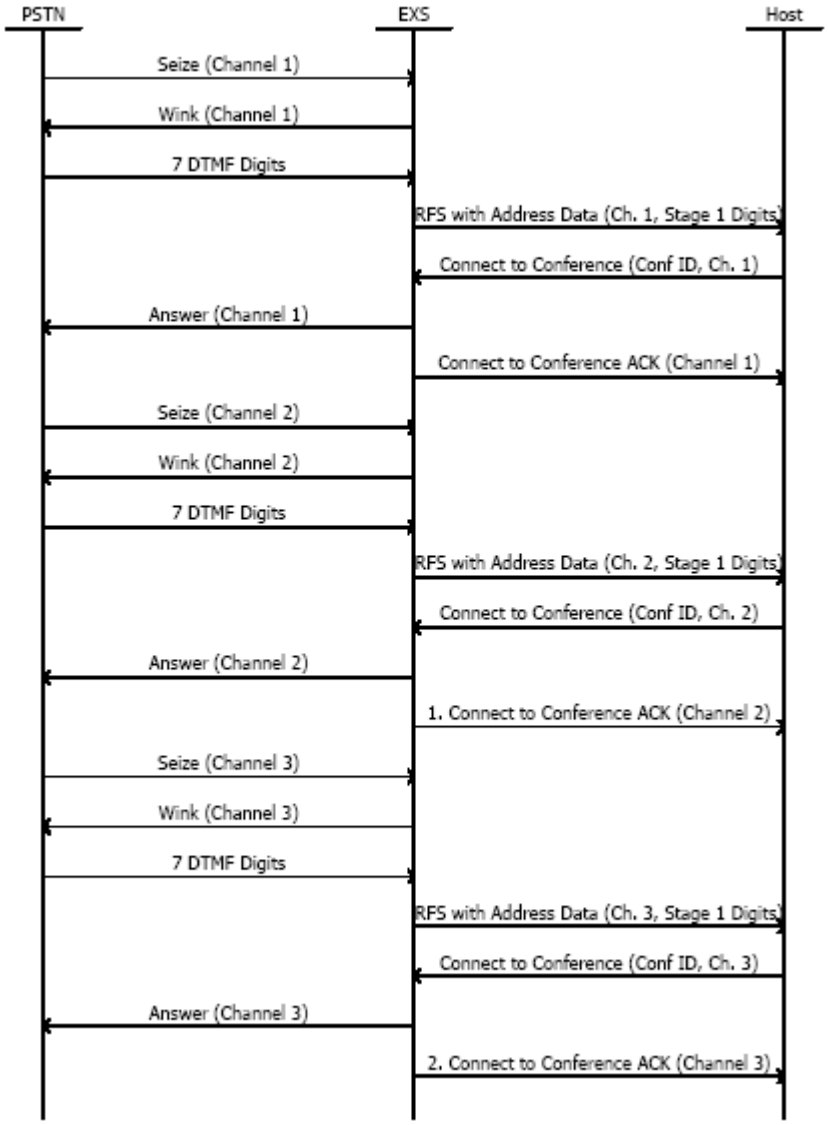
1. After this Acknowledgment, Channels 1 and 2 are conferencing.
2. After this Acknowledgment, Channels 1, 2, and 3 are conferencing.

Conferencing Incoming Calls - Host Created

This example shows conferencing of incoming calls when the host has already created a 3-way conference.



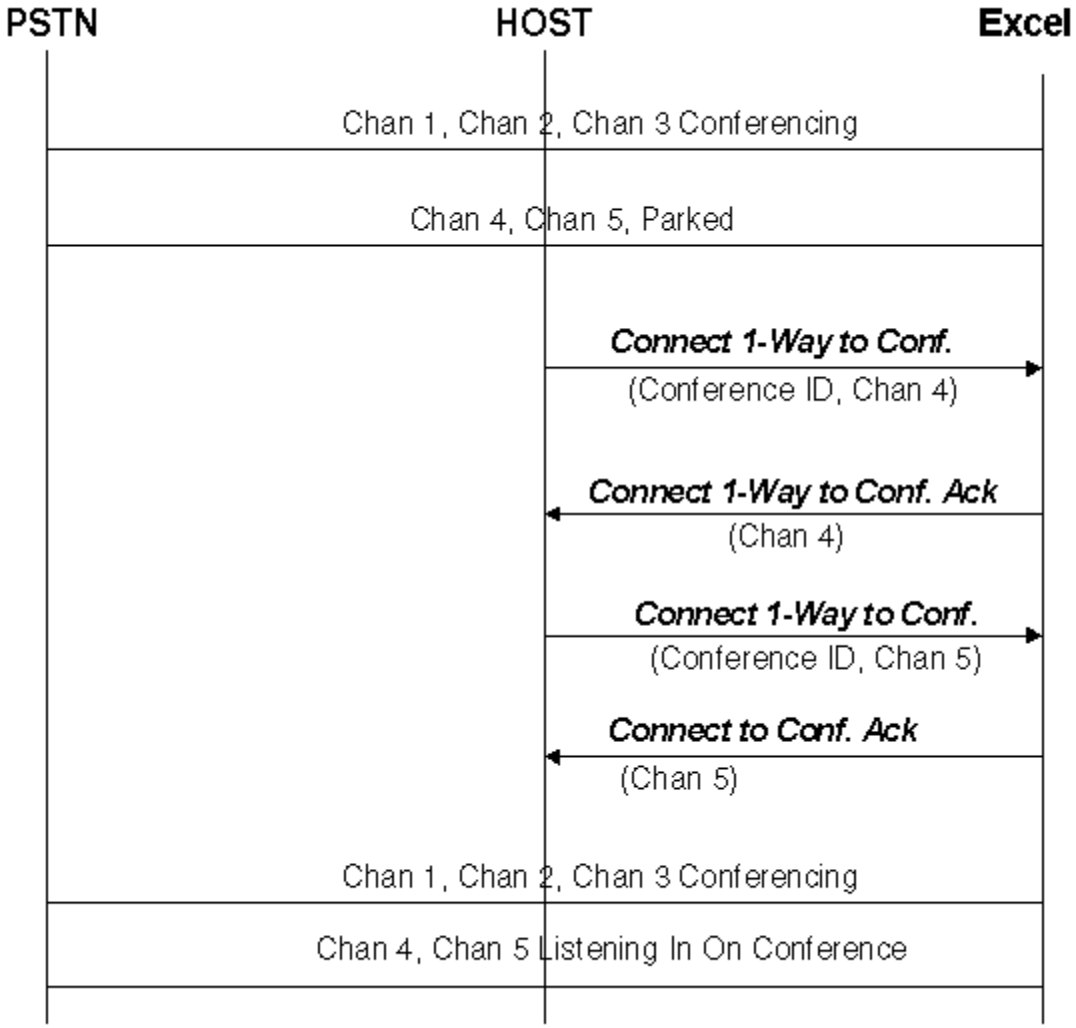
Conferencing Parked Channels



1. After this Acknowledgment, Channels 1 and 2 are conferencing
2. After this Acknowledgment, Channels 1, 2, and 3 are conferencing.

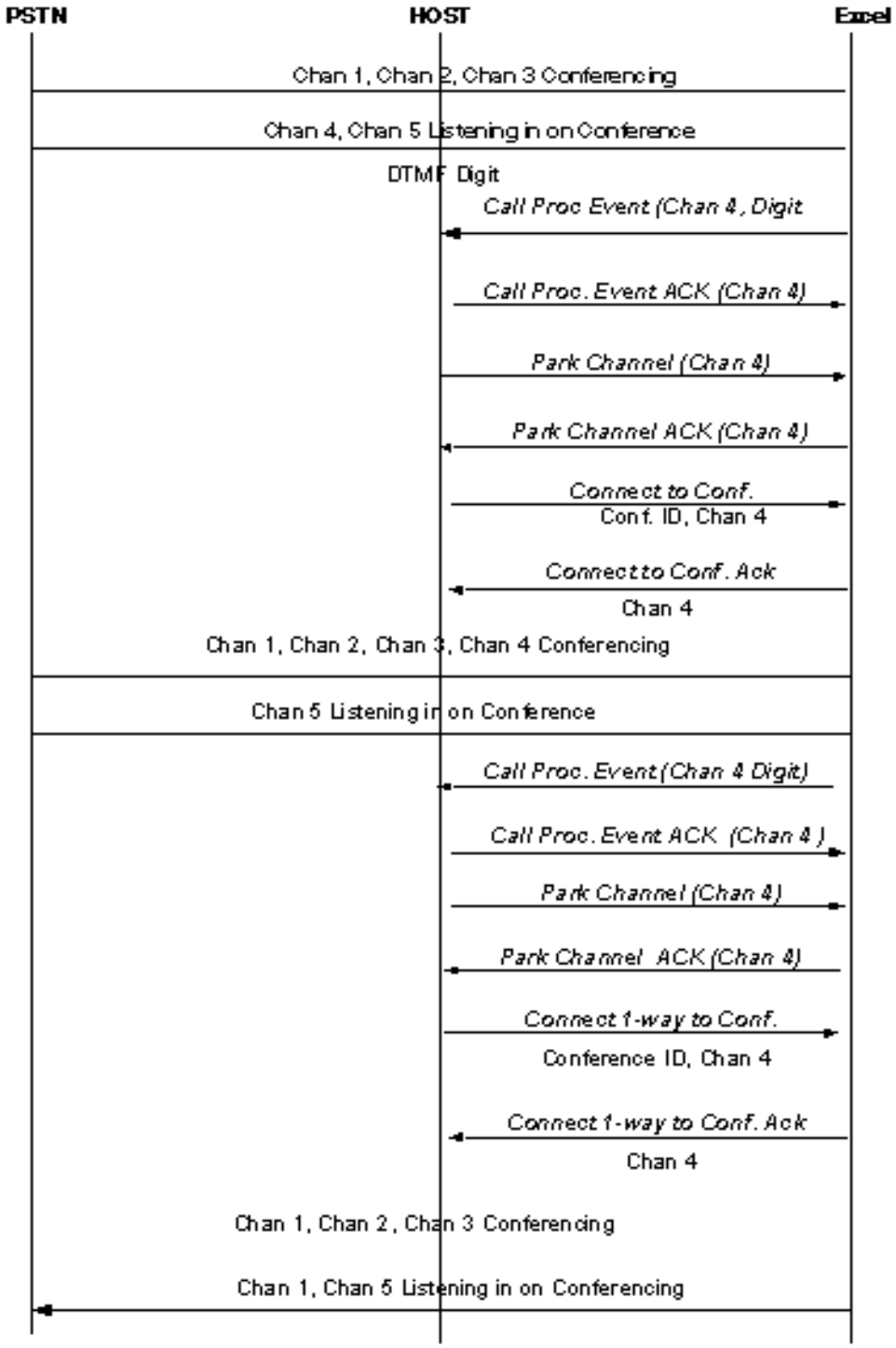
Broadcasting an Established Conference

This example shows broadcasting an established conference to any number of channels



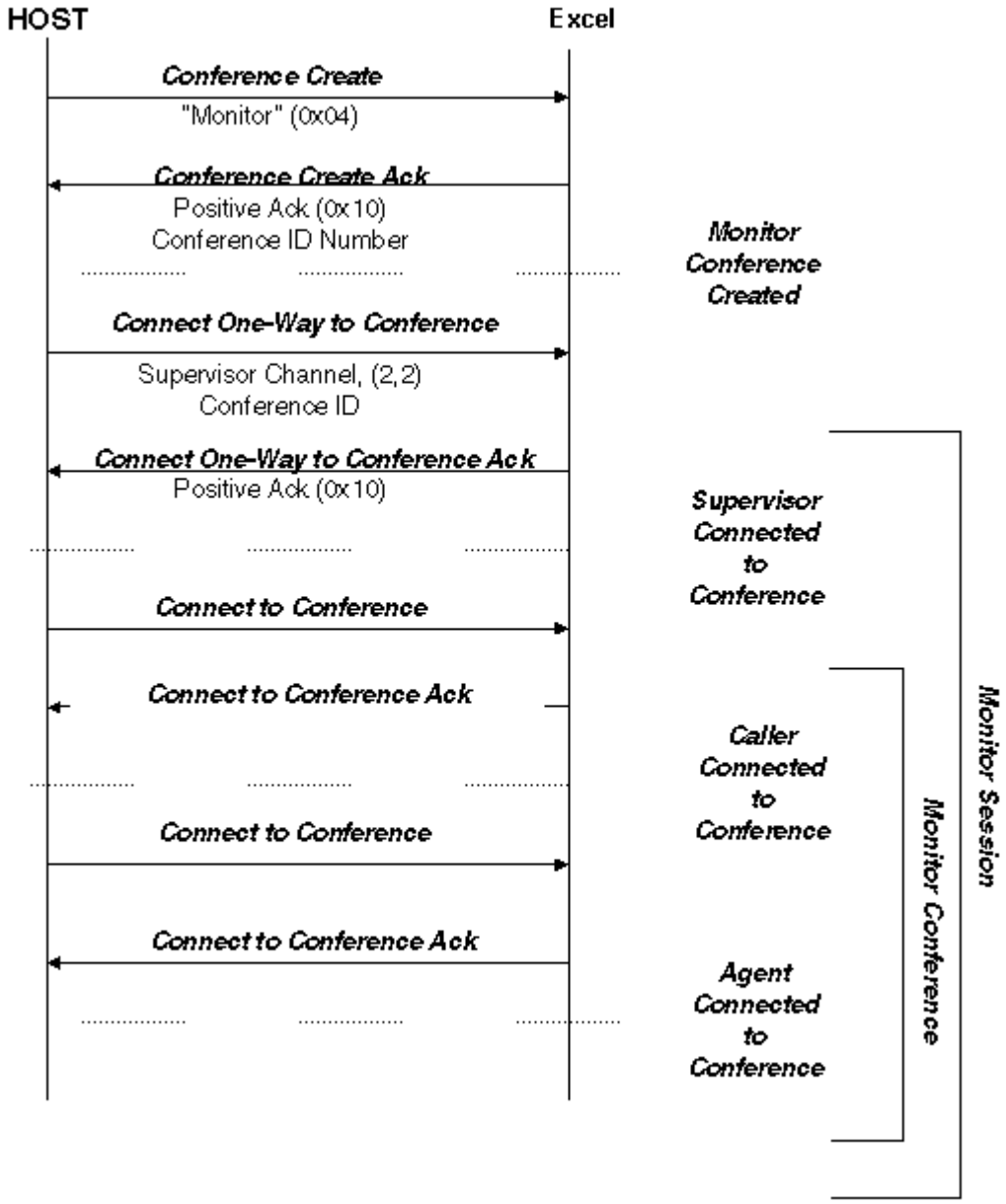
Switching From Conference To Broadcast (Listen only)

This example shows the API messages transferring a channel from broadcast (listen only) to conference and back again. Dialing “Add Me To Conference” and “Remove Me From Conference” DTMF digit is also shown.



Establishing a Monitor Conference

The example below shows a connection between a caller on Channel A (0, 0) an agent on Channel B (1, 1) and a supervisor on Channel C (2, 2) monitoring the conversation:


HOST
Excel
Conference Create
"Monitor" (0x04)
Conference Create Ack
Positive Ack (0x10)
Conference ID Number
Monitor Conference Created
Connect One-Way to Conference
Supervisor Channel, (2,2)
Conference ID
Connect One-Way to Conference Ack
Positive Ack (0x10)
Supervisor Connected to Conference
Connect to Conference
Connect to Conference Ack
Caller Connected to Conference
Connect to Conference
Connect to Conference Ack
Agent Connected to Conference
Monitor Conference
Monitor Session

Unified Conferencing

Overview

Unified Conferencing provides the following functionality:

- Mixed conferencing (µ-Law and A-Law, DSP-ONE card only)
- Unlimited Broadcast or “listen only” capability (µ-Law and A-Law)
- Ability to increase conference size dynamically
- Unified Conference algorithm

You can create, connect to, and delete a Unified conference by using the same techniques that are used for a Standard conference. But the Unified conference presents significant advantages over the Standard conference, including more channels, µ-Law and A-Law channels, and dynamic growth. Dynamic growth means that a Unified conference can automatically grow beyond its creation size when additional parties are connected to the conference.

Example

On the DSP-ONE card, an individual Unified conference can grow dynamically until the maximum Unified conference size of 25 is reached, or until all 31 channels on the DSP chip are used, whichever comes first. For example, a particular DSP chip can support one conference with 6 channels, a second conference with 10 channels, and a third conference with 15 channels, for a total of 31 channels.

All of the channels on this DSP chip are now in use, so none of the conferences on that chip can grow until one of these conferences frees a channel. If the 6-channel conference and the 10-channel conference were deleted, then 16 channels on that DSP chip would be freed. The remaining 15-party conference could then grow to the system-imposed maximum conference size of 25 parties, using 10 of the 16 freed channels. The six remaining channels on that DSP chip would then be available for new conferences.

Choosing Among Unified Conference Types

For the conference type Unified Dynamic Conference (0xN5), two local timeslots per conference are allocated to broadcast over the ring in **both** µ-law and A-law. In contrast, using one of the conference types 0xN7, 0xN8, 0xN9, or 0xNA, you can create a unified dynamic conference over the CSP that broadcasts over the ring in either A-law or µ-law, but **not both**. This way, only one local timeslot per conference is allocated to broadcast.

If you use µ-law exclusively, you should use one of these conference types:

- 0xN7 - Unified Dynamic, µ-law Broadcast
- 0xN8 - Unified Dynamic, µ-law Broadcast with DTMF Clamping

If you use A-law exclusively, you should use one of these conference types:

- 0xN9 - Unified Dynamic, A-law Broadcast
- 0xNA - Unified Dynamic, A-law Broadcast with DTMF Clamping

These two conference types behave the same way as the Unified Conference:

- Unified Dynamic, µ-law Broadcast
- Unified Dynamic, A-law Broadcast

These two conference types behave the same way as Unified Dynamic with DTMF Clamping:

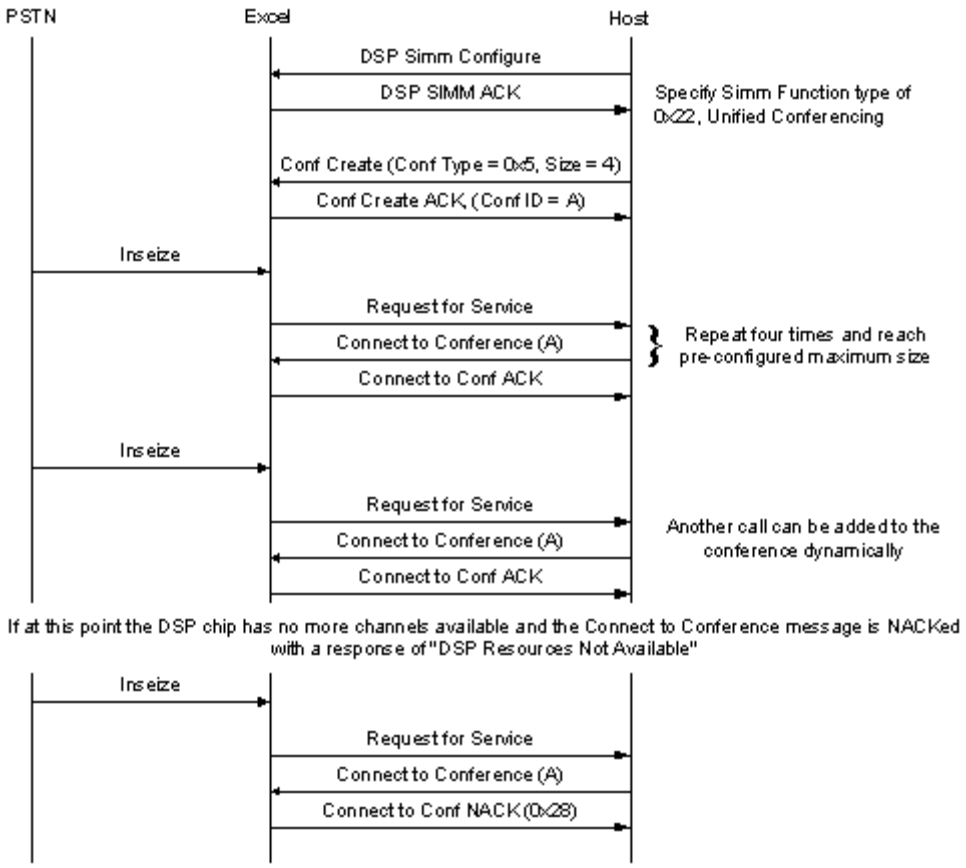
- Unified Dynamic, µ-law Broadcast with DTMF Clamping
- Unified Dynamic, A-law Broadcast with DTMF Clamping

You receive a NACK of 0x2D (Incompatible PCM Encoding for Conference) if a channel from a remote node tries to connect 1-Way to a conference and the encoding type does not match the broadcast type set during the conference creation.

When the CSP creates a Unified conference, it places the conference on whichever DSP chip has the largest number of free channels at that moment. For example, if one DSP chip has 10 free channels and another has 20 free channels, the CSP places the new conference on the DSP chip with 20 free channels. This logic maximizes the opportunity for that conference to grow.

The call flow below illustrates how the host application can configure the CSP to use Unified conferencing. It also illustrates how to dynamically increase the size of a conference beyond the number originally specified in the *Conference Create* (0x004B) message.

Table 9-28 Configuring Unified Conferencing



For the Unified Dynamic Conference (0xN5) type, two local timeslots per conference are allocated to broadcast over the ring in both µ-law and A-law. In contrast, conference types 0xN7, 0xN8, 0xN9, and 0xNA, use only one local timeslot per conference to broadcast over the ring, in either µ-law or A-law, but not both. Users who use only µ-law do not need to allocate midplane ports for A-law, and vice versa.

Unified Dynamic Conferencing with DTMF Clamping

Unified Dynamic Conferencing with DTMF Clamping is a unified conference that prevents DTMF echo. DTMF echo causes false DTMF indications on conference legs. This feature enables users to provide DTMF digit receivers on each conference leg without one conference participant inadvertently controlling another participant's conference parameters.

This conferencing type is like Unified Dynamic Conferencing except that on the DSP-ONE card, a DSP chip supports 24 channels of DTMF-clamped conferencing.

The following conference types behave the same way as Unified Dynamic with DTMF Clamping:

- Unified Dynamic, μ-law Broadcast with DTMF Clamping
- Unified Dynamic, A-law Broadcast with DTMF Clamping

Please refer to the following API messages to implement this feature:

- *DSP SIMM Configure* (0x00C0):
  Additional DSP Function Type
- *Conference Create* (0x004B):
  Additional DTMF-Clamped Conference
- *Card Status Report* (0x00A6):
  Additional Function Type

10 Configuring Multi-Node Systems

Purpose The CSP uses the proprietary EXNET® to connect multiple CSP chassis into a single virtual CSP. EXNET® is an open architecture, 1.3 Gbps fiber optic network that uses a packet-based protocol. When you use EXNET® to link up to seven CSP chassis into a single virtual programmable platform, the CSPs are then called nodes within the system.

EXNET-ONE Card

Overview

The EXNET-ONE card is used to connect up to seven EXNET® Nodes (CSP 2040 and CSP 2090) together over a high-speed optical fiber ring to form a single virtual CSP. This card is capable of switching any port on the local node to any other port on an attached node, and vice-versa.

The EXNET-ONE is a full-size card that can be installed into any CSP 2040 and CSP 2090 I/O slot that does not have a line card in the corresponding front slot.

CAUTION

A line card must not be inserted in the slot in front of the EXNET-ONE card, or vice-versa. The EXNET-ONE card uses the front line card slot for bus resources, identifying itself (Card ID 0x54) as residing in the front line card slot rather than the I/O. If this occurs, the Matrix Controller sends an alarm to the host, indicating that resources are not available to the newly inserted line card. The alarmed card remains out of service until it can be removed from the system.

The major differences between the original CPU kernel and the EXNET-ONE are as follows:

- RAM is no longer battery-backed, but will not be corrupted by a push button or watchdog timer reset.
- Processor upgraded to PowerPC resulting in a new load image.

Card Set Differences

The following information defines specific differences between the original EXNET Controller card set and the EXNET-ONE.

- As mentioned above, a line card cannot be inserted into the front line card slot associated with EXNET-ONE and vice-versa. Typically, a blockout card should be installed in front of the EXNET-ONE card. If any card is installed in front of the EXNET-ONE card, the system sends an *Alarm* (0xB9) message to the host. The conflicting card remains out of service until it can be removed from the chassis. If the chassis is powered up with both an EXNET-ONE and a corresponding line card inserted, the line card is always given precedence.
- EXNET-ONE is also an I/O card.

- New Card ID (0x54)
- Image Load ID (0xE)
- New TFTP load type and binary file have been created for EXNET-ONE:
  - EXNET-ONE TFTP Binary Image:exntone.bin
  - EXNET-ONE TFTP Load Type:EXNETONE_LOAD
- EXNET-ONE card identifies itself as residing in a front line card slot to the system software and host. A line card is not allowed in the corresponding front line card slot, since the EXNET-ONE card identifies itself as residing in the front line card slot, not the I/O slot. This minimizes the amount of host changes needed to support the EXNET-ONE card.

The following is a sample line from a TFTP Configuration File:

```
  EXNTONE_LOAD=c:\tftpboot\exntone.bin=SAVE_LOAD_FALSE
‘EXNET-ONE support
```

Refer to *Creating a TFTP Configuration File* for more information.

Installation Procedures

Follow the steps below to install the EXNET-ONE card.

1. Make necessary changes to application.
2. Allocate slot with open line card as well as I/O.
3. Put Blockout panel in line card slot.
4. Place EXNET-ONE into corresponding I/O slot.
5. Check EXNET-ONE front panel LED sequence for proper operation.

EXNET-ONE Card Software Functions

The EXNET-ONE card is responsible for the following functionality:

- System Monitoring
- Diagnostics
- Ring Control
- Ring Configuration
- Ring Switching
- Inter-Board Communications

The EXNET-ONE card requires system software Release 5.5 or later version. The system software treats the EXNET-ONE card as a line interface card. When the system software detects an EXNET-ONE card (when the system powers up or when the EXNET-ONE is hot-inserted), it notifies the host using the *Card Status Report* (0xA6) message. At start-up, the system software recognizes the EXNET-ONE card's unique ID and begins a card start sequence downloading the applicable EXNET-ONE system software. At this point, the ROM resident code is running on the EXNET-ONE card as it begins execution.

EXNET® Features

The EXNET® features the following:

- Single Socket Host Control - allows all host/node messaging through a single node. Messages to remote nodes (slaves) are routed using the Logical Node ID.
- EXNET® Redundancy - installing a second EXNET® ring provides ring redundancy. If one ring goes out-of-service, the second ring carries all traffic.
- Fault Recovery - provides recovery capability if an EXNET® ring or node fails. The EXNET-ONE cards loops back their EXNET® ports where necessary, and the complete ring is maintained. If a node's connection to the ring is broken, the ring connection among the remaining nodes is maintained.
- CSP Conferencing - allows channels on any node in multi-node CSP to connect to a conference, using a single DSP resource. The DSP resource can be located in any node on the EXNET® ring.

Configuration

Purpose This section describes how to bring an CSP into service. It assumes that all hardware configuration and cabling is complete, that all CSP nodes are powered-up, and that a communication link exists between the host and nodes.

Before you begin to configure an CSP, map out your configuration plan clearly, and record information on each node. See the *Hardware Installation and Maintenance Guide* for more information.

Important! Perform all configuration first, before bringing the system into service.

CAUTION

You must use the same API format (Basic or Extended) on all nodes on a single EXNET® ring.

Step	Action	API Message
1	Download System Software	See *EXS API Application Development*.
2	Re-establish Socket Connections	
3	Assign Logical Node IDs	*Assign Logical Node ID*
4	Configure Nodes	*EXS Node Configure*
5	Configure Ring	*EXNET Ring Configure*
6	Bring Ring In-Service	*Service State Configure*

The following steps detail the procedure for configuring your CSP as shown in the table above.

1. Download System Software

When an IP socket is opened to a node, the host begins to receive *Poll* messages indicating that the node requires a system software load. The *Node ID* field is 0xFF, indicating that a node ID has not yet been assigned. Download the system software to each node individually, following the procedure detailed in *Application Development* chapter.

Downloading the system software purges all local and distributed traffic. After the system software is downloaded, you must reconfigure all nodes and all EXNET® rings. The host must verify that each node has received the new software load.

2. Re-establish Socket Connections

After each Matrix Controller resets, you must re-establish each Matrix Controller card's socket connection. Each node then sends *Poll* and *Node Status Report* messages, indicating that the node requires a Logical Node ID, which you must assign. The *Node Status Report* message is sent only if a node is an CSP node. The software determines that a node is an CSP node by the presence of an EXNET-ONE card. So you must install the EXNET-ONE card before you power up the CSP. The CSP continues sending *Node Status Report* messages, about every five seconds, until a Logical Node ID is assigned.

3. Assign a Logical Node ID

Assign a Logical Node ID (0-63) to each node, using the *Assign Logical Node ID* message. All nodes on the same Ethernet segment must have a unique Logical Node ID and there must be one Logical Node ID for each Physical Node ID. If a node ID is already assigned, the *Assign Logical Node ID* message receives a negative acknowledgment (NACK).

> **Important!** The Logical Node ID must be in the range of (0x00–0x3F).

The first node to be assigned a Logical Node ID becomes the master node of the EXNET® ring, and it obtains transmit bandwidth on the ring. As subsequent nodes are assigned Logical Node IDs, they also obtain bandwidth on the ring.

If a message to a node is addressed with an incorrect Logical Node ID, the CSP responds with a status of "Invalid Logical Node ID." The *Logical Node ID* field in the header of the response indicates the correct Logical Node ID.

4. Configuring a Node

You can now configure each node the same way that you configure a stand-alone CSP, with the following considerations that are specific to the multi-node CSP:

Logical Span IDs

Logical Span IDs must be unique across all nodes in an CSP. If a duplicate ID is assigned, a response status of Logical Span ID Already Assigned is returned to the host. Each node keeps a record of every other node's Logical Span ID.

Service Resources

ISDN PRI, DASS2, and SS7 PQ card resources are available and addressed to ports that reside in the same node as those cards. However, you can use a DSP resource on a single node for conference connections among multiple nodes. Other DSP resources can be used locally only. Please refer to *DSP Resource Specifications* for more information.

5. Configuring the Ring

Use the *EXNET Ring Configure* message to perform the following tasks on each node:

1. Assign a Logical Ring ID.
2. Configure four packets to be transmitted.
3. Configure Ring Timing (See *Expanding the EXNET® Ring (10-35)* for more information).
4. Configure the Transmit/Receive mode of each ring on a node to Transmit and Receive. If you have a multi-ring configuration, please contact Dialogic Technical Support for assistance.
5. Configure CSP Conferencing, if required.

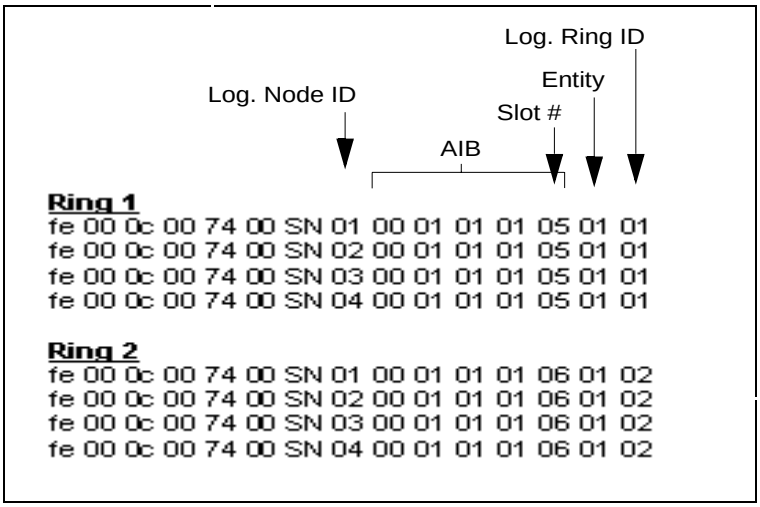
The following examples show a T1 configuration with four nodes and a redundant EXNET® ring. The EXNET-ONE for Ring 1 is in Slot 5 and the EXNET-ONE for Ring 2 is in Slot 6.

Assign Logical Ring ID

You must send an *EXNET Ring Configure* message to the EXNET-ONE card attached to each ring. You must assign the same Logical Ring ID to all EXNET-ONE cards on that ring.

The sequence of messages shown in Figure 10-1 is sent to assign Logical Ring ID 1 to the four nodes on the ring.

Figure 10-1 Assign Logical Ring ID



The following example shows details of the first message from the sequence below.

BYTE	Field Description	Value and Indication
0	Frame	0xFE
1, 2	Length (MSB, LSB)	0x000c
3, 4	Message Type	0x0074 (EXNET Ring Configure)
5	Reserved	0x00
6	Sequence Number	0xSN
7	Logical Node ID	0x01 (Logical Node 1)
8	AIB Address Method	0x00 (Single Entity)
9	Number of Address Elements	0x01
10	AIB Type	0x01 (Slot)
11	Address Data Length	0x01
12	Address Data[0] Slot Number	0x01 (Slot 5)
13	Entity	0x01 (Assign Logical Ring ID)

BYTE	Field Description	Value and Indication
14	Data[0] Logical Ring ID	0x01 (Logical Ring 1)
15	Checksum	

Configuration

The sequence shown in Figure 10-2 configures a node to transmit four packets.

Figure 10-2 Configure Number of Packets

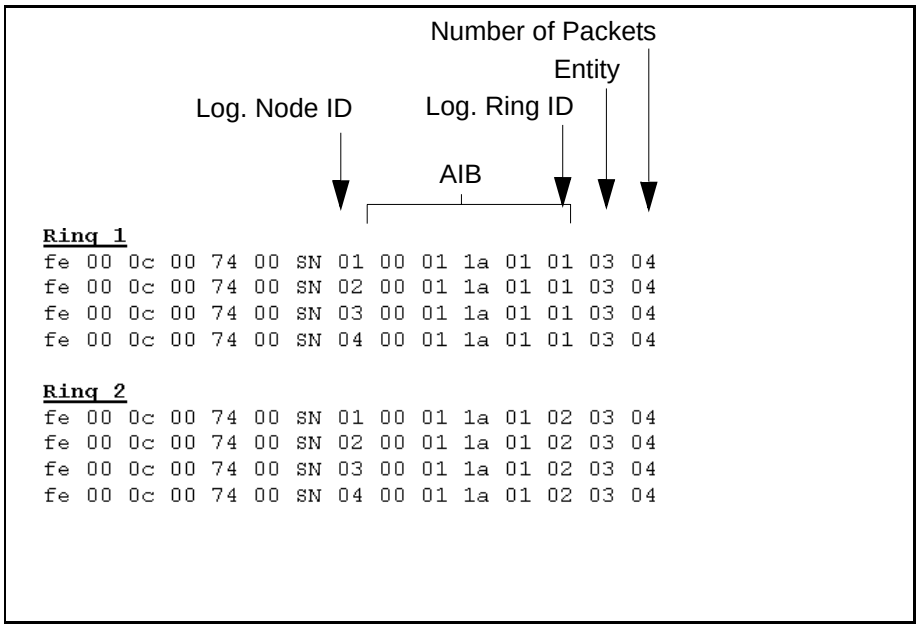


Table 10-1 Message Example

BYTE	Field Description	Value and Indication
0	Frame	0xfe
1, 2	Length (MSB, LSB)	0x000c
3, 4	Message Type	0x0074 (EXNET Ring Configure)
5	Reserved	0x00
6	Sequence Number	0xSN
7	Logical Node ID	0x01 (Logical Node 1)
8	AIB Address Method	 0x00 (Single Entity)
9	Number of Address Elements	0x01
10	AIB Type	0x1A (Logical Ring)
11	Address Data Length	0x01
12	Address Data[0] Logical Ring ID	0x01 (Logical Ring ID 1)

BYTE	Field Description	Value and Indication
13	Entity	0x02 (Configure Number of Packets)
14	Data[0] Number of Packets	0x04 (4 Packets)
15	Checksum	

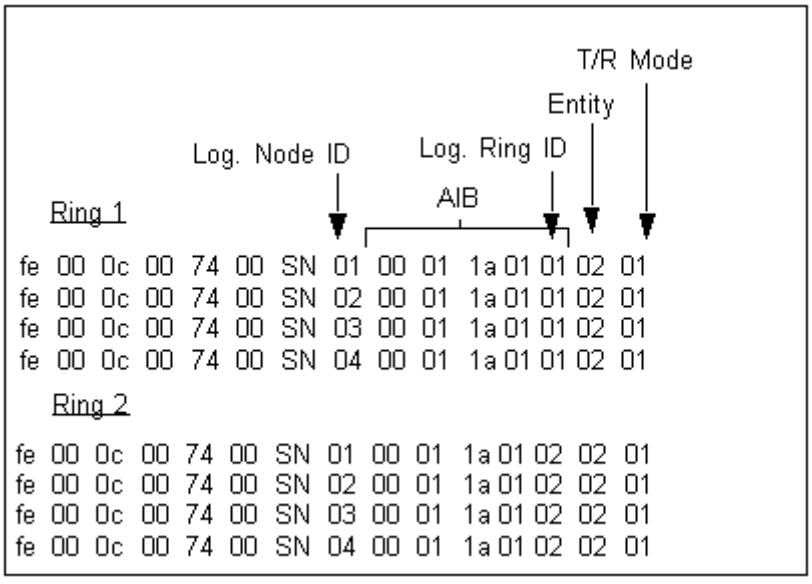
Configure Transmit/ Receive Mode

Configure the Transmit/Receive mode of each ring on a node to Transmit and Receive. If you have a multi-ring configuration, please contact Dialogic Technical Support for assistance.

Important! Before you change the Transmit/Receive mode, you must set the number of packets transmitted.

The sequence shown below configures all nodes in the system for Transmit/Receive mode.

Figure 10-3 EXNET® Ring Configure - Configure Transmit and Receive Mode



The following example of the *EXNET Ring Configure* message shows the first message from the sequence:

BYTE	Field Description	Value and Indication
0	Frame	0xFE
1, 2	Length (MSB, LSB)	0x000c
3, 4	Message Type	0x0074 (EXNET® Ring Configure)
5	Reserved	0x00
6	Sequence Number	0xSN
7	Logical Node ID	0x01 (Logical Node 1)
8	AIB Address Method	0x00 (Single Entity)
9	Number of Address Elements	0x01
10	AIB Type	0x1a (Logical Ring)
11	Address Data Length	0x01
12	Address Data[0] Logical Ring ID	0x01 (Logical Ring ID 1)
13	Entity	0x02 (Con Transmit Mode)
14	Data[0] Transmit Mode	0x01 (Transmit/Receive)
15	Checksum	

6. Bring the Ring In Service

After you finish all CSP configuration, bring the ring in service, using the *Service State Configure* message. You must send this message to only one node of all the nodes on a given EXNET® ring.

Each node informs all of the other nodes of its Logical Node ID. Each node stores this information for API messaging and PCM data routing. The messaging among nodes during ring setup is performed over the same Ethernet link as the host-to-nodes communication link.

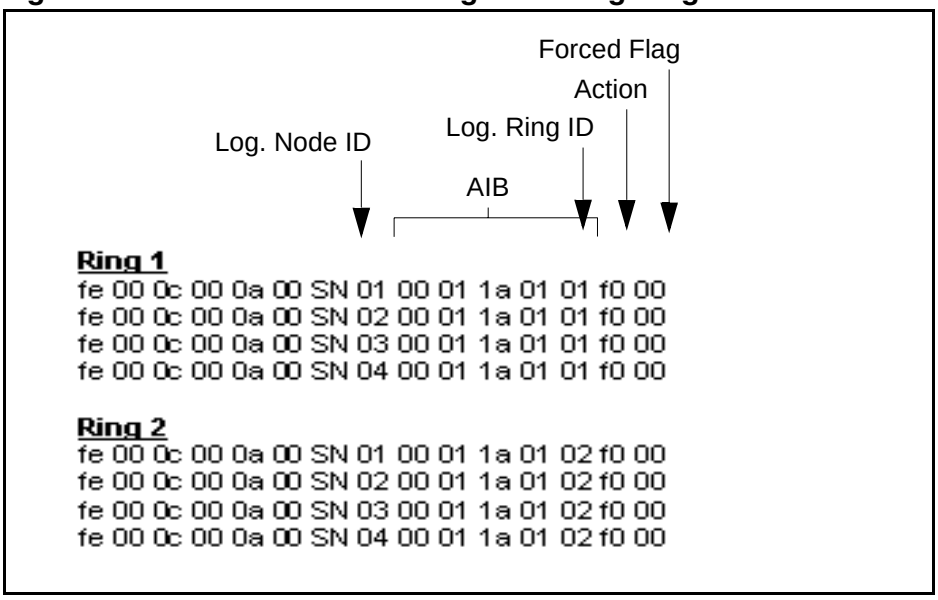
When all nodes are assigned and established on the ring, the ring begins a sequence to come in service. The host receives *Ring Status Report* messages as the ring transitions through states of Connected, Initializing, and In Service.

If the ring fails, it is taken out of service automatically. Until the ring is brought back in service, no connections can be made among nodes.

Example: The sequence shown in Figure 10-4 brings Ring 1 and Ring 2 in service.

You should receive a sequence of *Ring Status Report* messages from each node, indicating that the ring is initializing, connected, and in service and that the EXNET® ports are in Normal node. The Signal Detect and Fault Detect LEDs on the EXNET-ONE cards should be green.

Figure 10-4 Service State Configure - Bring Rings In Service



Distributed Layer 4 Call Processing

Host Managed

The host manages call processing on an CSP with a single service access point. If a connection is made between channels on different nodes, internal call control messaging is automatically sent between the nodes. This strategy allows for answer/release to propagate internally. It also allows for outseizure to initiate on a remote node (with the *Connect* message) so the host can manage calls as though the multi-node CSP were a single-chassis CSP.

When a connection is being made between two nodes (A, B) the host needs to send only a *Connect* message (or other connection management message) to Node A. Node A then establishes a connection with Node B using the Ethernet link. For routing messages between nodes like this, each node keeps a record of all Logical Span IDs in the CSP.

> **Important!** When cross connecting spans or channels among nodes in an CSP, a separate message must be sent to each span or channel in the connection.
>
> **Example:** Message 1: to Node 1, *Cross Connect Channel* A, B
> Message 2: to Node 2, *Cross Connect Channel* B, A

When cross connecting spans or channels in the same node, only one message is required. But you can still use the two-message internodal method. Having one method (internodal) instead of two simplifies your application.

Example Message

This example shows the format of a *Connect* message that connects Span 0/Channel 0 on Node 1 to Span 65/Channel 0 on Node 4:

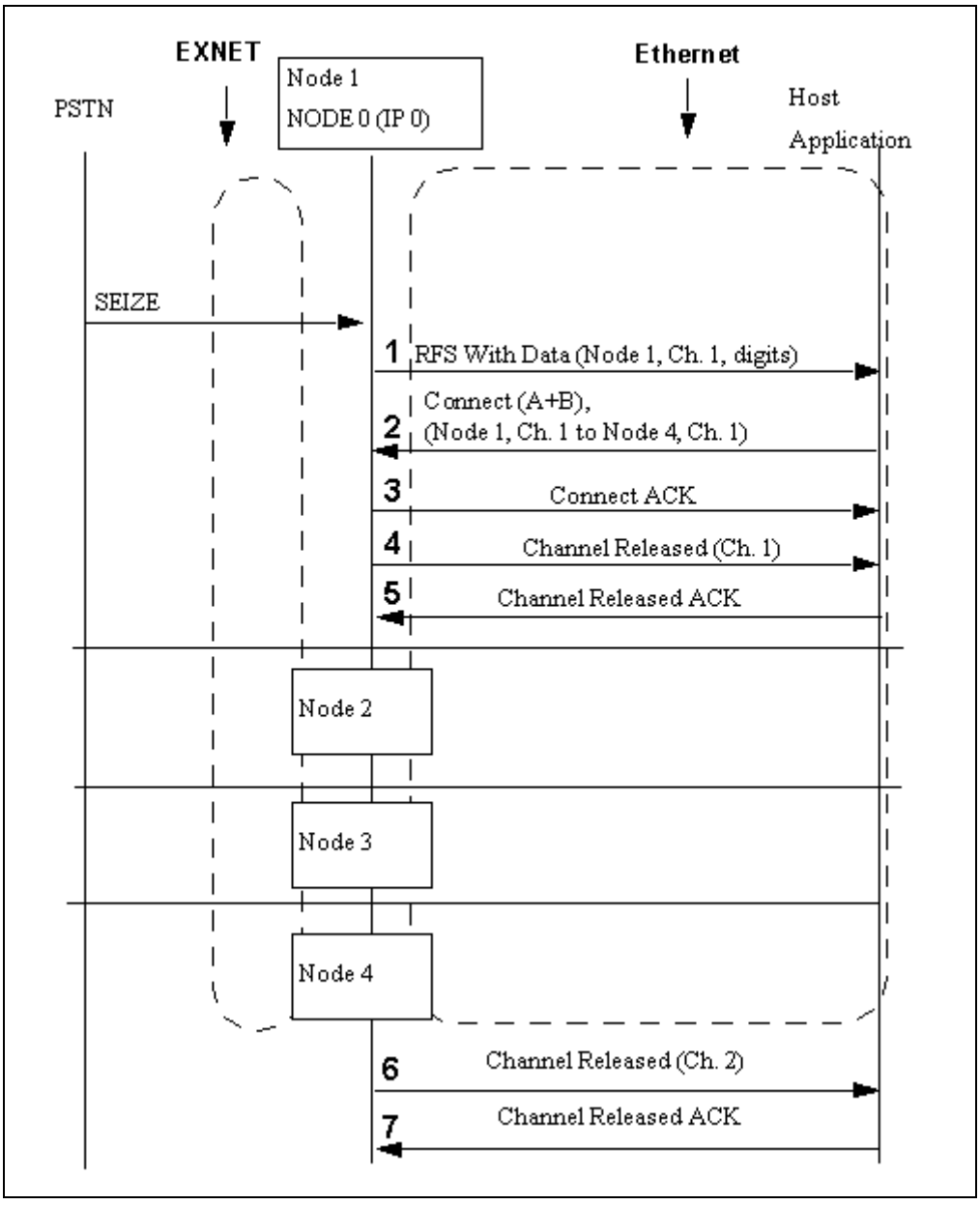
Table 10-2 API

BYTE	Field Description	Value and Indication
0	Frame	0xFE
1, 2	Length (MSB, LSB)	0x0010 (16)
3, 4	Message Type (MSB, LSB)	0x0000 (Connect)
5	Reserved	0x00
6	Sequence Number	0xSN
7	Node ID	0x01 (Node 1)
8	AIB Address Method	 0x10 (Positive ACK)
9	Number of Address Elements	
10	Address Element 1: Channel A Address Type	 0x0D (Channel)
11, 12	Logical Span (MSB, LSB)	0x0000 (Span 0)
13	Channel	0x00 (Channel 0)
14	Address Element 2: Channel B Address Type	 0x0D (Channel)
15, 16	Logical Span (MSB, LSB)	0x0041 (Span 65)
17	Channel	0x00 (Channel 0)
18	Checksum	0xCS

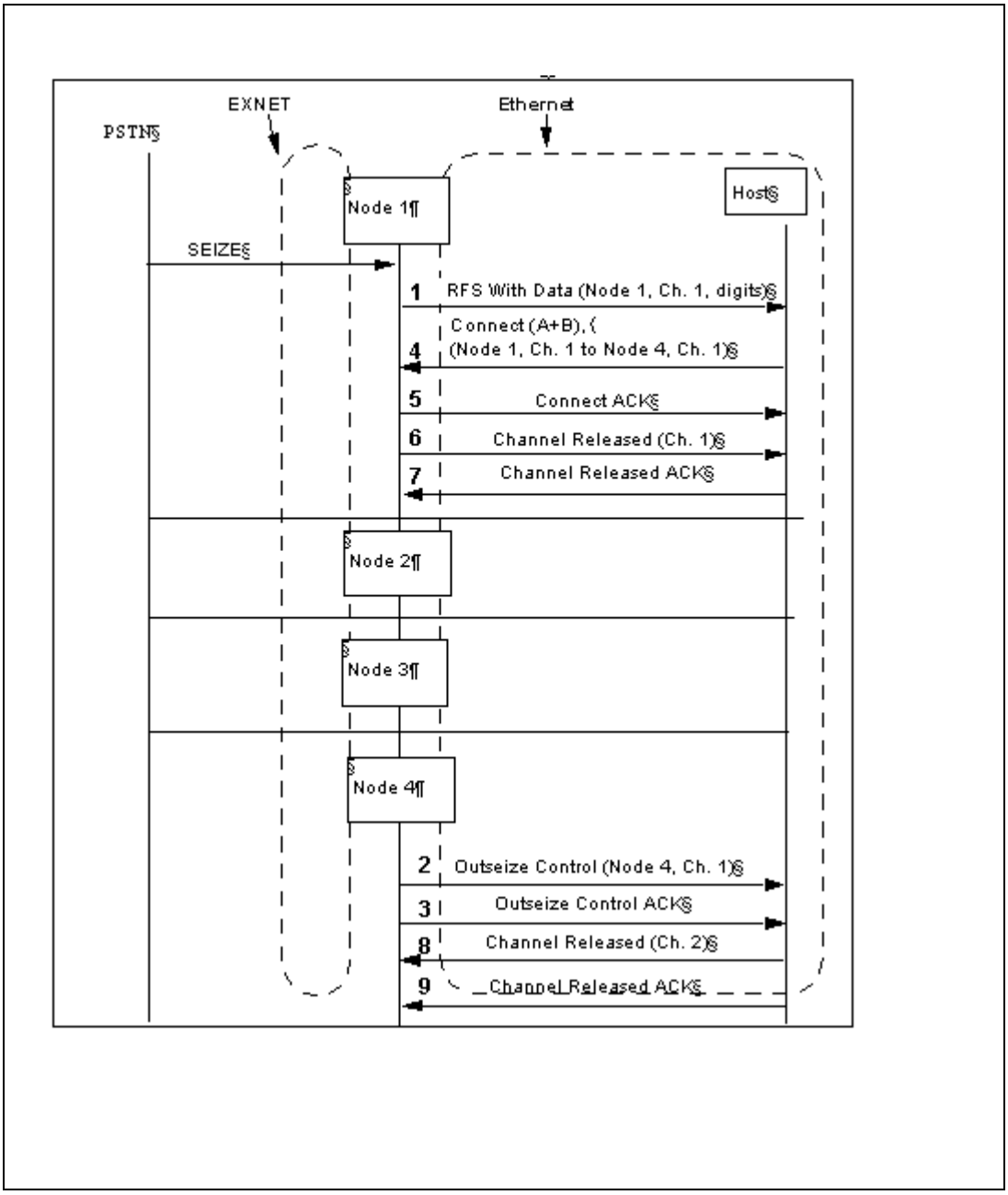
Call Flows

The following two examples illustrate how the host manages call control over the CSP.

Connect



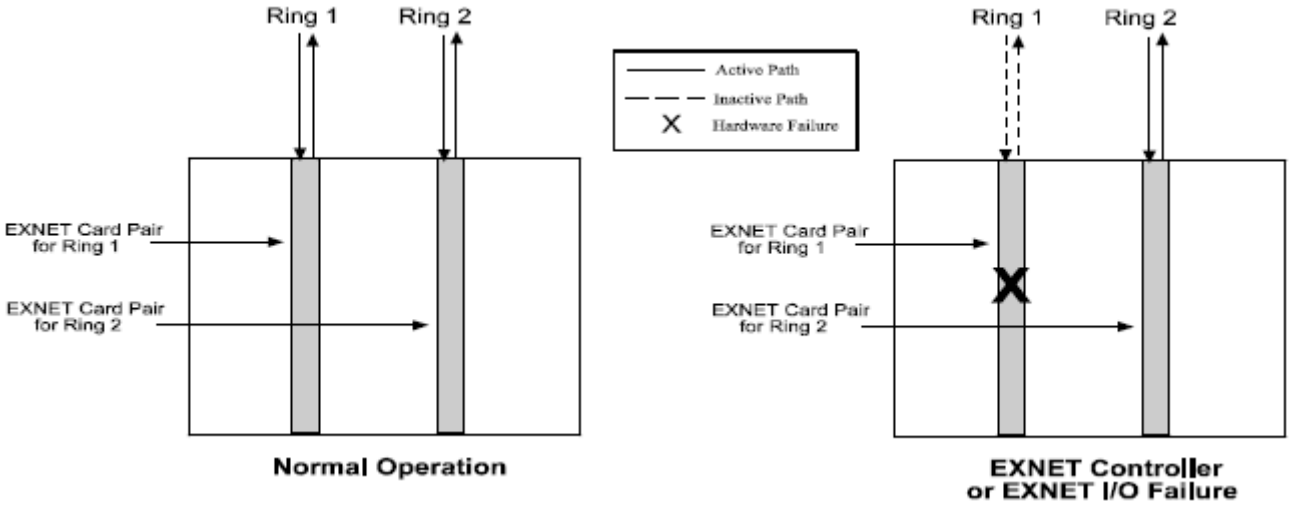
Outseize Control and Connect



EXNET® Redundancy

For redundancy, each node in an CSP can be configured with two EXNET-ONE card pairs and rings. Under normal operation, traffic is mirrored on both rings. If a node's EXNET-ONE fails, all of the traffic for that node continues to be routed through the other EXNET-ONE card and ring, and no connected calls are lost.

Figure 10-5 EXNET® Redundancy



Configuration

Install the EXNET® hardware for each ring as described in the Hardware section of the *Installation and Maintenance Guide*.

After the Logical Node IDs are assigned and the nodes are configured, you must configure each ring with the *EXNET Ring Configure* message. The host should maintain a record of the relationships between physical and logical EXNET® rings (Logical Ring ID/Slot # of EXNET-ONE card).

See the *Configuration (10-6)* section for details and examples for configuring the ring.

CSP Conferencing

CSP Conferencing allows channels on any CSP node to be connected to a conference. The conference can use a single DSP resource located on any node on the ring. All of the conferencing capabilities of a single CSP are supported.

Channels can use a remote DSP resource on the ring if:

- Both nodes have access to the EXNET® ring.
- The conference node has available CSP conferencing timeslots to the remote channel onto its local bus.

Hardware Requirements

The EXNET-ONE card (EXS-XIO-1201) and the DSP-ONE card are required for CSP conferencing.

Terminology

The following terms are used for CSP Conferencing:

Conference Node

The node where the DSP resource for the conference is located.

Timeslot

A space on the TDM bus that may or may not contain voice or data information. The CSP 2090 has 2,048 timeslots; the CSP 2040 has 1,024.

Allocated Timeslot

A local timeslot on the Conference Node that is available for CSP conferencing.

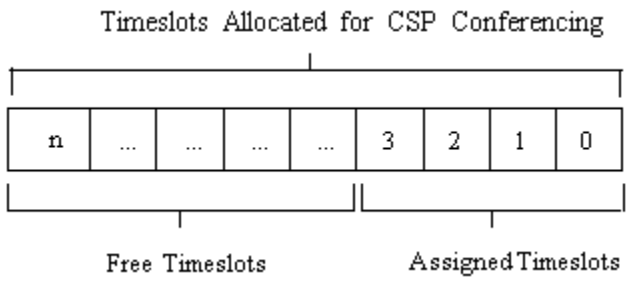
Assigned Timeslot

A local timeslot allocated for CSP Conferencing and assigned to a channel in a conference.

Free Timeslot

A timeslot allocated for CSP Conferencing but not assigned to a conference.

Figure 10-6 Free Timeslots



Channel A specific timeslot on the bus that is associated with a call.

Local Channel A channel located on the Conference Node.

Remote Channel A channel located on a node other than the Conference Node.

Configuration The only special configuration required for CSP Conferencing is the allocation of available timeslots on each node for conferencing, using the *EXS Node Configure* message. When these available timeslots are allocated, conferences are created and set up the same way as they are for a single CSP. Information about timeslot configuration is stored on the Matrix Controller card(s) and is maintained even if the Matrix Controller card is switched over.

Allocating Timeslots for CSP Conferencing

To allocate timeslots for CSP Conferencing, use the *EXS Node Configure* message with a value of 0xFFFF in the Number of Timeslots field. Entering any other value returns a response status of Invalid Value for Entity.

When a conference is created, timeslots are allocated for all parties in the conference (local and remote channels) and broadcasting if broadcasting is supported by the conference type. A Standard Conference requires one additional timeslot, and a Monitor Conference requires two additional timeslots.

Calculating the Number of Available Timeslots

The number of timeslots used by line cards is determined by the number of spans supported by the card, not by the number of spans assigned by the host.

To calculate the number of timeslots available for EXS conferencing in your CSP, use the following formula:

Total Number of free Timeslots = (2048 minus (the number of T1 spans multiplied by 24) minus (the number of E1 spans multiplied by 32) minus (the number of J1 spans multiplied by 32) minus (the number of VDAC spans multiplied by 32) minus (the number of DS3 spans multiplied by 24))

For example, you would calculate free timeslots on a node with three T-ONE-16 line cards as follows:

2048 - (3* (16 * 24)) = 896 timeslots available for EXS conferencing.

Card:	Spans * Channels/Span	= Time Slots Used
ST1LC 8-span	8 * 24	192

Card:	Spans * Channels/Span	= Time Slots Used
SE1LC 8-span	8 * 32	256
T-ONE 8-span	8 * 24	192
T-ONE 16-span	16 * 24	384
E-ONE	16 * 32	512
DS3	28 * 24	672
VDAC-1 w/2 modules	(2 * 32) + 16 odd channels	80
VDAC-1 w/4 modules	5 * 32	160
VDAC-2 profile 1	32 * 32	1024 (port consuming mode)
VDAC-2 profile 2	16 * 32	512 (port consuming mode)

If the allocation is successful, a positive acknowledgment (ACK) is returned, with data bytes that indicate the number of timeslots allocated. If the allocation is unsuccessful, a negative acknowledgment (NACK) is returned, with data bytes that indicate the number of unallocated timeslots available for CSP conferencing. To successfully allocate timeslots after a negative acknowledgment, re-issue the message with the number of free timeslots that were indicated in the response.

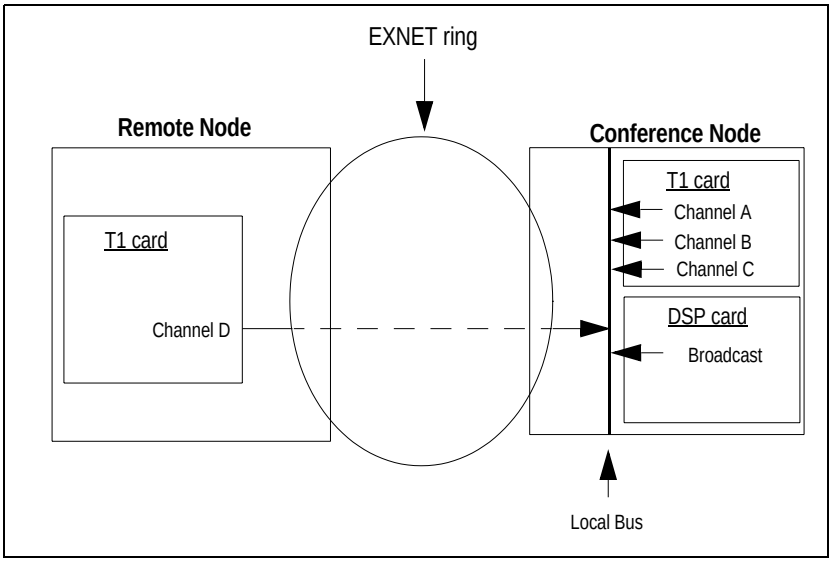
Example

This example shows a four-party standard conference established on a node where three parties are local and one party is on a remote node.

The conference requires six CSP Conferencing timeslots on the local bus, in addition to the three physical timeslots initially assigned to the local channels in the line card:

- Three physical time slot are used on the line card (T1/E1, which is already allocated)
- Six time slots need to be reserved because of the conferencing:
  - Two for conferencing to broadcast to the ring (one for u-law, one for a-law) Note: only one time slot is needed if you only use a-law or u-law.
  - Four for each conference channels to attach to the ring

Figure 10-7 Allocating CSP Conferencing Timeslots



Alarms

The CSP uses the following alarms to monitor CSP Conferencing Timeslots:

- Available CSP Conferencing Timeslot Count Changed (0x02)
- Resource Utilization Threshold Reached (0x13)

Available Timeslots

If a line card is inserted or removed, the number of timeslots available for an CSP Conference changes, and the host receives the CSP Node alarm, Available CSP Conferencing Timeslot Count Changed. The data in the alarm indicates the new timeslot count.

Free Timeslots

When the number of free timeslots decreases to a threshold pre-set by the host, the host receives a General Alarm of Resource Utilization Threshold Reached. The *Data* field in the alarm indicates a Resource Type of CSP Conferencing Timeslots.

The host sets the upper and lower threshold of the alarm, with the *System Configuration* message, are as follows:

Table 10-3 Upper and Lower Threshold

Configuration Type	Resource Threshold (0x07)
Data[0]	Resource Type: Conferencing Timeslots (0x01)
Data[1, 2]	Number of Assigned Timeslots to Generate an *Alarm Cleared*
Data[3, 4]	Number of Assigned Timeslots to Generate an *Alarm*

You can query the CSP Conferencing Resources with the *System Resource Usage Query* message, which reports the following:

- Number of timeslots currently used by CSP conferencing
- Number of timeslots available for CSP conferencing
- Percent of timeslots used for CSP conferencing

The *System Resource Utilization Query* message reports the number of timeslots currently used by CSP conferencing. You can calculate the number of free timeslots by subtracting this number from the total number of timeslots that are allocated for CSP Conferencing.

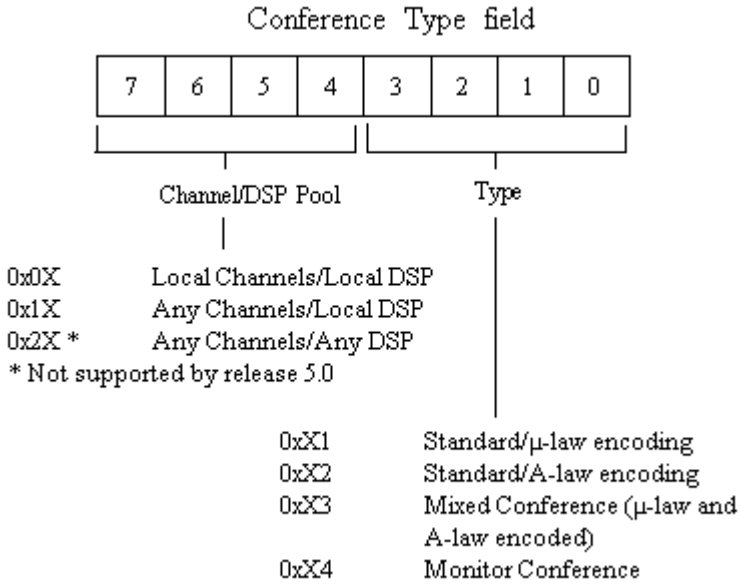
Conference Management

As with a single CSP, conferences are created using the *Conference Create* message.

Conference Type Field

You can use the Conference Type field in the *Conference Create* message not only to specify a conference type, but also to specify the channels and DSP resources to be used to establish the conference. The lower nibble (the right-most four bits) of the Conference Type field indicates the conference type. The upper nibble (the left-most four bits) indicates which channels and DSP resources can be used to establish the conference.

Figure 10-8 The *Conference Type* Field in the *Conference Create* Message



Channel/DSP Pool

This nibble indicates whether the conference can be established on the designated node only, or on any node with an available DSP resource. The nibble also indicates whether the channels in the conference must be located on the designated node, or on any node with available channels. The options are listed below:

Local Channels/Local DSP (0x0X)

- Only local channels can connect to the conference
- Only a DSP residing on the node that receives the *Conference Create* message can be used to establish the conference. If no DSP is available on the node, a Response Status is returned of No DSP Resource Available.

Any Channels/Local DSP (0x1X)

- Any local or remote channels can be connected to the conference.
- Only a DSP residing on the node that receives the *Conference Create* message can be used to establish the conference. If no DSP is available on the node, a Response Status is returned of No DSP Resource Available.

Any Channels/Any DSP (0x2X)

- Any local or remote channels can connect to the conference.
- If available, a DSP located on the node receiving the *Conference Create* message is used to establish the conference. If no DSP is available on the node, a remote node is chosen, based on the number of available resources.

Conference ID

The response to the *Conference Create* message contains a two-byte Conference ID. Within this Conference ID, the bits 2-4 (zero based) of the MSB indicate the Logical Node ID of the Conference Node.

You can use the *System Configuration Query* message to query the currently-active conference IDs for each node.

The conference exists until the host specifically deletes it, or until a system event deletes it, as described in the System Events section below.

Please refer to the *Conference Create 0x004B* message in the *API Reference* for more information.

System Events

This section describes the following effects of various system events on CSP Conferencing, and any required host interaction:

- EXNET® ring failure
- Insert, Remove, or Reset:
  - Line cards
  - DSP-ONE card
  - Matrix Controller card

Line Cards

Insert, Remove, or Reset

If a line card is removed or reset, any channels on the card involved in a conference are purged from the conference.

CAUTION

If a line card is removed or inserted, the number of timeslots available on a node for CSP Conferencing is dynamically

updated. The host is informed of the new number of available timeslots with an Alarm message.

CAUTION

If inserting a line card brings the number of timeslots available for CSP conferencing below the number assigned to CSP conferences, some channels connected to conferences are purged. This situation is a concern only for systems with many line cards installed and many CSP conference connections.

You can query the use of CSP conferencing resources, with the *System Resource Utilization Query* message.

DSP-ONE Card

Insert

When a DSP-ONE card is inserted into a node, it must be configured for conferencing with the *DSP SIMM Configure* message before it can be used to establish a conference.

Remove

If a DSP-ONE card is removed from a node, any conferences active on the card are deleted and all channels associated with the conference are purged.

Matrix Controller Card

Removal/Reset

If the conferencing node has a redundant Matrix Controller, and the active Matrix Controller is removed or reset, all conferences and conferees are maintained through the switchover.

EXNET® Ring Failure

EXNET® ring failure is defined as a node becoming inaccessible to the ring due to:

- Failure, removal, or reset of an EXNET-ONE card.
- Failure of the fiber optic cable between nodes.

Redundant Ring

All conferences and connections are maintained.

Non-redundant Ring

All inter-nodal conferees are purged. Local connections for conferences in each node are maintained.

Ethernet Link Failure

If communication between nodes fails, conferences on the Conference Node are maintained with any local channels connected to the conference. Conferees on remote nodes are purged from the conference.

Host Control

Overview

There are two basic options for configuring host control:

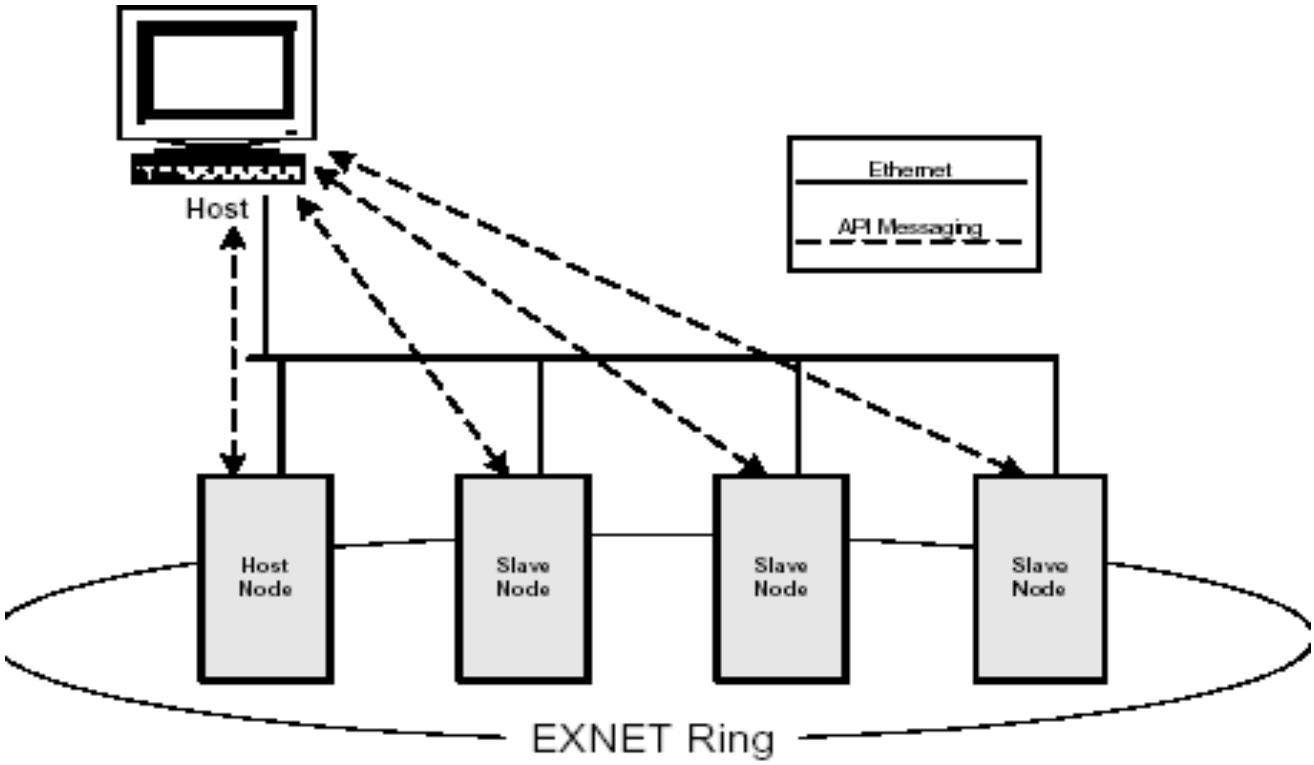
- Multiple Points of Host Control - one host, multiple host nodes
- Distributed Host Control - multiple hosts, multiple host nodes

Important! Single point of Host Control is no longer supported.

Multiple Points of Host Control

A single host can control each node directly through a dedicated TCP socket using the API messages. All messaging is sent directly between each node and the host.

Figure 10-9 Multiple Points of Host Control



Distributed Host Control

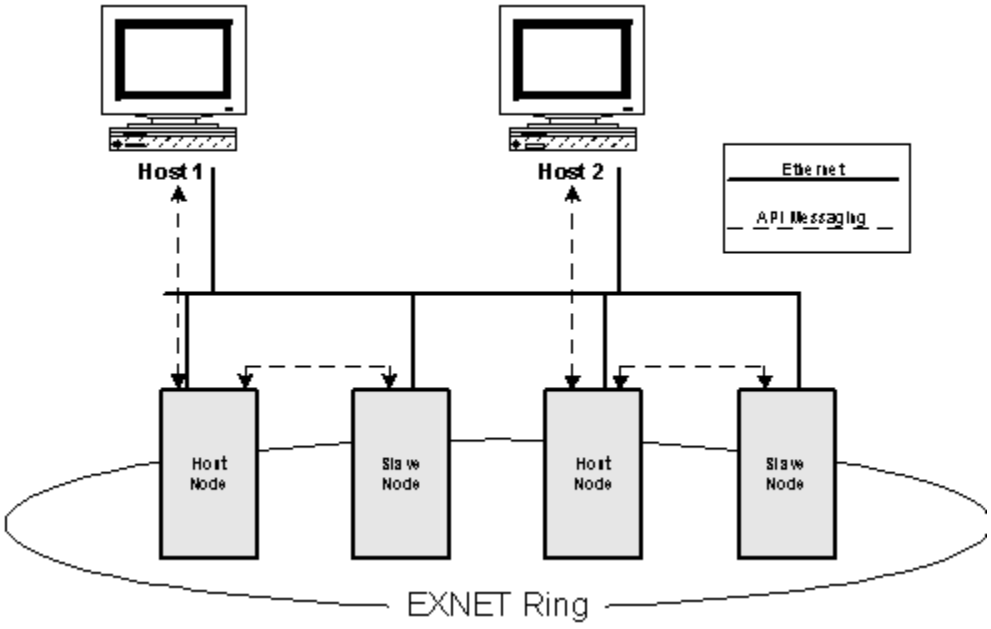
To distribute host and host node processing, multiple host nodes can be established, as shown in Figure 10-10. Any CSP node can be established as a host node by opening a TCP socket and sending any API message, such as a *Poll Request*, to the node.

Use the *Node Status Query* message to determine an EXNET® ring's host node. You can change the host/slave relationships using the *Assign EXS Host/Slave* message.

> **Important!** Do not use the *Assign EXS Host/Slave* message to establish a node as a host node. A node automatically becomes a host node when any API message, such as a *Poll Request*, is sent to it through a direct TCP connection.

The *Assign EXS Host/Slave* message establishes a relationship between an existing host node and a slave node. If the same node is entered as the host node and the slave node in the AIB, a response status of Invalid Logical Node ID (0xFC) is returned.

Figure 10-10 Distributed Host Control



Example The following is an example of the *Assign EXS Host/Slave* message.

Table 10-4 Example of the Assign EXS Host/Slave message

BYTE	Field Description	Value and Indication
0	Frame	0xFE
1, 2	Length (MSB, LSB)	0x000D
3, 4	Message Type	0x006E (Assign EXS Host/ Slave)
5	Reserved	0x00
6	Sequence Number	0xSN
7	Logical Node ID	0x01 (Logical Node 1)
8	AIB Address Method	 0x00 (Single Entity)
9	Number of Address Elements	0x02
10	Address Element 1: Host Node Address Type	 0x0E (Logical Node)
11	Address Data Length	0x01
12	Address Data[0] Logical Node ID	0x01 (Logical Node 1))
13	Address Element 2: Slave Node Address Type	 0x0E (Logical Node)
14	Address Data Length	0x01
15	Address Data[0] Logical Node ID	0x05 (Logical Node 5)
16	Checksum	CS

In-Service Upgrades

This section explains the procedures and contingencies for adding components to an active multi-node or single-chassis CSP.

Adding an EXNET® Ring to Stand-alone Chassis

You can create a multi-node CSP from two or more stand-alone CSPs by adding EXNET® hardware to each node without affecting existing calls on any nodes.

1. Add EXNET® hardware to each node
   - EXNET-ONE
   - Fiber optic cable
2. Assign Logical Node IDs to each node (*Assign Logical Node ID*)
3. Configure the EXNET® ring (*EXNET Ring Configure*)
   - Assign Logical Ring ID to each node
   - Configure Transmit/Receive Mode of each node
4. Bring the ring in service (*Service State Configure*)
   - This message only needs to be sent to one node to bring the entire ring in service

Calls can now be connected across nodes.

Adding a Node

The procedure for adding a node is the same for a redundant or non-redundant EXNET® ring configuration. However, adding a node to a non-redundant ring configuration causes some inter-nodal calls to purge while the ring resets.

See also *Adding a Node to a Multi-Node System (7-60)* in the *EXS SwitchKit Converged Services Administrator (CSA) User's Guide* for instructions on completing this procedure with CSA.

Adding a Node to a Non-redundant EXNET® Ring

Follow the steps below to add a node to a non-redundant ring configuration. This procedure assumes the configuration explained in the section *Configure Transmit/Receive Mode (10-12)*. For this example, assume that the EXNET® ring is assigned Logical Ring ID 1.

1. Disconnect the EXNET® ring where necessary.
2. Disconnect the fiber optic cable from the EXNET B port of the node that will be adjacent to the new node. The system is placed in loopback mode and all connections are maintained.
3. Make the EXNET® connections to the new node.
4. Configure the node as follows:

- Establish the Ethernet connection.
- Download the system software.
- Re-establish the socket connection.
- Assign Logical Node ID.
- Configure CSP options on the node, if necessary (*EXS Node Configure*).
- Assign Logical Ring ID 1 to the ring (*EXNET Ring Configure*).

5. Configure the Transmit/Receive Mode of the ring (*EXNET Ring Configure*).

CAUTION

When the Transmit/Receive Mode is configured, the EXNET ring resets and comes back in service. All inter-nodal calls are purged during the reset.

Adding a Node to a Redundant EXNET® Ring

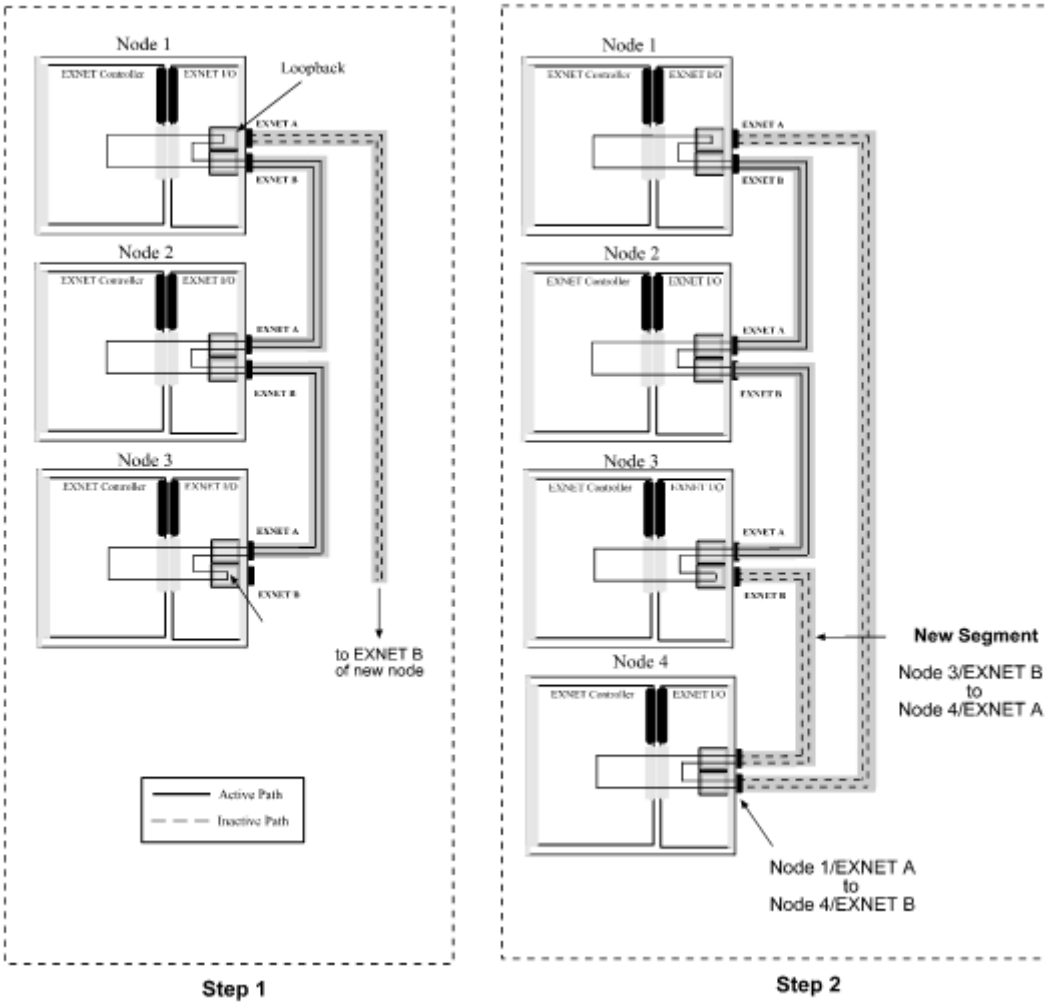
Follow the steps below to add a node to a redundant ring configuration. This procedure assumes the configuration shown in Figure 10-11. For this example, assume that the EXNET® rings are assigned Logical Ring IDs 1 and 2.

1. Disconnect the EXNET® ring where necessary: Disconnect the fiber optic cable from the EXNET B port of the EXNET-ONE cards in the node that will be adjacent to the new node (both Ring 1 and Ring 2). Both rings are placed in loopback mode and all connections are maintained.
2. Make the EXNET® connections to the new node for both rings
3. Configure Node
   - Establish the Ethernet connection
   - Download the system software
   - Re-establish the socket connection
   - Assign the Logical Node ID
   - Configure CSP options on the node, if necessary (*EXS Node Configure*)
4. Configure Ring 1
   - Assign Logical Ring ID 1 to the EXNET-ONE that is attached to Ring 1 (*EXNET Ring Configure*)
   - Configure the Transmit/Receive Mode of Ring 1 (*EXNET Ring Configure*)

Ring 1 resets and comes in service with the new node attached. All inter-nodal calls are maintained on Ring 2 while Ring 1 is resetting.

5. Configure Ring 2
   - Assign Logical Ring ID 2 to the EXNET-ONE that is attached to Ring 2 (*EXNET Ring Configure*).
   - Configure the Transmit/Receive Mode of Ring 2 (*EXNET Ring Configure*). Ring 2 resets and comes in service with the new node attached. All inter-nodal calls are maintained on Ring 1 while Ring 2 is resetting.

Figure 10-11 Adding a Node to an EXNET® Ring



Expanding the EXNET® Ring

This section describes the passive addition of an CSP node to an active, non-redundant EXNET® ring. This feature allows for easy ring expansion without disruption of voice traffic in an operating system. This feature describes the normal ring mode. However, if you want to use this procedure to use the enhanced fault tolerance ring mode feature, omit steps 6 and 10 from the following procedure. (The Enhanced Fault Tolerance Ring Mode is described in the following section.)

Follow the steps below:

1. Power Up New CSP Node.

The new CSP node must be powered up, before any connections (Ethernet, fiber optic) are made. The CSP Power-On Diagnostics will routinely verify the performance of the CSP with installed cards.

2. Establish Ethernet Connection to host.

In order to configure the new CSP node, the node's Ethernet interface must be connected to the common Ethernet network before being connected to the host.

> **Important!** Before making an Ethernet connection to the new CSP node which is about to be added to the active EXNET® ring, make sure that the new CSP node has default configuration settings (for example, Power-On Reset).

3. Download System Software.

The new CSP node must run the same CSP software version as the CSP nodes in the active network. If not, download the correct CSP software version from the host to the new CSP node. After downloading is complete, the new CSP node automatically restarts execution of the new system software, and is ready for further configuration.

4. Assign Logical Node ID.

Assign the new CSP node a new Logical Node ID. This is done by the host sending an *Assign Logical Node ID* (0x10) message, to the new CSP node. The Logical Node ID allows the new CSP node to use API messages in further configuration steps.

5. Perform RIC Runtime Diagnostics.

To ensure proper functionality of the new CSP node with the EXNET Ring Interface Card (RIC), perform the RIC Runtime Diagnostic procedure. This procedure is based on the new CSP node receiving the *Ring Configure* (0x74) message with the *RIC Diagnostics* (0x04) entity.

API Message Name	API Message Code	Entity Name	Entity Code	Diagnostic Type
Ring Configure	0x74	RIC Diagnostics	0x04	[0x01.0x04]

The Diagnostic Type in the table above specifies one of four different types of the RIC diagnostic procedure:

- *Normal mode, Internal loop back* (where the I/O hardware is instructed to internally loop back the I/O port)
- *Normal mode, External loop back* (where the fiber opticfiber optic cable is required to loop back externally both I/O ports)
- *Forced mode, Internal loop back* (where the I/O hardware is instructed to internally loop back the I/O port)
- *Forced mode, External loop back* (where the fiber optic cable is required to loop back externally both I/O ports)

Important! The *Diagnostic Type* [*normal mode*] does not allow execution of RIC Diagnostic Procedure while the EXNET® ring is up and running. Performing the RIC Diagnostic will bring the EXNET® ring down.

The *Diagnostic Type* [*external loop back mode*] can be used when both I/O ports of CSP node are externally looped back with fiber optic cable.

6. Prepare New CSP Node for Addition to EXNET® Ring (Not used or ignored in Enhanced Fault Tolerance).

To configure the new CSP node for non-disruptive addition into the active EXNET® ring, the host notifies the new CSP node by sending the *Ring Configure* (0x74) message with the *Prepare For Addition* (0x05) entity.

API Message Name	API Message Code	Entity Name	Entity Code
Ring Configure	0x74	Prepare for Addition	0x05

7. Set-up EXNET® Ring Configuration Parameters for the New CSP Node.

In order to prepare the CSP node and the EXNET® Ring Interface Card for active participation in data exchange over the EXNET® ring, the following configuration elements must be defined:

- *Logical Ring ID* must be assigned. It must be the same ID as the ring to be expanded.
- The RIC I/O port *Transmit mode* must be set according to user requirements for the *Transmit-Receive* or *Receive-Only* mode.
- The new CSP node's *Number Of Packets* must be defined (if the *Transmit-Receive* mode is chosen for the RIC). If the *Transmit-Receive mode* is set, there must be enough available bandwidth on the EXNET® Ring to accommodate the new CSP node.

All parameters specified above are configurable by sending the *Ring Configure* message with selective entities to the new CSP node as follows:

API Message Name	API Message Code	Entity Name	Entity Code
Ring Configure	0x74	Logical Ring ID	0x01
		Transmit Mode	0x02
		Number of Packets	0x03
		Ring Mode	0x0A

8. Send *Loop Back Port* message.

Before sending the *Loop Back Port* message make sure that there are no other ports in existing CSP node configuration already in the looped back mode. (Sending the *Loop Back Port* message in this case, could cause some nodes to become isolated.)

Send the *Loop Back Port* message before making any change in physical connection scheme regarding the new CSP node.

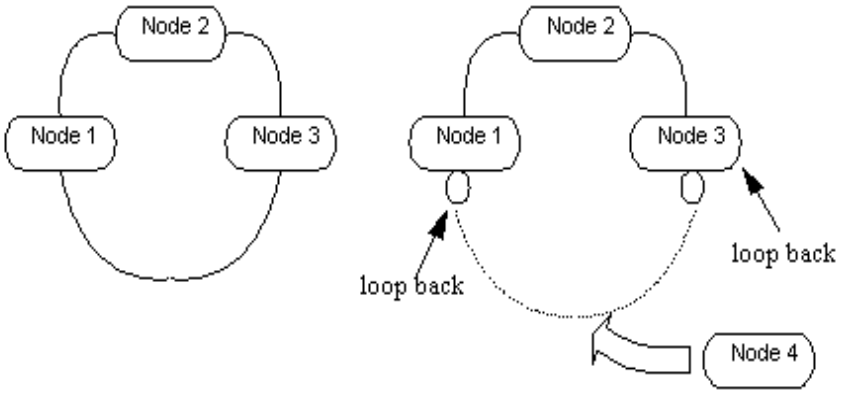
Disconnect the existing fiber optic ring between two adjacent nodes (selected by a user), to allow the new CSP node to join the EXNET® ring. The two adjacent nodes in existing configuration must loop back to their respective I/O ports. For example Node 1 and Node 3 in Figure 10-12 so that the transmission path over the EXNET® ring still exists, keeping the ring up and running.

In order to force respective I/O ports (of two adjacent nodes) working in loop back mode, the *Ring Configure* (0x74) message with *Loop Back Port* (0x07) entity must be sent to one of adjacent nodes.

API Message Name	API Message Code	Entity Name	Entity Code
Ring Configure	0x74	Loop Back Port	0x07

Important! The *Loop Back Port* (0x07) message must be sent to only one CSP node, which is expected to change its “A” I/O port, to work in loop back mode. The adjacent CSP node will be automatically notified about this message being processed, and its “B” I/O port will become looped back. The ID of this CSP node is included into the *Loop Back Port* message as a parameter.

Figure 10-12 Loop Back of Adjacent CSP Nodes

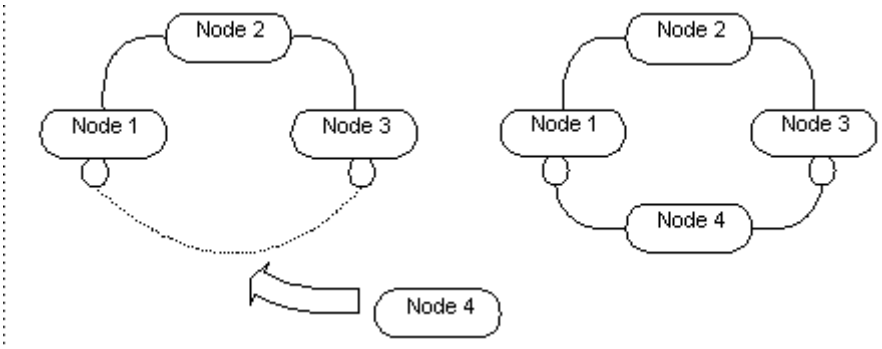


9. Connect fiber optic Cable to New CSP Mode.

Once the fiber optic cable between Node 1 and Node 3 does not carry ring traffic because the respective ports of these two nodes work in looped back mode, the existing fiber optic connections can be expanded to incorporate the new Node 4 into the ring.

The Node 4 I/O ports are both connected with fiber optic cables to Node 1 and Node 3 as shown in Figure 10-13. This connection does not provide any voice traffic exchange to and from Node 4, because the respective I/O ports of Node 1 and Node 3 still remain looped back.

Figure 10-13 fiber optic Connections to Add New CSP Node



10. Send *Expand Ring* message (Not used or ignored in ERFT. In ERFT the node signals to its adjacent nodes to initiate port expansion.)

At this point, the existing EXNET® ring is about to be expanded with the new CSP Node 4. To allow Node 4 to become active in the EXNET® ring, both Node 1 and Node 3 must remove the loop back from their respective I/O ports. To avoid the possibility of ring going down, removal of the loop backs on both Node 1 and Node 3 must be done essentially at the same time.

For the purpose of expansion of the EXNET® ring with the new CSP Node 4, the *Ring Configure* (0x74) message with *Expand Ring* (0x06) entity must be sent down to either Node 1 or Node 3, which has one looped back I/O port.

API Message Name	API Message Code	Entity Name	Entity Code
Ring Configure	0x74	Expand Ring	0x06

When both CSP nodes, Node 1 and Node 3, are not looped back, the EXNET® ring data is routed through Node 4. At this point, Node 4 is internally configured not to transmit its own packets onto the EXNET® ring.

11. Send the *Add Node* Message.

At this point there are no looped back I/O ports in the EXNET® ring, and the ring data is routed through Node 4 (both I/O ports remain open). For Node 4 to actively participate in the EXNET® ring, the *Ring Configure* (0x74) message with *Add Node* (0x08) entity must be sent to Node 4.

API Message Name	API Message Code	Entity Name	Entity Code
Ring Configure	0x74	Add Node	0x08

Upon receiving this message, Node 4 attempts to verify if the Master Node of the EXNET® ring acknowledges its presence on the EXNET® ring. When this is verified successfully, Node 4 turns on its transmitter (if configured for Transmit/Receive mode) to allow its own packets to appear on the EXNET® ring.

12. Perform Final Verification.

The final verification must be performed on all CSP nodes directly involved in the passive addition of the new CSP node to active ring (Node 1, Node 3 and Node 4):

- The EXNET® ring should be in service.
- Respective I/O port on Node 1, which prior to the reconfiguration was looped back, should be open.
- Respective I/O port on Node 3, which prior to the reconfiguration was looped back, should be open.
- Both I/O ports of the Node 4 should be open.

13. User configuration on the new CSP node.

At this point, you can configure required CSP options on the new CSP node. For this purpose, the API messages can be used.

Enhanced Ring Fault Tolerance

Overview

This section describes the Enhanced Ring Fault Tolerance (ERFT) and Single Ring Redundancy (SRR) features.

The ERFT feature enhances the EXNET® ring robustness, performance, and tolerance to unexpected faults.

The EXNET-ONE card has the ability to signal to its adjacent nodes when it has opened or looped back its port. This one feature enables the ring controller software to support online recovery from a single point of failure, seamless single ring redundancy switchover, and improved performance.

Primarily, ERFT supports the ring interface controller (EXNET-ONE card) component, which is responsible for establishing and maintaining the EXNET® ring using the ERFT ring mode.

The ERFT ring mode supports the EXNET-ONE ring interface controller card, activating full ERFT support. This includes enhanced fault detection, isolation, and recovery, increased performance and response time, enhanced SRR with seamless switchover support, and enhanced live node insertion (passive addition).

Features

The following enhanced fault tolerance features are provided by the EXNET® ring controller software:

EXNET® Live Node Insertion (Passive Addition)

The EXNET® live node insertion (passive addition) has been merged into the ERFT ring mode state machine. The EXNET-ONE card synchronizes the two adjacent nodes to loop back and expand their ports, so as to not bring down the ring. The EXNET-ONE card also has the ability to turn off the fiber optic port light to synchronize and signal between adjacent nodes. Passive addition, using the ERFT state machine, allows physical, link, and data validation on each port independently.

Single Ring Redundancy (SRR) with Seamless Switchover

While remaining transparent to the host, SRR performs a switchover seamlessly, without dropping calls due to the fact that a node can removed itself (or be removed) from the ring, allowing for a standby node to take over without disrupting the ring or local bus.

Debugging Capabilities

A real-time logging function has been created to trace various conditions. Also, since the logging function requires no resources and a small amount of CPU overhead, ISRs are permitted to log events. The logging function is similar to "printf", except that a log type is also supplied for filtering. Each event is time stamped with a resolution of five milliseconds and stored in a real-time circular buffer that is retained through a push button reset. While the ring is up, each node can synchronize their times to analyze timing sensitive scenarios.

The ring monitor is accessible via the debug console. At any time, monitor statistics can be queried, including TX and RX slips, node accessibility, and acquired failures.

ERFT Ring Mode

After the EXNET-ONE card has been assigned to a Logical Ring ID, the host uses the EXNET® Ring Configure message to enable the ERFT Ring Mode, which supports the following functionality:

- Multi-ring Redundancy/Expanded Switching
- Ring Timing/Mastership Configurability
- Enhanced Ring Initialization
- Enhanced Live Node Insertion/Removal
- Runtime Software Link Validation
- Enhanced Single Ring Redundancy (“Hot” Standby)
- Improved Performance
- Improved Ring Status Report Generation
- Improved Mastership Re-arbitration Logic

Failure Processing

The ERFT ring mode state machine performs the following failure processing. A failure can be defined as an unrecoverable fault that does not enable the EXNET-ONE card from participating on the ring. Anything that causes the Ring Controller State Machine to reset other than the host bringing the ring out of service (OOS) is considered a fault.

Ring Status Report

When a failure occurs, the host is notified that the EXNET-ONE card has gone Out of Service by the *Ring Status Report (0x72)* message. A global failure count is incremented and internal loop back diagnostics are performed. If the diagnostics were successful and the failure threshold was not reached, the ring state machine is reset.

Reset

All slave nodes transition to the “waiting for addition” state, while the master node brings the ring back up. Before starting ring initialization, the master sends a “reset” message to all slaves. This generates an “in service” event, causing all slaves to re-participate on the ring. After the third unsuccessful attempt to bring the ring up, the master node, will no longer be the master node.

A slave node will not intentionally bring down the ring. If the slave node fails to be brought onto the ring, the slave will notify the host it has reset and is waiting to be passively added.

If there is a standby node available, a switchover is performed. If there is no standby node, mastership re-arbitration is performed.

Failed state

If at anytime, diagnostics fail, or the failure count exceeds a threshold (currently 12), the EXNET-ONE alarms to the host and transitions to the “failed” state. Once in the “failed” state, the EXNET-ONE will not participate on the ring until the “host service state” is toggled and it can pass diagnostics.

Mastership Re-Arbitration Logic

With ERFT, a node can be removed from the ring without that node intentionally bringing down the ring. Only the master node can induce a re-arbitration. If the master node fails to bring the ring up after three attempts or transitions to the “failed” state, the master node will no longer be the master node. If the master EXNET-ONE is removed or brought out of service, the Matrix Controller card induces a re-arbitration.

When the master node induces a re-arbitration, it does not participate unless a new master has not declared itself within 30 seconds. After 30 seconds, if another node has not become master, the previous master node tries to become master again. This happens when there is only a single master configurable node.

There is one case where a slave node can force a re-arbitration. This occurs if the slave node stops hearing from the master node (for example, the master node chassis is off). In this case, all slave nodes force a re-arbitration.

Real-time Software Link Validation

ERFT provides the state machine the ability to detect, isolate, and recover from a single point of failure without dropping existing connections. Software link validation is identical to ring initialization, except that the data layer is not being validated. Only the physical and link layer are validated. A faulty link can be isolated and the ring healed while calls are maintained. Each link can be revalidated at a rate of approximately 20 milliseconds per node, resulting in an online revalidation of a 32 node system within about 700 milliseconds.

Ring Controller State Machine Global States

The ERFT Ring Mode supports the following eight global states. Figure 10-14 also defines the global states of the ring controller state machine.

1. **RESET -** This is the start state for the state machine. This state unconditionally resets the hardware and transitions to the WAITING FOR CONFIGURATION state.
2. **WAITING FOR CONFIGURATION -** Until the host configures the EXNET-ONE card (assigns the EXNET-ONE card to the ring, specifies transmit (TX) mode, number of packets and so forth.), the ring state machine resides in this state. Once the EXNET-ONE card is configured, the host In Service state is checked.
3. **OUT OF SERVICE (OOS) -** If the host has the ring out of service, the ring state machine waits in this state until the host brings the ring in service.
4. **WAITING FOR ADDITION** - If the ring state machine resets while the host has the ring In Service, the ring state machine waits in the WAITING FOR ADDITION state until the host passively adds the node to the ring or the master node resets. Only slave nodes enter this state. The master node automatically brings the ring back up.
5. **CONNECTING TO RING -** When the host brings the ring in service or the master mode resets (bringing down the entire ring), the EXNET-ONE card enters this state to participate on the ring. To the host, this is seen as a single state. Actually, this is a series of sub-states based off of the following factors:

- Whether the EXNET-ONE card is configured as a master or slave node, and is it in a SRR configuration.
- Whether the EXNET-ONE card is taking the role of the active or standby card.

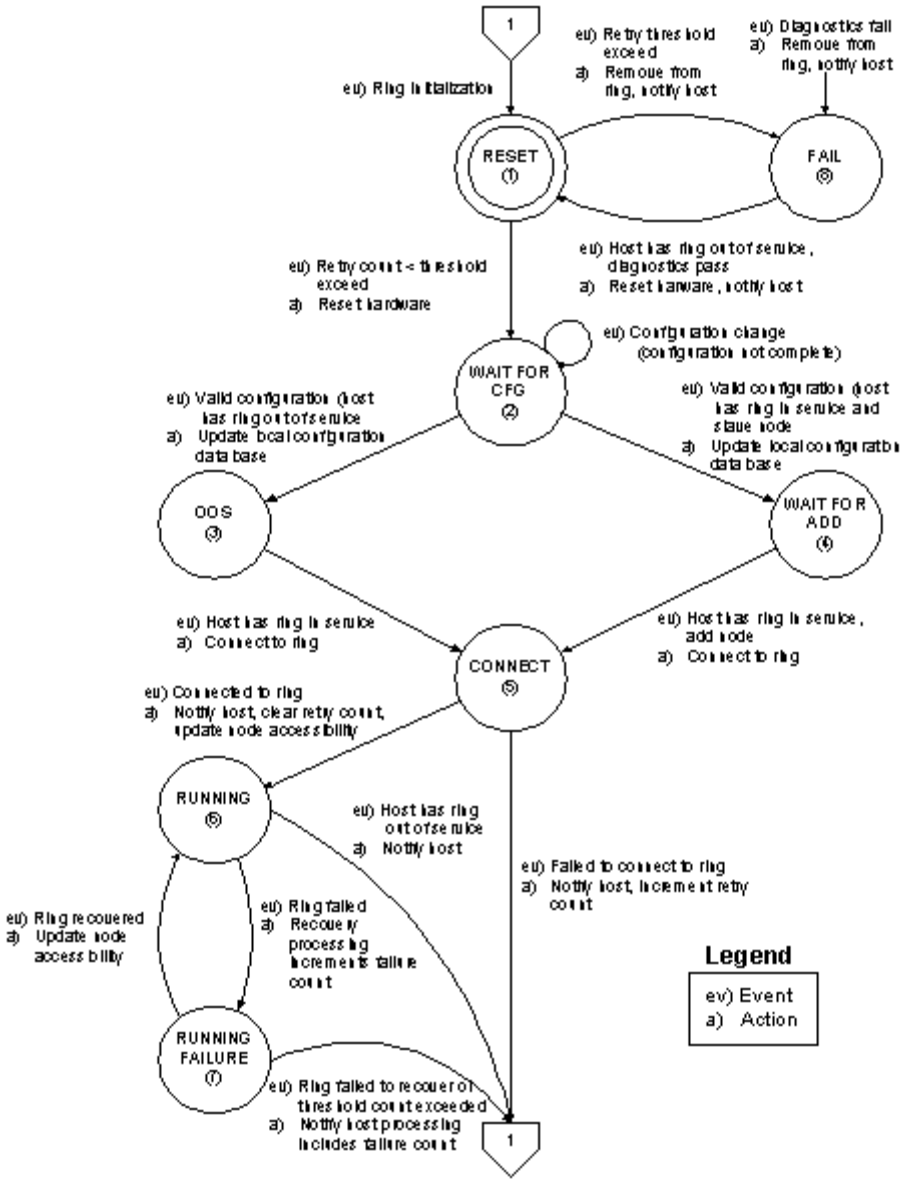
6. **RUNNING (INS) -** The EXNET-ONE card is connected to the ring. If it is configured to transmit data, the nodes entire payload is transmitted onto the ring (1 or 4 packets). Once the ring is running, the ring state machine interrogates the ring to determine which nodes are transmitting on the ring. The node accessibility information is sent to the Matrix Controller card so that Layer 4 can route calls over the ring. While in this state, the ring is continuously monitored and its health maintained. Different features are supported based off the ring state machine variant.

7. **RUNNING FAILURE -** When a condition occurs that causes the EXNET® ring to lose frame synchronization, the ring state machine enters the RUNING FAILURE state to try and recover the ring.

   If the EXNET-ONE card is in the ERFT Ring Mode, the master node re-validates each link until the faulty one is found and looped out of the ring. If the faulty link cannot be found and the ring can not recover, the ring state machine is reset. The standby EXNET-ONE card monitors its node's packet information on the ring. If the active EXNET-ONE card is removed from the ring, the standby EXNET-ONE card will no longer see itself on the ring and switchover to the active EXNET-ONE card.

8. **FAIL -** An abnormal condition that causes the ring to reset induces failure processing to be performed. If the EXNET-ONE card cannot recover, or has failed too many times, it transitions to the FAIL state. Until the host brings the ring out of service or successfully runs diagnostics on the EXNET-ONE card, the EXNET-ONE card will not participate on the ring.

Figure 10-14 Ring Controller State Machine - High Level



The ring controller follows the basic state machine and shows primarily how the EXNET-ONE cards connect to the ring and how they recover from a failure described. Every time the EXNET-ONE card changes its ring state, the host is notified with the Ring Status Report (0x72). This API message notifies the host of the EXNET-ONE card's current status on the EXNET® ring. See Table 10-5 as an example.

Table 10-5 Ring Status Report (0x72)

API Name	API Number	Field(s)	Data
RING STATUS REPORT	0x72	Slot AIB	Slot Number
		Logical Ring ID	0x0 – 0x7
		Ring Status	0xFF – EXNET-ONE Not Connected To Ring 0x01 – EXNET-ONE Connecting to Ring 0x00 – EXNET-ONE Connected To Ring 0xFE – EXNET-ONE Failure
		More Status	See API
		Port A Status	0x00 – Open, 0x01 - Looped
		Port B Status	0x00 – Open, 0x01 - Looped
		Master Status	0x00 – Slave, 0x01 – Master
		Transmit Mode	0x00 – Rx Only, 0x01 – Rx/Tx 0xFF – Not Configure
		Source Packet Address	0x00 – 0x1F, 0xFF Unassigned
		Master Arbitration Flag	0x00 – Not Master Configurable 0x01 – Master Configurable
		Redundancy State	0xFF – Not Configured 0x00 – No Redundancy 0x01 – Active EXNET-ONE 0x02 – Slave EXNET-ONE
		EXNET Ring Mode	0x00 – Not Used 0x01 – ERFT Ring Mode

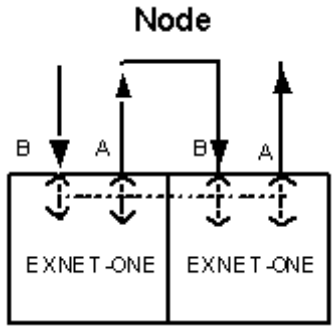
A *Ring Status Report (0x72)* message is also sent whenever the port status or configuration data changes. The Ring Status Report message, however, does not provide the host service state, which node is the master node, and the number of packets. The More Status field along with the Ring Status can be used to determine the current state of the EXNET-ONE card.

SRR Configuration

Follow the steps below to set up the Single Ring Redundancy (SRR):

- Install the second EXNET-ONE card hardware.
- Connect the two EXNET-ONE cards as shown in Figure 10-15.
- For further information, see the *Expanding the EXNET Ring (6-38)* for instructions on configuring an expanded switching system.

Figure 10-15 Single Ring Redundancy EXNET-ONE card Connections



The SRR feature is enabled by assigning two EXNET-ONE cards to the same Ring ID (API 0x74, Entity 0x01). The first EXNET-ONE card assigned to the ring takes the role of the active EXNET-ONE card; the second EXNET-ONE card assigned to the ring takes the role of the standby card. All subsequent ring configuration messages are applied to both EXNET-ONE cards assigned to the ring, allowing the node to support SRR configurations with little if any host intervention.

No Redundancy (0x00)

Until a redundant EXNET-ONE card is assigned to the ring, the EXNET-ONE card is in the active state (non-SRR). The active EXNET-ONE card knows that it is not in a redundant configuration.

Active (0x01)

Once a redundant EXNET-ONE card is assigned to the ring, one EXNET-ONE card takes the active state and the other, the standby state. The active EXNET-ONE card still acknowledges a non-redundant configuration, but indicates an awareness of the standby EXNET-ONE card.

Standby (0x02) The standby EXNET-ONE card, although configured identically as the active EXNET-ONE card, does not transmit on the ring or local PCM bus.

The standby EXNET-ONE card:

- Operates in the Receive Only mode. The participates in initializing the ring as any other Receive Only slave, receiving and processing all of the data from the ring.
- Passes data through from one adjacent EXNET-ONE card to the other.
- Will not intentionally bring down the ring. If the ring is already up, the standby EXNET-ONE card will wait until it can be passively added to the ring.

Undefined (0xFF) This is a transition state. When a EXNET-ONE card is brought in service (not the ring), the Matrix Controller notifies the EXNET-ONE card of its redundancy state. For the case where the active EXNET-ONE card resets, a switchover is performed. When the EXNET-ONE card comes back up, it does not know if its current redundancy state is valid. For cases where the EXNET-ONE card resets and was previously defined as either active or standby, it updates its redundancy state to Undefined, waiting until the Matrix Controller updates it. While in this state, the EXNET-ONE card will not participate on the ring. The host should never see a EXNET-ONE card in this state.

The SRR feature is enabled when the host configures a second EXNET-ONE card to a ring. This causes the Matrix Controller to configure a second EXNET-ONE card as the redundant standby and notifies the other EXNET-ONE card that it is active in a redundant configuration.

The Matrix Controller downloads the configuration and connection data of the active EXNET-ONE card to the standby EXNET-ONE card, therefore a EXNET-ONE card can be introduced at anytime. Either before the active EXNET-ONE card is fully configured, halfway in between, afterwards, or even after the ring is brought in service and calls are established. The standby EXNET-ONE card automatically synchronizes itself with the active EXNET-ONE card.

Once the standby EXNET-ONE card is assigned to a ring, all subsequent messages identified by the Ring ID are forwarded to all EXNET-ONE cards assigned to the ring.

Except for assigning the EXNET-ONE card to the ring (*EXNET Ring Configure* (0x74), Entity Assign Ring), Passive Node Addition, and querying the ring (*Ring Status Query* 0x71), all ring configuration and maintenance messages can reference the Ring ID. From the position of the host, this allows the ring to be configured the same way whether it is in an SRR or non-SRR configuration. When passively adding the standby EXNET-ONE card or a SRR configured node, both addressing schemes are supported, slot-based, and ring-based. See *Passive Additions Process*. All messages addressed by the Ring ID are processed by all EXNET-ONE cards assigned to the ring. All messages addressed by the slot number are processed by the individual EXNET-ONE card, regardless of the ring.

The major impact the SRR plays on the host is the increase of CSP solicited *Ring Status Report (0x72)* messages. Currently, the EXNET-ONE card sends a Ring Status Report to the host when the following conditions occur:

- The EXNET-ONE card sees the ring as out-of-service (ring status 0xFF)
- The EXNET-ONE card is connected to the ring (ring status 0x01)
- The EXNET-ONE card sees the ring as in service and valid data is being passed (ring status 0x00)
- The EXNET-ONE card detects a port status change (port loops back or opens)

The differences are as follows:

- A report will be sent from both EXNET-ONE cards (number of messages has doubled)
- A report is sent every time a EXNET-ONE card changes its redundancy state
- The redundancy state is supplied

The other impact that the SRR plays on the host is that the *Ring Status Query (0x71)* now provides the EXNET-ONE cards redundancy state.

Under normal conditions, the Ring Status Report (0x72) received from the standby EXNET-ONE card should match the active EXNET-ONE card. Some fields are common to the ring and some are specific to the EXNET-ONE card:

Ring Common Fields

- Logical Ring ID
- Ring Status

- More Status
- Master Status
- Transmit Mode
- Source Packet Mode
- Master Arbitration Flag
- Ring Mode

EXNET-ONE Card Specific Fields

- Slot Number
- EXNET Port A Status
- EXNET Port B Status
- Redundancy State

If the Ring Common Fields deviate, it indicates that something is abnormal. The only normal deviation is when a standby is introduced while the current status of the ring is in service. For this case, the ring status of the standby EXNET-ONE card should indicate that it is out-of-service. This should trigger the host to passively add to the standby EXNET-ONE card to the ring or display a notice to the operator.

Fault Detection and Switchover

CSP-Initiated Switchovers

The CSP initiates a switchover when any of the following occurs:

- The active EXNET-ONE card detects its own failure while the ring is in service.
- The standby EXNET-ONE card no longer detects its node's packet information.
- If the active EXNET-ONE card gets isolated due to a link re-validation.

When this occurs, the host is notified with a Ring Status Report (0x72) and Alarm (0xB9) indicating that the active EXNET-ONE card is out-of-service and is not host-initiated. The possible failures, as indicated by the More Status field, are as follows:

- (0xFF) Unknown Failure (Bad Ring)
- (0xFC) Internal Error (Bad Receive Matrix)
- (0xFB) Isolated From Ring
- (0xFA) Internal Diagnostics Failed
- (0xF9) Time-out During Ring Initialization
- (0xF7) Ring Mastership Configuration Change
- (0xF6) Unrecoverable Failure (Looped Out)
- (0xF5) Unsupported Ring Mode

When any of the above failures occur the CSP performs a switchover as described below:

- The active EXNET-ONE card is looped off of the ring, disabling write access to the ring and the local PCM bus. The standby EXNET-ONE card takes over the active role, maintaining the node's presence on the ring, including all calls established. Internal diagnostics are run on the "Looped Out" EXNET-ONE card. If the diagnostics fail, the EXNET-ONE card will stay in the "Looped Out" state. If diagnostics pass, the EXNET-ONE card will go to the standby state. Since the EXNET-ONE card is looped off of the ring, it can be replaced with another EXNET-ONE card, removed from the system, or, if it is determined that the EXNET-ONE card is still operational, passively added without affecting the ring state. This is also true for the standby EXNET-ONE card, if it unexpectedly removed from the ring.

- The switchover logic for an unexpected failure of the active EXNET-ONE card is handled differently for the master node than it is for a slave node. For the condition where the active slave node fails, the standby node takes over 50 milliseconds after it detects that its node's packets are no longer being transmitted on the ring. When the active master node fails, the ring looses Frame 125 Sync, resulting in the loss of all data on the ring. The standby master node switches over as master if the ring has not come back into service within 50 milliseconds. Once the switchover has occurred, the new master EXNET-ONE card revalidates each node, which takes about 20 milliseconds per node. As each node is added back to the ring, data between those nodes is allowed to pass again.

Host-Initiated Switchovers

The host can initiate a switchover in the following ways:

- The host sends a *Line Card Switchover* message (0x24) specifying the originating slot as the active EXNET-ONE card and the destination slot as the standby EXNET-ONE card. If both EXNET-ONE cards are assigned to the same Ring ID, a switchover is performed, where the active EXNET-ONE card goes to the standby state and the standby EXNET-ONE card goes to the active state. The switchover is synchronized between the two EXNET-ONE cards to minimize down time. The standby EXNET-ONE card takes over as active EXNET-ONE card within 10 milliseconds, usually around 2 milliseconds.
- The host de-assigns by removing, resetting, or taking the active EXNET-ONE card out-of-service. When this occurs, the standby EXNET-ONE card automatically switches to the active state, maintaining its node's presence on the ring.

Important! Removing the active slave EXNET-ONE card in this manner is treated as an unexpected failure, resulting in a switchover after 50 milliseconds or more.

Passive Addition Process

Passive Addition of a Standby EXNET-ONE Card

Passive addition has been enhanced and simplified for SRR2. While still compatible with the old procedure, the following two commands are needed:

1. Prepare for Addition – This prevents the EXNET-ONE card from trying to get added to the ring.
2. Add Node – Once the EXNET-ONE card is configured and ready to participate on the ring, the node is added to the ring after issuing this command.

During EXNET-ONE card passive addition, each port is individually added, validating the physical, link, and data layer of the protocol stack. If either layer fails to validate, the port is looped back and not added to the ring. This allows for future enhancement, including automated passive addition and port expansion.

Normally, the Passive Addition Command entity messages are addressed by specifying the Ring ID. In a SRR configuration, these messages will get sent to both the active and standby EXNET-ONE cards. This process works when adding an entire node to a ring or adding another node because both the active and standby EXNET-ONE cards process the same commands. When passively adding the standby EXNET-ONE card, the active EXNET-ONE card is playing the role of adjacent node and the standby EXNET-ONE card is playing the role of the passively added node, so both EXNET-ONE cards process different passive addition commands. When passively adding the standby EXNET-ONE card, the host needs the ability to send the EXNET Ring Configure (0x74), Entity messages directly to the standby EXNET-ONE card, based off of its Slot Number, not the Ring ID. The EXNET Ring Configure (0x74), Entity messages now supports the slot AIB (Address type 0x01) or the Ring ID AIB (Address type 0x1A) for the following entities:

- EXNET-ONE card Diags (0x04)
- Prepare For Addition (0x05)
- Expand Ring (0x06)
- Loop Back Port (0x07)
- Add Node (0x08)

The SRR2 Passive Addition Process is backward compatible with the existing (CSP Software Release 5.5) non-SRR2 Passive Addition Process.

For the host to passively add a standby EXNET-ONE card to an operational ring, the original procedure would be:

1. Assign a redundant EXNET-ONE card to the desired Ring ID. The EXNET-ONE card will be placed in the standby state. Since the ring is operational, the standby EXNET-ONE card waits for the host to passively add it or bring down the ring.
2. Send a Loop Back Port (0x07) entity command to the active EXNET-ONE card (or the EXNET-ONE card with “A” port will be looped back).
3. Send a Prepare For Addition (0x05) entity command to the standby EXNET-ONE card.
4. Physically connect the standby EXNET-ONE card to the ring.
5. Send an Expand Ring (0x06) entity command to the active EXNET-ONE card.
6. Send an Add Node (0x08) entity command to the standby EXNET-ONE card.

A node can also be added using these simplified procedures:

1. Assign the EXNET-ONE card a Ring ID. If this is a standby EXNET-ONE card, it will automatically inherit the configuration of the active EXNET-ONE card.
2. Send a Prepare For Addition (0x05) entity command to the slot of the EXNET-ONE card to be Passively Added.
3. If not already configured, configure the Transmit Mode and Number of Transmit Packets.
4. Physically connect the EXNET-ONE card to the ring.
5. Send an Add Node (0x08) entity command to the EXNET-ONE card.

Important! If you are passively adding both the active EXNET-ONE card and the standby EXNET-ONE card, we recommend that each EXNET-ONE card is added one at a time, obviously starting with the active EXNET-ONE card (if you passively add the standby first, it will switchover to active once on the ring), followed by the standby.

Diagnostics

The internal loop back diagnostics run automatically upon card power-up and when the host brings the ring in and out-of-service. The internal loop back diagnostic performs the following function: It configures the EXNET-ONE card as the master node, bringing the ring up while both ports are looped back. The test passes when a canned frame is validated on the ring. This test does not test the fiber drivers. To validate the external fiber drivers, the external loopback test needs to be performed. See the *API Reference* for the messages indicated below.

If the internal diagnostics fail, the CSP notifies the host by sending a major Alarm (0xB9). The Alarm Type field, Card (0x02), indicates the following message:

- *Internal Diagnostics Failure (0x1B).*

If the EXNET-ONE card is assigned to a ring, Ring Status Report (0x72) is also sent. The More Status field indicates the following message:

- *Internal Diagnostics Failed (0xFA)*

Adding a Redundant EXNET® Ring

Adding a redundant EXNET® ring to an CSP has no effect on calls that are connected or that are setting up. Use the following procedure to add a second EXNET® ring.

1. Add an EXNET-ONE card to each node
2. Connect the EXNET® cable to each node the same way the active ring is connected
3. Configure the EXNET® ring (*EXNET Ring Configure*)
4. Assign a unique Logical Ring ID and configure the Transmit/ Receive Mode
5. Bring the ring in service (*Service State Configure*)

Adding a Line Card

Adding a line card to a CSP has no effect on traffic, but note that timeslots used for conferencing may be affected. You can add a redundant Matrix Controller card to a node with no effect on local or distributed traffic. However, in the event of an Matrix Controller switchover, some calls setting up on the node may be dropped.

EXNET® Ring Timing

This section explains how to configure the system software for the EXNET® Ring Timing Enhancement.

Overview

The EXNET® Ring Timing Enhancement provides the CSP with following features:

- CSP nodes can select higher quality timing sources.
- CSP nodes can derive loop timing from the EXNET® Ring.
- CSP nodes can be configured for each ring, to determine whether they will arbitrate to become the EXNET® Ring Master. By default, CSP 1000 and CSP 2000 nodes arbitrate, while EXNET-ONE nodes do not arbitrate.
- CSP nodes can optionally be configured as the master of multiple EXNET® Rings.
- CSP nodes can be configured so that if they are configured to derive their loop timing from the EXNET® Ring, they will be prevented from arbitrating for EXNET® Ring Master.

Terminology

EXNET® Ring Master is the CSP node that supplies timing to the EXNET® Ring.

CSP refers to both single and multi-node configurations, and encompasses CSP 2040 and CSP 2090 models.

Master-configurable refers to any CSP node that can serve as the EXNET® Ring Master.

Network Synchronization / Prioritizing Clock Source

EXNET® Ring Timing Enhancement adds the EXNET® Ring as a valid timing source for Loop Timing 1 and Loop Timing 2. The *Loop Timing Configure* message is used to identify the slot of the EXNET-ONE card.

The CSP continuously monitors all synchronization clock sources and uses the highest priority clock available. The host determines the current clock source from the Current Synchronization Mode byte in the *Synchronization Priority List Query* message.

When the highest priority source is unavailable, the second highest priority source takes its place, and the Matrix Controller sends a Clock Mode Switched Alarm message. If neither of the reference clock

sources are available, the Matrix Controller also sends a Reference Clock Source Lost Alarm. If the higher priority clock source resumes, the Matrix Controller automatically switches back to it, and sends the Clock Mode Switched Alarm to notify the host.

CSP Node Configuration

Some network conditions affect how you configure CSP nodes for Ring Timing Enhancement. Consider these conditions before considering the details of timing and arbitration:

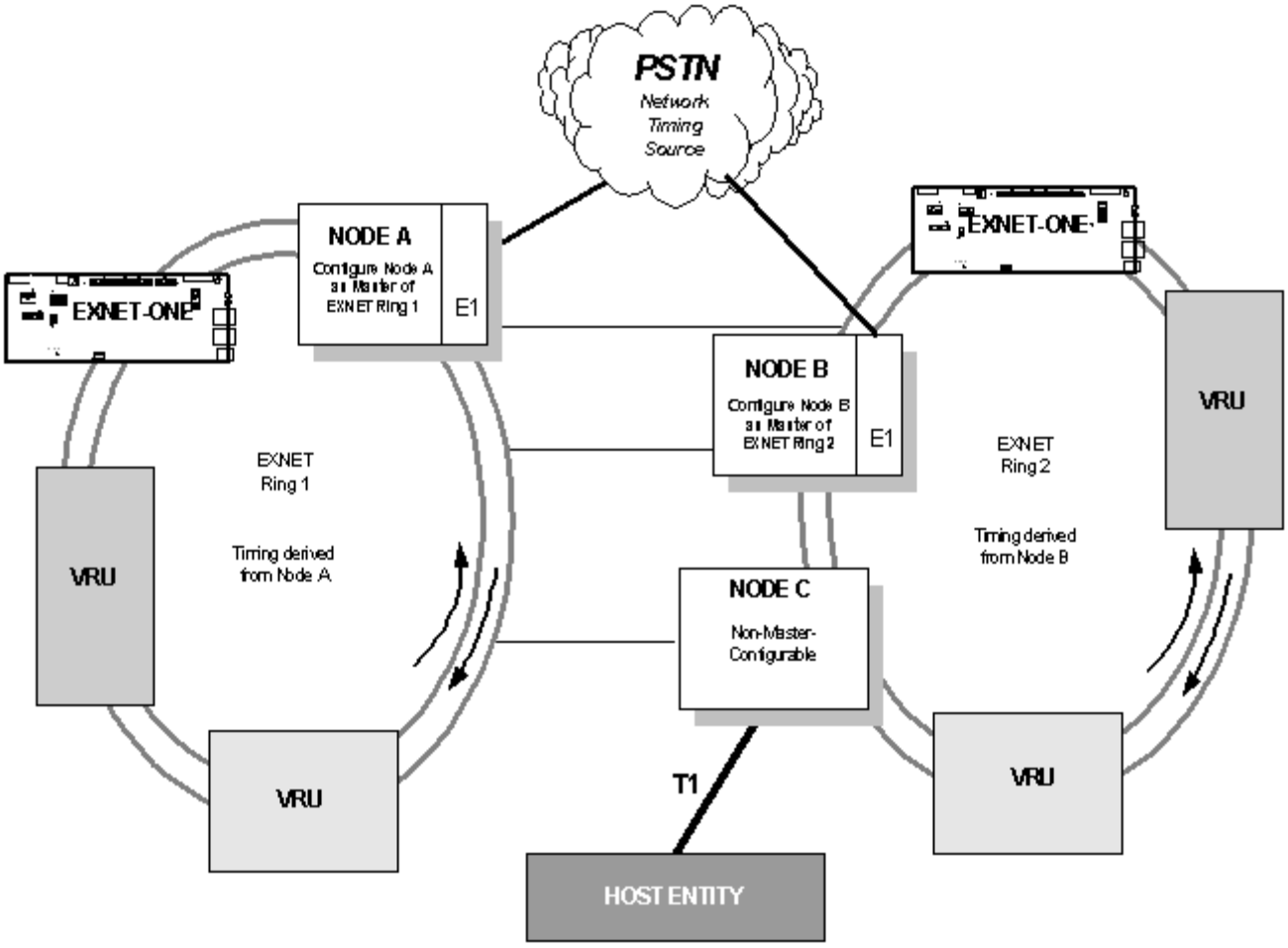
- Any CSP node that is to serve as an EXNET® Ring Master that provides timing should have a hardware line, which can be T1, E1, J1, or an External Reference Clock.
- The host should configure all CSP nodes that can derive loop timing from the network (PSTN or an External Reference Clock) to do so. These nodes should derive their secondary loop timing from the network as well.
- The host should configure all CSP nodes that cannot derive loop timing from the network to derive their loop timing from the EXNET® Ring.
- With redundant EXNET® Rings, the host should configure all CSP nodes that cannot derive their loop timing from the network to derive their secondary loop timing from the second EXNET-ONE card.
- The host should configure all CSP nodes to be excluded from artbitrating for EXNET® Ring Master if they:
  - derive loop timing from an EXNET-ONE card.
  - use the Free-running clock.

Ring Timing and Arbitration

All CSP nodes that are master-configurable can arbitrate to become the EXNET® Ring Master.

In Figure 10-16, only Nodes A and B can arbitrate to become EXNET® Ring Master. They were configured to have a high quality timing source (the PSTN). During EXNET® Ring initialization, Node A successfully arbitrated to become EXNET® Ring Master for Ring 1, and Node B successfully arbitrated to become EXNET® Ring Master for Ring 2. If multiple CSP nodes on a ring are configured to arbitrate to become the EXNET® Ring Master, the node that initializes and comes in service first becomes the EXNET® Ring Master.

Figure 10-16 CSP Master Nodes Connected to PSTN providing timing to EXNET® Rings



When all CSP nodes have received timing from the EXNET® Ring Master, EXNET® Ring 1 and EXNET® Ring 2 serve as the primary clock sources for the EXNET-ONE card and for the Voice Response Units (VRUs) as shown in the diagram.

CSP Ring Timing Enhancement

CSP Ring Timing Enhancement allows all EXNET® devices to be master-configurable, although by default they are not master-configurable.

Considerations

EXNET® Ring Master Arbitration

EXNET® Ring Timing Enhancement provides some control of the arbitration process with these configuration options:

- You can configure an CSP node as not available to arbitrate to become EXNET® Ring Master
- You can allow an CSP node that is already the Master of one EXNET® Ring to arbitrate to become Master of other EXNET® Rings, depending upon the type of configuration

When an EXNET® Ring is set up, all master-configurable CSP nodes on the ring arbitrate equally to become master of the EXNET® Ring.

Master of Multiple Rings

Node C is unavailable because it has been configured so as not to function as an EXNET® Ring Master. In this scenario, only Node A remains as an EXNET® Ring Master. To ensure that both rings can remain in service, you should configure both Node A and Node B as master-configurable for both rings.

Configuration

To configure EXNET® Ring Timing Enhancement, perform the following procedures:

1. Configure Selection of EXNET® Ring Master Candidates.
2. De-select an CSP node as master-configurable if it is a poor candidate for EXNET® Ring Master, using the *EXNET Ring Configure* message.
3. Select an CSP node as master-configurable to allow an EXNET to arbitrate for EXNET Ring Master, using the *EXNET Ring Configure* message.

   Do not send *EXNET Ring Configure* to Configure Selection of EXNET® Ring Master Candidates. Instead, send *EXS Node Configure* to configure the EXNET® Ring Master on multiple rings.
4. Designate an EXNET® Ring as a valid timing source, with the *Loop Timing Configure* message.

Configure the slot number of the EXNET-ONE card with the *Loop Timing Configure* message. The span offset byte in the message is ignored.

Frequently Asked Questions

Q1 Does a ring require an EXNET® Ring Master?

A1 Yes, every ring must be assigned an EXNET® Ring Master

Q2 Why does the ring require an EXNET® Ring Master?

A2 EXNET® Ring Master nodes drive the timing for the packets on the ring.

Q3 How do you find out which CSP node is the EXNET® Ring Master?

A3 Use the *Ring Status Query* message to identify which CSP node is the EXNET® Ring Master.

Q4 How is the EXNET® Ring Master determined?

A4 During ring initialization, the CSP nodes arbitrate to determine which becomes the EXNET® Ring Master.

Q5 Can an CSP node be Master of more than one EXNET® Ring?

A5 Yes. The default setting is for a node to be Master of only one EXNET® Ring, but the CSP Ring Timing Enhancement allows you to change this default.

Q6 Is there a relationship between the number of rings and the number of CSP nodes in an CSP network?

A6 There is no defined relationship. In a network there can be fewer rings than nodes, more rings than nodes, or the same number of rings as nodes.

Q7 What happens when the CSP node that provides the ring timing fails, and another CSP node on the ring has been configured to be master-configurable?

A7 The ring goes down and all master-configurable nodes arbitrate to become EXNET® Ring Master.

Q8 What happens when the CSP node providing the ring timing fails, and there IS NOT another CSP node on the ring that has been configured as an EXNET® Ring Master?

A8 The ring goes down and remains down until a master-configurable CSP node becomes available.

Q9 What API messages support the changes required for the CSP Ring Timing Enhancement?

A9 The following API messages support the changes: *EXNET Ring Configure* (0x74), *Ring Status Query* (0x71), *Ring Status Report* (0x72), *EXS Node Configure* (0x7F), *EXS Node Configuration Query* (0x7E), *Loop Timing Configure* (0x4A).

Q10 When should I configure an CSP node to be non-master configurable?

A10 When it or its timing source is considered too unreliable to drive the timing for the EXNET® Ring (for example, nodes with a free running clock, with loop timing from a non-network connected source, nodes frequently taken out of service, nodes not configured for redundancy).

Q11 Should I make all CSP nodes non-master-configurable except for the CSP node I want to become the EXNET® Ring Master?

A11 You should designate all CSP nodes with a quality clock source as master-configurable.

Q12 What happens if none of my active CSP nodes are master-configurable?

A12 No CSP nodes arbitrate to become EXNET® Ring Master, and the ring will not come up.

Q13 Why not configure one CSP node to derive its timing from the PSTN and have all other CSP nodes on that EXNET® Ring derive their timing from the ring?

A13 To ensure accurate synchronization, all CSP nodes connected to the PSTN should derive their timing from the PSTN. While deriving timing from the ring is superior to free-running, the less equipment between the original timing source and the CSP node, the better the synchronization.

Q14 Why not permit all CSP nodes on redundant EXNET® Rings to become EXNET® Ring Masters on both rings?

A14 Loss of an EXNET® Ring Master brings the ring down. If a Master of multiple EXNET® Rings comes out of service for any reason, all rings go down for which the node is Master.

11 Call Routing

Purpose The Router feature of Call Control supports diverse internal call routing algorithms with user-defined routing, translation tables, and resource group tables.

This chapter provides basic information on Call Routing. If you require more advanced information, please contact Dialogic Technical Support.

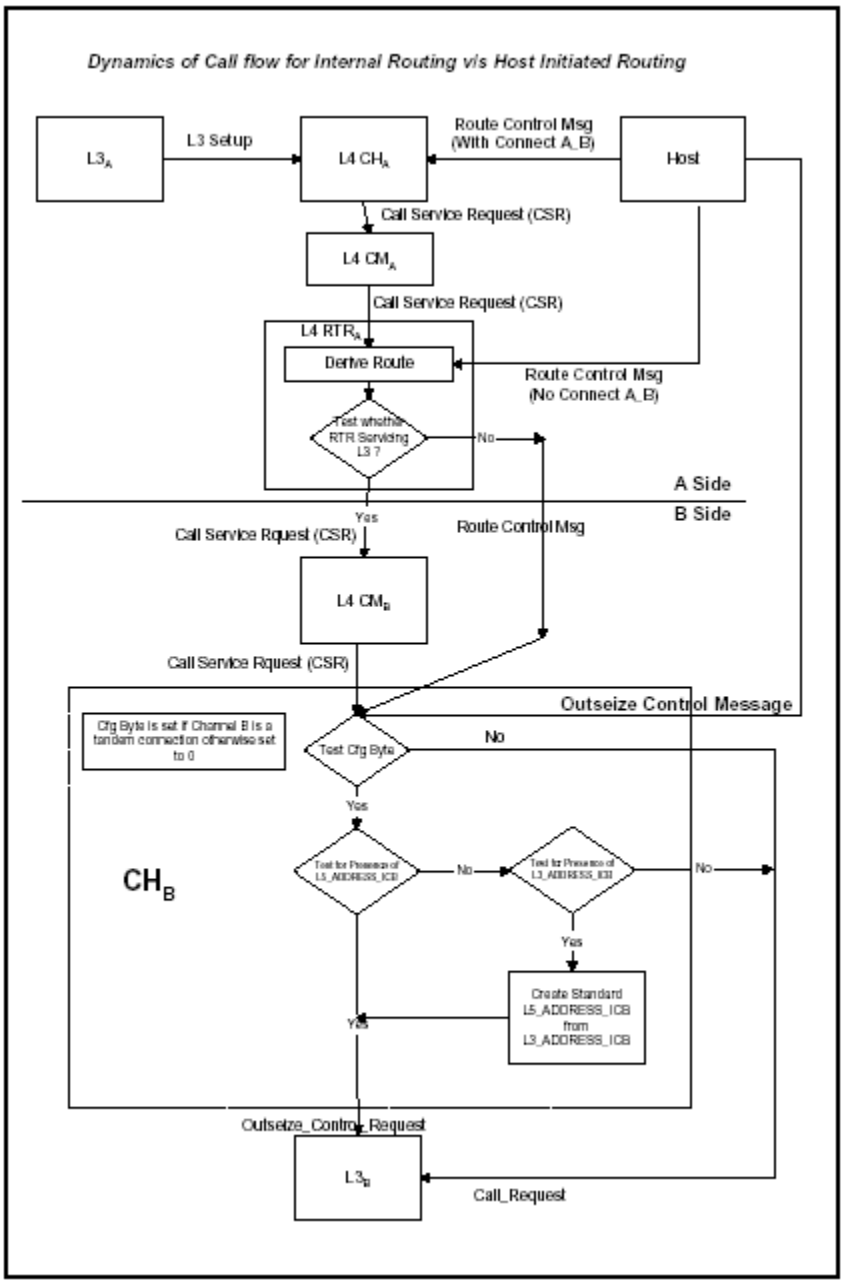
Introduction to Call Routing

You can configure the CSP to route calls without host intervention, and you can route calls interactively using the *Route Control* message. You use the Router Configuration Tool to create route tables that the CSP uses to route calls. Calls can be routed based upon the following:

- Incoming Channel
- Resource Group (Group of spans/channels)
- Digits (Called or Calling Party)
- Time of Day

Call Routing Operation

Figure 11-1 Routing Flow Chart



Incoming Call Handling

You can process incoming calls two ways:

- Send the *Request For Service* (or *Request for Service with Data*) message to the host This option is the system default.
- Use Internal Router

When the CH component receives a call, it sends a Call Service Request (CSR) to the CM component, which forwards it to the router. To configure the PPL Config Bytes of the CH component, you must use the Router Protocol and Route Method TLVs, along with other TLVs required for the specific route method.

The Route methods you can use include the following:

- Route Group (0x06)
- Criteria (0x08)
- Resource Group

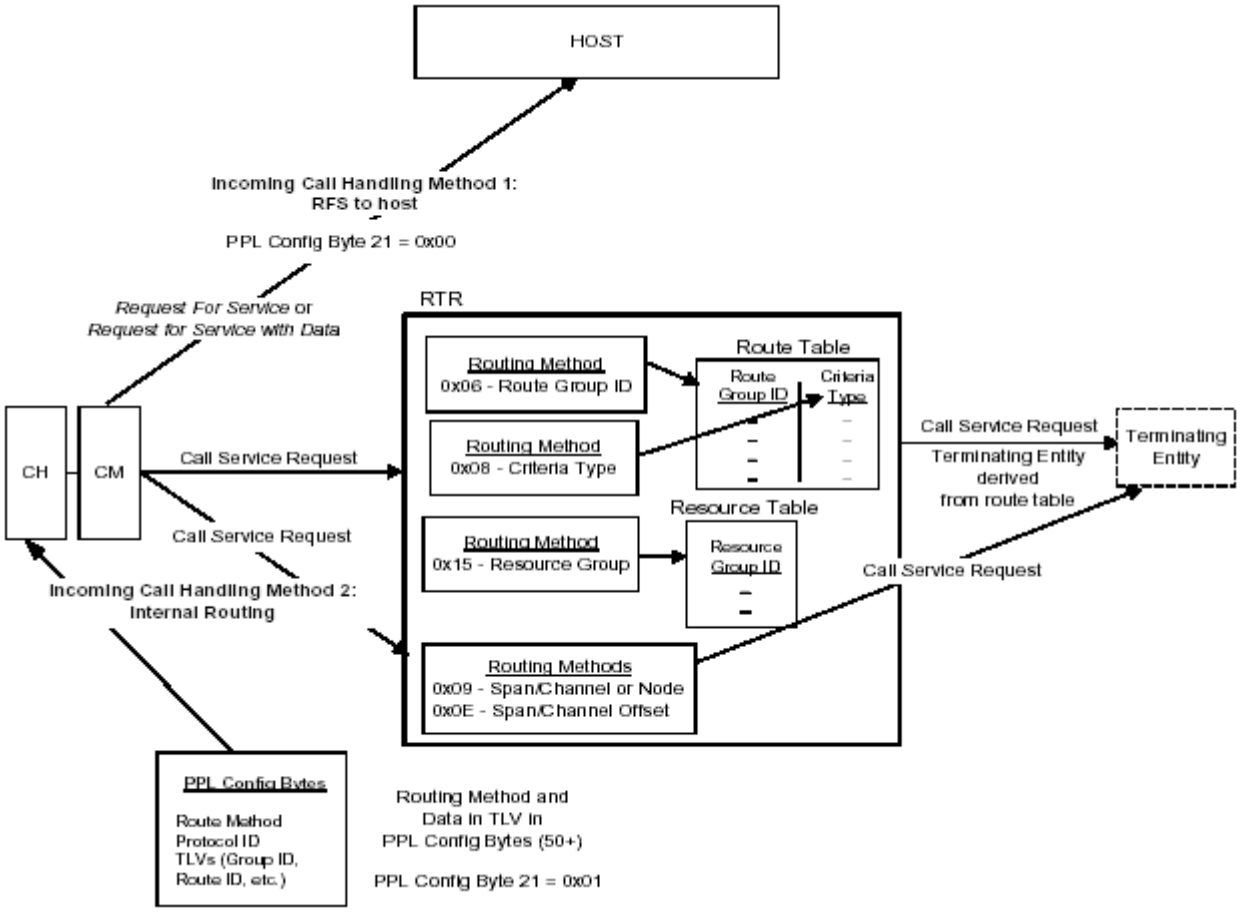
If you choose to use Route Group (0x06) or Criteria (0x08), the router uses the route table to find a terminating channel. If you use the Resource Group method, the Resource Group table determines the terminating channel. If you use any other route method, the CSR from the CH component indicates the terminating entity.

When the terminating entity is derived, a CSR is sent to the remote CH component, and a connection is made. If a CSR ACK is not received within 3 minutes, the switch sends a *Request For Service* message. This is CH Timer 8 (Router Call Service ACK Wait), and you can use the *PPL Timer Configure* message to configure it.

To enable internal routing, set PPL Config Byte 21 of the CH component to 0x01

There are two methods for handling incoming calls, shown in Figure 11-2.

Figure 11-2 Incoming Call Handling



Host-Initiated Outseizures

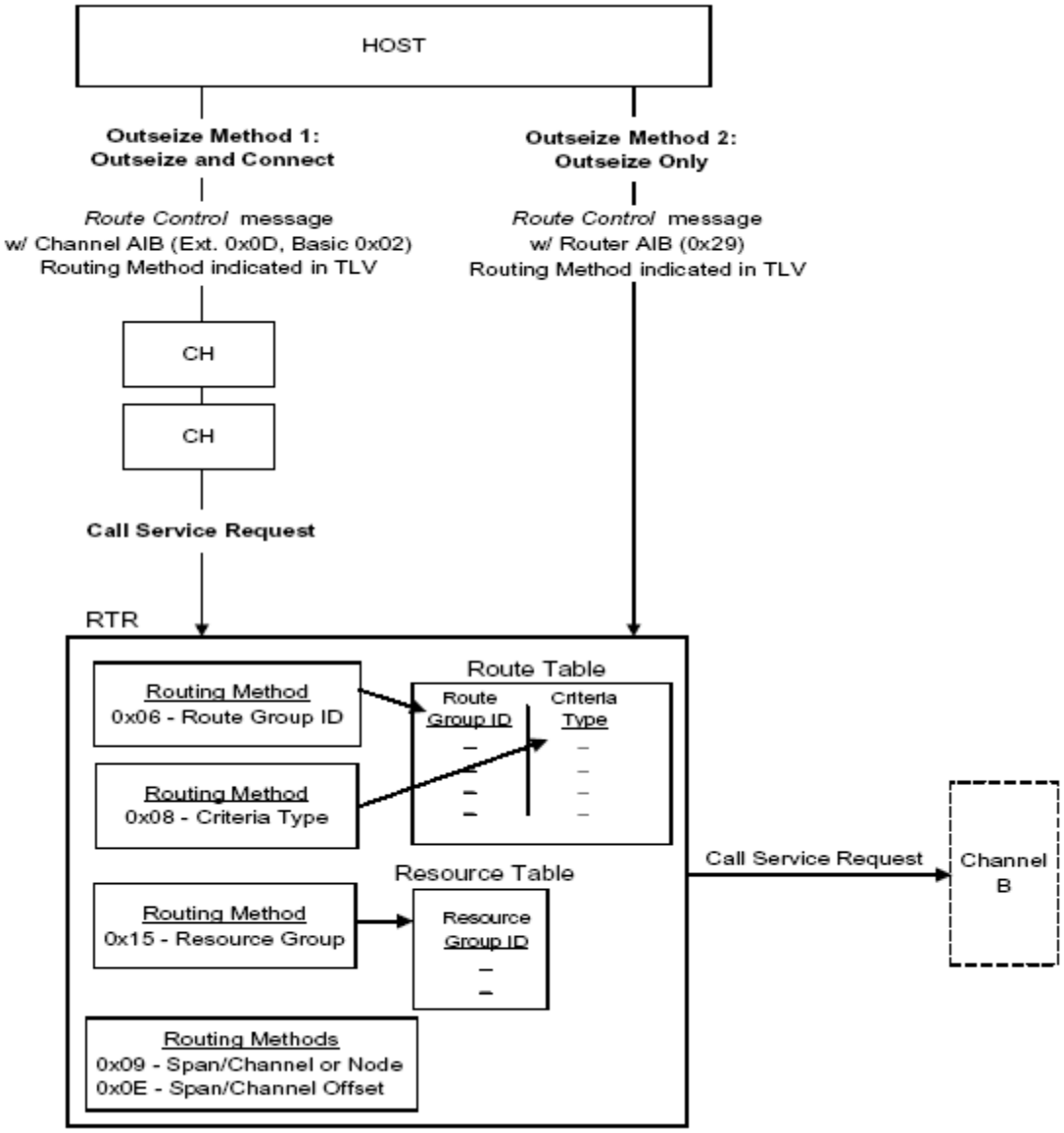
Use the *Route Control* message to initiate inseizures. Use TLVs to indicate the Route Method and associated data.

Where Channel A is known, use the Channel AIB. When a terminating channel is derived, the two channels are connected.

If Channel A is not known, use the Route AIB. When a terminating channel is derived, it is seized and no further action is taken by the switch.

> **Important!** For backward compatibility, you can also use the *Outseize Control* message if internal routing is not required. You specify the terminating channel in the message.

Figure 11-3 Host-Initiated Outseize Methods

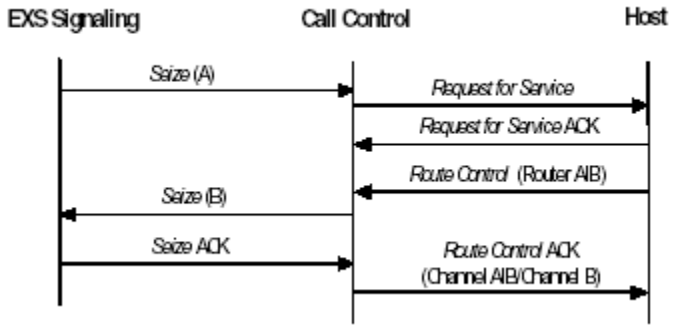


Call Routing Call Flows

The call flow diagrams that follow illustrate the two scenarios of host-initiated routing: Outseize Only and Outseize and Connect.

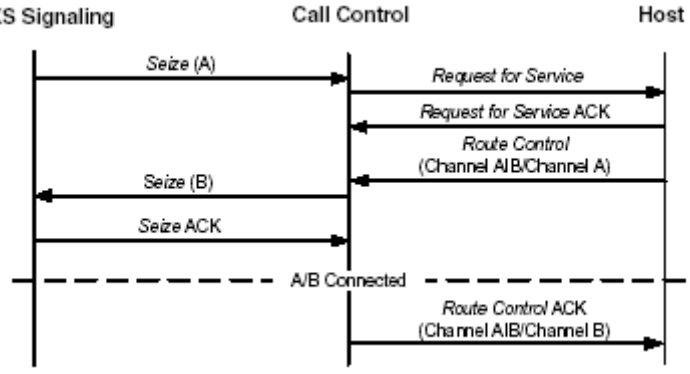
In Outseize Only scenarios, the router selects a channel, which is then seized. The CSP performs no further call setup.

Outseize Only.



In Outseize and Connect scenarios, the router selects a channel, which is seized and then connected to the specified channel.

Outseize and Connect



Outseize Modes for Host-Initiated Routing

You can use PPL Config Byte 20 to configure two outseize modes:

- Call Service Request (0x00)

This mode is the default. The CSP seizes the terminating channel, but takes no further action. This mode is usually used for terminating calls to a VRU or subscriber.

- Outseize (0x01)

The terminating channel is seized and digits or other information may be outpulsed. This mode is usually used when initiating or relaying a call.

You must configure outseize instructions for the channel and the *Route Control* message must contain digits and any other information required for outseizing, as noted in the message description format.

Outpulse Signal Type

If the Outseize Mode is set to Outseize (0x01), PPL Config Byte 8 indicates the outpulse signal type. MFR1 is the default. (Host does not send KP ST) (0x02).

0x01 - DTMF

0x02 - MFR1 (Host does not send KP ST)

0x03 - MFR2

0x04 - MFR1 (Host sends KP ST)

0x05 - Dial Pulse

Call Routing Scenarios

The diagrams in this section show the internal routing feature in various scenarios.

Single Node

The diagrams that follow show call routing for calls routed within a single node.

The three scenarios described are as follows:

- Successful Route Request
- Route found but rejected by Remote CM
- Route Request rejected by Router

Figure 11-4 shows the router finding a route, and the remote CM accepting the route request.

Figure 11-4 Successful Route Request

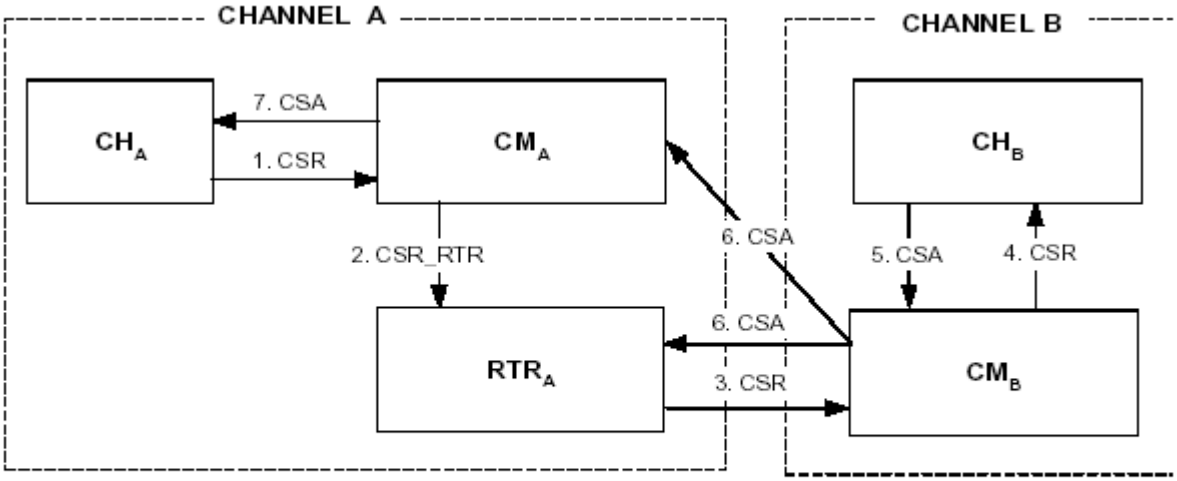


Figure 11-5 shows the router finding a route but the remote CM rejecting the route request.

Figure 11-5 Route Found but Rejected by Remote CM

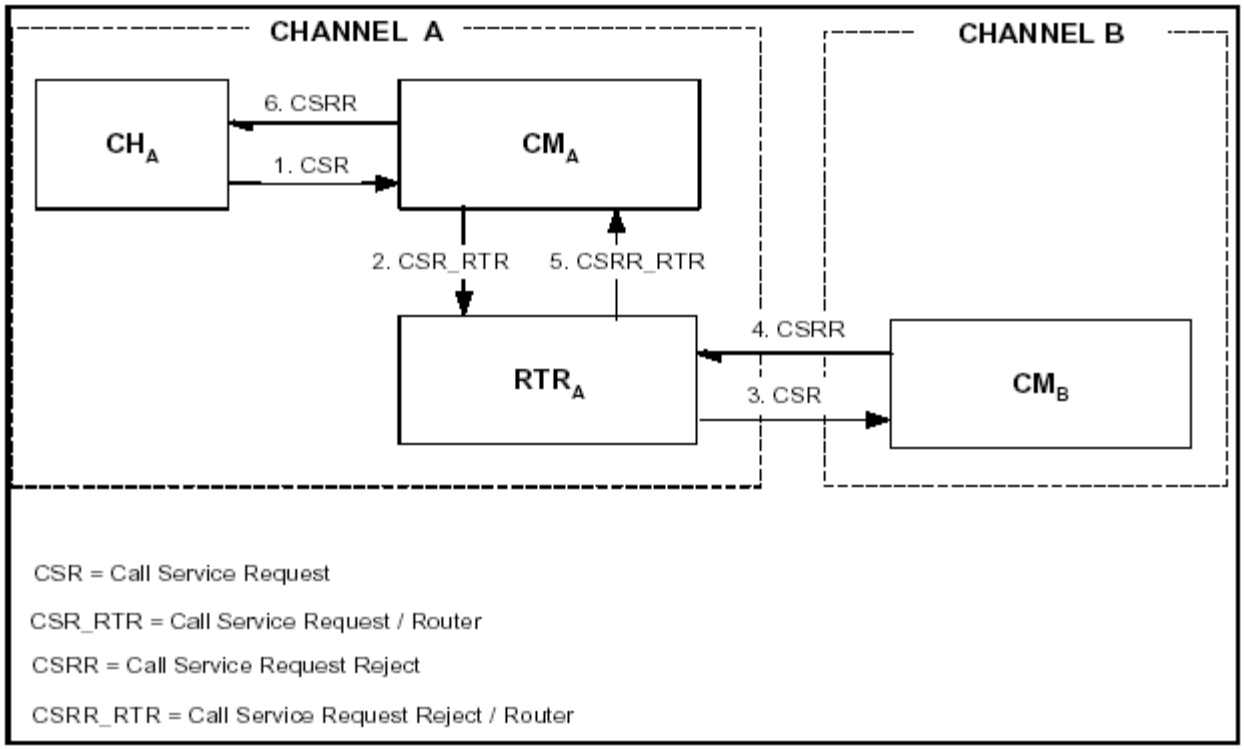
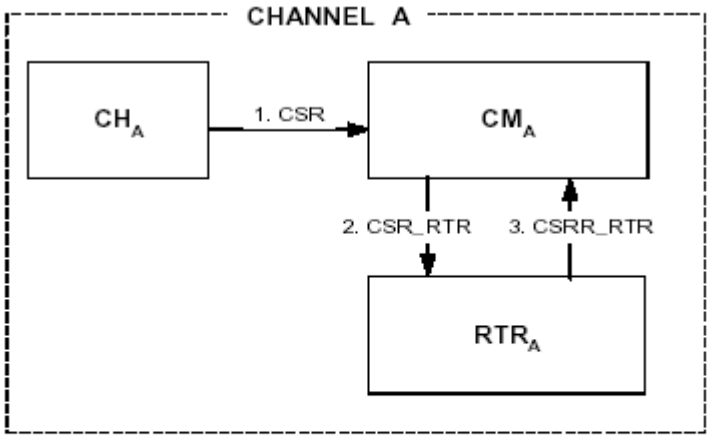


Figure 11-6 shows the router rejecting a route request from the local CM.

Figure 11-6 Route Request Rejected by Router



CSR = Call Service Request

CSR_RTR = Call Service Request / Router

CSRR_RTR = Call Service Request Reject / Router

Across Nodes

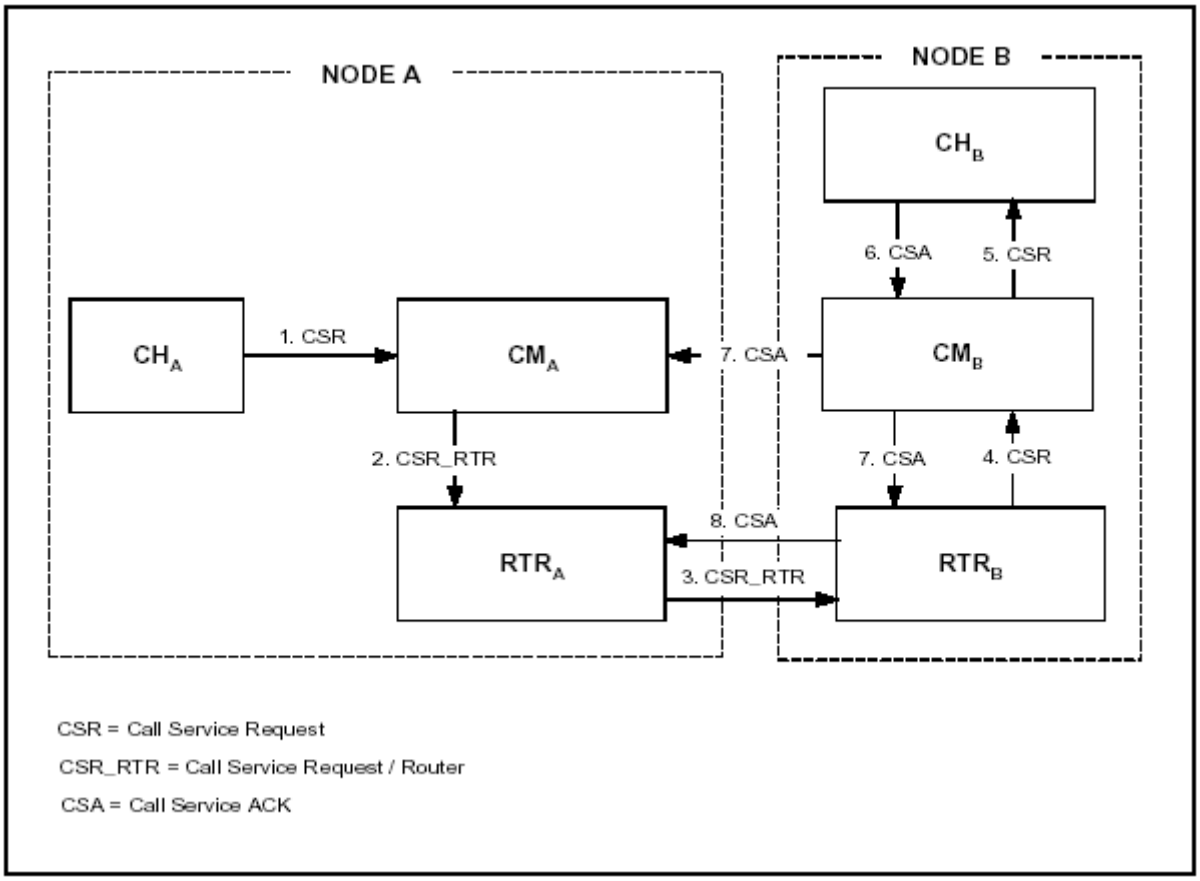
This diagrams in this section show calls routed across two nodes.

The two scenarios described are as follows:

- Successful route request
- Route found but rejected by remote node's CM

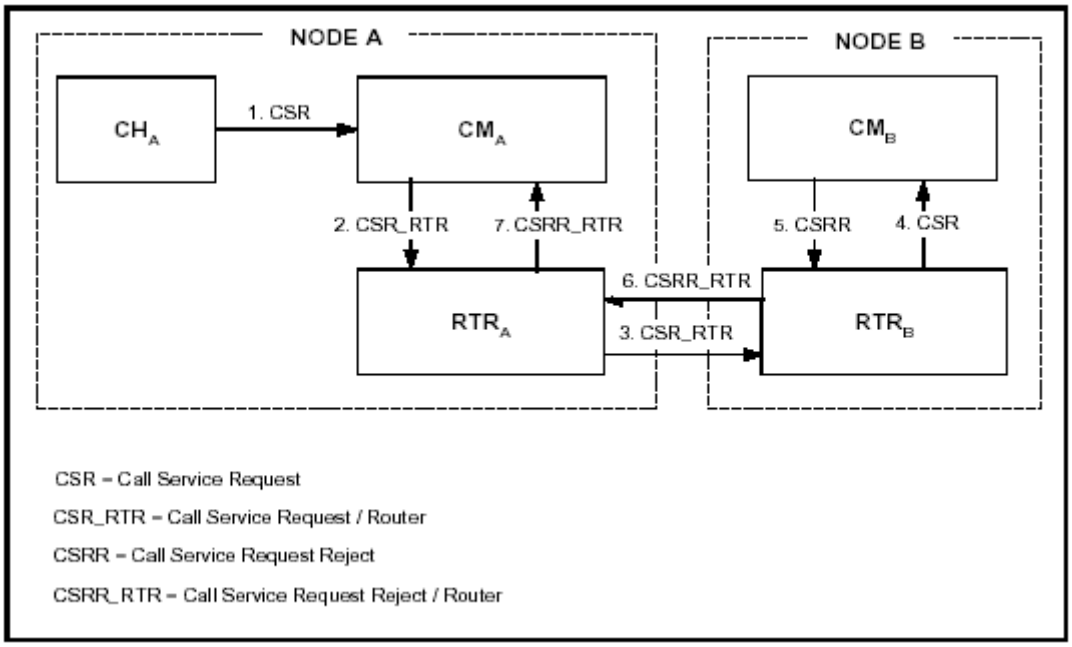
Figure 11-7 shows the router finding a route on a remote node and the remote CM accepting the route request.

Figure 11-7 Successful Route Request Between Two Nodes



The diagram in Figure 11-8 shows the router finding a route on a remote node, and the remote CM rejecting the route request.

Figure 11-8 Route Found on Remote Node but Rejected by Remote CM



Call Routing Configuration

Configuration for internal routing involves the following:

- Enabling internal routing
- Using TLVs

Enabling Internal Routing for Incoming Calls

To enable automatic internal routing for incoming calls, change PPL Config Byte 21 of the CH component (0x61) to 0x01. Upon receiving an incoming call, the CH component sends a CSR to the CM component, which passes it to the router. No *Request for Service* message is sent to the host.

Using TLVs

TLVs specify the router protocol and routing method, and pass data required by the router. You can use TLVs in two ways:

- For host-initiated outseizures using the internal router, TLVs are passed in the PPL Variable Data ICB (0x1E) in the *Route Control* message
- For handling incoming calls with the internal router, TLVs are preconfigured in the PPL Config Bytes of the CH component

The format of the TLVs is identical for each method, except that the TLVs used in the *Route Control* message have two bytes for the *Tag* and *Length* fields, while the TLVs in the PPL Config Bytes have one byte for each.

Whether in the *Route Control* message or in PPL Config Bytes, you must include the Router Protocol and Routing Method TLVs:

- Router Protocol (0x0F) - Mandatory

This TLV indicates the router component to be initiated. The default protocol is 0x0B. Currently, no router protocol variants are supported.

- Routing Method (0x13) - Mandatory

The following routing method options are supported:

0x06 - Route Group

0x08 - Criteria

0x09 - Terminating Channel or Node

0x10 - Incoming Resource Group

0x15 - Terminating Resource Group

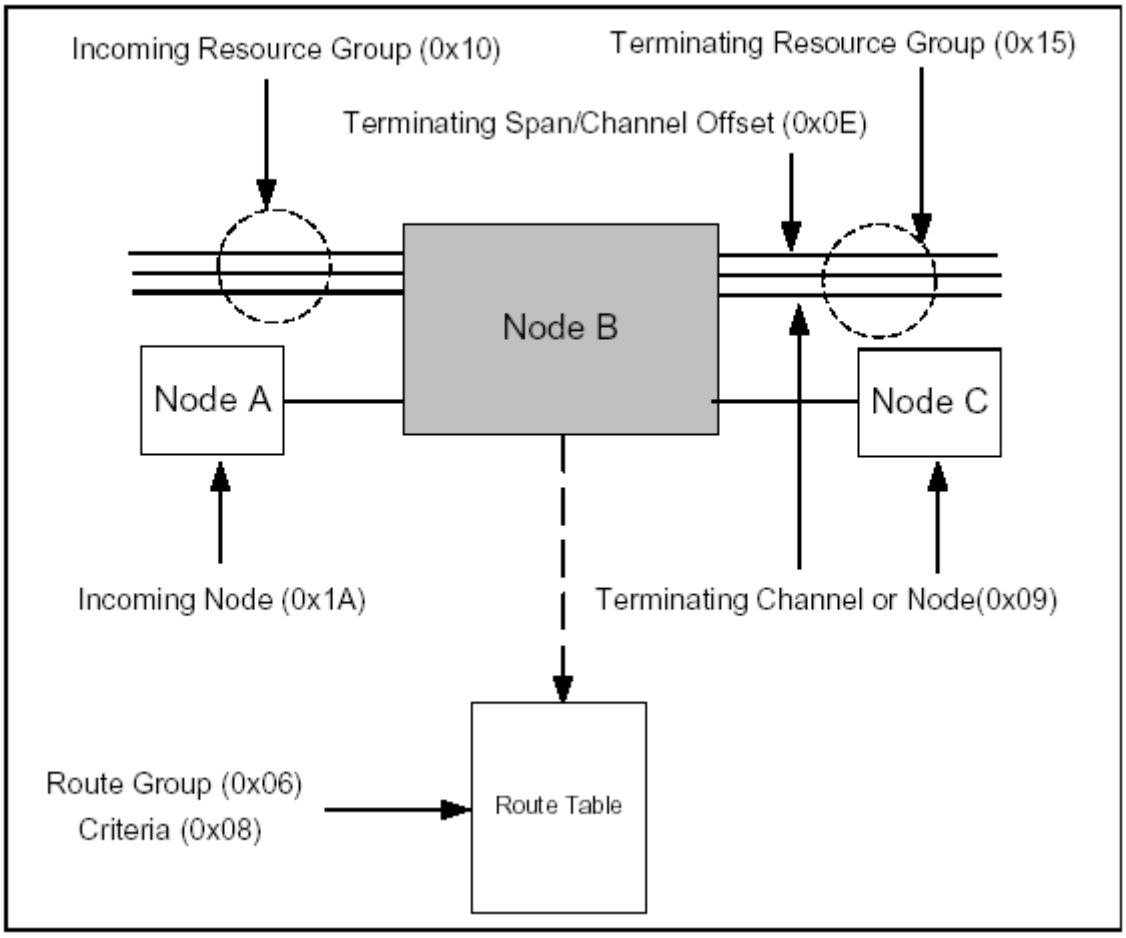
0x0E - Terminating Span/Channel Offset

0x1A - Incoming Node

0x19 - Multiple Resource Group

The next diagram shows the various routing methods in relation to the node handling the call (Node B). Only the Route Group and Criteria methods require the router to use the route table. For all other methods, the terminating entity is indicated to the router in a TLV.

Figure 11-9 Routing Methods



Depending on the route method used, you must include TLVs to provide required information to the router. The TLVs required for each routing method are as follows:

- Route Group (Method 0x06)

You must include the Route Group TLV (0x06) specifying the Route Group ID. To have the router match both a route group and criteria, include the Criteria with Route Group TLV (0x12). To specify multiple route groups, use the Multiple Route Groups TLV (0x0C).

- Criteria (Method 0x08)

You must include the Criteria Type TLV (0x08) specifying the criteria type to be used by the router, as well as an additional TLV defining the criteria data. The criteria options and the required TLVs for each are listed below:

For Address Digits criteria types, you must also include the Address Digits TLV (0x01–0x04):

- 0x01 - Address Digits 1 (Called Party)
- 0x02 - Address Digits 2 (Calling Party)
- 0x03 - Address Digits 3
- 0x04 - Address Digits 4

0x05 - Time of Day

- No additional TLV is required. The router derives the time of day from the system clock.

0x07 - Incoming Channel

- You must also include the Incoming Channel TLV (0x07).

0x10 - Incoming Resource Group

- You must also include the Incoming Resource Group TLV (0x10).

0x1A - Incoming Node

- You must also include the Incoming Node TLV (0x1A).

0x03E9–0x03F8 - User-defined

- To have the router match both a route group and criteria, use the Route Group TLV (0x06) and the Criteria with Route Group TLV (0x12).
- To specify multiple criteria, use the Multiple Criteria TLV (0x0D).

Terminating Channel or Node (Method 0x09)

- You must include the Terminating Channel or Node TLV (0x09) with an AIB indicating the terminating span/channel or node ID.

Incoming Resource Group (0x10)

- You must include the Incoming Resource Group TLV (0x10) indicating the incoming resource group ID.

Terminating Resource Group (Method 0x15)

- You must include the Terminating Resource Group TLV (0x15) indicating the terminating resource group ID.

Terminating Span/Channel Offset (0x0E)

- You must include the Span/Channel Offset TLV (0x0E) indicating the offset value.

Incoming Node (0x1A)

- You must include the Incoming Node TLV (0x1A) indicating the node ID.

Multiple Resource Group (0x19)

Sample Routing TLV Configurations

This section includes examples of PPL Config Byte configurations of the CH component required for various routing methods.

Terminating Channel Routing

The example in the table below shows the following PPL Config Byte configuration:

- Byte 20 configures the Outseize Method to Send Call Request to Layer 4 (0x00).
- Byte 21 configures the Routing Flag to Use Internal Router (0x01).
- TLV 1: Bytes 51–54 configure the Routing Method to Terminating Channel (0x09).
- TLV 2: Bytes 55–57 configure the Router Protocol ID to 0x0B, which is the default and the only protocol currently supported.
- TLV 3: Bytes 58–62 configure the value of the Terminating Span/ Channel.

Table 11-1 PPL Config Byte Configuration

BYTE	Description	Value
20	Outseize Method	0x00 - Send Call Request to L4
21	Routing Flag	0x01 - Use Internal Router
50	TLV Count	0x03
51	TLV 1 Tag	0x13 (Routing Method)
52	TLV 1 Data Length	0x02
53	TLV 1 Data[0] Routing Method (MSB)	0x00
54	TLV 1 Data[1] Routing Method (LSB)	0x09 (Terminating Channel or Node)
55	TLV 2 Tag	0x0F (Protocol ID)
56	TLV 2 Length	0x01
57	TLV 2 Data[0] Protocol ID	0x0B (Default)
58	TLV 3 Tag	0x09 (Terminating Channel or Node)

BYTE	Description	Value
59	TLV 3 Data Length	0x03
60	TLV 3 Data[0] Terminating Span MSB	
61	TLV 3 Data[1] Terminating Span LSB	
62	TLV 3 Data[2] Terminating Channel	

Terminating Resource Group Routing

The example in the table below shows the following PPL Config Byte configuration:

- Byte 20 configures the Outseize Method to Send Call Request to Layer 4 (0x00).
- Byte 21 configures the Routing Flag to Use Internal Router (0x01).
- TLV 1: Bytes 51–54 configure the Routing Method to Terminating Resource Group (0x15).
- TLV 2: Bytes 55–57 configure the Router Protocol ID to 0x0B, which is the default and the only protocol currently supported.
- TLV 3: Bytes 58–61 configure the value of the Terminating Resource Group ID.

Table 11-2 PPL Config Bytes

BYTE	Description	Value
20	Outseize Method	0x00 - Send Call Request to L4
21	Routing Flag	0x01 - Use Internal Router
50	TLV Count	0x03
51	TLV 1 Tag	0x13 (Routing Method)
52	TLV 1 Data Length	0x02
53	TLV 1 Data[0] Routing Method (MSB)	0x00
54	TLV 1 Data[1] Routing Method (LSB)	0x15 (Terminating Resource Group)
55	TLV 2 Tag	0x0F (Protocol ID)
56	TLV 2 Length	0x01
57	TLV 2 Data[0] Protocol ID	0x0B (Default)
58	TLV 3 Tag	0x15 (Terminating Resource Group)
59	TLV 3 Data Length	0x02
60	TLV 3 Data[0] Resource Group MSB	
61	TLV 3 Data[1] Resource Group LSB	

Terminating Span/Channel Offset Routing

The example in the table below shows the following PPL Config Byte configuration:

- Byte 20 configures the Outseize Method to Send Call Request to Layer 4 (0x00).
- Byte 21 configures the Routing Flag to Use Internal Router (0x01)
- TLV 1: Bytes 51–54 configure the Routing Method to Span/ Channel Offset (0x0B).
- TLV 2: Bytes 55–57 configure the Router Protocol ID to 0x0B, which is the default and the only protocol currently supported.
- TLV 3: Bytes 58–62 configure the value of the Terminating Span/ Channel Offset.

Table 11-3 PPL Config Byte Configuration

BYTE	Description	Value
20	Outseize Method	0x00 - Send Call Request to L4
21	Routing Flag	0x01 - Use Internal Router
50	TLV Count	0x03
51	TLV 1 Tag	0x13 (Routing Method)
52	TLV 1 Data Length	0x02
53	TLV 1 Data[0] Routing Method (MSB)	0x00
54	TLV 1 Data[1] Routing Method (LSB)	0x09 (Terminating Channel)
55	TLV 2 Tag	0x0F (Protocol ID)
56	TLV 2 Length	0x01
57	TLV 2 Data[0] Protocol ID	0x0B (Default)
58	TLV 3 Tag	0x0E (Span/Channel Offset)
59	TLV 3 Data Length	0x03
60	TLV 3 Data[0] Terminating Span Offset MSB	
61	TLV 3 Data[1] Terminating Span Offset LSB	

BYTE	Description	Value
62	TLV 3 Data[2] Terminating Channel Offset	

Route Group Routing with Address Digits

The example in the table below shows the following PPL Config Byte configuration:

- Byte 20 configures the Outseize Method to Send Call Request to Layer 4 (0x00)
- Byte 21 configures the Routing Flag to Use Internal Router (0x01)
- TLV 1: Bytes 51–54 configure the Routing Method to Route Group (0x06).
- TLV 2: Bytes 55–57 configure the Router Protocol ID to 0x0B, which is the default and the only protocol currently supported.
- TLV 3: Bytes 58–61 indicate the Route Group ID.
- TLV 4: Bytes 62–65 indicate the stage and string number for the address digits.

Table 11-4 Sample PPL Config Byte Configuration

BYTE	Description	Value
20	Outseize Method	0x00 - Send Outseize Control to L3
21	Routing Flag	0x01 - Use Internal Router
50	TLV Count	0x04
51	TLV 1 Tag	0x13 (Routing Method)
52	TLV 1 Data Length	0x02
53	TLV 1 Data[0] Routing Method (MSB)	0x00
54	TLV 1 Data[1] Routing Method (LSB)	0x06 (Route Group)
55	TLV 2 Tag	0x0F (Protocol ID)
56	TLV 2 Length	0x01
57	TLV 2 Data[0] Protocol ID	0x0B (Default)
58	TLV 3 Tag	0x06 (Route Group)
59	TLV 3 Length	0x02
60	TLV 3 Data[0] Route Group ID MSB	
61	TLV 3 Data[1] Route Group ID LSB	
62	TLV 4 Tag	0x01 (Address Digits)

BYTE	Description	Value
63	TLV 4 Data Length	0x02
64	TLV 4 Data[0] Stage #	0x01
65	TLV 4 Data[1] String #	0x01

Terminating Channel Routing with Called Party Digits

The example in the table below shows the following PPL Config Byte configuration:

- Byte 20 configures the Outseize Method to Send Call Request to Layer 4 (0x00)
- Byte 21 configures the Routing Flag to Use Internal Router (0x01)
- TLV 1: Bytes 51–54 configure the Routing Method to Terminating Channel (0x09).
- TLV 2: Bytes 55–57 configure the Router Protocol ID to 0x0B, which is the default and the only protocol currently supported.
- TLV 3: Bytes 58–61 indicate the criteria type of Address Digits 1 (0x01).
- TLV 4: Bytes 62–65 indicate the stage and string number of the address digits.

Table 11-5 Sample PPL Config Byte Configuration

BYTE	Description	Value
20	Outseize Method	0x00 - Send Outseize Control to L3
21	Routing Flag	0x01 - Use Internal Router
50	TLV Count	0x04
51	TLV 1 Tag	0x13 (Routing Method)
52	TLV 1 Data Length	0x02
53	TLV 1 Data[0] Routing Method (MSB)	0x00
54	TLV 1 Data[1] Routing Method (LSB)	0x09 (Terminating Channel)
55	TLV 2 Tag	0x0F (Protocol ID)
56	TLV 2 Length	0x01
57	TLV 2 Data[0] Protocol ID	0x0B (Default)
58	TLV 3 Tag	0x08 (Criteria)
59	TLV 3 Length	0x02
60	TLV 3 Data[0] Criteria Type MSB	0x00
61	TLV 3 Data[1] Criteria Type LSB	0x01 (Address Digits 1)

BYTE	Description	Value
62	TLV 4 Tag	0x01 (Address Digits)
63	TLV 4 Data Length	0x02
64	TLV 4 Data[0] Stage #	0x01
65	TLV 4 Data[1] String #	0x01

Incoming Channel Routing

The example in the table below shows the following PPL Config Byte configuration:

- Byte 20 configures the Outseize Method to Send Call Request to Layer 4 (0x00)
- Byte 21 configures the Routing Flag to Use Internal Router (0x01)
- TLV 1: Bytes 51–54 configure the Routing Method to Terminating Channel (0x09).
- TLV 2: Bytes 55–57 configure the Router Protocol ID to 0x0B, which is the default and the only protocol currently supported.
- TLV 3: Bytes 58–61 indicate the criteria type of Incoming Channel (0x07).
- TLV 4: Bytes 62–65 indicate the stage and string number of the address digits.
- TLV 5: Bytes 66–68 are the Incoming Channel TLV. No data is required.

Table 11-6 Sample PPL Config Byte Configuration

Byte	Description	Value
20	Outseize Method	0x00 - Send Outseize Control to L3
21	Routing Flag	0x01 - Use Internal Router
50	TLV Count	0x05
51	TLV 1 Tag	0x13 (Routing Method)
52	TLV 1 Data Length	0x02
53	TLV 1 Data[0] Routing Method (MSB)	0x00
54	TLV 1 Data[1] Routing Method (LSB)	0x09 (Terminating Channel)
55	TLV 2 Tag	0x0F (Protocol ID)
56	TLV 2 Length	0x01
57	TLV 2 Data[0] Protocol ID	0x0B (Default)
58	TLV 3 Tag	0x08 (Criteria)
59	TLV 3 Length	0x02
60	TLV 3 Data[0] Criteria Type MSB	0x00

Byte	Description	Value
61	TLV 3 Data[1] Criteria Type LSB	0x07 (Incoming Channel)
62	TLV 4 Tag	0x01 (Address Digits 1)
63	TLV 4 Data Length	0x02
64	TLV 4 Data[0] Stage #	0x01
65	TLV 4 Data[1] String #	0x01
66	TLV 5 Tag	0x07 (Incoming Channel)
67	TLV 5 Length	0x01
68	TLV 5 Data[0]	X (Reserved)

Incoming Resource Group Routing

The example in the table below shows the following PPL Config Byte configuration:

- Byte 20 configures the Outseize Method to Send Call Request to Layer 4 (0x00)
- Byte 21 configures the Routing Flag to Use Internal Router (0x01)
- TLV 1: Bytes 51–54 configure the Routing Method to Criteria (0x08).
- TLV 2: Bytes 55–57 configure the Router Protocol ID to 0x0B, which is the default and the only protocol currently supported.
- TLV 3: Bytes 58–61 indicate the criteria type of Incoming Resource Group (0x10).
- TLV 4: Bytes 62–65 indicate the Incoming Resource Group ID.
- TLV 5: Bytes 66–68 indicate the stage and string number of the address digits.

Table 11-7 Sample PPL Config Bytes Configuration

Byte	Description	Value
20	Outseize Method	0x00 - Send Outseize Control to L3
21	Routing Flag	0x01 - Use Internal Router
50	TLV Count	0x03
51	TLV 1 Tag	0x13 (Routing Method)
52	TLV 1 Data Length	0x02
53	TLV 1 Data[0] Routing Method (MSB)	0x00
54	TLV 1 Data[1] Routing Method (LSB)	0x08 (Criteria)
55	TLV 2 Tag	0x0F (Protocol ID)
56	TLV 2 Length	0x01
57	TLV 2 Data[0] Protocol ID	0x0B (Default)
58	TLV 3 Tag	0x08 (Criteria Type)
59	TLV 3 Length	0x02
60	TLV 3 Data[0] Criteria Type MSB	0x00

Byte	Description	Value
61	TLV 3 Data[1] Criteria Type LSB	0x10 (Incoming Resource Group)
62	TLV 4 Tag	0x10 (Incoming Resource Group ID)
63	TLV 4 Length	0x02
64	TLV 4 Data[0] Incoming Resource Group ID MSB	
65	TLV 4 Data[1] Incoming Resource Group ID LSB	
66	TLV 5 Tag	0x01 (Address Digits 1)
67	TLV 5 Data Length	0x02
68	TLV 5 Data[0] Stage #	0x01
69	TLV 5 Data[1] String #	0x01

Incoming Resource Group Routing with Multiple Criteria (Address Digit Type 2)

The example in the table below shows the following PPL Config Byte configuration:

- Byte 20 configures the Outseize Method to Send Call Request to Layer 4 (0x00)
- Byte 21 configures the Routing Flag to Use Internal Router (0x01)
- TLV 1: Bytes 51–54 configure the Routing Method to Criteria (0x08).
- TLV 2: Bytes 55–57 configure the Router Protocol ID to 0x0B, which is the default and the only protocol currently supported.
- TLV 3: Bytes 58–61 indicate the criteria type of Incoming Resource Group (0x10).
- TLV 4: Bytes 62–65 indicate the Incoming Resource Group ID.
- TLV 5: Bytes 66–69 indicate the stage and string number of the address digits for digit type 1.
- TLV 6: Bytes 70–73 is a Multiple Criteria TLV (0x0C) indicating the criteria type of Address Digits Type 2 (0x02).
- TLV 7: Bytes 74–77 indicate the stage and string number of the address digits for digit type 2

Table 11-8 PPL Config Bytes Configuration.

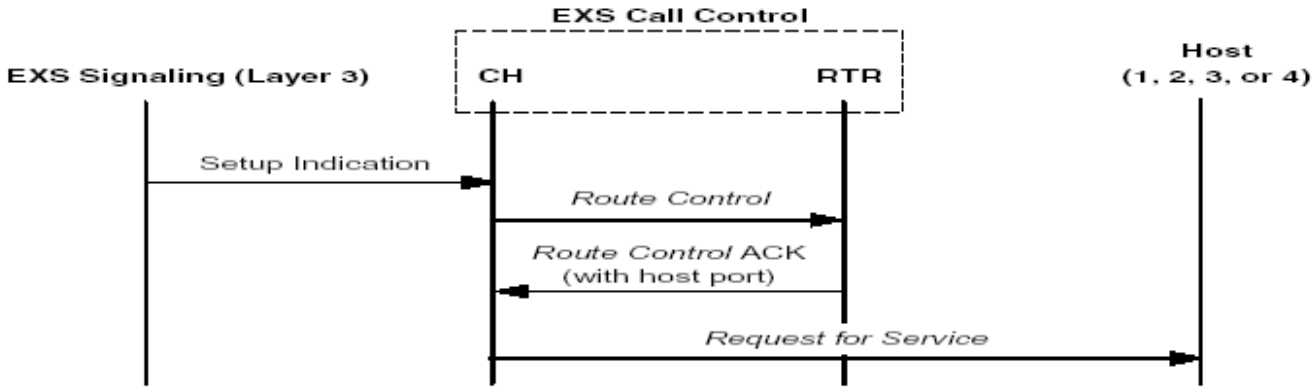
Byte	Description	Value
20	Outseize Method	0x00 - Send Outseize Control to L3
21	Routing Flag	0x01 - Use Internal Router
50	TLV Count	0x07
51	TLV 1 Tag	0x13 (Routing Method)
52	TLV 1 Data Length	0x02
53	TLV 1 Data[0] Routing Method (MSB)	0x00
54	TLV 1 Data[1] Routing Method (LSB)	0x09 (Terminating Channel)
55	TLV 2 Tag	0x0F (Protocol ID)
56	TLV 2 Length	0x01
57	TLV 2 Data[0] Protocol ID	0x0B (Default)
58	TLV 3 Tag	0x08 (Criteria)

Byte	Description	Value
59	TLV 3 Length	0x02
60	TLV 3 Data[0] Criteria Type MSB	0x00
61	TLV 3 Data[1] Criteria Type LSB	0x10 (Incoming Resource Group)
62	TLV 4 Tag	0x10 (Incoming Resource Group ID)
63	TLV 4 Length	0x02
64	TLV 4 Data[0] Incoming Resource Group MSB	
65	TLV 4 Data[1] Incoming Resource Group LSB	
66	TLV 5 Tag	0x01 (Address Digits 1)
67	TLV 5 Data Length	0x02
68	TLV 5 Data[0] Stage #	0x01
69	TLV 5 Data[1] String #	0x01
70	TLV 6 Tag	0x0C (Multiple Criteria)
71	TLV 6 Length	0x02
72	TLV 6 Data[0] Criteria Type MSB	0x00
73	TLV 6 Data[1] Criteria Type LSB	0x02 (Digit Type 2)
74	TLV 7 Tag	0x02 (Address Digits 2)
75	TLV 7 Data Length	0x02
76	TLV 7 Data[0] Stage #	0x02
77	TLV 7 Data[1] String #	0x01

Host Notification of Route Derivation

You can configure the router to send a *Request For Service* message to a specific host port after a route has been derived.

Figure 11-10 Host Port Routing of Request for Service



Configuration

To enable this feature, insert the following TLVs into the PPL Config Bytes of the CH component:

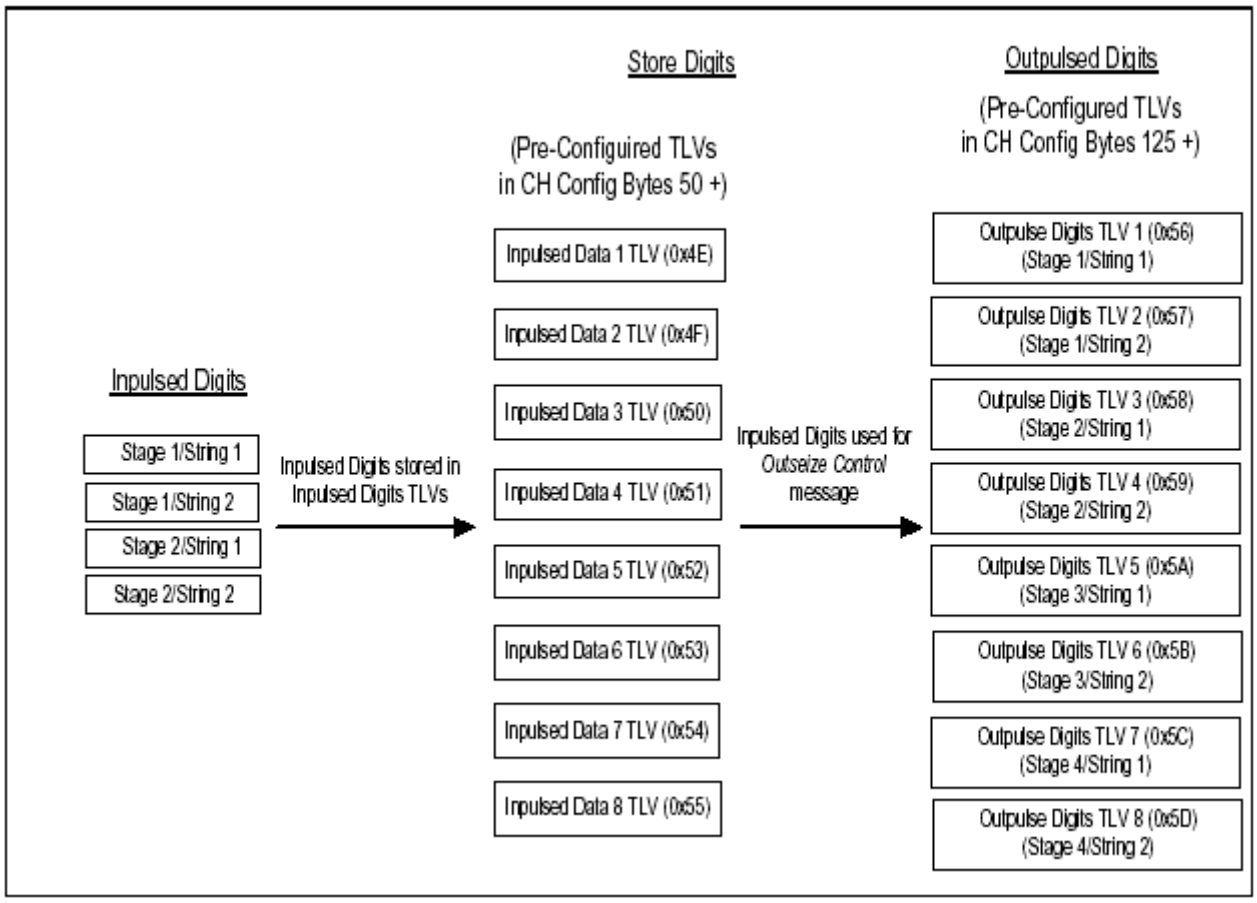
- Special Instructions TLV (0x001D)
- Instruction # 0x01 (Allocate Entity)
- Send Route Control to Router TLV (0x004D)

See the *API Reference* for information on TLV formats.

Stage/String Translation

You can configure the CH component to translate inpulsed data to be used for an outbound call. Inpulsed digits are stored into specified Inpulsed Data TLVs, and then reassembled in a specific order for the outbound call. There are eight generic Inpulsed Digits TLVs with which you can store incoming digits, and eight Outpulse Digits TLVs with which you can place the digits for outgoing calls.

Figure 11-11 Stage/String Translation



Inpulsed Digits You configure Inpulsed Digits TLVs to store specific stage/string data from incoming calls. For example, you could configure the Inpulsed Digits 1 TLV (0x4E) to store Stage 1/String 1 data. Or you could

configure it to store Stage 2/String 1 and Stage 4/String 1 data. Or, you could configure Inpulsed Digits TLV 1 to store Stage 2/String 1 data, and configure Inpulsed Digits TLV 2 to store Stage 4/String 1 data.

Outpulse Digits

Each Outpulse Digits TLV is used to outpulse a specific stage/string combination. You configure each Outpulse Digits TLVs to use the data from one or more Inpulsed Digits TLVs to send in the *Outseize Control* message. For example, you could configure Outpulsed Digits TLV 1 to outpulse the digits stored in Inpulsed Digits TLV 1. You could configure Outpulse Digits TLV 2 to outpulse the digits stored in Inpulsed Digits TLVs 1 and 2.

TLV Formats

Inpulsed Digits TLVs (0x4E–0x55)

Use these TLVs in conjunction with Outpulse Digits TLVs (0x56–5D) for Stage/String translation for tandem routing. The inpulsed digits stored in these TLVs are used to construct the *Outseize Control* message. You configure each TLV to store any stage/string combination(s) from the inpulsed digits.

Inpulsed Digits TLV values are as follows:

0x4E = Inpulsed Digits 1

0x4F = Inpulsed Digits 2

0x50 = Inpulsed Digits 3

0x51 = Inpulsed Digits 4

0x52 = Inpulsed Digits 5

0x53 = Inpulsed Digits 6

0x54 = Inpulsed Digits 7

0x55 = Inpulsed Digits 8

Table 11-9 TLV Formats

Description	Byte (Value)
Tag	Data[0] (0x4E–55)
Length	Data[1] Variable

Description	Byte (Value)
Value	Data[3] Stage Number
	Data[4] String Number
	Data[:] Stage Number
	Data[:] String Number
	:
	:

Outpulse Digits TLVs (0x56–5D)

These TLVs are used to construct the outpulse data in the *Outseize Control* message from the inpulsed digits stored in the Inpulsed Digits TLVs (0x4E–55). Each TLV outpulses a specific stage/string combination. You configure each TLV for which Inpulsed Digits TLV(s) it retrieves the digits to outpulse. The digits are then outpulsed in the *Outseize Control* message in the stage/string specified by that TLV.

The Outpulse Digits TLVs begin at PPL Config Byte 125 of the CH component. Byte 125 specifies the number of TLVs to follow. You configure PPL Config Bytes with the *PPL Configure* message.

Outpulse Digits TLV values are as follows:

0x56 = Outpulse Digits 1 (Stage 1/String 1)

0x57 = Outpulse Digits 2 (Stage 1/String 2)

0x58 = Outpulse Digits 3 (Stage 2/String 1)

0x59 = Outpulse Digits 4 (Stage 2/String 2)

0x5A = Outpulse Digits 5 (Stage 3/String 1)

0x5B = Outpulse Digits 6 (Stage 3/String 2)

0x5C = Outpulse Digits 7 (Stage 4/String 1)

0x5D = Outpulse Digits 8 (Stage 4/String 2)

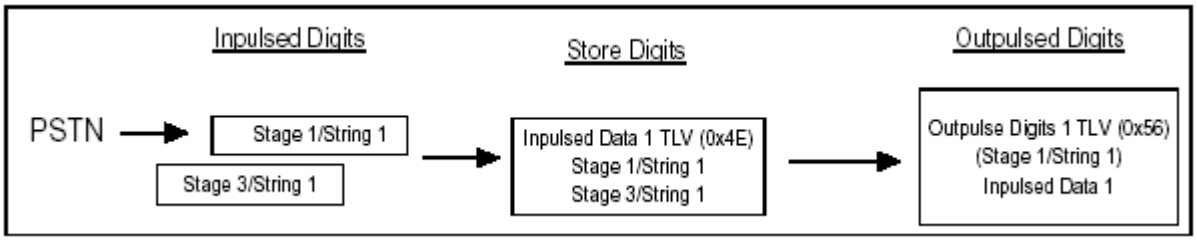
Table 11-10 TLV Formats

Description	Byte (Value)
Tag	Data[0] (0x56–5D)

Description	Byte (Value)
Length	Data[1] Variable
Value	Data[2] Inpulsed Digits TLV #
	Data[3] Inpulsed Digits TLV #
	:

Example This example shows the configuration required for stage/string translation for an E1 R2 to ISDN tandem call. A block diagram representation of the sequence is shown next.

Figure 11-12 Stage/String Translation Example



The following are traces of an example message sequence for configuring TLVs for this scenario.

> **Important!** Values that are part of a string, such as header data, AIBs, or config byte/value combinations, are enclosed in brackets. Bytes important to this configuration are in bold, with descriptive information following in parenthesis.

The following *PPL Configure* message configures Inpulse Digits TLV 1 (0x4E) to store stage 1/string 1 and stage 3/string 1 incoming digits.

```
[00 1d  00 d7  00  00   ff]  [01 02 0d 03 00 00 00 0d 03
   00 1f 17]  [00 61]   01   04 [47 4e] (Inpulse Digits
   TLV 1)  [48 02]  [49  01] (stage 1) [ 4a  01] (string
   1)  [4b  03] (stage 3) [4c  01]  (string 1)
```

The following *PPL Configure* messages configure Outpulse Data TLVs to reassemble inpulsed digits for outbound calls.

This *PPL Configure* message configures byte 125 (Number of TLVs) to 2.

```
[00 17  00 d7  00  00   ff]    [01 02 0d 03 00 00 00 0d 03
   00 1f 17]   [00 61]  01  01   [7d  02]
```

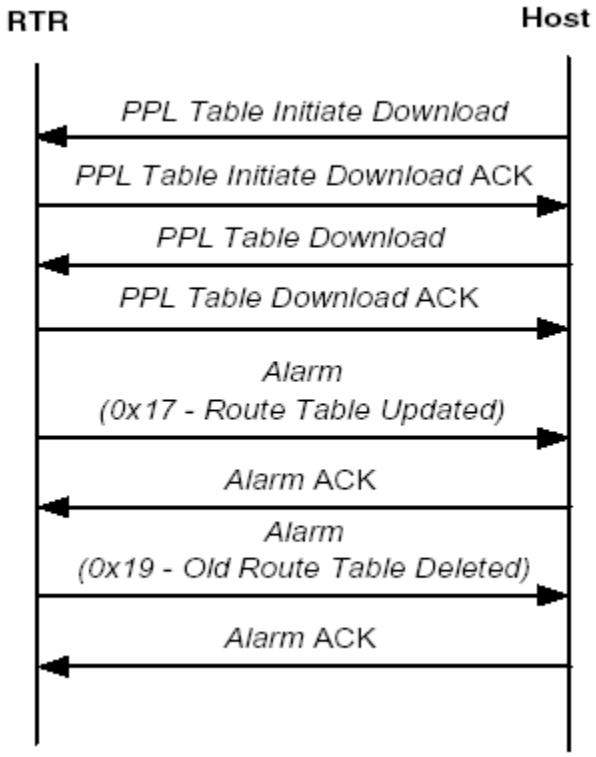
The following *PPL Configure* message configures Outpulse Data TLV 1 (0x4E) to use the digits in Inpulse Digits TLV 1 for the outbound call. The digits will be placed in stage 1/string 1 of the *Outseize Control* message.

```
[00 1b  00 d7  00  00   ff]    [01 02 0d 03 00 00 00 0d 03
   00 1f 17]   [00 61]  01  03 [7e  56] (Outpulse Digits
   TLV 1)  [7f  01]  [80  4e]  (Inpulse Digits TLV 1)
```

Real-time Updating of Route and Resource Group Tables

You can download incremental updates to an existing route or resource group table without deleting the table or disrupting service. Use the *PPL Initiate Download* and *PPL Table Download* messages just as you would to download a new table. In response, you receive an informative alarm of Route Table Updated (0x17), followed by an Old Route Table Deleted (0x19). The sequence of messages is shown in next.

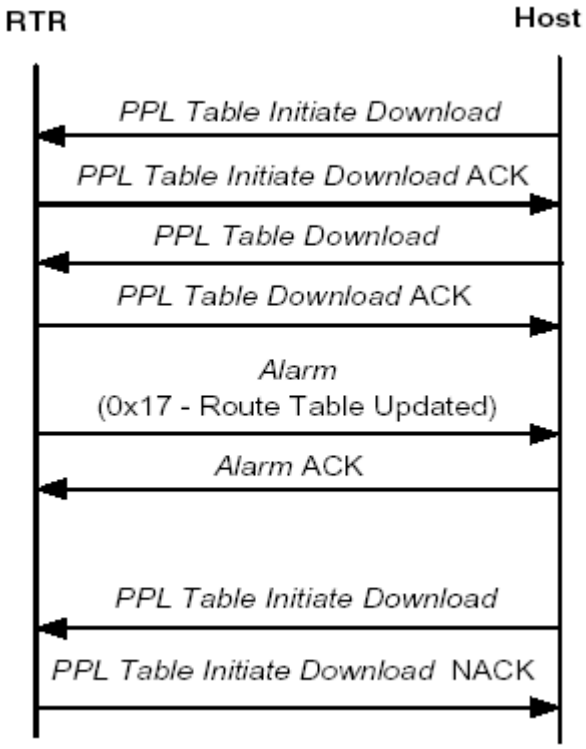
Figure 11-13 Updating Tables



Updating Table Error

If you attempt to download another table of the same type before you receive the Old Route Table Deleted alarm, the *PPL Table Initiate Download* message will not be not acknowledged. The interim between the table download and the Old Route Table Deleted alarm is usually a matter of milliseconds, but may be up to 30 seconds in the case of a forced delete.

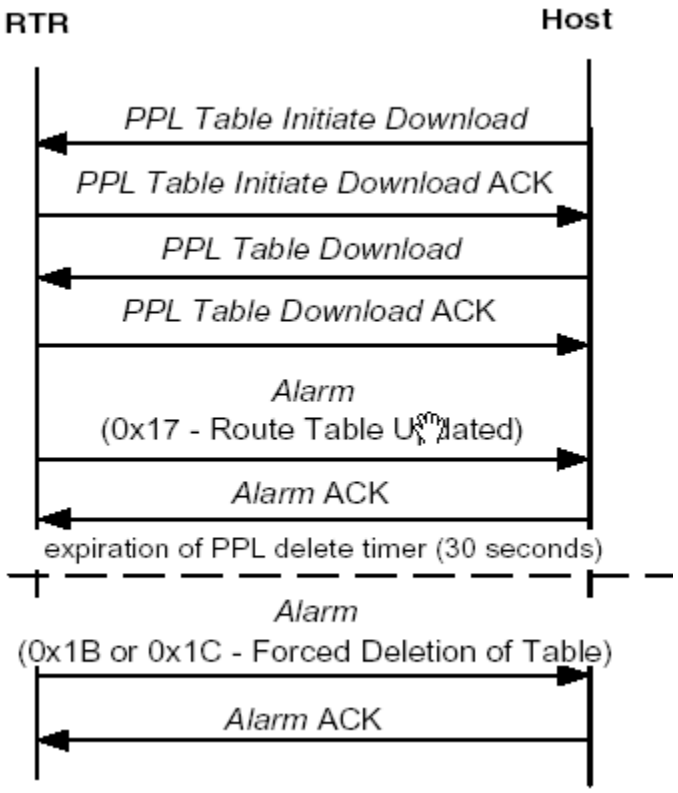
Figure 11-14 Updating Tables Error



Forced Deletion of Tables

A table is forcefully deleted if for any reason it cannot be gracefully deleted within 30 seconds. You will receive a general informative alarm of Route Table Forcefully Deleted (0x1B) or Resource Table Forcefully Deleted (0x1C) depending on the type of table deleted. The sequence of messages is shown next.

Figure 11-15 Forced Deletion of Tables



Uploading Route and Resource Group Tables

You can upload Route or Resource Group table data from the CSP using either the *PPL Data Query* message or the Route Configuration Tool.

This is useful for the following purposes:

- Diagnostics
- Reconstructing tables inadvertently deleted on the host
- Modifying existing tables

Route Configuration Tool

From the Main screen of the Route Configuration Tool, click the Upload button. The Upload Manager screen appears. Enter the IP Address of the CSP, the API format, and the Table IDs then click the Start button. The table data is displayed in the text box at the bottom of the window.

PPL Data Query Message

To upload a table, send the *PPL Data Query* message with the Entity field set to 0x04 (PPL Table). You can retrieve either a portion of a table or the entire table. If the response data cannot be sent by the CSP in one message, you must resend the message, incrementing the range, until a response status of 0xD7 is returned. See the *PPL Data Query 0x00DE* message in the *API Reference* for data.

Route Configuration Tool

This section includes information on using the Route Configuration Tool for editing or creating the following:

- Route Tables
- Resource Tables
- PPL Configuration Bytes

Installing and Running the Program

To install the Route Configuration Tool, copy the **rct**.**exe** file to a local drive. To run the tool, double-click the **.exe** file or type **.rct** from a command line.

Creating/Editing Files

To create a new file, click on the NEW button on the right side of the main screen. Then click the **SAVE AS** button to save the file.

To edit an existing **.rtr** file, click the **OPEN** button, select the desired file, and click the **OK** button.

You can create multiple route tables and resource tables within a single **.rtr** file.

Configuring a Route Table

1. Create a New Route Table

Click in the white box in the Route Table section and then click the **ADD** button. The Route Table box appears with the next available Table ID (you can change this). Click the **OK** button to add the Table ID to the route table. Valid Table IDs are 1–10.

To edit an existing Table ID, click on the **EDIT** button or double-click on the entry.

2. Route Entry Information

In the Route Table Entries box, click on the **ADD** button. The Route Entry Editor screen appears.

To edit an existing entry, double-click on the entry in the Resource Table Entries box or select the entry and click the **EDIT** button.

From the Route Entry Editor screen, you enter the following information:

- Route Group ID
- Criteria Type and Data

- Route Entry Instructions

Criteria Editor

When you click the EDIT button in the Criteria section of the Route Entry Editor screen, the Criteria Editor screen appears. The data that can be edited appears in white, depending on the Criteria Type selected. When you have edited the data, click the **OK** button.

Route Entry Instruction Editor

Click in the Route Entry Instructions box in the **Route Entry Editor** screen to enter or edit Route Entry Instructions. Select a Tag from the Tag drop-down menu, enter the required data (which appears in white), and click the **OK** button.

When you have completed all entries for the table, return to the main screen where you can add new route tables. When the configuration is complete, create a configuration file. See *Creating a Configuration File (11-45)*.

Configuring a Resource Table

1. Create a New Resource Table

Click in the white box in the **Resource Tables** section and click the **ADD** button. The Resource Table box will appear. Enter the Table ID and Group ID. Click the **OK** button.

To edit an existing Table ID or Group ID, double-click on the entry or select the entry and click the **EDIT** button.

2. Resource Entry Information

In the Resource Table Entries box, click on the ADD button. The Resource Entry Editor screen appears. Select the Resource Type and enter the required data, which appears in the Data section in white. When entries are complete, click on the OK button.

To edit an existing table, double-click on the entry or select the entry and click the **EDIT** button.

When you have completed all entries for the table, return to the main screen. You can add new route tables from here. When all configuration is complete, create a configuration file.

Configuring PPL Config Bytes

The Configuration Byte Editor allows you to configure router-related PPL Config Bytes of the CH component. The following bytes can be edited with the tool:

20 - Outseize Mode

21 - Enable Internal Routing

50+ - Router TLVs

To edit a Config Byte, click in the Configuration Bytes box and click the **ADD** button. To edit an existing entry, double-click on the entry or select the entry and click the **EDIT** button. The Configuration Byte Editor screen appears.

Enter the Byte Offset, Config Data, and the resources to which the configuration is to be assigned. Click the **OK** button.

Creating a Configuration File

When all configuration is complete, click the .**cfg** file button on the right side of the Main screen. The **Create Configuration File** box appears. Select the path to where the file is to be saved and then click the **CREATE** button.

12 EXNET Connect® Card

Purpose This chapter provides an information on the PCI H.100 EXNET Connect® card (referred to in this document as the EXNET Connect® card) including:

- overview
- configuring
- requirements

For more information on the EXNET-ONE card and the EXNET® ring, refer to *Configuring Multi-Node Systems (10-1)*.

EXNET Connect® Card Overview

Local Switching Matrix

The PCI H.100 EXNET Connect® card is a single card that allows you to connect remote CSP nodes to Computer Telephony (CT) bus voice/media processing resources within the voice/media processing unit. The EXNET Connect® card becomes a local switching matrix within the PC chassis, switching traffic between the EXNET® ring and the CT bus resources.

EXNET Connect® accomplishes the connection by switching remote node data off the EXNET® high-speed fiber optic ring and onto the bi-directional CT bus, which makes the data accessible to any of the voice processing resources in the EXNET Connect® node. In a similar manner, EXNET Connect® switches local CT bus data onto the EXNET® ring, making this data available to any of the remote CSP nodes.

The EXNET Connect® plugs into any voice processing subsystem with an industry standard CT bus slot.

The following are compatibility issues and other factors to consider:

- The EXNET Connect® card requires a minimum of System Software Release 8.1. Clear channel protocol is the default.
- The PCI H.100 EXNET Connect® card and the CSP must be running the same version of system software.
- The Parallel Bus, although described in this chapter, will not be supported in Release 8.1.
- To use TFTP downloading with both EXNET Connect® cards, the cards must have version 2.2 ROMs or later.

Communicating with the Host

The EXNET Connect® cards are designed to communicate with the voice processing host externally via Ethernet, or internally via the PC parallel bus. The voice processing host can configure the EXNET Connect® over the parallel bus along with the other CT bus cards in the VRU.

Important! All CSP internodal communications use Ethernet; therefore, the user must be sure to make all Ethernet connections.

Downloading System Software

The same procedure used to download system software to CSP 1000/ 2000 nodes is used to download system software to EXNET Connect® cards, wit the following exceptions:

Refer to *Downloading System Software*.

1. EXNET Connect® uses a different system software executable image from CSP 1000/2000 nodes.
2. EXNET Connect® does not support system software downloading in S-Record format. Only a binary file is supplied.
3. As with CSP 1000/2000 nodes, system software is downloaded to volatile memory (RAM). However, during initialization and initial operation, EXNET Connect® copies the download image from volatile to non-volatile memory (Flash) for local reloads of the executable image. This copying into Flash memory takes about two minutes, during which bit 4 in the *Poll* message from the EXNET Connect® is set to 1 (not ready for power down). This bit is reset to 0 when the Flash copy is complete.

Important! The CSP 1000/2000 system software is available as both a .bin (binary) and .abs (S-Record) file, which is then downloaded. EXNET Connect® is available as a .bin file only. EXNET Connect® does not support downloading S-Record files.

Configuring the IP Address and Subnet Mask

The procedure to configure the IP Address and Subnet Mask for the EXNET Connect® card is the same as for the CSP Matrix Series 3 Card.

Refer to the *Host Communication Module* for the procedure.

Configuring EXNET Connect®

Overview

You can configure EXNET® using the API messages as described below. You can also use the Converged Services Administrator (CSA). Refer to the *Converged Services Administrator User's Guide*.

Sending API Messages

1. Send the *Assign Logical Node ID* message (0x10), using the last four digits of the serial number (which must be converted to hexadecimal) of the EXNET Connect® card as the basis for the physical node ID. See *Assigning Physical Node IDs (12-6)* for more information.
2. Send the *EXS Node Configure* message (`0x7F`) to set the H.100 clock master and H.100 clock speed.

   Important! You must first assign the H.100 clock master before you configure any H.100 resource.

3. Send the *EXNET Ring Configure* message (`0x74`) with entity *Assign Ring* (0x01) to assign the logical ring ID.
4. Send the *EXNET Ring Configure* message (`0x74`) with entity *Configure Number of Packets* (0x03) to set the number of packets to be transmitted on the ring.

   Important! The number of packets should be set to 1 for the PCI H.100 EXNET Connect® in the SCSA compatibility mode. For the full H.100 mode the number of packets should be set to 4.

5. Send the *EXNET Ring Configure* message (`0x74`) with entity *Configure Transmit Mode* (0x02) to set the EXNET Connect® to transmit mode.
6. Send the *EXNET Ring Configure* message (`0x74`) with entity Ring Mode (0x0A) to set the EXNET Connect® to Enhanced Fault Tolerance Ring (EFTR) mode.
7. Send the *Assign Logical Span ID* message (`0xA8`) to assign spans to physical span offsets on the PCI H.100 card.
8. Bring the spans and channels into service.
9. Bring the EXNET® ring into service.

   Important! An CSP node ring configuration for all other non-EXNET Connect® nodes on the ring must be performed before ring configuration of the EXNET Connect® node, to avoid bandwidth fragmentation.

Assigning Logical Node IDs

You assign a logical node ID to the PCI H.100 card just as you would with the CSP nodes on the EXNET® ring.

Assign a Logical Node ID (0-63) to each card, using the *Assign Logical Node ID* message. All nodes on the same Ethernet segment must have a unique Logical Node ID and each Physical Node ID must have one Logical Node ID. If a node ID is already assigned, the Assign Logical Node ID message receives a negative acknowledgment (NACK).

Important! The Logical Node ID must be in the range of (0x00–0x3F).

Assigning Physical Node IDs

Because the PCI H.100 is a card within a PC chassis instead of an CSP, you assign the physical node ID by performing the following:

1. Use the serial number of the card for the last four digits.
2. Set the two most significant bytes to 0x0001.

Example: If the serial number of the PCI H.100 card was 0x03AD, in the *Assign Logical Node ID* message (`0x10`) you define the value of the *Physical Node ID* field as 0x000103AD (which is 0x03AD ORd with 0x10000).

Setting Up EXNET Connect® on the Ring

The PCI H.100 card appears as a normal CSP node on the ring; however, it cannot become the ring master. Therefore, you must have at least one other CSP on the ring to act as the ring master.

Configuring the Clock Master

You configure the EXNET Connect® like any other CSP node by sending the *EXS Node Configure* message (0x7F). This message allows the host to make the following two assignments:

- Number of channels per span on the H.100 bus
- H.100 clock mastership

You can configure any voice processing resource card, including EXNET Connect®, as the H.100 clock master. You can configure the system to assign EXNET Connect® as the clock master should the current master fail.

Making a Connection

One-way Connection

To create a one-way connection to a voice processing resource, the host must send an *Outseize Control* message to the EXNET Connect® span/channel. This span/channel must be associated with the voice processing resource timeslot span/channel that is assigned to a physical offset on the H.100 bus.

To create the connection, send a *Connect One-Way Forced* message (0x50) to another CSP node. In the message, designate the EXNET Connect® channel as Channel B and the channel on the other node as Channel A.

Two-way Connection

To create a two-way connection between a channel on an CSP node and a voice processing resource, send a *Connect* message (0x00) to the CSP node. In the message, designate the CSP node channel as Channel A and the EXNET Connect® channel as Channel B.

Sending a *Connect* message allows the channel on the CSP node to receive data from the voice processing resource and data from the CSP node to be transmitted on the H.100 bus timeslot associated with the logical span and channel assigned to the EXNET Connect®.

PCI/H.100 Specific Requirements

The following are requirements for the PCI H.100 EXNET Connect®:

- The EXNET® (ring) interface is similar for both the EXNET Connect® product and the CSP. See the *Hardware Installation and Maintenance Guide* information for details on CSP system setup.
- EXNET Connect® appears as a normal CSP node on the ring. You can test it as an CSP node with the following exceptions:
  - The EXNET Connect® will not attempt to become the ring master; therefore, the system must have at least one other CSP 2040 or CSP 2090 on the ring to act as the ring master.
  - The EXNET Connect® always behaves like an CSP node with a single EXNET-ONE card installed (for example, ring redundancy is not supported).
- You assign a card slot number between 0 and 31 (0x00 and 0x1F) to EXNET Connect® using DIP Switch S4. The host for subsequent slot-related messages will use this slot number to set, for example, span assignments and the H.100 master clock.
- The host can assign a maximum of 16 E1 or 21 T1 logical spans on the PCI H.100 card operating in the SCSA Compatibility Mode because the SCSA Bus (4MHZ) supports 1024 timeslots or 512 CSP channels. However, the host supports a maximum of 64 E1 or 80 T1 logical spans in the full H.100 mode because the H.100 Bus (8MHZ) supports 2048 timeslots or 1024 CSP channels.
- The EXNET Connect's Layer 3 is PPL-driven. The default PPL Protocol included in the system software is Clear Channel protocol, PPL component ID 75 (0x4B).
- EXNET Connect® can support multiple protocols. It allows the host application to download up to 10 protocols. If you attempt to download more than 10 protocols, EXNET Connect® may send the following responses:
  – 0xD3 - *Invalid Protocol ID*
  (If the protocol ID is greater than 10)
  – 0xA7 - *PPL Table Already Exists*
  (If the protocol table ID is 1–10 but has already been downloaded)

- The EXNET Connect® uses the *EXS Node Configure* message (0x7F) with new entities defined to configure PCI H.100 mastership.
- All configuration is maintained on a push button reset of the EXNET Connect® card. This includes card-specific information (assigned spans, channels per span, H.100 bus, download software revisions, downloaded PPL Protocols, and so on) and ring-specific information (logical ring ID, ring service state, logical node ID, and so on).
- When the host assigns spans on the PCI H.100 card, the span assignments are broadcast to all other nodes on the ring. Broadcasting the assignments guarantees unique span assignments among the nodes on the ring.

H.100 Bus Timeslot Mapping

SCSA Compatibility Mode

There are 1024 timeslots available on the H.100 for transmitting and receiving data. The PCI/H.100 EXNET Connect® transmits over a maximum of 16 E1 logical spans (512 channels) or 21 T1 spans (504 channels). It internally maps logical span/channels to H.100 timeslots depending on the physical span offsets by the host. Timeslots 0 - 511 are reserved for PCI/H.100 to transmit to voice processing resources over the H.100 bus. Timeslots 512 — 1023 are available for the other H.100 bus resources to transmit data.

Also, EXNET Connect® logically pairs transmit and receive Timeslots offset by 512 into full duplex channels. For example, Timeslot 0 (transmit) is paired with Timeslot 512 (receive) to make up Channel 0, Timeslot 1 is paired with Timeslot 513 to make up Channel 1, etc. EXNET Connect® logically groups channels into spans to establish connections based on span/channel assignments.

Example

T1 Logical Spans

Channels 0-23 = Span 0, Channels 24-47 = Span 1, etc.

E1 Logical Spans

Channels 0-31 = Span 0, Channels 32-63 = Span 1, etc.

The EXNET Connect® driven timeslots and the voice resource-driven timeslots have a one-to-one correspondence, offset by 512. For example, timeslots 0 and 512 create a logical full duplex channel (span 0, channel 0). Figure 12-1 illustrates this relationship.

Figure 12-1 H.100 Bus Logical Channel Mapping (SCSA Compatibility Mode)

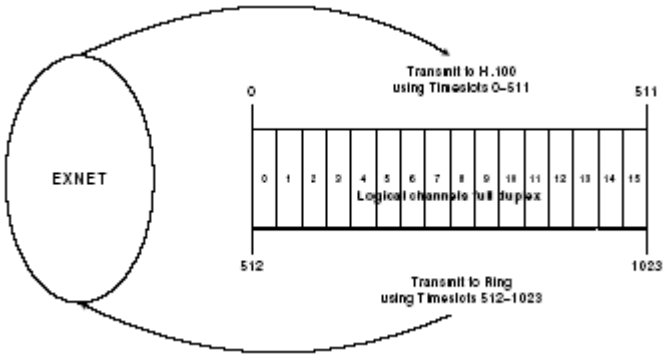


Table 12-1 Transmit and Receive (SCSA Compatibility Mode)

H.100 Bus	EXNET Connect XMIT		EXNET Connect XMIT		Voice Resources XMIT		Voice Resources XMIT	
Timeslots	**E1 Span**	**E1 Channel**	**T1 Span**	**T1 Channel**	**E1 Span**	**E1 Channel**	**T1 Span**	**T1 Channel**
0	0	0	0	0	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV
3	0	3	0	3	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV
31	0	31	1	7	RCV	RCV	RCV	RCV
32	1	0	1	8	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV
63	1	31	2	:	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV
510	15	30	Not used	Not used	RCV	RCV	Not used	Not used
511	15	31	Not used	Not used	RCV	RCV	Not used	Not used
512	RCV	RCV	RCV	RCV	0	0	0	0
:	:	:	:	:	:	:	:	:

H.100 Bus	EXNET Connect XMIT		EXNET Connect XMIT		Voice Resources XMIT		Voice Resources XMIT	
Timeslots	**E1 Span**	**E1 Channel**	**T1 Span**	**T1 Channel**	**E1 Span**	**E1 Channel**	**T1 Span**	**T1 Channel**
543	RCV	RCV	RCV	RCV	0	31	1	7
:	:	:	:	:	:	:	:	:
547	RCV	RCV	RCV	RCV	1	3	1	11
:	:	:	:	:	:	:	:	:
576	RCV	RCV	RCV	RCV	2	0	2	16
:	RCV	RCV	RCV	RCV	:	:	:	:
1023	RCV	RCV	Not used	Not used	15	31	Not used	Not used

Full H.100 Mode

There are 4096 timeslots available on the H.100 for transmitting and receiving data. The PCI/H.100 EXNET Connect® transmits over a maximum of 64 E1 logical spans (2048 channels) or 80 T1 spans (1920 channels). It internally maps logical span/channels to H.100 timeslots depending on the physical span offsets by the host. Timeslots 0 — 2047 are reserved for PCI/H.100 to transmit to voice processing resources over the H.100 bus. Timeslots 2048 — 4095 are available for the other H.100 bus resources to transmit data.

Also, EXNET Connect® logically pairs transmit and receive Timeslots offset by 2048 into full duplex channels. For example, Timeslot 0 (transmit) is paired with Timeslot 2048 (receive) to make up Channel 0, Timeslot 1 is paired with Timeslot 2049 to make up Channel 1, etc. EXNET Connect® logically groups channels into spans to establish connections based on span/channel assignments.

Example

T1 Logical Spans

Channels 0-23 = Span 0, Channels 24-47 = Span 1, etc.

E1 Logical Spans

Channels 0-31 = Span 0, Channels 32-63 = Span 1, etc.

The EXNET Connect® driven timeslots and the voice resource-driven timeslots have a one-to-one correspondence, offset by 2048. For example, timeslots 0 and 2048 create a logical full duplex channel (span 0, channel 0). Figure 12-2 illustrates this relationship.

Figure 12-2 H.100 Bus Logical Channel Mapping

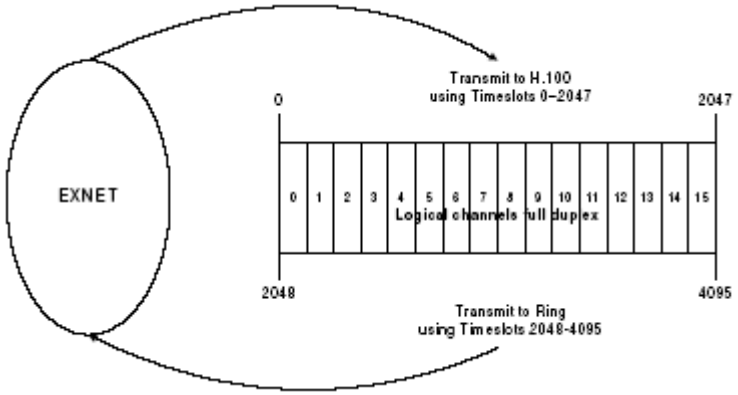


Figure 12-3 Transmit and Receive (Full H.100 Mode)

H.100 Bus	EXNET Connect XMIT		EXNET Connect XMIT		Voice Resources XMIT		Voice Resources XMIT	
Timeslots	E1 Span	E1 Channel	T1 Span	T1 Channel	E1 Span	E1 Channel	T1 Span	T1 Channel
0	0	0	0	0	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV
3	0	3	0	3	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV
31	0	31	1	7	RCV	RCV	RCV	RCV
32	1	0	1	8	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV
63	1	31	2	:	RCV	RCV	RCV	RCV
:	:	:	:	:	RCV	RCV	RCV	RCV

H.100 Bus	EXNET Connect XMIT		EXNET Connect XMIT		Voice Resources XMIT		Voice Resources XMIT	
2046	63	30	Not used	Not used	RCV	RCV	Not used	Not used
2047	63	31	Not used	Not used	RCV	RCV	Not used	Not used
2048	RCV	RCV	RCV	RCV	0	0	0	0
:	:	:	:	:	:	:	:	:
2079	RCV	RCV	RCV	RCV	0	31	1	7
:	:	:	:	:	:	:	:	:
2082	RCV	RCV	RCV	RCV	1	3	1	11
:	:	:	:	:	:	:	:	:
2111	RCV	RCV	RCV	RCV	2	0	2	16
:	RCV	RCV	RCV	RCV	:	:	:	:
4095	RCV	RCV	Not used	Not used	63	31	Not used	Not used