Dialogic®

Dialogic® Global Call API

Demo Guide

May 2008

05-2439-003

Copyright © 2002-2008 Dialogic Corporation. All rights reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation.

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.

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.**

Dialogic, Dialogic Pro, Brooktrout, Cantata, SnowShore, Eicon, Eicon Networks, Eiconcard, Diva, SIPcontrol, Diva ISDN, TruFax, Realblocs, Realcomm 100, NetAccess, Instant ISDN, TRXStream, Exnet, Exnet Connect, EXS, ExchangePlus VSE, Switchkit, N20, Powering The Service-Ready Network, Vantage, Connecting People to Information, Connecting to Growth, Making Innovation Thrive, and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic. 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. Microsoft, NetMeeting, and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries. Other names of actual companies and products mentioned herein are the trademarks of their respective owners.

Publication Date: May 2008

Document Number: 05-2439-003

Contents

Tables

Revision History

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

Document No.	Publication Date	Description of Revisions
05-2439-003	May 2008	Made global changes to reflect Dialogic brand and changed title to "Dialogic® Global Call API Demo Guide."
05-2439-002	July 2005	Starting the Demo section: Updated the instructions for starting the demo.
05-2439-001	April 2005	Initial version of document. Much of the information contained in this document was previously published in the *Dialogic® Global Call API Demo Guide*, document number 05-1818-002.

About This Publication

The following topics provide information about this publication:

Purpose

This publication provides information about the Dialogic® Global Call API demonstration program available with the Dialogic® Host Media Processing (HMP) Software. This guide describes the demo, its requirements and details how it works.

Applicability

This document version (05-2439-003) is published for Dialogic® Host Media Processing Software Release 3.1LIN.

This document may also be applicable to later software releases (including service updates) on Linux or Windows®. Check the Release Guide for your software release to determine whether this document is supported.

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 Dialogic® HMP software, which includes the Dialogic® 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 Dialogic® 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

See the following for additional information:

- *http://www.dialogic.com/manuals/* (for Dialogic® product documentation)
- *http://www.dialogic.com/support/* (for Dialogic technical support)
- *http://www.dialogic.com/* (for Dialogic® product information)

Demo Description 1

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

The Dialogic Global Call API demo program sets up and tears down calls on the virtual boards and channels specified by the user. The program demonstrates call control functionality only and uses the Global Call basic call state model. The Dialogic Global Call API demo program configuration file enables a user to specify:

- The channels to be used by the demo
- The protocol (H.323 or SIP) to be used by each device
- The protocol type (inbound or outbound) for each device
- The IP destination address to associate with each device
- The transmit (Tx) and receive (Rx) codec parameters (type, rate and Voice Activity Detection [VAD])

When the Dialogic 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.

System Requirements 2

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

2.1 Hardware Requirements

The Dialogic Global Call API demo program can run in two modes: Loopback mode and Non-Loopback mode. To run the demo program in Loopback mode, no additional hardware is required. To run the demo program in Non-Loopback mode, a separate PC hosting an external endpoint such as Microsoft® NetMeeting or SJPhone is required. See Section 3.2, "Connecting to External Equipment", on page 15 for more information.

2.2 Software Requirements

To run the Dialogic Global Call API demo, you need the Dialogic® Host Media Processing (HMP) Software, which includes the Dialogic® Global Call software.

Note: To ensure that the Dialogic Global Call software is installed as part of the Dialogic® HMP Software, see the software installation documentation for the Dialogic® HMP Software release.

For a list of operating system requirements and supported compilers, see the *Release Guide* for the Dialogic® HMP Software.

Preparing to Run the Demo 3

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

3.1 Overview

The Dialogic Global Call API demo program can run in one of two modes:

Loopback Mode

In this mode, IP calls are made and disconnected within the Dialogic® Host Media Processing (HMP) Software system.

Non-Loopback Mode

In this mode, IP calls are made between the Dialogic® HMP Software system and an external IP endpoint, such as Microsoft® NetMeeting® or SJPhone.

3.2 Connecting to External Equipment

When running the Dialogic Global Call API demo in Loopback mode, no connection to external equipment is required.

When running the Dialogic Global Call API demo in Non-Loopback mode, use a standard Ethernet cable to connect the NIC of the system hosting the Dialogic® HMP Software to the same IP network as the system hosting the other endpoint (for example, Microsoft® NetMeeting® or SJPhone).

3.3 Editing Configuration Files

In both Linux and Windows® environments, you must edit the demo configuration file to customize it to your specific configuration. Depending on the mode of operation you are using, you edit the configuration differently.

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.

3.3.1 Editing gc_basic_call_model.cfg

Depending on the protocol and operating mode you are using, you edit different sections of the *gc_basic_call_model.cfg* configuration file as described in the following subsections.

When Using IP Protocols in Loopback Mode

To configure the demo to run in Loopback mode, you must edit the appropriate lines in the H.323 or SIP sections of the *gc_basic_call_model.cfg* configuration file using a standard text editor before running the demo program.

Note: The demo is designed to run using channel numbers in the range of 0 to 120. The demo program fails if this condition is not met.

The parameters that you can configuration include the following:

1. Network device
2. Protocol name
3. Direction
4. IP Media device
5. IP destination address
6. Inter-call delay
7. Transmit codec parameters including; type, rate and VAD
8. Receive codec parameters including; type, rate and VAD

The following is the H.323-specific section of the *gc_basic_call_model.cfg* file:

```
###############################################################################
# For H.323 protocol
# Board #1
# CAUTION: Entries are case sensitive.            Inter-Call       TX CODEC              RX CODEC
#                                                      Delay     (See NOTE below)      (See NOTE below)
#ipt dev:   Prot: Dir: ipm dev:  Dest Address:      (secs)     Type:   Rate: VAD:   Type:   Rate: VAD:
#--------   ----- ---- --------- ---------------- ---------- --------- ----- ---- --------- ----- ----
#Loopback (where loopback IP address is 127.0.0.1):
iptB1T1    H323  In   ipmB1C1   TA:127.0.0.1          0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
iptB1T2    H323  Out  ipmB1C2   TA:127.0.0.1          0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
#
#Non-loopback call to/from another H.323 endpoint, NetMeeting, Messenger, etc.:
#NOTE: Change dest. address from X.X.X.X to specify IP address of remote endpoint:
#iptB1T1    H323  In   ipmB1C1   TA:X.X.X.X            0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
#iptB1T2    H323  Out  ipmB1C2   TA:X.X.X.X            0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
###############################################################################
```

The two uncommented lines under the “#Loopback...” comment specify that the demo will run in H.323 Loopback mode.

The first uncommented line specifies that channel 1 on virtual board 1 (iptB1T1) is an inbound channel (In), running the H.323 protocol, using the channel 1 media resource on virtual board 1 (ipmB1C1), with an associated IP destination address of TA:127.0.0.1 (the loopback address). Both the Transmit (Tx) and Receive (Rx) codecs are of type G.711 Alaw, at a rate of 20 frames per packet (fpp), without Voice Activity Detection (VAD).

The second uncommented line specifies that channel 2 on virtual board 1 (iptB1T1) is an outbound channel (Out), running the H.323 protocol, using the channel 2 media resource on virtual board 1 (ipmB1C2), with an associated IP destination address of TA:127.0.0.1 (the loopback address). Both the Transmit (Tx) and Receive (Rx) codecs are of type G.711 Alaw, at a rate of 20 frames per packet (fpp), without Voice Activity Detection (VAD).

There is a similar section in the *gc_basic_call_model.cfg* file for SIP operation which contains the following lines for Loopback mode:

```
#iptB1T1    SIP   In   ipmB1C1   SIP:userA@127.0.0.1      0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
#iptB1T2    SIP   Out  ipmB1C2   SIP:userB@127.0.0.1      0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
```

Uncommenting these lines configures the Dialogic Global Call demo for SIP operation in Loopback mode. The parameters on these lines are similar to their H.323 counterparts described above with the exception of the destination addresses, which are in a standard SIP address format, for example, userA@127.0.0.1.

When Using IP Protocols in Non-Loopback Mode

To configure the demo to run in Non-Loopback mode, you must edit the appropriate line in the H.323 or SIP section of the *gc_basic_call_model.cfg* configuration file using a standard text editor before running the demo program.

Note: The demo is designed to run using channel numbers in the range of 0 to 120. The demo program fails if this condition is not met.

The parameters that you can configuration include the following:

1. Network device
2. Protocol name
3. Direction
4. IP Media device
5. IP destination address
6. Inter-call delay
7. Transmit codec parameters including; type, rate and VAD
8. Receive codec parameters including; type, rate and VAD

The following is the SIP-specific section of the *gc_basic_call_model.cfg* file:

```
#############################################################################
# For SIP protocol
# Board #1
# CAUTION: Entries are case sensitive.              Inter-Call       TX CODEC             RX CODEC
#                                                      Delay     (See NOTE below)      (See NOTE below)
#ipt dev:   Prot: Dir: ipm dev:  Dest Address:        (secs)     Type:   Rate: VAD:   Type:   Rate: VAD:
#--------   ----- ---- --------- ------------------- ---------- --------- ----- ---- --------- ----- ----
#Loopback (where loopback IP address is 127.0.0.1):
#iptB1T1    SIP   In   ipmB1C1   SIP:userA@127.0.0.1      0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
#iptB1T2    SIP   Out  ipmB1C2   SIP:userB@127.0.0.1      0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
#
#Non-loopback call to/from another H.323 endpoint, NetMeeting, Messenger, etc.:
#NOTE: Change dest. address from X.X.X.X to specify IP address of remote endpoint:
```

```
#iptB1T1    SIP   In   ipmB1C1   SIP:userA@X.X.X.X        0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
iptB1T2     SIP   Out  ipmB1C2   SIP:userB@x.x.x.X        0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
##############################################################################
```

The uncommented line under the "#Non-loopback..." comment specifies that the demo will run in SIP Non-Loopback mode. The external endpoint (for example, Microsoft® NetMeeting® or SJPhone) is set to automatically accept calls.

The uncommented line specifies that channel 2 on virtual board 1 (iptB1T2) is an outbound channel (Out), running the SIP protocol, using the channel 2 media resource on virtual board 1 (ipmB1C2), with an associated SIP address of userB@x.x.x.x. Note that you must set the IP address x.x.x.x appropriately. Both the Transmit (Tx) and Receive (Rx) codecs are of type G.711 Alaw, at a rate of 20 frames per packet (fpp), without Voice Activity Detection (VAD).

There is a similar section in the *gc_basic_call_model.cfg* file for H.323 operation which contains the following line for Non-Loopback mode:

```
#iptB1T2    H323  Out  ipmB1C2   TA:X.X.X.X           0     G.711Alaw  20   N/A  G.711Alaw  20   N/A
```

Uncommenting this line configures the Dialogic Global Call demo for H.323 operation in Non-Loopback mode. The parameters on these lines are similar to their SIP counterparts described above with the exception of the destination addresses, which for H.323 operation is the standard IP address format. Again, you must set the IP address x.x.x.x appropriately.

3.4 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 subdirectory:

   Linux:
   /demos/gc_basic_call_model

   Windows®:
   \demos\gc_basic_call_model

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.

Running the Demo 4

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

4.1 Starting the Demo

Note: Ensure the *.cfg* file is modified (if necessary) before running the demo.

Start the demo as follows:

1. Open a command prompt window.
2. Move to one of the following subdirectories:
   In Linux systems, */demos/gc_basic_call_model*
   In Windows® systems, *\demos\gc_basic_call_model*
3. Type `gc_basic_call_model`

The Dialogic Global Call API demo program displays the status of the program as it runs. The following is an example of the output generated for a demo run in Non-Loopback mode:

```
C:\Program Files\Dialogic\HMP\demos\gc_basic_call_model>gc_basic_call_model
 [MISC]:                SRL Model Set to SR_STASYNC | SR_POLLMODE
gc_basic_call_model_B1T1.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_SS7_LIB - available
                GC_DM3CC_LIB - available
                GC_IPM_LIB - available
                GC_H3R_LIB - available
                GC_CUSTOM1_LIB - configured
                GC_CUSTOM2_LIB - configured

 [MISC]:                H.323 device being opened
 [GC_APICALL]: gc_OpenEx(devicename=:N_iptB1T1:P_H323:M_ipmB1C1, mode=EV_ASYNC) Success
 [MISC]:                Specifying codecs
 [MISC]:                ********* Received a GC event *********
 [EVENT]:               GCEV_OPENEX
 [STATE]:               GCST_NULL is the current GC call state
```

```
 [MISC]:                 GCEV_OPENEX received
 [MISC]:                 gc_SetAlarmNotifyAll(linedev=1, ALARM_SOURCE_ID_NETWORK_ID,
ALARM_NOTIFY) PASSED
 [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=1, numberstr=TA:146.152.84.233, mode=EV_ASYNC) Success
 [STATE]:                GCST_NULL is the new GC call state after processing the event
 [MISC]:                 ********* Received a GC event *********
 [EVENT]:                GCEV_CONNECTED
 [STATE]:                GCST_NULL is the current GC call state
 [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=0x8000001, cause=GC_NORMAL_CLEARING, mode=EV_ASYNC) Success
 [MISC]:                 ********* Received a GC event *********
 [EVENT]:                GCEV_DROPCALL
 [STATE]:                GCST_CONNECTED is the current GC call state
 [GC_APICALL]: gc_ReleaseCallEx(crn=0x8000001, 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=1, numberstr=TA:146.152.84.233, mode=EV_ASYNC) Success
 [STATE]:                GCST_NULL is the new GC call state after processing the event
 [MISC]:                 ********* Received a GC event *********
 [EVENT]:                GCEV_CONNECTED
 [STATE]:                GCST_NULL is the current GC call state
 [MISC]:                 gc_DropCall() will be issued in 1-2 seconds
 [STATE]:                GCST_CONNECTED is the new GC call state after processing the event
 [MISC]:                  *******Received User Interrupted Signal *******

 [MISC]:                 ********** Exit initiated **********

 [GC_APICALL]: gc_ResetLineDev(linedev=1, mode=EV_ASYNC) Success
 [MISC]:                 Please wait while 1 gc_ResetLineDev()s are being done...
 [MISC]:                 ********* Received a GC event *********
 [EVENT]:                GCEV_RESETLINEDEV
 [STATE]:                GCST_CONNECTED is the current GC call state
 [MISC]:                 One less GCEV_RESETLINEDEV to wait for, 0 left
 [STATE]:                GCST_NULL is the new GC call state after processing the event
 [MISC]:                 ********* Program Exiting *********
 [MISC]:                 The total no of Outbound calls on this device is: 2
 [MISC]:                 The total no of Inbound calls over all devices is: 0
 [MISC]:                 The total no of Outbound calls over all devices is: 2
 [MISC]:                 The total duration taken by the test is:  0.13 minutes

C:\Program Files\Dialogic\samples\gc_demos\gc_basic_call_model>
```

The demo program also generates log files. For Non-Loopback mode, a log is generated for the outbound channel only. For Loopback mode, a log is generated for both the inbound and outbound channels. See Section 4.2, “Using the Demo”, on page 20 for examples.

4.2 Using the Demo

The Dialogic 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 is *bx* is the virtual 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

An example of the activity log for an inbound channel is shown below. This activity log is only produced when the demo is run in Loopback mode.

```
04/05 13:53:39.494   [MISC]: ********** GC DEMO - BASIC CALL MODEL ***********

04/05 13:53:40.936   [GC_APICALL]: gc_Start(startp = NULL) Success
04/05 13:53:40.936   [MISC]:  Call Control Library Status:
  GC_ICAPI_LIB - available
  GC_ISDN_LIB - available
  GC_ANAPI_LIB - available
  GC_PDKRT_LIB - available
  GC_SS7_LIB - available
  GC_DM3CC_LIB - available
  GC_IPM_LIB - available
  GC_H3R_LIB - available
  GC_CUSTOM1_LIB - configured
  GC_CUSTOM2_LIB - configured

04/05 13:53:40.936  [MISC]: H.323 device being opened
04/05 13:53:41.017  [GC_APICALL]: gc_OpenEx(devicename=:N_iptB1T1:P_H323:M_ipmB1C1,
mode=EV_ASYNC) Success
04/05 13:53:41.017  [MISC]: Specifying codecs

04/05 13:53:41.097  [MISC]: ********* Received a GC event *********
04/05 13:53:41.097  [EVENT]: GCEV_OPENEX
04/05 13:53:41.097  [STATE]: GCST_NULL is the current GC call state
04/05 13:53:41.097  [MISC]: GCEV_OPENEX received
04/05 13:53:41.097  [MISC]: gc_SetAlarmNotifyAll(linedev=1, ALARM_SOURCE_ID_NETWORK_ID,
ALARM_NOTIFY) PASSED
04/05 13:53:41.097  [STATE]: GCST_NULL is the new GC call state after processing the event

04/05 13:53:41.097  [MISC]: ********* Received a GC event *********
04/05 13:53:41.097  [EVENT]: GCEV_UNBLOCKED
04/05 13:53:41.097  [STATE]: GCST_NULL is the current GC call state
04/05 13:53:41.097  [GC_APICALL]: gc_WaitCall(linedev=1, crnp=NULL, waittime=0, mode=EV_ASYNC)
Success
04/05 13:53:41.097  [STATE]: GCST_NULL is the new GC call state after processing the event

04/05 13:53:41.197  [MISC]: ********* Received a GC event *********
04/05 13:53:41.197  [EVENT]: GCEV_OFFERED
04/05 13:53:41.197  [STATE]: GCST_NULL is the current GC call state
04/05 13:53:41.207  [GC_APICALL]: gc_GetCallInfo(crn=0x8000001) Success - called party =
04/05 13:53:41.207  [GC_APICALL]: gc_GetCallInfo(crn=0x8000001) Success - calling party =
TA:146.152.86.110:3595
04/05 13:53:41.217  [GC_APICALL]: gc_AcceptCall(crn=0x8000001, mode=EV_ASYNC) Success
04/05 13:53:41.217  [STATE]: GCST_OFFERED is the new GC call state after processing the event

04/05 13:53:41.217  [MISC]: ********* Received a GC event *********
04/05 13:53:41.217  [EVENT]: GCEV_ACCEPT
04/05 13:53:41.217  [STATE]: GCST_OFFERED is the current GC call state
04/05 13:53:41.237  [GC_APICALL]: gc_AnswerCall(crn=0x8000001, mode=EV_ASYNC) Success
04/05 13:53:41.237  [STATE]: GCST_ACCEPTED is the new GC call state after processing the event

04/05 13:53:41.247  [MISC]: ********* Received a GC event *********
04/05 13:53:41.247  [EVENT]: GCEV_ANSWERED
04/05 13:53:41.247  [STATE]: GCST_ACCEPTED is the current GC call state
```

```
04/05 13:53:41.247  [STATE]: GCST_CONNECTED is the new GC call state after processing the event

04/05 13:53:43.320  [MISC]: ********* Received a GC event *********
04/05 13:53:43.320  [EVENT]: GCEV_DISCONNECTED
04/05 13:53:43.320  [STATE]: GCST_CONNECTED is the current GC call state
04/05 13:53:43.320  [GC_APICALL]: gc_DropCall(crn=0x8000001, cause=GC_NORMAL_CLEARING,
mode=EV_ASYNC) Success
04/05 13:53:43.320  [STATE]: GCST_DISCONNECTED is the new GC call state after processing the
event

04/05 13:53:43.320  [MISC]: ********* Received a GC event *********
04/05 13:53:43.320  [EVENT]: GCEV_DROPCALL
04/05 13:53:43.320  [STATE]: GCST_DISCONNECTED is the current GC call state
04/05 13:53:43.320  [GC_APICALL]: gc_ReleaseCallEx(crn=0x8000001, EV_ASYNC) Success
04/05 13:53:43.320  [STATE]: GCST_IDLE is the new GC call state after processing the event

04/05 13:53:43.360  [MISC]: ********* Received a GC event *********
04/05 13:53:43.360  [EVENT]: GCEV_RELEASECALL
04/05 13:53:43.360  [STATE]: GCST_IDLE is the current GC call state
04/05 13:53:43.360  [STATE]: GCST_NULL is the new GC call state after processing the event

04/05 13:53:43.370  [MISC]: ********* Received a GC event *********
04/05 13:53:43.370  [EVENT]: GCEV_OFFERED
04/05 13:53:43.370  [STATE]: GCST_NULL is the current GC call state
04/05 13:53:43.370  [GC_APICALL]: gc_GetCallInfo(crn=0x8000001) Success - called party =
04/05 13:53:43.380  [GC_APICALL]: gc_GetCallInfo(crn=0x8000001) Success - calling party =
TA:146.152.86.110:3598
04/05 13:53:43.380  [GC_APICALL]: gc_AcceptCall(crn=0x8000001, mode=EV_ASYNC) Success
04/05 13:53:43.380  [STATE]: GCST_OFFERED is the new GC call state after processing the event

04/05 13:53:43.380  [MISC]: ********* Received a GC event *********
04/05 13:53:43.380  [EVENT]: GCEV_ACCEPT
04/05 13:53:43.390  [STATE]: GCST_OFFERED is the current GC call state
04/05 13:53:43.400  [GC_APICALL]: gc_AnswerCall(crn=0x8000001, mode=EV_ASYNC) Success
04/05 13:53:43.400  [STATE]: GCST_ACCEPTED is the new GC call state after processing the event

04/05 13:53:43.410  [MISC]: ********* Received a GC event *********
04/05 13:53:43.410  [EVENT]: GCEV_ANSWERED
04/05 13:53:43.410  [STATE]: GCST_ACCEPTED is the current GC call state
04/05 13:53:43.410  [STATE]: GCST_CONNECTED is the new GC call state after processing the event
04/05 13:53:44.572   [MISC]:  *******Received User Interrupted Signal *******

04/05 13:53:44.942   [MISC]: ********** Exit initiated ***********

04/05 13:53:44.942  [GC_APICALL]: gc_ResetLineDev(linedev=1, mode=EV_ASYNC) Success

04/05 13:53:45.002  [MISC]: ********* Received a GC event *********
04/05 13:53:45.002  [EVENT]: GCEV_RESETLINEDEV
04/05 13:53:45.002  [STATE]: GCST_CONNECTED is the current GC call state
04/05 13:53:45.002   [MISC]: One less GCEV_RESETLINEDEV to wait for, 1 left
04/05 13:53:45.002  [STATE]: GCST_NULL is the new GC call state after processing the event
04/05 13:53:45.002   [MISC]: One less GCEV_RESETLINEDEV to wait for, 0 left

04/05 13:53:45.002   [MISC]: ********* Program Exiting *********
04/05 13:53:45.002  [MISC]: The total no of Inbound calls on this device is: 2
04/05 13:53:45.002  [MISC]: The total no of Inbound calls over all devices is: 2
04/05 13:53:45.002  [MISC]: The total no of Outbound calls over all devices is: 2
04/05 13:53:45.002  [MISC]: The total duration taken by the test is:  0.10 minutes
```

Example of Outbound Channel Activity Log

An example of the activity log for an outbound channel is shown below. This activity log is produced when the demo is run in both the Loopback and Non-Loopback modes.

```
04/05 13:53:39.494   [MISC]: ********** GC DEMO - BASIC CALL MODEL ***********

04/05 13:53:40.936   [GC_APICALL]: gc_Start(startp = NULL) Success
04/05 13:53:40.936   [MISC]:  Call Control Library Status:
  GC_ICAPI_LIB - available
  GC_ISDN_LIB - available
  GC_ANAPI_LIB - available
  GC_PDKRT_LIB - available
  GC_SS7_LIB - available
  GC_DM3CC_LIB - available
  GC_IPM_LIB - available
  GC_H3R_LIB - available
  GC_CUSTOM1_LIB - configured
  GC_CUSTOM2_LIB - configured

04/05 13:53:41.017  [MISC]: H.323 device being opened
04/05 13:53:41.027  [GC_APICALL]: gc_OpenEx(devicename=:N_iptB1T2:P_H323:M_ipmB1C2,
mode=EV_ASYNC) Success
04/05 13:53:41.027  [MISC]: Specifying codecs

04/05 13:53:41.127  [MISC]: ********* Received a GC event *********
04/05 13:53:41.127  [EVENT]: GCEV_OPENEX
04/05 13:53:41.127  [STATE]: GCST_NULL is the current GC call state
04/05 13:53:41.127  [MISC]: GCEV_OPENEX received
04/05 13:53:41.127  [MISC]: gc_SetAlarmNotifyAll(linedev=5, ALARM_SOURCE_ID_NETWORK_ID,
ALARM_NOTIFY) PASSED
04/05 13:53:41.127  [STATE]: GCST_NULL is the new GC call state after processing the event

04/05 13:53:41.127  [MISC]: ********* Received a GC event *********
04/05 13:53:41.127  [EVENT]: GCEV_UNBLOCKED
04/05 13:53:41.127  [STATE]: GCST_NULL is the current GC call state
04/05 13:53:41.127  [GC_APICALL]: gc_MakeCall(linedev=5, numberstr=TA:127.0.0.1, mode=EV_ASYNC)
Success
04/05 13:53:41.127  [STATE]: GCST_NULL is the new GC call state after processing the event

04/05 13:53:41.237  [MISC]: ********* Received a GC event *********
04/05 13:53:41.237  [EVENT]: GCEV_ALERTING
04/05 13:53:41.237  [STATE]: GCST_NULL is the current GC call state
04/05 13:53:41.237  [STATE]: GCST_ALERTING is the new GC call state after processing the event

04/05 13:53:41.277  [MISC]: ********* Received a GC event *********
04/05 13:53:41.277  [EVENT]: GCEV_CONNECTED
04/05 13:53:41.277  [STATE]: GCST_ALERTING is the current GC call state
04/05 13:53:41.277  [MISC]: gc_DropCall() will be issued in 1-2 seconds
04/05 13:53:41.277  [STATE]: GCST_CONNECTED is the new GC call state after processing the event

04/05 13:53:43.280  [MISC]: ********* Dropping outbound call from
drop_outbound_calls_if_required() *********
04/05 13:53:43.280  [GC_APICALL]: gc_DropCall(crn=0x8000002, cause=GC_NORMAL_CLEARING,
mode=EV_ASYNC) Success

04/05 13:53:43.320  [MISC]: ********* Received a GC event *********
04/05 13:53:43.320  [EVENT]: GCEV_DROPCALL
04/05 13:53:43.320  [STATE]: GCST_CONNECTED is the current GC call state
04/05 13:53:43.320  [GC_APICALL]: gc_ReleaseCallEx(crn=0x8000002, EV_ASYNC) Success
04/05 13:53:43.320  [STATE]: GCST_IDLE is the new GC call state after processing the event

04/05 13:53:43.350  [MISC]: ********* Received a GC event *********
04/05 13:53:43.350  [EVENT]: GCEV_RELEASECALL
04/05 13:53:43.350  [STATE]: GCST_IDLE is the current GC call state
04/05 13:53:43.360  [GC_APICALL]: gc_MakeCall(linedev=5, numberstr=TA:127.0.0.1, mode=EV_ASYNC)
Success
04/05 13:53:43.360  [STATE]: GCST_NULL is the new GC call state after processing the event

04/05 13:53:43.360  [MISC]: ********* Received a GC event *********
04/05 13:53:43.360  [EVENT]: GCEV_DISCONNECTED
04/05 13:53:43.360  [STATE]: GCST_NULL is the current GC call state
```

```
04/05 13:53:43.360  [GC_APICALL]: gc_DropCall(crn=0x8000002, cause=GC_NORMAL_CLEARING,
mode=EV_ASYNC) Success
04/05 13:53:43.360  [STATE]: GCST_DISCONNECTED is the new GC call state after processing the
event

04/05 13:53:43.360  [MISC]: ********* Received a GC event *********
04/05 13:53:43.360  [EVENT]: GCEV_DROPCALL
04/05 13:53:43.360  [STATE]: GCST_DISCONNECTED is the current GC call state
04/05 13:53:43.360  [GC_APICALL]: gc_ReleaseCallEx(crn=0x8000002, EV_ASYNC) Success
04/05 13:53:43.360  [STATE]: GCST_IDLE is the new GC call state after processing the event

04/05 13:53:43.360  [MISC]: ********* Received a GC event *********
04/05 13:53:43.360  [EVENT]: GCEV_RELEASECALL
04/05 13:53:43.360  [STATE]: GCST_IDLE is the current GC call state
04/05 13:53:43.370  [GC_APICALL]: gc_MakeCall(linedev=5, numberstr=TA:127.0.0.1, mode=EV_ASYNC)
Success
04/05 13:53:43.370  [STATE]: GCST_NULL is the new GC call state after processing the event

04/05 13:53:43.400  [MISC]: ********* Received a GC event *********
04/05 13:53:43.400  [EVENT]: GCEV_ALERTING
04/05 13:53:43.400  [STATE]: GCST_NULL is the current GC call state
04/05 13:53:43.400  [STATE]: GCST_ALERTING is the new GC call state after processing the event

04/05 13:53:43.440  [MISC]: ********* Received a GC event *********
04/05 13:53:43.440  [EVENT]: GCEV_CONNECTED
04/05 13:53:43.440  [STATE]: GCST_ALERTING is the current GC call state
04/05 13:53:43.440  [MISC]: gc_DropCall() will be issued in 1-2 seconds
04/05 13:53:43.440  [STATE]: GCST_CONNECTED is the new GC call state after processing the event
04/05 13:53:44.572   [MISC]:  *******Received User Interrupted Signal *******

04/05 13:53:44.942   [MISC]: ********** Exit initiated **********

04/05 13:53:44.942  [GC_APICALL]: gc_ResetLineDev(linedev=5, mode=EV_ASYNC) Success
04/05 13:53:44.942  [MISC]: Please wait while 2 gc_ResetLineDev()s are being done...
04/05 13:53:45.002   [MISC]: One less GCEV_RESETLINEDEV to wait for, 1 left

04/05 13:53:45.002  [MISC]: ********* Received a GC event *********
04/05 13:53:45.002  [EVENT]: GCEV_RESETLINEDEV
04/05 13:53:45.002  [STATE]: GCST_CONNECTED is the current GC call state
04/05 13:53:45.002   [MISC]: One less GCEV_RESETLINEDEV to wait for, 0 left
04/05 13:53:45.002  [STATE]: GCST_NULL is the new GC call state after processing the event

04/05 13:53:45.002   [MISC]: ********* Program Exiting *********
04/05 13:53:45.002  [MISC]: The total no of Outbound calls on this device is: 2
04/05 13:53:45.002  [MISC]: The total no of Inbound calls over all devices is: 2
04/05 13:53:45.002  [MISC]: The total no of Outbound calls over all devices is: 2
04/05 13:53:45.002  [MISC]: The total duration taken by the test is:  0.10 minutes
```

4.3 Stopping the Demo

The Dialogic Global Call API demo runs continuously. You can press Ctrl-C at any time to exit the Dialogic 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.

Demo Details 5

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

Table 1 lists the files used by the Dialogic 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 */demos/gc_basic_call_model*
- In a Windows® environment, the files are located in *\demos\gc_basic_call_model*

Table 1. Files Used by the Dialogic® 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 (not applicable to Dialogic® HMP Software)
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.

codec: A device that converts analog voice signals to a digital form and vice versa. In this context, analog signals are converted into the payload of UDP packets for transmission over the internet. The codec also performs compression and decompression on a voice stream.

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.

H.323: H.323 is an ITU recommendation for a standard for interoperability in audio, video and data transmissions as well as Internet phone and voice-over-IP (VoIP). H.323 addresses call control and management for both point-to-point and multipoint conferences as well as gateway administration of IP Media traffic, bandwidth and user participation.

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.

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.

SIP: Session Initiated Protocol. An ASCII-based, peer-to-peer protocol designed to provide telephony services over the Internet.

VAD: Voice Activation Detection. In Voice over IP (VoIP), voice activation detection (VAD) is a technique that allows a data network carrying voice traffic over the Internet to detect the absence of audio and conserve bandwidth by preventing the transmission of silent packets over the network.

Index