



Global Call API for Linux and Windows Operating Systems

Demo Guide

September 2002



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

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

This Global Call API for Linux and Windows Operating Systems Demo Guide as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without express written consent of Intel Corporation.

Copyright © 2002 Intel Corporation. All Rights Reserved.

AlertVIEW, AnyPoint, AppChoice, BoardWatch, BunnyPeople, CablePort, Celeron, Chips, CT Media, Dialogic, DM3, EtherExpress, ETOX, FlashFile, i386, i486, i960, iCOMP, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Create&Share, Intel GigaBlade, Intel InBusiness, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel Play, Intel Play logo, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel TeamStation, Intel Xeon, Intel XScale, IPLink, Itanium, LANDesk, LanRover, MCS, MMX, MMX logo, Optimizer logo, OverDrive, Paragon, PC Dads, PC Parents, PDCharm, Pentium, Pentium II Xeon, Pentium III Xeon, Performance at Your Command, RemoteExpress, Shiva, SmartDie, Solutions960, Sound Mark, StorageExpress, The Computer Inside., The Journey Inside, TokenExpress, Trillium, VoiceBrick, Vtune, and Xircom are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others.

Publication Date: September 2002

Document Number: 05-1818-001

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

For **Technical Support**, visit the Intel Telecom Support Resources website at:

<http://developer.intel.com/design/telecom/support/>

For **Products and Services Information**, visit the Intel Communications Systems Products website at:

<http://www.intel.com/network/csp/>

For **Sales Offices** and other contact information, visit the Intel Telecom Building Blocks Sales Offices page at:

<http://www.intel.com/network/csp/sales/>



Contents

	About This Publication	5
1	Demo Description	7
2	System Requirements	9
	2.1 Hardware Requirements	9
	2.2 Software Requirements	9
3	Preparing to Run the Demo	11
	3.1 Connecting to External Equipment	11
	3.2 Editing Configuration Files	11
	3.3 Compiling and Linking	13
4	Running the Demo	15
	4.1 Starting the Demo	15
	4.2 Using the Demo	18
	4.3 Stopping the Demo	21
5	Demo Details	23
	Glossary	25
	Index	27

Tables

1	Files Used by the Global Call API Demo	23
---	--	----



About This Publication

The following topics provide information about this publication:

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

Purpose

This publication provides information about the Global Call API demonstration program provided with the Global Call software and gives instructions for setting up and running the program.

Intended Audience

This publication is written for the following audience:

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

How to Use This Publication

Refer to this publication after you have installed the hardware and the Intel® Dialogic® system software that includes the Global Call software.

This publication assumes that you understand computer telephony terms and concepts, and are familiar with the Linux or Windows operating system and the C programming language.

The information in this guide is organized as follows:

- [Chapter 1, “Demo Description”](#) provides a brief overview of the Global Call API demo.
- [Chapter 2, “System Requirements”](#) discusses the hardware and software required to run the demo.
- [Chapter 3, “Preparing to Run the Demo”](#) lists the procedures you must follow before running the demo.

- [Chapter 4, “Running the Demo”](#) describes the steps to run the demo and how to stop the demo.
- [Chapter 5, “Demo Details”](#) provides additional information about the demo, such as the files used by the demo.

Related Information

Refer to the following documents and web site for more information on developing applications using the Global Call software:

- *Global Call API Programming Guide* - describes the Global Call software and provides guidelines for building applications that use the Global Call software
- *Global Call API Library Reference* - provides reference information on all Global Call functions and parameters
- *Global Call E-1/T-1 Technology User's Guide* - provides technology-specific information when using the Global Call software with E-1 or T-1 interfaces
- *Global Call ISDN Technology User's Guide* - provides technology-specific information when using the Global Call software with ISDN interfaces
- *Global Call SS7 Technology User's Guide* - provides technology-specific information when using the Global Call software with SS7 interfaces
- *System Release Guide* - provides information on the Intel® Dialogic system software, system requirements, software and hardware features, supported hardware, and release documentation
- *System Release Update* (available on the Technical Support web site only) - describes compatibility issues, restrictions and limitations, known problems, and late-breaking updates or corrections to the release documentation
- <http://developer.intel.com/design/telecom/support/> - Technical Support web site that contains developer support information, downloads, release documentation, technical notes, application notes, a user discussion forum and more

This chapter describes the capabilities of the Global Call API demo program.

The Global Call API demo program sets up and tears down calls on the boards and channels specified by the user. The program uses the Global Call basic call state model. Using the Global Call API demo program configuration file, the user can specify:

- The channels to be used by the demo
- The protocol to be used by each device
- The protocol type (inbound or outbound) for each device
- The voice resource associated with each device if a voice resource is required by the technology
- The phone number to associate with each device

When the Global Call API demo program is run, one device waits for calls while another device makes calls. The sequence of function calls, events received, and the call states are displayed as the program proceeds.

When the user presses `Ctrl-C` to interrupt the process, the program prints a summary of the activity including information such as, the total number of inbound calls, the total number of outbound calls, the amount of time the demo program was running.



This chapter describes the requirements for running the Global Call API demo program. Topics include:

- [Hardware Requirements](#) 9
- [Software Requirements](#) 9

2.1 Hardware Requirements

To run the Global Call API demo program, you need:

- At least one board that supports the Global Call software. The board may be a board that supports Analog, E-1/T-1, ISDN or SS7 technology.
- If you are using Analog interfaces, a ring generator such as a Teltone is required.

Note: The Global Call API demo program supports DM3 and Springware boards.

2.2 Software Requirements

To run the Global Call API demo, you need the Intel® Dialogic® system software which includes the Global Call software.

Note: To ensure that the Global Call software is installed as part of the system software, see the software installation documentation for your system release.

When running the demo program on boards containing Analog, E-1/T-1, or ISDN interfaces, the appropriate protocol software must also be installed. For information on installing protocols, see one of the following:

- *Global Call Analog Technology User's Guide*
- *Global Call E-1/T-1 Technology User's Guide*
- *Global Call ISDN Technology User's Guide*

The Global Call Protocols Package, a separately orderable package, contains all the protocols that the Global Call software supports.

For a list of operating system requirements and supported compilers, see the Release Guide for the Intel® Dialogic system software.

This chapter provides information on preparations to follow before running the Global Call API demo. Topics include:

- Connecting to External Equipment 11
- Editing Configuration Files 11
- Compiling and Linking 13

3.1 Connecting to External Equipment

To run the Global Call API demo program, you need one or more of the following:

- For protocols other than Analog, a back-to-back connection is required. For Analog protocols, a ring generator such as a Teltone is required.
- A technology simulator.

Two boards can be connected back to back or a board can be connected to a technology simulator. Connections can be made either before or after installing the Intel® Dialogic® system software that includes the Global Call software.

3.2 Editing Configuration Files

In both Linux and Windows environments, you must edit the demo configuration file to customize it to your specific configuration.

Note: In Linux environments, there is also a UNIX makefile, but it is preconfigured for operation in a Linux environment and does **not** need to be modified.

The executable demo programs use the *gc_basic_call_model.cfg* configuration file. You must edit this configuration file before running the demo program using a standard text editor to include the protocols and products used by your application.

The protocol and resource information for each channel and the telephone number dialed (up to 24 digits) are defined in these configuration files. The configuration is specified in the following order:

1. Network device
2. Protocol name
3. Direction
4. Voice device (for technologies that require a voice resource to make a call)
5. Phone number

A digital network interface is not used for an analog call.

Note: Do not enter a digital interface for an analog board or channel. Use the value “NONE” instead.

The following is an extract from the *gc_basic_call_model.cfg* file:

```
#####
# For ICAPI Protocols
# Board #1
dtiB1T1      ar_r2_i      In      dxxxB1C1      1234567
#
# Board #2
dtiB2T1      ar_r2_o      Out     dxxxB9C1      1234567
#####
```

The first uncommented line specifies that time slot 1 on digital network interface board 1 (dtiB1T1) is the inbound channel (In), running the Argentina for ICAPI inbound protocol (ar_r2_i), using the voice resource on virtual board 1, channel 1 (dxxxB1C1), with an associated phone number (1234567).

The second uncommented line specifies that time slot 1 on digital network interface board 2 (dtiB2T1) is the outbound channel (Out), running the Argentina for ICAPI outbound protocol (ar_r2_o), using the voice resource on virtual board 9, channel 1 (dxxxB9C1), with an associated phone number (1234567).

Example of Configuration File

The example configuration file (*gc_basic_call_model.cfg*) shown below configures one channel on one T-1 span to handle inbound calls using the United States T-1 for ICAPI inbound protocol (*us_mf_i*) and one channel on one T-1 span to handle outbound calls using the United States T-1 for ICAPI outbound protocol (*us_mf_o*).

Note: No voice resources are dedicated to either digital interface.

```
#####
#      gc_basic_call_model.cfg      #
#      DEFAULT CONFIGURATION FILE   #
#####
#
#
#####
#NetworkDev      ProtName      Direction      VoiceDevice(If any)      Phone #
# CAUTIONS: NONE is case sensitive
#      Follow the case sensitivity as provided in the sample fields
#      All fields must be used, including the phone # for the inbound direction
#####

#####
# For PDK protocols
# Board #1
#Caution: For PDK protocols: Outbound channels must be listed before the
#associated Inbound channel
#This is a timing issue related to doing back-to-back testing using one application
#and the unmodified, cdp protocol file as shipped.
#This is NOT an issue if you have both an inbound and outbound app where the inbound
#side starts before the outbound side OR your application is attached to a live trunk
#
```

```
#dtiB1T1      pdk_ar_r2_io      Out      dxxxB1C1      1234567
#
# Board #2
#dtiB2T1      pdk_ar_r2_io      In       dxxxB2C1      1234567
#####

#####
# For ICAPI Protocols
# Board #1
dtiB1T1      us_mf_i      In      dxxxB1C1      1234567
#dtiB1T1      ar_r2_i      In      dxxxB1C1      1234567
#
# Board #2
dtiB2T1      us_mf_o      Out      dxxxB2C1      1234567
#dtiB2T1      ar_r2_o      Out      dxxxB2C1      1234567
#####

#####
# For ISDN Protocols
# Board #1
#dtiB1T1      ISDN      In      NONE      1234567
#
# Board #2
#dtiB2T1      ISDN      Out      NONE      1234567
#####
# For SS7 protocols
# Board #1
#dtiB1T1      SS7      In      dxxxB1C1      1234567
#
# Board #2
#dtiB2T1      SS7      Out      dxxxB2C1      1234567
#####

#####
# For PDK Analog protocols
# Board #1
#NONE      pdk_na_an_io      In      dxxxB73C1      102
#
# Board #2
#NONE      pdk_na_an_io      Out      dxxxB73C2      101
#####

#####
# For ANAPI protocols
#Board #1
#NONE      na_an_io      In      dxxxB1C1      102
#
# Board #2
#NONE      na_an_io      Out      dxxxB2C2      101
#####
```

3.3 Compiling and Linking

To compile the demonstration program using edited configuration files, follow these instructions:

1. While logged on to the system with root privileges, change to the following installation directory:
 - Linux
/usr/dialogic/demos/gc_demos

- Windows
\Program Files\Dialogic\Samples\gc_demos
2. To compile the program, type the following command and press enter:
- Linux
`make all`
 - Windows
`nmake -f makefile.win32 all`
- The demo program compiles.

This chapter describes how to run the Global Call API demo program. Topics include:

- Starting the Demo 15
- Using the Demo 18
- Stopping the Demo 21

4.1 Starting the Demo

Start the Global Call API demo program by typing the program name at the command line. The program name is:

```
gc_basic_call_model
```

- Notes:**
1. A protocol package must be installed on the system prior to running the Global Call API demo program. The configuration file must specify an installed protocol. Refer to the documentation accompanying the separately orderable Global Call Protocols Package for information on installing protocols.
 2. Before running the demonstration program with a T-1 robbed bit protocol and the ICAPI call control library, disable the DTI Wait Call firmware function by changing the default selection in the *icapi.cfg* file. See the **\$14** parameter description in the “ICAPI Configuration File” chapter of the *Global Call E-1/T-1 Technology User's Guide* for details.
 3. When using PDK protocols, disconnecting and reconnecting the cable causes the demo to exit.

The Global Call API demo program displays the status of the program as it runs. The following is an example of the output generated on a Linux system:

```
./gc_basic_call_model
[MISC]:          SRL mode ID set to SR_POLLMODE
gc_basic_call_model_B1T1.log successfully opened
gc_basic_call_model_B2T1.log successfully opened
[MISC]:          ***** GC DEMO - BASIC CALL MODEL *****

[GC_APICALL]: gc_Start(startp = NULL) Success
[MISC]:          Call Control Library Status:
                GC_ICAPI_LIB - available
                GC_ISDN_LIB - available
                GC_ANAPI_LIB - available
                GC_PDKRT_LIB - available
                GC_DM3CC_LIB - available
                GC_SS7_LIB - is not available for use
                GC_IPM_LIB - is not available for use
                GC_CUSTOM1_LIB - configured
                GC_CUSTOM2_LIB - configured

[MISC]:          E1 or T1 device being opened
[GC_APICALL]: gc_OpenEx(devicename=:N_dtiB1T1:P_ar_r2_i:V_dxxxB1C1, mode=EV_SYNC) Success
[MISC]:          E1 or T1 device being opened
[GC_APICALL]: gc_OpenEx(devicename=:N_dtiB2T1:P_ar_r2_o:V_dxxxB9C1, mode=EV_SYNC) Success
[MISC]:          gc_SetCallingNum(linedev=15, phone_num = 7654321) Success
```

```

[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_UNBLOCKED
[STATE]:         GCST_NULL is the current GC call state
[GC_APICALL]:    gc_WaitCall(linedev=7, crnp=NULL, waittime=0, mode=EV_ASYNC) Success
[STATE]:         GCST_NULL is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_UNBLOCKED
[STATE]:         GCST_NULL is the current GC call state
[GC_APICALL]:    gc_MakeCall(linedev=15, numberstr=1234567, mode=EV_ASYNC) Success
[STATE]:         GCST_NULL is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_OFFERED
[STATE]:         GCST_NULL is the current GC call state
[GC_APICALL]:    gc_GetDNIS(crn=0x1000007) Success - dnis = 1234
[GC_APICALL]:    gc_GetANI(crn=0x1000007) Success - ANI = 7654321
[GC_APICALL]:    gc_AcceptCall(crn=0x1000007, mode=EV_ASYNC) Success
[STATE]:         GCST_OFFERED is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_ALERTING
[STATE]:         GCST_NULL is the current GC call state
[STATE]:         GCST_ALERTING is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_ACCEPT
[STATE]:         GCST_OFFERED is the current GC call state
[GC_APICALL]:    gc_AnswerCall(crn=0x1000007, mode=EV_ASYNC) Success
[STATE]:         GCST_ACCEPTED is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_ANSWERED
[STATE]:         GCST_ACCEPTED is the current GC call state
[STATE]:         GCST_CONNECTED is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_CONNECTED
[STATE]:         GCST_ALERTING is the current GC call state
[MISC]:          call connected - call progress not applicable
[MISC]:          gc_DropCall() will be issued in 1-2 seconds
[STATE]:         GCST_CONNECTED is the new GC call state after processing the event
[MISC]:          ***** Dropping outbound call from drop_outbound_calls_if_required() *****
[GC_APICALL]:    gc_DropCall(crn=0x100000f, cause=GC_NORMAL_CLEARING, mode=EV_ASYNC) Success
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_DISCONNECTED
[STATE]:         GCST_CONNECTED is the current GC call state
[GC_APICALL]:    gc_DropCall(crn=0x1000007, cause=GC_NORMAL_CLEARING, mode=EV_ASYNC) Success
[STATE]:         GCST_DISCONNECTED is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_DROP_CALL
[STATE]:         GCST_DISCONNECTED is the current GC call state
[GC_APICALL]:    gc_ReleaseCallEx(crn=0x1000007, EV_ASYNC) Success
[STATE]:         GCST_IDLE is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_RELEASE_CALL
[STATE]:         GCST_IDLE is the current GC call state
[STATE]:         GCST_NULL is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_DROP_CALL
[STATE]:         GCST_CONNECTED is the current GC call state
[GC_APICALL]:    gc_ReleaseCallEx(crn=0x100000f, EV_ASYNC) Success
[STATE]:         GCST_IDLE is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_RELEASE_CALL
[STATE]:         GCST_IDLE is the current GC call state
[GC_APICALL]:    gc_MakeCall(linedev=15, numberstr=1234567, mode=EV_ASYNC) Success
[STATE]:         GCST_NULL is the new GC call state after processing the event
[MISC]:          ***** Received a GC event *****
[EVENT]:         GCEV_OFFERED
[STATE]:         GCST_NULL is the current GC call state
[GC_APICALL]:    gc_GetDNIS(crn=0x1000007) Success - dnis = 1234
[GC_APICALL]:    gc_GetANI(crn=0x1000007) Success - ANI = 7654321

```



```

[GC_APICALL]: gc_AcceptCall(crn=0x1000007, mode=EV_ASYNC) Success
[STATE]:      GCST_OFFERED is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_ALERTING
[STATE]:      GCST_NULL is the current GC call state
[STATE]:      GCST_ALERTING is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_ACCEPT
[STATE]:      GCST_OFFERED is the current GC call state
[GC_APICALL]: gc_AnswerCall(crn=0x1000007, mode=EV_ASYNC) Success
[STATE]:      GCST_ACCEPTED is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_ANSWERED
[STATE]:      GCST_ACCEPTED is the current GC call state
[STATE]:      GCST_CONNECTED is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_CONNECTED
[STATE]:      GCST_ALERTING is the current GC call state
[MISC]:      call connected - call progress not applicable
[MISC]:      gc_DropCall() will be issued in 1-2 seconds
[STATE]:      GCST_CONNECTED is the new GC call state after processing the event
[MISC]:      ***** Dropping outbound call from drop_outbound_calls_if_required() *****
[GC_APICALL]: gc_DropCall(crn=0x100000f, cause=GC_NORMAL_CLEARING, mode=EV_ASYNC) Success
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_DISCONNECTED
[STATE]:      GCST_CONNECTED is the current GC call state
[GC_APICALL]: gc_DropCall(crn=0x1000007, cause=GC_NORMAL_CLEARING, mode=EV_ASYNC) Success
[STATE]:      GCST_DISCONNECTED is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_DROPCALL
[STATE]:      GCST_DISCONNECTED is the current GC call state
[GC_APICALL]: gc_ReleaseCallEx(crn=0x1000007, EV_ASYNC) Success
[STATE]:      GCST_IDLE is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_RELEASECALL
[STATE]:      GCST_IDLE is the current GC call state
[STATE]:      GCST_NULL is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_DROPCALL
[STATE]:      GCST_CONNECTED is the current GC call state
[GC_APICALL]: gc_ReleaseCallEx(crn=0x100000f, EV_ASYNC) Success
[STATE]:      GCST_IDLE is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_RELEASECALL
[STATE]:      GCST_IDLE is the current GC call state
[GC_APICALL]: gc_MakeCall(linedev=15, numberstr=1234567, mode=EV_ASYNC) Success
[STATE]:      GCST_NULL is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_OFFERED
[STATE]:      GCST_NULL is the current GC call state
[GC_APICALL]: gc_GetDNIS(crn=0x1000007) Success - dnis = 1234
[GC_APICALL]: gc_GetANI(crn=0x1000007) Success - ANI = 7654321
[GC_APICALL]: gc_AcceptCall(crn=0x1000007, mode=EV_ASYNC) Success
[STATE]:      GCST_OFFERED is the new GC call state after processing the event
[MISC]:      ***** Received a GC event *****
[EVENT]:      GCEV_ALERTING
[STATE]:      GCST_NULL is the current GC call state
[STATE]:      GCST_ALERTING is the new GC call state after processing the event
*****Received User Interrupted Signal*****
*****Received User Interrupted Signal*****
[MISC]:      ***** Program Exiting *****
[MISC]:      The total no of Inbound calls on this device is: 2
[MISC]:      The total no of Inbound calls over all devices is: 2
[MISC]:      The total no of Outbound calls over all devices is: 2

```

```
[MISC]:      The total duration taken by the test is:  0.70 minutes
[MISC]:      The total no of Outbound calls on this device is:  2
[MISC]:      The total no of Inbound calls over all devices is:  2
[MISC]:      The total no of Outbound calls over all devices is:  2
[MISC]:      The total duration taken by the test is:  0.70 minutes
```

Also, see [Section , “Example of Inbound Channel Activity Log”](#), on page 18 and [Section , “Example of Outbound Channel Activity Log”](#), on page 20 for more information.

4.2 Using the Demo

The Global Call API demo program provides a trace of the activity on each channel as it runs. No user interaction is required. The information is captured in an activity log, one for each channel being used. The log files are named according to the following convention *gc_basic_call_model_bxty.log*, where *bx* is the board and *ty* is the channel number of the channel being used. Examples of the output generated by the demo are described in the following topics:

- [Example of Inbound Channel Activity Log](#)
- [Example of Outbound Channel Activity Log](#)

Example of Inbound Channel Activity Log

The activity log for the inbound channel of the demo program run shown in [Section 4.1, “Starting the Demo”](#), on page 15 is shown below:

```
06/25 21:09:25.817 [MISC]:      ***** GC DEMO - BASIC CALL MODEL *****

06/25 21:09:26.578 [GC_APICALL]:  gc_Start(startp = NULL) Success
06/25 21:09:26.578 [MISC]:      Call Control Library Status:
                        GC_ICAPI_LIB - available
                        GC_ISDN_LIB - available
                        GC_ANAPI_LIB - available
                        GC_PDKRT_LIB - available
                        GC_DM3CC_LIB - available
                        GC_SS7_LIB - is not available for use
                        GC_IPM_LIB - is not available for use
                        GC_CUSTOM1_LIB - configured
                        GC_CUSTOM2_LIB - configured

06/25 21:09:26.588 [MISC]:      E1 or T1 device being opened
06/25 21:09:27.640 [GC_APICALL]:  gc_OpenEx(devicename=:N_dtiB1T1:P_ar_r2_i:V_dxxxB1C1,
                        mode=EV_SYNC) Success

06/25 21:09:27.920 [MISC]:      ***** Received a GC event *****
06/25 21:09:27.930 [EVENT]:      GCEV_UNBLOCKED
06/25 21:09:27.950 [STATE]:      GCST_NULL is the current GC call state
06/25 21:09:27.970 [GC_APICALL]:  gc_WaitCall(linedev=5, crnp=NULL, waittime=0,
                        mode=EV_ASYNC) Success
06/25 21:09:28.000 [STATE]:      GCST_NULL is the new GC call state after processing
                        the event
```

```

06/25 21:09:30.925 [MISC]:          ***** Received a GC event *****
06/25 21:09:30.945 [EVENT]:          GCEV_OFFERED
06/25 21:09:30.965 [STATE]:          GCST_NULL is the current GC call state
06/25 21:09:30.985 [GC_APICALL]:      gc_GetDNIS(crn=0x1000005) Success - dnis = 1234
06/25 21:09:31.005 [GC_APICALL]:      gc_GetANI(crn=0x1000005) Success - ANI = 7654321
06/25 21:09:31.025 [GC_APICALL]:      gc_AcceptCall(crn=0x1000005, mode=EV_ASYNC) Success
06/25 21:09:31.045 [STATE]:          GCST_OFFERED is the new GC call state after
                                processing the event

06/25 21:09:42.752 [MISC]:          ***** Received a GC event *****
06/25 21:09:42.762 [EVENT]:          GCEV_ACCEPT
06/25 21:09:42.782 [STATE]:          GCST_OFFERED is the current GC call state
06/25 21:09:42.812 [GC_APICALL]:      gc_AnswerCall(crn=0x1000005, mode=EV_ASYNC) Success
06/25 21:09:42.822 [STATE]:          GCST_ACCEPTED is the new GC call state after
                                processing the event

06/25 21:09:42.862 [MISC]:          ***** Received a GC event *****
06/25 21:09:42.882 [EVENT]:          GCEV_ANSWERED
06/25 21:09:42.902 [STATE]:          GCST_ACCEPTED is the current GC call state
06/25 21:09:42.922 [STATE]:          GCST_CONNECTED is the new GC call state after
                                processing the event

06/25 21:09:45.155 [MISC]:          ***** Received a GC event *****
06/25 21:09:45.175 [EVENT]:          GCEV_DISCONNECTED
06/25 21:09:45.195 [STATE]:          GCST_CONNECTED is the current GC call state
06/25 21:09:45.215 [GC_APICALL]:      gc_DropCall(crn=0x1000005, cause=GC_NORMAL_CLEARING,
                                mode=EV_ASYNC) Success
06/25 21:09:45.255 [STATE]:          GCST_DISCONNECTED is the new GC call state after
                                processing the event

06/25 21:09:45.285 [MISC]:          ***** Received a GC event *****
06/25 21:09:45.305 [EVENT]:          GCEV_DROP_CALL
06/25 21:09:45.325 [STATE]:          GCST_DISCONNECTED is the current GC call state
06/25 21:09:45.345 [GC_APICALL]:      gc_ReleaseCallEx(crn=0x1000005, EV_ASYNC) Success
06/25 21:09:45.365 [STATE]:          GCST_IDLE is the new GC call state after processing
                                the event

06/25 21:09:45.395 [MISC]:          ***** Received a GC event *****
06/25 21:09:45.415 [EVENT]:          GCEV_RELEASE_CALL
06/25 21:09:45.436 [STATE]:          GCST_IDLE is the current GC call state
06/25 21:09:45.456 [STATE]:          GCST_NULL is the new GC call state after processing
                                the event

06/25 21:09:48.490 [MISC]:          ***** Received a GC event *****
06/25 21:09:48.510 [EVENT]:          GCEV_OFFERED
06/25 21:09:48.530 [STATE]:          GCST_NULL is the current GC call state
06/25 21:09:48.550 [GC_APICALL]:      gc_GetDNIS(crn=0x1000005) Success - dnis = 1234
06/25 21:09:48.570 [GC_APICALL]:      gc_GetANI(crn=0x1000005) Success - ANI = 7654321
06/25 21:09:48.590 [GC_APICALL]:      gc_AcceptCall(crn=0x1000005, mode=EV_ASYNC) Success
06/25 21:09:48.600 [STATE]:          GCST_OFFERED is the new GC call state after
                                processing the event

.
.
.
06/25 21:13:00.847 [MISC]:          ***** Program Exiting *****
06/25 21:13:00.867 [MISC]:          The total no of Inbound calls on this device is: 12
06/25 21:13:00.887 [MISC]:          The total no of Inbound calls over all devices is: 12
06/25 21:13:00.897 [MISC]:          The total no of Outbound calls over all devices is: 12
06/25 21:13:00.917 [MISC]:          The total duration taken by the test is: 3.58 minutes

```

Example of Outbound Channel Activity Log

The activity log for the outbound channel of the demo program run described in [Section 4.1, “Starting the Demo”](#), on page 15 is shown below:

```
06/25 21:09:25.817 [MISC]:          ***** GC DEMO - BASIC CALL MODEL *****

06/25 21:09:26.578 [GC_APICALL]:  gc_Start(startp = NULL) Success
06/25 21:09:26.578 [MISC]:          Call Control Library Status:
                                GC_ICAPI_LIB - available
                                GC_ISDN_LIB - available
                                GC_ANAPI_LIB - available
                                GC_PDKRT_LIB - available
                                GC_DM3CC_LIB - available
                                GC_SS7_LIB - is not available for use
                                GC_IPM_LIB - is not available for use
                                GC_CUSTOM1_LIB - configured
                                GC_CUSTOM2_LIB - configured

06/25 21:09:27.640 [MISC]:          E1 or T1 device being opened
06/25 21:09:27.850 [GC_APICALL]:  gc_OpenEx(devicename=:N_dtiB2T1:P_ar_r2_o:V_dxxxB9C1,
                                mode=EV_SYNC) Success
06/25 21:09:27.860 [MISC]:          gc_SetCallingNum(linedev=12, phone_num = 7654321)
                                Success

06/25 21:09:28.041 [MISC]:          ***** Received a GC event *****
06/25 21:09:28.061 [EVENT]:        GCEV_UNBLOCKED
06/25 21:09:28.071 [STATE]:        GCST_NULL is the current GC call state
06/25 21:09:28.101 [GC_APICALL]:  gc_MakeCall(linedev=12, numberstr=1234567,
                                mode=EV_ASYNC) Success
06/25 21:09:28.131 [STATE]:        GCST_NULL is the new GC call state after processing
                                the event

06/25 21:09:31.385 [MISC]:          ***** Received a GC event *****
06/25 21:09:31.405 [EVENT]:        GCEV_ALERTING
06/25 21:09:31.425 [STATE]:        GCST_NULL is the current GC call state
06/25 21:09:31.435 [STATE]:        GCST_ALERTING is the new GC call state after
                                processing the event

06/25 21:09:42.952 [MISC]:          ***** Received a GC event *****
06/25 21:09:42.972 [EVENT]:        GCEV_CONNECTED
06/25 21:09:42.992 [STATE]:        GCST_ALERTING is the current GC call state
06/25 21:09:43.012 [MISC]:          call connected - call progress not applicable
06/25 21:09:43.022 [MISC]:          gc_DropCall() will be issued in 1-2 seconds
06/25 21:09:43.042 [STATE]:        GCST_CONNECTED is the new GC call state after
                                processing the event

06/25 21:09:45.085 [MISC]:          ***** Dropping outbound call from
                                drop_outbound_calls_if_required() *****
06/25 21:09:45.115 [GC_APICALL]:  gc_DropCall(crn=0x100000c, cause=GC_NORMAL_CLEARING,
                                mode=EV_ASYNC) Success

06/25 21:09:45.496 [MISC]:          ***** Received a GC event *****
06/25 21:09:45.516 [EVENT]:        GCEV_DROP_CALL
06/25 21:09:45.526 [STATE]:        GCST_CONNECTED is the current GC call state
06/25 21:09:45.546 [GC_APICALL]:  gc_ReleaseCallEx(crn=0x100000c, EV_ASYNC) Success
06/25 21:09:45.566 [STATE]:        GCST_IDLE is the new GC call state after processing
                                the event
```

```

06/25 21:09:45.606 [MISC]:      ***** Received a GC event *****
06/25 21:09:45.626 [EVENT]:    GCEV_RELEASECALL
06/25 21:09:45.636 [STATE]:    GCST_IDLE is the current GC call state
06/25 21:09:45.666 [GC_APICALL]: gc_MakeCall(linedev=12, numberstr=1234567,
                                mode=EV_ASYNC) Success
06/25 21:09:45.696 [STATE]:    GCST_NULL is the new GC call state after processing
                                the event

06/25 21:09:48.951 [MISC]:      ***** Received a GC event *****
06/25 21:09:48.961 [EVENT]:    GCEV_ALERTING
06/25 21:09:48.981 [STATE]:    GCST_NULL is the current GC call state
06/25 21:09:49.001 [STATE]:    GCST_ALERTING is the new GC call state after
                                processing the event

.
.
.

06/25 21:13:00.847 [MISC]:      ***** Program Exiting *****
06/25 21:13:00.927 [MISC]:      The total no of Outbound calls on this device is: 12
06/25 21:13:00.957 [MISC]:      The total no of Inbound calls over all devices is: 12
06/25 21:13:00.967 [MISC]:      The total no of Outbound calls over all devices is: 12
06/25 21:13:00.987 [MISC]:      The total duration taken by the test is: 3.58 minutes

```

4.3 Stopping the Demo

The Global Call API demo runs continuously. You can press Ctrl-C at any time to exit the Global Call API demo. All channels and files are properly closed by the demo and a summary of the activity during the session is displayed.



This chapter provides more detail about the Global Call API program.

Table 1 lists the files used by the Global Call API demo. The directories in which these files are found vary according to the operating system as follows:

- In a Linux environment, the files are located in *usr/dialogic/demos/gc_demos/gc_basic_call_model*
- In a Windows environment, the files are located in *\Program Files\Dialogic\Samples\gc_demos\gc_basic_call_model*

Table 1. Files Used by the Global Call API Demo

File Name	Purpose
dxchan.vcp	Used to demonstrate the use of the gc_LoadDxParm() function, which sets voice parameters associated with a line device that operates as a dedicated or shared resource
gc_basic_call_model.c	Demo program source code
gc_basic_call_model.cfg	Demo program configuration file
gc_basic_call_model.exe	Demo program executable in Windows
makefile.win32	Windows makefile
makefile	Linux makefile





Glossary

activity log: A file used to record activity on a channel, such as changes in state, as the demo runs.

configuration file: A file that enables the demo user to customize demo parameters including, the boards and channels to be used, the protocol, phone numbers etc.

ICAPI protocol: The Dialogic® Interface Control Application Programming Interface. Provides a device specific telephony and signaling interface for the Global Call API to control Intel® Dialogic network interface boards using E-1 CAS or T-1 Robbed Bit signaling schemes. Also the name of a Global Call call control library. This library cannot be accessed directly.

inbound call: A call received by the local end point from a remote end point.

makefile: A software project management file that is used to determine which parts of a program to compile.

PDK protocol: A protocol developed using the Intel® Dialogic Protocol Development Kit (PDK). PDK protocols supersede ICAPI and other older protocols.

protocol: A set of rules that apply to the signaling between the end points in a connection that has been established for communication. Both end points must recognize and observe a protocol. Protocols are often described in an industry or international standard.

outbound call: A call made by the local end point to a remote end point.

technology simulator: Test equipment used to simulate a specific technology such as, E-1 or T-1.



A

- activity log
 - file 18
 - inbound 18
 - outbound 20

C

- compiling the demo 13
- configuration file
 - editing 11
- connecting
 - external equipment 11

D

- demo
 - configuration file 12
 - description 7
 - files used 23
 - makefiles 13
 - physical connections 11
 - preparation to run 11
 - prerequisites 13
 - running 15
 - starting 15
 - stopping 21
 - system requirements 9
- DTI Call Wait
 - disabling 15

E

- external equipment
 - connecting 11

F

- files used by the demo 23

G

- gc_basic_call_model.cfg
 - configuration file 11
 - example 12

H

- hardware requirements 9

L

- log file
 - inbound channel activity 18
 - naming convention 18
 - outbound channel activity 20

R

- recompiling the demo 13
- related information 6
- requirements
 - hardware 9
 - software 9

S

- sample
 - configuration file 12
 - demo output 15
- software requirements 9
- stopping the demo 21
- system requirements 9
 - hardware 9
 - software 9

