Dialogic® NaturalAccess™ ISDN Software Installation Manual

October 2009 64-0507-01

www.dialogic.com

Copyright and legal notices

Copyright © 2000-2009 Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation at the address provided below.

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

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

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

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at www.dialogic.com.

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

Any use case(s) shown and/or described herein represent one or more examples of the various ways, scenarios or environments in which Dialogic® products can be used. Such use case(s) are non-limiting and do not represent recommendations of Dialogic as to whether or how to use Dialogic products.

Dialogic, Dialogic Pro, Brooktrout, Diva, Cantata, SnowShore, Eicon, Eicon Networks, NMS Communications, NMS (stylized), Eiconcard, SIPcontrol, Diva ISDN, TruFax, Exnet, EXS, SwitchKit, N20, Making Innovation Thrive, Connecting to Growth, Video is the New Voice, Fusion, Vision, PacketMedia, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation or its subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. The names of actual companies and product mentioned herein are the trademarks of their respective owners.

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

Revision history

Revision	Release date	Notes
9000-60010-10	July, 2000	EPS / SJC, support for Fusion 4.0
9000-60010-11	September, 2000	SJC, for CT Access 4.0
9000-60010-12	March, 2001	SJC, for NACD 2000-2
9000-60010-13	April, 2001	SJC, for NACD 2001-1 Beta
9000-60010-14	August, 2001	SJC, for NACD 2001-1
9000-60010-15	November, 2001	SJC, for NACD 2002-1 Beta
9000-60010-16	May, 2002	LBG, NACD 2002-1
9000-60010-17	November, 2002	LBG, Natural Access 2003-1 Beta
9000-60010-18	April, 2003	MCM, Natural Access 2003-1
9000-60010-19	April, 2004	SRR, Natural Access 2004-1
9000-60010-20	October 2005	DEH, Natural Access 2005-1, SP 1
64-0507-01	October 2009	LBG< NaturalAccess R9.0
Last modified: September 8, 2009		

Refer to www.dialogic.com for product updates and for information about support policies, warranty information, and service offerings.

Table Of Contents

1 Introduction

The *Dialogic® NaturalAccess™ ISDN Software Installation Manual* describes how to install NaturalAccess ISDN Software on your system. This software enables boards to connect to a T1, E1, or BRI trunk using primary (PRI) or basic (BRI) rate ISDN.

This document is intended for telephony and voice application developers who are using NaturalAccess. This manual defines telephony terms where applicable, but assumes that you are familiar with telephony concepts. It also assumes that you are familiar with the C programming language.

2 Terminology

Note: The product to which this document pertains is part of the NMS Communications Platforms business that was sold by NMS Communications Corporation ("NMS") to Dialogic Corporation ("Dialogic") on December 8, 2008. Accordingly, certain terminology relating to the product has been changed. Below is a table indicating both terminology that was formerly associated with the product, as well as the new terminology by which the product is now known. This document is being published during a transition period; therefore, it may be that some of the former terminology will appear within the document, in which case the former terminology should be equated to the new terminology, and vice versa.

Former terminology	Dialogic terminology
CG 6060 Board	Dialogic® CG 6060 PCI Media Board
CG 6060C Board	Dialogic® CG 6060C CompactPCI Media Board
CG 6565 Board	Dialogic® CG 6565 PCI Media Board
CG 6565C Board	Dialogic® CG 6565C CompactPCI Media Board
CG 6565e Board	Dialogic® CG 6565E PCI Express Media Board
CX 2000 Board	Dialogic® CX 2000 PCI Station Interface Board
CX 2000C Board	Dialogic® CX 2000C CompactPCI Station Interface Board
AG 2000 Board	Dialogic® AG 2000 PCI Media Board
AG 2000C Board	Dialogic® AG 2000C CompactPCI Media Board
AG 2000-BRI Board	Dialogic® AG 2000-BRI Media Board
NMS OAM Service	Dialogic® NaturalAccess™ OAM API
NMS OAM System	Dialogic® NaturalAccess™ OAM System
NMS SNMP	Dialogic® NaturalAccess™ SNMP API
Natural Access	Dialogic® NaturalAccess™ Software
Natural Access Service	Dialogic® NaturalAccess™ Service
Fusion	Dialogic® NaturalAccess™ Fusion™ VoIP API
ADI Service	Dialogic® NaturalAccess™ Alliance Device Interface API
CDI Service	Dialogic® NaturalAccess™ CX Device Interface API
Digital Trunk Monitor Service	Dialogic® NaturalAccess™ Digital Trunk Monitoring API
MSPP Service	Dialogic® NaturalAccess™ Media Stream Protocol Processing API
Natural Call Control Service	Dialogic® NaturalAccess™ NaturalCallControl™ API
NMS GR303 and V5 Libraries	Dialogic® NaturalAccess™ GR303 and V5 Libraries

Former terminology	Dialogic terminology
Point-to-Point Switching Service	Dialogic® NaturalAccess™ Point-to-Point Switching API
Switching Service	Dialogic® NaturalAccess™ Switching Interface API
Voice Message Service	Dialogic® NaturalAccess™ Voice Control Element API
NMS CAS for Natural Call Control	Dialogic® NaturalAccess™ CAS API
NMS ISDN	Dialogic® NaturalAccess™ ISDN API
NMS ISDN for Natural Call Control	Dialogic® NaturalAccess™ ISDN API
NMS ISDN Messaging API	Dialogic® NaturalAccess™ ISDN Messaging API
NMS ISDN Supplementary Services	Dialogic® NaturalAccess™ ISDN API Supplementary Services
NMS ISDN Management API	Dialogic® NaturalAccess™ ISDN Management API
NaturalConference Service	Dialogic® NaturalAccess™ NaturalConference™ API
NaturalFax	Dialogic® NaturalAccess™ NaturalFax™ API
SAI Service	Dialogic® NaturalAccess™ Universal Speech Access API
NMS SIP for Natural Call Control	Dialogic® NaturalAccess™ SIP API
NMS RJ-45 interface	Dialogic® MD1 RJ-45 interface
NMS RJ-21 interface	Dialogic® MD1 RJ-21 interface
NMS Mini RJ-21 interface	Dialogic® MD1 Mini RJ-21 interface
NMS Mini RJ-21 to NMS RJ-21 cable	Dialogic® MD1 Mini RJ-21 to MD1 RJ-21 cable
NMS RJ-45 to two 75 ohm BNC splitter cable	Dialogic® MD1 RJ-45 to two 75 ohm BNC splitter cable
NMS signal entry panel	Dialogic® Signal Entry Panel

3 Overview of NMS ISDN

NMS ISDN software overview

NMS ISDN protocol software enables you to write Natural Access applications that communicate on T1, E1, or BRI trunks to perform voice processing functions and call control using ISDN common channel signaling (CCS) protocols.

Use NMS ISDN with one or more AG or CG boards, which provide the physical interface to trunk lines. Most of these boards also feature on-board digital signal processing (DSP) resources that can perform call control and voice processing functions.

You can configure NMS ISDN software to access ISDN services in any of the following ways:

Configuration	Description
Channelized	The application performs call control and other operations using Natural Call Control (NCC). For more information about this configuration, see the *NMS ISDN for Natural Call Control Developer's Manual.*
ACU	The application accesses ISDN services at the ACU SAP using the NMS ISDN Messaging service. This configuration enables the application to perform a wide range of Q.931 ISDN D channel functions. For more information about this configuration, see the *NMS ISDN Messaging API Developer's Reference Manual.*
LAPD	The application accesses ISDN services at the data link layer (layer 2) using the NMS ISDN Messaging service. This setup enables the application to send and receive I-frame data in LAPD messages. This data typically consists of Q.931 messages. For more information about this configuration, see the *NMS ISDN Messaging API Developer's Reference Manual*.

Choose a method for accessing ISDN services when you initialize the NMS ISDN protocol stack. The access methods are described in both the *NMS ISDN Messaging API Developer's Reference Manual* and the *NMS ISDN for Natural Call Control Developer's Manual*.

NMS ISDN software components

ISDN is implemented in different ways around the world. For this reason, NMS Communications provides several variants of its NMS ISDN software for different regions. The package for a variant contains the software modules you need to enable an AG or CG board to communicate on a T1, E1, or BRI trunk in your particular country or region.

The NMS ISDN software package contains the following:

- A *readme* file
- NMS ISDN function libraries for Natural Access
- Header files
- Downloadable object modules containing the NMS ISDN protocol stack software and the NMS ISDN management software
- Sample board keyword files
- Demonstration programs and utilities (including the source code files and makefiles)
- A trunk control program (TCP)
- Several binary parameter files (*.pf* files) and several ASCII parameter files (*.par* files)

readme file

The *readme_isdn.txt* file contains release information that does not appear in other documentation. Consult this file to learn where the NMS ISDN software components are located.

You can find *readme_isdn.txt* in the following location:

Operating system	***readme* file location**
Windows	*\nms\doc*
UNIX	*/opt/nms/doc*

NMS ISDN function libraries

The NMS ISDN function libraries run on the host computer. The application uses the function libraries to interact with ISDN protocol stacks running on a board and to communicate with the NCC and ADI services.

The following table lists the NMS ISDN function libraries:

Windows	**UNIX**
isdnapi.lib, *isdnapi.dll*	*libisdnapi.so*
nccisdn.lib, *nccisdn.dll*	*libnccisdn.so*
imgtapi.lib, *imgtapi.dll*	*libimgtapi.so*

Header files

The following header files are supplied with NMS ISDN software:

File name	Contents	Used in these configurations
isdnval.h	Definitions for Q.931 messages created by the stack.	ACU, channelized
isdndef.h	Event code definitions and ISDN function prototypes.	ACU, LAPD
isdnparm.h	Parameter structure definitions and manifest constants for parameter structure fields.	ACU, LAPD
isdntype.h	NMS Communications type definitions, basic and derived types, and entity identifiers.	ACU, LAPD
isdnacu.h	ACU SAP message structure definitions and macros to build ACU messages.	ACU
isdndl.h	SAPI SIG configuration for raw LAPD and macros to build LAPD messages.	LAPD
nccxadi.h	NCC parameter structures.	Channelized
nccadi.h	NCC values for mediamask, connectmask, and disconnectmask.	Channelized
nccxisdn.h	ISDN parameter structures and values for NCC.	Channelized
adiisd.h	TCP parameter structures.	ADI
imgtdef.h	Management service function prototypes and management event code definitions.	IMGT For more information, refer to the *NMS ISDN Management API Developer's Reference Manual*.
imgtsvc.h	Management service primitives and related data structures.	IMGT For more information, refer to the *NMS ISDN Management API Developer's Reference Manual*.

Downloadable object modules

A downloadable object module file contains the basic low-level software that an AG or CG board requires to support ISDN. The module is transferred from the host to on-board memory when the board boots. NMS ISDN provides downloadable variant modules that are specific to the configuration and country. For a complete list of the NMS ISDN variant modules, see *DLMFiles[x]* on page 30.

NMS ISDN also provides the downloadable module files *imgt.leo*, *imgt.dlm*, and *c65imgt.dlm*, which contain the basic low-level software required by an AG or CG board to support the NMS ISDN Management service. For more information about these files and about the Management service, refer to the *NMS ISDN Management API Developer's Reference Manual*.

Demonstration programs

The following demonstration programs, source code, and makefiles are included in the software package:

Name	Description	For more information, see the...
dectrace	Decodes and displays messages sent or received by the NMS ISDN protocol stack that were previously captured in a log file by the *oammon* utility.	*NMS ISDN Messaging API Developer's Reference Manual*
isdncta	ISDN *daemon* program to start and stop the NMS ISDN protocol stack in channelized mode.	*NMS ISDN for Natural Call Control Developer's Manual*
isdndemo	• Provides an example of a digital trunk application where the NMS ISDN software runs in an ACU configuration. • Shows how to use the NMS ISDN Messaging service to start this configuration, and to receive and place calls on an ISDN trunk.	*NMS ISDN Messaging API Developer's Reference Manual*
isdnncc	• Uses a TCP with Natural Access to receive and/or place calls. • Demonstrates the ISDN TCP on a live trunk. • Provides an example of a two-way trunk application using the NCC service.	*NMS ISDN for Natural Call Control Developer's Manual*
itrace	Runtime filter for NMS ISDN messages from the stack that are sent to the OAM service.	*NMS ISDN Messaging API Developer's Reference Manual*
lapddemo	• Provides an example of a digital trunk application where the NMS ISDN software runs in a LAPD configuration. • Shows how to use the Messaging service to start this configuration, establish a data link, and send and receive Q.931 messages.	*NMS ISDN Messaging API Developer's Reference Manual*
imgtdemo	Shows how to use the NMS ISDN Management service to control and configure B channels. For use with PRI only.	*NMS ISDN Management API Developer's Reference Manual*

Trunk control program (TCP)

NMS ISDN software includes one trunk control program (TCP), *isd0.tcp*. This TCP is useful only if you are configuring the NMS ISDN software to access ISDN call control services in a channelized configuration.

The TCP is transferred to on-board memory by the OAM service. An instance of the TCP is associated with each context. The TCP is used to mediate transactions between Natural Access, DSP resources, and the NMS ISDN protocol stack. For more information, see the *NMS ISDN for Natural Call Control Developer's Manual*.

Parameter files

NMS ISDN software includes several files that contain parameters and values to configure the NMS ISDN TCP. Some of these parameters are country specific (different values are supplied for them depending upon the target country).

Country-specific parameter files are useful only if you are configuring the NMS ISDN software in channelized configuration. For more information, see the *NMS ISDN for Natural Call Control Developer's Manual*.

Three types of parameter files are installed with each NMS ISDN package for use with the NCC service:

File type and name	Description
*nccxadi**cty**.pf* *nccstart**cty**.pf* ***cty*** is the three character code of the target country. For example, the code for Australia is aus. Thus, the versions of these files for Australia are *nccxadiaus.pf* and *nccstartaus.pf*.	Binary parameter files containing a set of country-specific values for NCC service parameters. **Note:** Most of the values in these files should not be changed. Changing certain values may affect the regulatory approvals in the target country.
nccxisdn.pf	A binary parameter file containing a set of NMS ISDN parameters and default values. These values can safely be changed without affecting the regulatory approvals in the target country.
*nccxadi**cty**.par* *nccstart**cty**.par* *nccxisdn.par*	ASCII versions of the binary files.

For Natural Access to load the binary parameter file, both of the binary parameter files (*.pf* files) for the target country must be in one of the directories specified with the AGLOAD environment variable. *nccxisdn.pf* must also be in this directory.

When you install NMS ISDN, the installation program asks you to specify a default country. It creates copies of the country-specific parameter files for that country, renames them, and places them in the AGLOAD path, as follows:

These files...	In this operating system...	Are copied to...
*nccxadi**cty**.pf* *nccxadi**cty**.par* where ***cty*** is the code for the default country	Windows	*\nms\ag\cfg\nccxadi.pf* *\nms\ctaccess\cfg\nccxadi.par*
	UNIX	*/opt/nms/ag/cfg/nccxadi.pf* */opt/nms/ctaccess/cfg/nccxadi.par*
*nccstart**cty**.pf* *nccstart**cty**.par*	Windows	*nms\ag\cfg\nccstart.pf* *\nms\ag\ctaccess\nccstart.par*
	UNIX	*/opt/nms/ag/cfg/nccstart.pf* */opt/nms/ctaccess/cfg/nccstart.par*
nccxisdn.pf *nccxisdn.par*	Windows	*nms\ag\cfg\nccxisdn.pf* *\nms\ctaccess\cfg\nccxisdn.par*
	UNIX	*/opt/nms/ag/cfg/nccxisdn.pf* */opt/nms/ctaccess/cfg/nccxisdn.par*

Ensure that only one set of files is copied to the *AGLOAD* directory. If the directory contains two sets or parameter files, the parameters will not load.

Changing parameter values

To change parameter values in a *.pf* file, modify the value in the corresponding *.par* file. Then load the changes as follows:

Step	Action
1	Parse the *.par* file.
2	Set parameters in one of the following ways: • Call **ctaSetParmByName** for each parameter specified in the file, to set a new default value. For an example, see the **DemoLoadParameters** function in the demonstration library supplied with Natural Access. • Call **ctaLoadParameterFile** from within your application.

You can also use the *ctdaemon* program to set the parameters on a system-wide basis. See the *Natural Access Developer's Reference Manual* for more information on this method.

Parameter modification must take place before **nccStartProtocol** is called to start the TCP. When you call this function, specify parameters to configure the TCP.

For more information about parameters in the *nccxidsn.pf* parameter file, see the *NMS ISDN for Natural Call Control Developer's Manual*.

Related products

In addition to the NMS ISDN software, you need the following components to build an ISDN protocol application:

- One or more AG or CG boards with one or more trunk interfaces
- Natural Access
- NMS OAM
- *oamsys* and *oammon* - the board loading and monitoring programs

Natural Access

Natural Access is a complete development environment for voice applications. It provides a standard set of functions grouped into logical services. Each service has a standard programming interface.

For general information about installing and using Natural Access, see the Natural Access documentation.

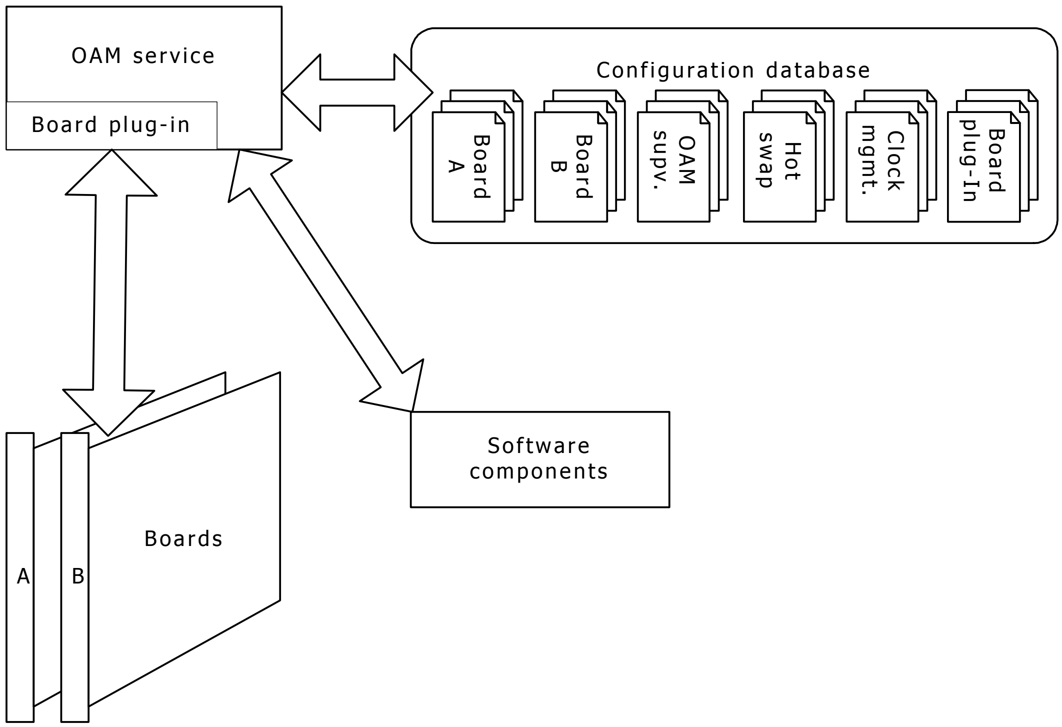
NMS OAM

NMS Operations, Administration, and Maintenance (OAM) manages and maintains the telephony resources in a system. These resources include hardware components, such as AG or CG boards, and low-level board management software modules.

Using NMS OAM, you can:

- Create, edit, delete, and query the configuration of a component
- Start, stop, and test a component
- Receive notifications from components

NMS OAM maintains a database containing records of configuration information for each component, as shown in the following illustration. This information consists of parameters and values.



Each NMS OAM database parameter and value is expressed as a keyword name and value pair (for example, Encoding = MU_LAW). You can query the NMS OAM database for keyword values for any component. Keywords and values can be added, modified, or deleted.

Note: Before using NMS OAM or any of its related utilities, ensure that the Natural Access Server (*ctdaemon*) is running. For more information about *ctdaemon*, refer to the *Natural Access Developer's Reference Manual*. For general information about NMS OAM and its utilities, refer to the *NMS OAM System User's Manual*.

NMS OAM and the configuration files

The NMS OAM system configuration file and referenced board keyword files contain information that *oamsys* reads to determine how to configure a board. These files contain information such as country-specific settings that must be tailored to describe your hardware and software setup.

When you set up your system, specify configuration information for all boards in your system with configuration files and board keyword files. These files specify

- Whether a board performs MVIP switching.
- Which board is the MVIP clock master.
- The software modules to transfer to the AG or CG board's memory on startup, including the TCPs to load.

The NMS ISDN software includes several sample files that describe configurations for different boards. Running the system with NMS OAM describes how to use the sample NMS OAM configuration files to create a file that configures your NMS ISDN software.

To initialize your boards based on the information in the configuration files, run the *oamsys* utility. *oamsys* transfers all software modules specified in the file to each board and performs other needed configuration activities. Run *oammon* to monitor boards for errors and other events. Use *oamcfg* to change system information or board parameters while the system is running.

oamsys, *oamcfg*, and *oammon* are installed with NMS OAM. For more information, see the *NMS OAM System User's Manual*.

Steps for developing an NMS ISDN application

To create an NMS ISDN application, follow these steps:

Step	Action	Where step is documented
1	Install AG or CG boards, and any other boards you need for the application in the system.	The hardware installation manual
2	Install Natural Access.	The Natural Access installation booklet
3	Install NMS ISDN software for each country or region where your application will be used.	The Natural Access installation booklet
4	Create the board keyword file for each board in the system.	• *Creating board keyword files* on page 22 • The hardware installation manuals • NMS OAM System User's Manual
5	Configure on-board software for NMS ISDN on the board keyword files.	*Configuring on-board software for NMS ISDN* on page 30
6	Configure data routing on the board keyword files.	• *Configuring data routing* on page 32 • The hardware installation manuals
7	Set up NFAS groups on the board keyword files.	• *Setting up NFAS groups* on page 32 • The hardware installation manuals
8	Create an OAM system configuration file that describes all boards in the system.	• *Creating a system configuration file for oamsys* on page 39 • The hardware installation manuals • NMS OAM System User's Manual
9	Run the *omasys* utility to set up the NMS OAM database.	• *Running oamsys* on page 40 • The hardware installation manuals • NMS OAM System User's Manual
10	Test the hardware installation.	• *Verifying the installation* on page 43 • The hardware installation manuals
11	Write the application.	• NMS ISDN for Natural Call Control Developer's Manual • NMS ISDN Messaging API Developer's Reference Manual • NMS ISDN Management API Developer's Reference Manual • The Natural Access documentation set

4 Preparing the system for ISDN development

Installing NMS ISDN software

Install NMS ISDN software from a package downloaded from the NMS Communications web site (www.nmscommunications.com). To install the software this way, follow the instructions on the web site to download and decompress the package.

Running the system with NMS OAM

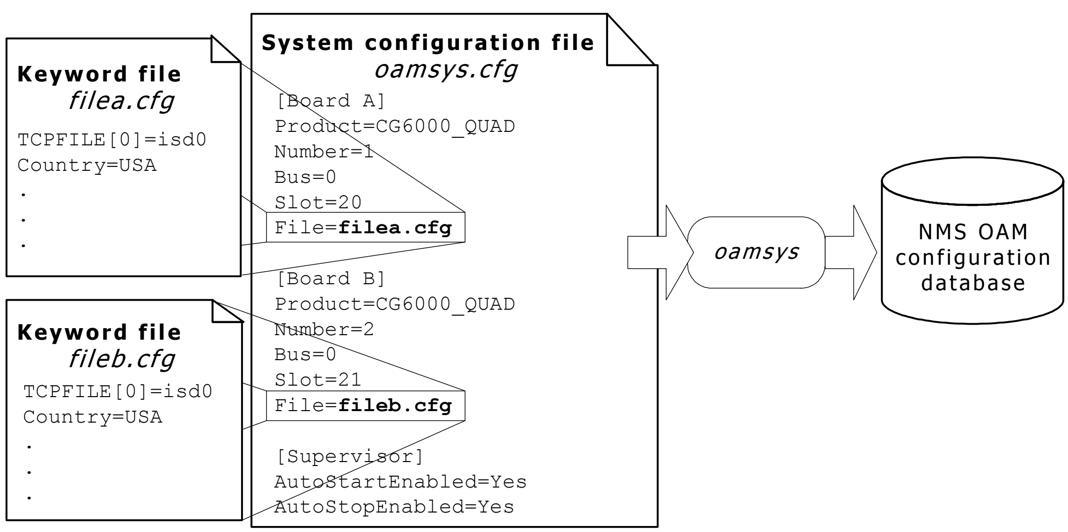
To be configured and started correctly, each board you install must have a separate set of configuration parameters and values entered into the NMS OAM configuration database. See *NMS OAM* on page 16 for details. Parameters and their assigned values are expressed as a keyword name and value pair (for example, Encoding = MU_LAW).

These configuration settings are specified in two types of files:

File type	Description
Board keyword	NMS OAM board keyword files contain parameters to configure the board. These settings are expressed as keyword name and value pairs. See *Creating board keyword files* on page 22.
System configuration	NMS OAM system configuration files contain a list of boards in the system and the name of one or more board keyword files for each board. See *Creating a system configuration file for oamsys* on page 39.

After you create and edit these files, load them into the NMS OAM configuration database by running the *oamsys* utility. Running oamsys describes this step.

The following illustration shows the process by which configuration files are sent to *oamsys* to create the NMS OAM configuration database:



After *oamsys* creates records in the NMS OAM configuration database, it starts and configures boards in your system according to the information now present in the database.

Creating board keyword files

A board keyword file contains a list of parameters and values used to configure a board. One board keyword file is assigned to each board. When *oamsys* runs, it creates a record for each board in the NMS OAM database.

NMS ISDN and Natural Access install a set of example board keyword files. To create a board keyword file for your setup, edit the appropriate example file installed on the system:

Operating system	Example file location
Windows	*\nms\ag\cfg*.cfg* *\nms\cg\cfg*.cfg*
UNIX	*/opt/nms/ag/cfg/*.cfg* */opt/nms/cg/cfg/*.cfg*

The sample board keyword files are named ***yyy****gi****x****zz**pi.cfg* (except for BRI and NFAS configurations), where:

- ***y*** is the board family (a if an AG board, c if a CG board, c61 if a CG 6100C board, c65 if a CG 6500C board)
- ***x*** is the board type (4 if an AG 4000 board, 6 if a CG 6000 board)
- ***zz*** is the line type (e1 or t1)

The following table lists some of the sample board keyword files:

File name	For use with...
agi4e1pi.cfg	AG 4000 E1 and AG 4040 E1
agi4t1pi.cfg	AG 4000 T1 and AG 4040 T1
cgi6e1.cfg	CG 6000 E1
cgi6t1.cfg	CG 6000 T1
aginfas.cfg	NFAS support for all AG T1 boards
cginfas.cfg	NFAS support for all CG T1 boards
agpi2bri.cfg	AG 2000-BRI

You can use the same board keyword file for multiple boards if the boards' configurations are identical (for example, both are CG 6000C T1 boards).

Sample board keyword files for the CG 6100C board and the CG 6500C board start with *c61*... and *c65*..., respectively.

For more information about board keyword files, refer to the *NMS OAM System User's Manual.*

Sample board keyword files

This topic presents sample board keyword files for NMS ISDN configurations. Use these sample files to better understand the format of and information in a board keyword file. These sample files are similar to the files provided with your software. This topic includes the following sample board keyword files:

- CG 6000C E1 board
- CG 6000C T1 board
- NFAS groups
- AG 4000 board
- AG 2000-BRI board

Sample configuration for a CG 6000C E1 board

The following sample board keyword file describes a single CG 6000C E1 board that is using the NOCC protocol:

```
#
# cgi6e1.cfg
#               CG 6000 configuration file
#
#     This file configures the board to run NMSVoice with NOCC.
#
#
Clocking.HBus.ClockMode                           = STANDALONE
Clocking.HBus.ClockSource                         = OSC
Clocking.HBus.ClockSourceNetwork                  = 1
TCPFiles                                          = nocc isd0
DSPStream.VoiceIdleCode[0..3]                     = 0x54
DSPStream.SignalIdleCode[0..3]                    = 0x09
Hdlc[0,3,6,9].Boot                                = YES
Hdlc[0,3,6,9].Comet.TxTimeSlot                    = 16
Hdlc[0,3,6,9].Comet.RxTimeSlot                    = 16
NetworkInterface.T1E1[0..3].Type                  = E1
NetworkInterface.T1E1[0..3].Impedance             = G703_120_OHM
NetworkInterface.T1E1[0..3].LineCode              = HDB3
NetworkInterface.T1E1[0..3].FrameType             = CEPT
NetworkInterface.T1E1[0..3].SignalingType         = PRI
NetworkInterface.T1E1[0..3].D_Channel             = ISDN
DSP.C5x[0..31].Libs[0]                            = cg6kliba
DSP.C5x[0..31].XLaw                               = A_LAW
DSP.C5x[1..31].Files = voice tone dtmf echo rvoice callp ptf wave oki ima
                       gsm_ms g726 mf
DSP.C5x[0].Files     = qtsignal tone dtmf echo NULL NULL NULL
Resource[0].Name     = RSC1
Resource[0].Size     = 120
Resource[0].TCPs     = nocc isd0
################################################################
# Before modifying this resource definition string refer to the CG6000
# Installation and Developers Manual.
#################################################################
Resource[0].Definitions
  = ( dtmf.det_all & echo.ln20_apt25 & ptf.det_2f & tone.gen &
callp.gnc & ptf.det_4f & \
( (rvoice.rec_mulaw & rvoice.play_mulaw) | \
  (rvoice.rec_alaw & rvoice.play_alaw) |    \
  (rvoice.rec_lin & rvoice.play_lin) |      \
  (voice.rec_16 & (voice.play_16_100 |      \
                   voice.play_16_150 |      \
                   voice.play_16_200)) |    \
  (voice.rec_24 & (voice.play_24_100 |      \
                   voice.play_24_150 |      \
                   voice.play_24_200)) |    \
  (voice.rec_32 & (voice.play_32_100 |      \
```

```
                   voice.play_32_150 |      \
                   voice.play_32_200)) |    \
  (voice.rec_64 & (voice.play_64_100 |      \
                   voice.play_64_150 |      \
                   voice.play_64_200)) |    \
  (wave.rec_11_16b & wave.play_11_16b) |    \
  (wave.rec_11_8b & wave.play_11_8b) |      \
  (oki.rec_24 & (oki.play_24_100 | oki.play_24_150 | oki.play_24_200)) | \
  (oki.rec_32 & (oki.play_32_100 | oki.play_32_150 | oki.play_32_200)) | \
  (ima.rec_24 & ima.play_24) | \
  (ima.rec_32 & ima.play_32) | \
  (gsm_ms.frgsm_rec & gsm_ms.frgsm_play) | \
  g726.rec_32 | g726.play_32) )

DLMFiles[0]                                        = cg6krun
DLMFiles[1]                                        = isdnetsi

# For other ISDN configurations uncomment one of the DMSFile[1] keywords
#DLMFiles[1]                                       = isdnvn6
#DLMFiles[1]                                       = isdnqsig
#DLMFiles[1]                                       = isdnaus1
#DLMFiles[1]                                       = isdnkor
DebugMask                                          = 0x0
```

Sample configuration for a CG 6000C T1 board

The following sample board keyword file describes a single CG 6000C T1 board using the NOCC protocol:

```
#
# cgi6t1.cfg
#               CG 6000 configuration file
#
#     This file configures the board to run NMSVoice with NOCC.
#
#
Clocking.HBus.ClockMode                           = STANDALONE
Clocking.HBus.ClockSource                         = OSC
Clocking.HBus.ClockSourceNetwork                  = 1
TCPFiles                                          = nocc isd0
DSPStream.VoiceIdleCode[0..3]                     = 0x7F
DSPStream.SignalIdleCode[0..3]                    = 0x00
Hdlc[0,3,6,9].Boot                                = YES
Hdlc[0,3,6,9].Comet.TxTimeSlot                    = 23
Hdlc[0,3,6,9].Comet.RxTimeSlot                    = 23
NetworkInterface.T1E1[0..3].Type                  = T1
NetworkInterface.T1E1[0..3].Impedance             = DSX1
NetworkInterface.T1E1[0..3].LineCode              = B8ZS
NetworkInterface.T1E1[0..3].FrameType             = ESF
NetworkInterface.T1E1[0..3].SignalingType         = PRI
NetworkInterface.T1E1[0..3].D_Channel             = ISDN
DSP.C5x[0..31].XLaw                               = MU_LAW
DSP.C5x[1..31].Files = voice tone dtmf echo rvoice callp ptf wave oki ima \
                       gsm_ms g726 mf
DSP.C5x[0].Files     = qtsignal tone dtmf echo NULL NULL NULL
Resource[0].Name     = RSC1
Resource[0].Size     = 120
Resource[0].TCPs     = nocc isd0
################################################################
# Before modifying this resource definition string refer to the CG6000
# Installation and Developers Manual.
#################################################################
Resource[0].Definitions
  = ( dtmf.det_all & echo.ln20_apt25 & ptf.det_2f & tone.gen &
callp.gnc & ptf.det_4f & \
( (rvoice.rec_mulaw & rvoice.play_mulaw) | \
  (rvoice.rec_alaw & rvoice.play_alaw) |    \
  (rvoice.rec_lin & rvoice.play_lin) |      \
  (voice.rec_16 & (voice.play_16_100 |      \
                   voice.play_16_150 |      \
```

```
                   voice.play_16_200)) |    \
  (voice.rec_24 & (voice.play_24_100 |      \
                   voice.play_24_150 |      \
                   voice.play_24_200)) |    \
  (voice.rec_32 & (voice.play_32_100 |      \
                   voice.play_32_150 |      \
                   voice.play_32_200)) |    \
  (voice.rec_64 & (voice.play_64_100 |      \
                   voice.play_64_150 |      \
                   voice.play_64_200)) |    \
  (wave.rec_11_16b & wave.play_11_16b) |    \
  (wave.rec_11_8b & wave.play_11_8b) |      \
  (oki.rec_24 & (oki.play_24_100 | oki.play_24_150 | oki.play_24_200)) | \
  (oki.rec_32 & (oki.play_32_100 | oki.play_32_150 | oki.play_32_200)) | \
  (ima.rec_24 & ima.play_24) | \
  (ima.rec_32 & ima.play_32) | \
  (gsm_ms.frgsm_rec & gsm_ms.frgsm_play) | \
  g726.rec_32 | g726.play_32) )

DLMFiles[0]                                       = cg6krun
DLMFiles[1]                                       = isdn4ess
# For other ISDN configurations uncomment one of the DMSFile[1] keywords
#DLMFiles[1]                                      = isdnqsig
#DLMFiles[1]                                      = isdn5ess
#DLMFiles[1]                                      = isdndms
#DLMFiles[1]                                      = isdnni2
#DLMFiles[1]                                      = isdnhkt
#DLMFiles[1]                                      = isdnntt
#DLMFiles[1]                                      = isdntwn

#DLMFiles[2]                                      = imgt
DebugMask                                         = 0x0
```

Sample configuration involving NFAS groups

The following sample board keyword file describes an CG T1 board configured for AT&T 4ESS ISDN variant, on a T1 trunk using ESF. The application is designed to access ISDN services in a channelized configuration using Natural Call Control.

There are two NFAS groups. Group 5 contains trunks 0 and 1. Group 3 contains trunks 2 and 3. Trunk 0 carries the D channel for Group 5. Trunk 2 carries the D channel for Group 3.

```
#
#
#     This file configures the board to run ISDN with NFAS
#     for CG 6000 board
#
#

TCPFiles                                         = nocc isd0

DLMFiles[0]                                      = cg6krun

# For USA ISDN configurations uncomment one of the DMSFile[1] keywords

DLMFiles[1]                                      = isdn4ess
#DLMFiles[1]                                     = isdnqsig
#DLMFiles[1]                                     = isdn5ess
#DLMFiles[1]                                     = isdndms
#DLMFiles[1]                                     = isdnni2

# Required if using IMGT service
#DLMFile[2]                                      = imgt

Clocking.HBus.ClockMode                          = STANDALONE
Clocking.HBus.ClockSource                        = OSC
Clocking.HBus.ClockSourceNetwork                 = 1
```

```
DSPStream.VoiceIdleCode[0..3]                    = 0x7F

DSPStream.SignalIdleCode[0..3]                   = 0x00

#
#   There are 3 HDLC controller cores per network interface.
#   PRI configurations use only one of the three HDLC cores on each network interface.
#   For PRI, configure Hdlc[x] where x is (network interface * 3)
#
Hdlc[0,3,6,9].Boot                               = YES
#
#   Hdlc[x] settings for E1
#
#Hdlc[0,3,6,9].Hardware.TxTimeSlot               = 16
#Hdlc[0,3,6,9].Hardware.RxTimeSlot               = 16
#
#   Hdlc[x] settings for T1
#
Hdlc[0,3,6,9].Hardware.TxTimeSlot                = 23
Hdlc[0,3,6,9].Hardware.RxTimeSlot                = 23

NetworkInterface.T1E1[0..3].Type                 = T1
NetworkInterface.T1E1[0..3].Impedance            = DSX1
NetworkInterface.T1E1[0..3].LineCode             = B8ZS
NetworkInterface.T1E1[0..3].FrameType            = ESF
#===============================================================================
#
#                   NFAS configuration
#
#===============================================================================
#
#  NFAS group information is associated with the trunk, on which the primary
#  D-channel is located. A unique number must be assigned to each NFAS group
#  for reference:
#
#     NetworkInterface.T1E1[x].ISDN.NFASGroup            - group number
#
#  For each NFAS group member (including primary D-channel) following
#  configuration information should be specified:
#
#     NetworkInterface.T1E1[x].ISDN.NFAS_Member[y].Board - board number
#     NetworkInterface.T1E1[x].ISDN.NFAS_Member[y].Trunk - trunk number (from 0)
#     NetworkInterface.T1E1[x].ISDN.NFAS_Member[y].NAI   - assigned NAI number
#                                                          unique within group
#  Note:
#     x - trunk number for primary D-channel
#     y - NFAS member number (starting with 0 without any gaps in numbering)
#
#===============================================================================
#===============================================================================
#   Group #5, D-channel on trunk 0
#
#   Members:
#      0 - NAI 4, Trunk 0, this board ( number 0 )
#      1 - NAI 6, Trunk 1, this board ( number 0 )
#
#===============================================================================
#
# Signaling settings for trunk with D-channel
#
NetworkInterface.T1E1[0].D_Channel                 = ISDN
NetworkInterface.T1E1[0].SignalingType             = PRI

#
# Signaling settings for trunk without D-channel
#
NetworkInterface.T1E1[1].D_Channel                 = ISDN_NONE
NetworkInterface.T1E1[1].SignalingType             = RAW

#
# Assigning group number (to D-channel trunk)
```

```
#
NetworkInterface.T1E1[0].ISDN.NFASGroup            = 5

#
# Configuring group members
#
NetworkInterface.T1E1[0].ISDN.NFAS_Member[0].Board = 0
NetworkInterface.T1E1[0].ISDN.NFAS_Member[0].Trunk = 0
NetworkInterface.T1E1[0].ISDN.NFAS_Member[0].NAI   = 4

NetworkInterface.T1E1[0].ISDN.NFAS_Member[1].Board = 0
NetworkInterface.T1E1[0].ISDN.NFAS_Member[1].Trunk = 1
NetworkInterface.T1E1[0].ISDN.NFAS_Member[1].NAI   = 6

#===============================================================================
#   Group #3, D-channel on trunk 2
#
#   Members:
#      0 - NAI 4, Trunk 2, this board ( number 0 )
#      1 - NAI 6, Trunk 3, this board ( number 0 )
#
#===============================================================================

#
# Signaling settings for trunk with D-channel
#
NetworkInterface.T1E1[2].D_Channel                 = ISDN
NetworkInterface.T1E1[2].SignalingType             = PRI

#
# Signaling settings for trunk without D-channel
#
NetworkInterface.T1E1[3].D_Channel                 = ISDN_NONE
NetworkInterface.T1E1[3].SignalingType             = RAW

#
# Assigning group number (to D-channel trunk)
#
NetworkInterface.T1E1[2].ISDN.NFASGroup            = 3

#
# Configuring group members
#
NetworkInterface.T1E1[2].ISDN.NFAS_Member[0].Board = 0
NetworkInterface.T1E1[2].ISDN.NFAS_Member[0].Trunk = 2
NetworkInterface.T1E1[2].ISDN.NFAS_Member[0].NAI   = 4

NetworkInterface.T1E1[2].ISDN.NFAS_Member[1].Board = 0
NetworkInterface.T1E1[2].ISDN.NFAS_Member[1].Trunk = 3
NetworkInterface.T1E1[2].ISDN.NFAS_Member[1].NAI   = 6
#===============================================================================


DSP.C5x[0..31].XLaw                              = MU_LAW
DSP.C5x[0].Files                                 = qtsignal
DSP.C5x[1..31].Files                             = voice tone dtmf echo rvoice
callp ptf wave oki ima gsm_ms g726 mf

Resource[0].Name                                 = RSC1
Resource[0].Size                                 = 120
Resource[0].TCPs                                 = nocc isd0

################################################################
# Before modifying this resource definition string refer to the CG6000
# Installation and Developers Manual.
#################################################################
Resource[0].Definitions         = ( dtmf.det_all & echo.ln20_apt25 & ptf.det_2f &
tone.gen & \
callp.gnc & ptf.det_4f & \
( (rvoice.rec_mulaw & rvoice.play_mulaw) | \
(rvoice.rec_alaw & rvoice.play_alaw) | \
```

```
(rvoice.rec_lin & rvoice.play_lin) | \
(voice.rec_16 & (voice.play_16_100 | voice.play_16_150 | voice.play_16_200)) | \
(voice.rec_24 & (voice.play_24_100 | voice.play_24_150 | voice.play_24_200)) | \
(voice.rec_32 & (voice.play_32_100 | voice.play_32_150 | voice.play_32_200)) | \
(voice.rec_64 & (voice.play_64_100 | voice.play_64_150 | voice.play_64_200)) | \
(wave.rec_11_16b & wave.play_11_16b) | \
(wave.rec_11_8b & wave.play_11_8b) | \
(oki.rec_24 & (oki.play_24_100 | oki.play_24_150 | oki.play_24_200)) | \
(oki.rec_32 & (oki.play_32_100 | oki.play_32_150 | oki.play_32_200)) | \
(ima.rec_24 & ima.play_24) | \
(ima.rec_32 & ima.play_32) | \
(gsm_ms.frgsm_rec & gsm_ms.frgsm_play) | \
g726.rec_32 | g726.play_32) )

DebugMask                                         = 0x0
```

Sample configuration for a single AG 4000 board

The following sample board keyword file describes a single AG 4000 board configured for the AT&T 4ESS ISDN variant, on a T1 trunk using extended superframe format (ESF). The application is designed to access ISDN services in a channelized configuration using Natural Call Control.

```
#---------------------------------------------------------------
# Originally created from input file ./agi4t1.cfg.
# Detailed board settings for:
#    Product = AG_4000_4T1
#---------------------------------------------------------------

TCPFiles[0] = nocc.tcp    # "no trunk control" protocol
TCPFiles[1] = isd0.tcp

DLMFiles[0] = gtp.leo
DLMFiles[1] = voice.leo
DLMFiles[2] = svc.leo

#   Uncomment the appropriate leo file

DLMFiles[3] = isdn4ess.leo    # use with "AT&T 4ESS"
# DLMFiles[3] = isdn5ess.leo  # use with "AT&T 5ESS"
# DLMFiles[3] = isdndms.leo   # use with "Northern Telecom DMS100"
# DLMFiles[3] = isdnhkt.leo   # use with "Hong Kong Telephone"
# DLMFiles[3] = isdnntt.leo   # use with "Nippon Telegraph Telephone"
# DLMFiles[3] = isdnni2.leo   # use with "US National ISDN 2"
# DLMFiles[3] = isdntwn.leo   # use with "Taiwanese Operator"

# DLMFiles[4] = imgt.leo      # required if using IMGT service

XLaw = MU-LAW

Clocking.HBus.ClockMode          = STANDALONE
Clocking.HBus.ClockSource        = NETWORK
Clocking.HBus.ClockSourceNetwork = 1

DSP.C5x[0..15].Files = callp ptf tone dtmf voice

NetworkInterface.T1E1[0..3].Type          = T1
NetworkInterface.T1E1[0..3].SignalingType = PRI
NetworkInterface.T1E1[0..3].FrameType     = ESF    # Extended SuperFrame
NetworkInterface.T1E1[0..3].LineCode      = AMI
NetworkInterface.T1E1[0..3].D_Channel     = ISDN
```

Sample configuration for an AG 2000-BRI board

The following sample board keyword file describes the configuration for an AG 2000-BRI board using the NOCC protocol:

```
#
#       AG Plug-in Config File for AG 2000 BRI
#

# TCP files are shipped with the NMS CAS sub-package of Natural Access.
# Be sure to install the protocols that are specified below before
# trying to start a board with this configuration file.

 TCPFiles[0] = nocc.tcp           # "no trunk control" protocol
 TCPFiles[1] = isd0.tcp           # ISDN protocol

# DSP (.m54) files to link in

 DSP.C5x.DSPFiles = callp.m54 dtmf.m54 mf.m54 ptf.m54 signal.m54 tone.m54 \ voice.m54


 DLMFiles[0] = gtp.leo
 DLMFiles[1] = voice.leo
 DLMFiles[2] = svc.leo
 DLMFiles[3] = isdnbri.leo   # use with "Euro ISDN with major ETSI variants"

 XLaw = A-LAW                # A-Law silence, idle signaling bit code

 RunFile = ag2bri.cor

 Clocking.HBus.ClockSource = NETWORK
 Clocking.HBus.ClockMode = STANDALONE
```

Configuring on-board software for NMS ISDN

To configure the NMS ISDN software with the boards, include the following keywords in each board keyword file:

- DSP.C5x[x].Files
- DLMFiles[x]
- TCPFiles[x]

DSP.C5x[x].Files

This statement specifies the DSP program files to be loaded to the board. The following files must be assigned through this keyword for all NMS ISDN installations:

- *dtmf.**xxx**,* or *dtmf_a.**xxx*** for A-law configurations

  Note: Substitute *dtmfe.**xxx*** or *dtmfe_a.**xxx*** if you are using echo cancellation.

- *callp.**xxx**,* or *callp_a.**xxx*** for A-law configurations
- *tone.**xxx**,* or *tone_a.**xxx*** for A-law configurations

  where ***xxx*** is the three-letter file extension the board requires:

Board	File extension
AG 2000/C, AG 4000/C, and AG 4040/C	*.m54*
CG family	*.f54*

To run the demonstration programs supplied with Natural Access and with your NMS ISDN software, specify *voice.**xxx***, or *voice_a.**xxx*** for A-law configurations.

Because these keyword statements apply to all boards in an NMS ISDN system, specify these DSP file assignments in each board keyword file.

DLMFiles[x]

This statement specifies the run module to be downloaded to the board. Run modules are specific to the protocol variant and country. Because the DLMFiles[***x***] keyword applies to all boards in an NMS ISDN system, specify this value in each board keyword file. See *Variant specifications* on page 45 for more information about variants.

Download only one run module to a particular board.

PRI run modules

The following table lists the PRI run modules.

Note: Europe includes the following countries: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Russia, Spain, Sweden, Switzerland, UK.

PRI run module for AG boards	PRI run module for CG boards	PRI run module for CG 6500C boards	Variant	Country
isdngen.leo	*isdngen.dlm*	*cg65igen.dlm*	ANSI T1.607 AT&T 4ESS AT&T 5ESS National ISDN 2 Northern Telecom DMS100/250	USA
			EuroISDN	Europe and other countries, where supported
			France Telecom VN6	France
			Australian Telecom 1	Australia
			Hong Kong Telephone	Hong Kong
			Korean Operator	Korea
			Nippon Telegraph Telephone	Japan
			Taiwanese Operator	Taiwan
			ECMA-QSIG	All listed countries
dpnss.leo	*dpnss.dlm*	*c65idpnss.dlm*	DPNSS	United Kingdom

BRI run modules

The following table lists the BRI run modules.

Note: Europe includes the following countries: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Russia, Spain, Sweden, Switzerland, UK.

BRI run module	Variant	Country
isdnbri.leo	Euro ISDN France Telecom VN6	Europe, China, Singapore, Australia France

TCPFiles[x]

The TCPFiles[***x***] keyword specifies a TCP to be downloaded to the board. For NMS ISDN installations, specify one of the following TCPs according to the layer from which the application accesses ISDN services:

Specify this TCP...	If...
isd0.tcp	NMS ISDN protocol stack runs in channelized configuration.
nocc.tcp	NMS ISDN protocol stack runs in ACU or LAPD configuration. *nocc.tcp* sets up your system for call control to be handled by the host application instead of by the TCP.

Because this keyword applies to all boards in an NMS ISDN system, specify this value in each board keyword file.

For more information about the ISDN TCP file, see *Trunk control program (TCP)* on page 14.

Configuring data routing

Use the NetworkInterface.T1E1[***x***].SignalingType (where ***x*** is the number of a trunk) keyword to configure routing of voice and signaling information between trunks and DSPs. Set this keyword as follows:

- If you are not setting up NFAS groups, set NetworkInterface.T1E1[***x***].SignalingType to PRI (or BRI, if applicable) for each board in your system. This setting routes D channel information on each trunk to the HDLC controllers and routes B channel information to DSP resources.
- If you are setting up NFAS groups, set NetworkInterface.T1E1[***x***].SignalingType as follows:

If the...	Set NetworkInterface.T1E1[*x*].SignalingType to...
Trunk carries D channel	PRI or BRI. This setting routes D channel information on the trunk to the HDLC controllers and routes B channel information to DSP resources.
Trunk does not carry D channel	RAW. This setting routes B channel information only.

If MVIP switching is enabled, routing does not take place regardless of the NetworkInterface.T1E1[***x***].SignalingType setting. Instead, your application must make the appropriate MVIP switch settings, as described in the *NMS ISDN for Natural Call Control Developer's Manual* and *NMS ISDN Messaging API Developer's Reference Manual*.

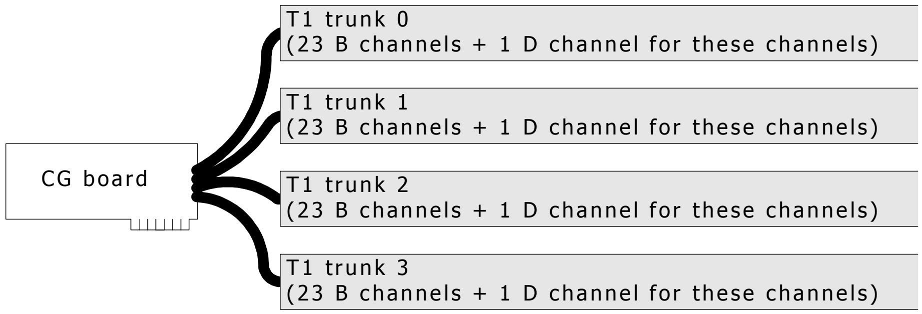
Setting up NFAS groups

In systems with multiple T1 ISDN trunks, a non-facility associated signaling (NFAS) configuration can be used. This topic describes:

- Non-facility associated signaling (NFAS)
- NFAS groups with hot swap configurations
- Specifying NFAS groups in board keyword files
- D channel backup

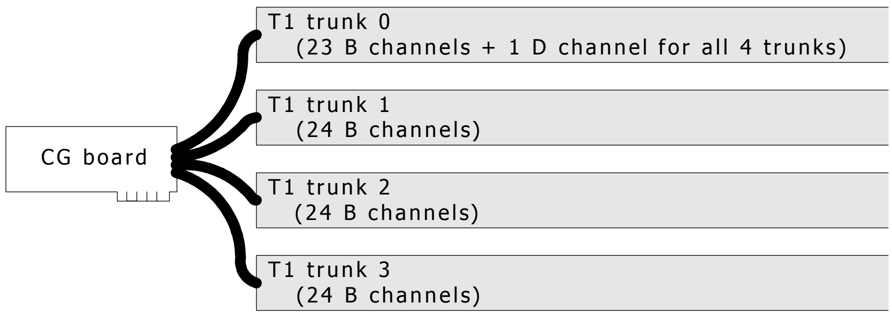
Data on a T1 trunk is transmitted in channels. For primary rate ISDN, T1 carries 24 channels. With primary-rate ISDN:

- 23 of the 24 channels carry voice, audio, data, and video signals. These channels are called bearer channels (B channels).
- On a T1 trunk, one channel carries signaling for all B channels. This is called the D channel. On T1 trunks, the D channel is typically carried in channel 24. See the following illustration:

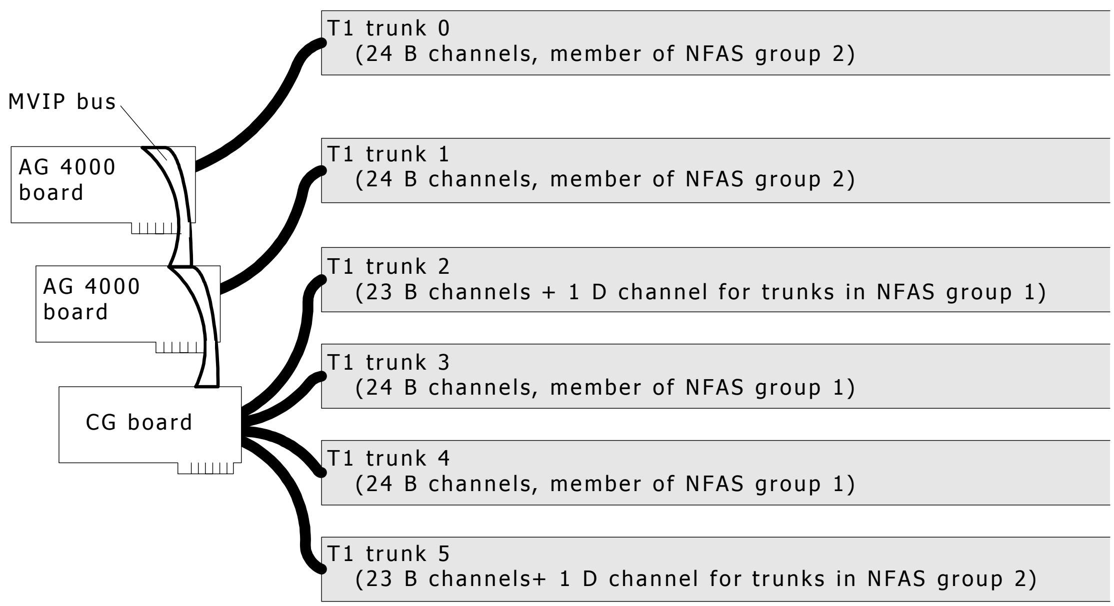


Non-facility associated signaling (NFAS)

In an NFAS configuration, trunks are grouped into one or more NFAS groups. One of the trunks in each group has a D channel carrying the signaling for all of the B channels on all of the trunks in the group, up to 20 trunks per group. This leaves channel 24 free on all other trunks in the NFAS group. This extra channel can be used as another B channel, as shown in the following illustration:



A single NFAS group can contain trunks from multiple boards, as shown in the following illustration:



If your application uses NMS ISDN software in a channelized configuration, all trunks in an NFAS group must be on the same board. An NFAS group cannot contain trunks from multiple boards. This restriction does not apply to the ACU or LAPD configurations.

NFAS groups with hot swap configurations

If an NFAS group spans multiple boards in a Hot Swap system and you remove or insert a board while the system is running, the NMS ISDN protocol stacks on the other boards are not affected. If you remove a board containing only B channels, the stack on the D channel board does not sense that the board is missing. It is the application's responsibility to sense this change and take appropriate action (to not access B channels on that board).

Specifying NFAS groups in board keyword files

To group T1 trunks into NFAS groups, your board keyword file must contain information regarding the NFAS-specific tasks the trunks will perform. To accomplish this, add the keywords described in this topic to your board keyword file for the trunk that carries the primary D channel.

When you set up NFAS groups, make sure to correctly set the NetworkInterface.T1E1[***x***].SignalingType keyword for each trunk, as described in *Configuring data routing* on page 32.

Note: These keywords are required only if you are setting up NFAS groups.

NetworkInterface.T1E1[x].ISDN.NFAS_Group

This keyword specifies the NFAS group number, as follows:

```
NetworkInterface.T1E1[x].ISDN.NFAS_Group= group_no
```

where ***x*** is the trunk that carries the primary D channel and ***group_no*** is the NFAS group number. There is a maximum of 16 groups.

This keyword must be accompanied by one or more groups of NetworkInterface.T1E1[***x***].ISDN.NFAS_Member[***y***].Board and NetworkInterface.T1E1[***x***].ISDN.NFAS_Member[***y***].Trunk statements to specify the board and trunk numbers for each member of this NFAS group.

NetworkInterface.T1E1[x].D_Channel

This keyword specifies the trunk that carries the primary D channel, as follows:

```
NetworkInterface.T1E1[x].D_Channel= ISDN
```

where ***x*** is the trunk that carries the primary D channel. In each NFAS group, only one trunk (***x***) can carry the D channel, unless D channel backup is used. Refer to *D channel backup* on page 37.

If this keyword is set to ISDN for a trunk, the NetworkInterface.T1E1[***x***].SignalingType keyword (described in *Configuring data routing* on page 32) must be set to PRI or BRI for that trunk. If a trunk is part of an NFAS group but does not carry a D channel, NetworkInterface.T1E1[***x***].SignalingType must be set to RAW for that trunk. Specify the backup D channel using the keyword NetworkInterface.T1E1[***x***].ISDN.D_Channel_Backup_Trunk, where ***x*** is the trunk that carries the primary D channel.

NetworkInterface.T1E1[x].ISDN.NFAS_Member[y].Board

This keyword specifies the board number of a member trunk of an NFAS group.

Specify one of these statements for every NFAS member (***y***) on every trunk (***x***) that carries a primary D channel. This field must be set in the board keyword file for the board where the D channel resides. The board number must match the board number specified in the OAM system configuration file, *oamsys.cfg.*

NetworkInterface.T1E1[x].ISDN.NFAS_Member[y].NAI

This keyword identifies the network access identifier (NAI) of a member trunk of an NFAS group, as follows:

```
NetworkInterface.T1E1[2].ISDN.NFAS_Member[1].NAI= nai
```

where ***nai*** is the NAI of the trunk and ranges from 0 to 127. The NAI of each trunk in an NFAS group must be unique.

Specify one of these statements for every NFAS member (***y***) on every trunk (***x***) that carries a primary D channel. This field must be set in the board keyword file for the board where the D channel resides.

NetworkInterface.T1E1[x].ISDN.NFAS_Member[y].Trunk

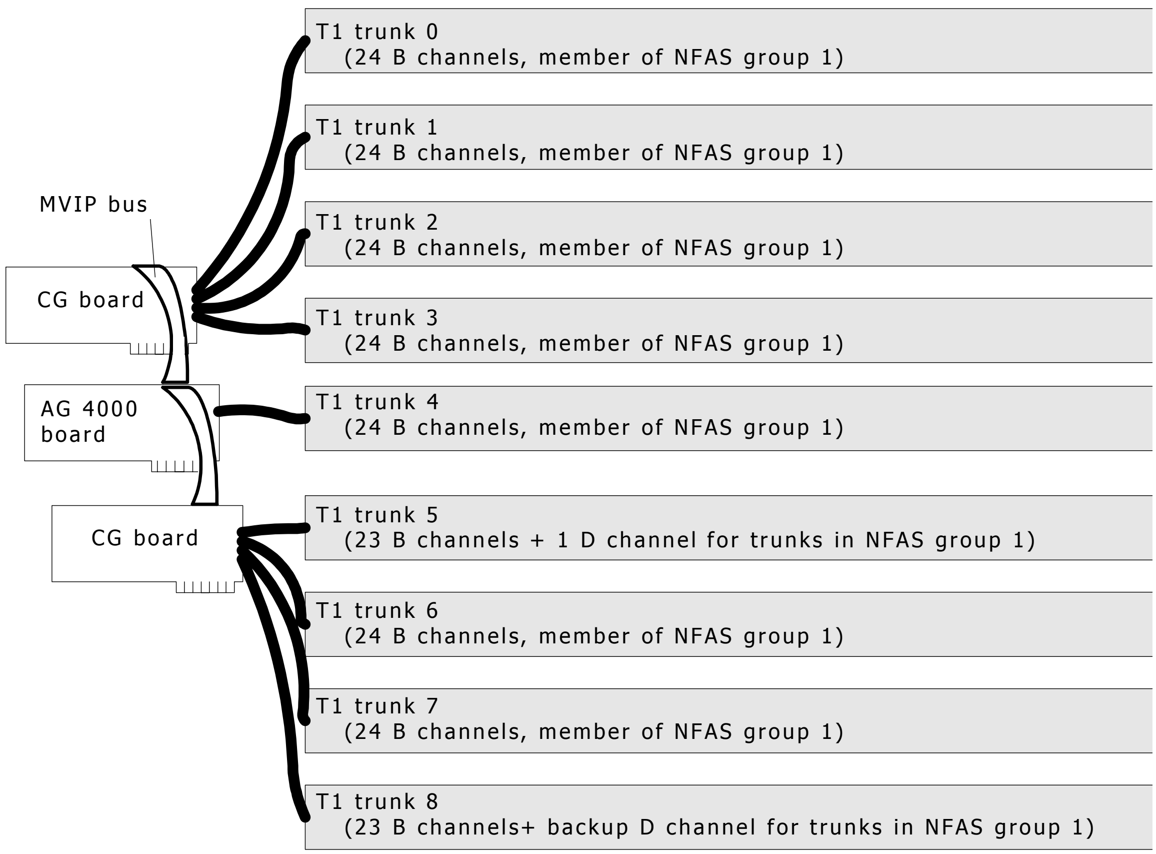
This keyword specifies the trunk number of a member trunk of an NFAS group.

Specify one of these statements for every NFAS member (***y***) on every trunk (***x***) that carries a primary D channel. This field must be set in the board keyword file for the board where the D channel resides.

D channel backup

When NFAS is used, the reliability of the signaling performance for the ISDN interfaces controlled by the D channel can be improved by creating a standby D channel, the D channel backup. The D channel backup feature transfers most of the signaling information to the backup D channel, enabling continued access to the ISDN network if the primary D channel fails.

The backup D channel must exist on a different trunk than the primary D channel, as shown in the following illustration:



At any point in time, only one D channel conveys B channel signaling information. The other D channel remains in a standby mode and is only active at the LAPD layer (layer 2). While the backup D channel is on standby, any layer 3 messages received on it are ignored.

Neither D channel can serve as a B channel while designated as a backup D channel. Each primary/backup D channel pair provides signaling only for the set of B channels within a specific NFAS group and cannot backup any other D channels in a different group.

When both D channels are out of service, the first D channel has priority as the channel to carry call control signaling. If the first D channel cannot be established, the backup D channel is chosen.

To configure D channel backup, edit the board keyword file as described in *Configuring data routing* on page 32. Then set the keywords as follows:

For trunk(s) with...	**Set NetworkInterface.T1E1[*x*].D_CHANNEL to...**	**Set NetworkInterface.T1E1[*x*].SignalingType to...**
Primary D channel	ISDN	PRI or BRI
Backup D channel	ISDN_NONE	PRI or BRI
Only B channels	ISDN_NONE	RAW

Both primary and backup D channels must be defined on the same board and belong to the same NFAS group. Set NetworkInterface.T1E1[***x***].ISDN.SignalingType to PRI (or BRI, where applicable) for both D channels.

Set the NetworkInterface.T1E1[***x***].ISDN.D_Channel_Backup_Trunk keyword to enable D channel backup. For example:

```
NetworkInterface.T1E1[x].ISDN.D_Channel_Backup_Trunk = trunkno
```

where ***trunkno*** is the number of the trunk to perform D channel backup.

When you start ISDN with Natural Access, you must enable option t309 in the ISDN_PROTOCOL_PARMS_Q931CC or ISDN_PROTOCOL_PARMS_CHANNELIZED structure, which appears as follows:

```
WORD t309;                /* T309 in use flag */
```

To enable this option, set this parameter to 1 in your application. When enabled, active calls are preserved while switching to the backup D channel when the primary D channel fails. For more information on this feature, refer to the *NMS ISDN Messaging API Developer's Reference Manual*.

Creating a system configuration file for oamsys

When you finish creating the board keyword files, create a system configuration file. The system configuration file lists the type and location of each of the boards and assigns a board keyword file to each board. *oamsys* uses this file to create records in the OAM database for each of the boards.

The following table describes the settings to include in the file for each board:

Keyword	Description
[***name***]	Name used to refer to the board in software. This name must be unique.
Product	Name of the board product (for example, CG_6000C_QUAD).
Number	Board number you use in the Natural Access application to refer to the board.
Bus	PCI bus number. The bus:slot location for each board must be unique.
Slot	PCI slot number. The bus:slot location for each board must be unique.
File	Name of the board keyword file containing settings for the board.

You can also specify keyword and value pairs directly in the system configuration file instead of in separate board keyword files. This method is often useful if the board configurations are identical except for one or two parameters (such as clocking information).

If you need to determine configuration information for the boards, use the *pciscan* utility. This utility identifies the NMS Communications PCI boards installed in the system, and returns each board's bus, slot, interrupt, and board type. For more information about *pciscan*, refer to the *NMS OAM System User's Manual.*

The system configuration file you create must be named *oamsys.cfg*. This is the file name *oamsys* looks for by default.

Sample system configuration file

The following sample system configuration file describes a CG 6000C board and an AG 4000C board:

```
[My_CG_Board]
Product = CG_6000C_QUAD
Number  = 0
Bus     = 0
Slot    = 15
File    = c:\nms\cg\cfg\mycgbrd.cfg

Clocking.HBus.ClockMode   = MASTER_A
Clocking.HBus.ClockSource = OSC


[My_4000_Board]
Product = AG_4000C_2T1
Number  = 1
Bus     = 0
Slot    = 16
File    = c:\nms\ag\cfg\my40brd.cfg

Clocking.HBus.ClockMode   = SLAVE
Clocking.HBus.ClockSource = A_CLOCK
```

The CG 6000C board is assigned board number 0 and is located at bus 0, slot 15. It is assigned a keyword file named *mycgbrd.cfg*. It is set up as the primary H.100 bus clock master, using its on board oscillator as its timing reference.

The AG 4000C board is assigned board number 1 and is located at bus 0, slot 16. It is assigned a keyword file named *my40brd.cfg*. It is set up as a slave to the primary master (the CG 6000C board).

Running oamsys

After you create all necessary configuration files, you are ready to use the *oamsys* utility. To run *oamsys*, enter the following command from the command line:

```
oamsys options
```

where

Option	Description
-f ***filename***	Specifies the file name (and path, if necessary) of a system configuration file to load. If you invoke *oamsys* without this option, it searches for a file named *oamsys.cfg* in the current directory, and then in the paths specified in the AGLOAD environment variable. If you specify a file name without an extension, *oamsys* assumes the extension to be *.cfg*.
-@ ***host***	Loads the configuration file on the specified resource host. ***host*** is an IP address or machine name. If unspecified, the operations are performed on the host on which the utility was initialized.

oamsys reads system configuration files, not board keyword files. Board keyword files to be added to the NMS OAM database must be specified within the system configuration file. Refer to *Creating a system configuration file for oamsys* on page 39.

When you invoke *oamsys* with a valid file name, *oamsys* performs the following tasks:

- Checks the syntax of the system configuration file, and ensures that all required keywords are present. *oamsys* reports any syntax errors it finds.

  Note: *oamsys* verifies the syntax of the system configuration file, but not of the board keyword files.

- Checks for uniqueness of board names, board numbers, bus numbers, and slot numbers within the system configuration file.
- Shuts down all boards referenced in the NMS OAM database.
- Deletes all board configuration information currently stored in the NMS OAM database.
- Sets up the NMS OAM database according to settings in the system configuration file and any referenced board keyword files.
- Attempts to start all boards according to the description in the database.

Natural Access Server (*ctdaemon*) must be running for *oamsys* to operate. For more information about Natural Access Server (*ctdaemon)*, refer to the *Natural Access Developer's Reference Manual.*

Changing configuration parameter settings

After initializing the database with *oamsys*, you can make further parameter changes as follows:

- Modify the board keyword file for the board, make sure the name is correctly specified in the File statement in *oamsys.cfg*, and run *oamsys* again.
- Specify parameter settings using the *oamcfg* utility. For information about this utility, refer to the *NMS OAM System User's Manual*.
- Specify the settings using the OAM service functions. See the *NMS OAM Service Developer's Reference Manual* for more information.
- Set individual keywords using the *oaminfo* utility. For information about this utility, refer to the *NMS OAM System User's Manual*.

5 Verifying NMS ISDN installation

Verifying the installation

After you install and configure the NMS ISDN software, verify that the software is operational before you continue.

When running the NMS ISDN verification procedure, it is assumed that:

- Natural Access is properly installed
- Your boards are installed
- Your configuration files have been edited to reflect your configuration

To verify your installation, run the *isdndemo* demonstration program supplied with your NMS ISDN software.

Running isdndemo

Perform the following steps to run *isdndemo* and verify the NMS ISDN installation:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Create an NMS OAM system configuration file and board keyword files specific to the installation.</td></tr>
<tr><td>2</td><td>Invoke oamsys to initialize the board.
Note: oammon should be left running at all times to capture hardware errors.</td></tr>
<tr><td>3</td><td>Change to the directory containing isdndemo:
Windows: \nms\ctaccess\demos\isdndemo\
UNIX: /opt/nms/ctaccess/demos/isdndemo/</td></tr>
<tr><td>4</td><td>Run isdndemo by entering the following command at the command line:
<code>isdndemo -p variant</code>
where variant is the protocol variant to run.
For example:
<code>isdndemo -p 23</code>
isdndemo responds with a message similar to the following:
<code>ISDN Multiple Thread Demo V.9 (Nov 14, 1999)
T1 board 0 found.
Protocol = AT&T 5E10
Number of inbound ports = 23
Number of outbound ports = 0
24 ports opened.
NOCC protocol started on 23 ports.
Started 23 threads...
Enter program test loop...</code></td></tr>
</table>

6 Variant specifications

PRI variants and specifications

The following table lists variants and specifications for the PRI interface:

Network protocol variant	Specification
AT&T 4ESS (AT4)	TR 41459 (June 1999), PRI or BRI only
AT&T 5ESS10 (E10)	Custom now edited by LUCENT AT&T 235-900-342 (January 1996): PRI or BRI
Northern Telecom DMS-100 (DMS)	NIS A211-1 Standard 08.01: PRI or BRI
Bellcore National 2 (NI2)	SR-3887 (November 1996): PRI or BRI
France Telecom Euro ISDN and Euro Numeris (VN6)	ETS 300 102-1 (December 1990) + Addendum ETS 300 103-1/A2 (October 1993) CSE P 10-21A (June 1994): French deltas
NTT INS 1500 (NTT)	INS-NET-64 (March 1993)
Hong Kong (HKT)	HKTA 2015 Issue 1 (1996)
Korea (KOR)	Similar to ETS. There is no Korean specification, but ETSI specifications have been implemented in NMS ISDN with some changes requested by Korean Samsung.
Taiwan (TWN)	ME 0200-2 May 1997, edited by Chunghwa Telecom Co. Ltd.
Australia (AUS)	TS-013.1 1990 and TS-014.2 1990
Signaling at the Q reference point (QSIG)	ECMA 143 (June 1997)
ANSI T1.607 (ANSI)	T1.607-2000

BRI variants and specifications

The following table lists variants and specifications for the BRI interface:

Network protocol variant	Specification
Euro ISDN	ETS 300 102-1 (December 1990) and Amendment ETS 300 102-1/A2 (October 1993)
France Telecom VN6	CSE P 10-21A (June 1994)
NTT INS 1500 (NTT)	INS-NET-64 (March 1993)

Other common channel signaling protocols

Network protocol variant	Specification
DPNSS	BTNR 188 Issue 6, January 1995

Index