



Global Call API for Host Media Processing

Demo Guide

April 2005



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 Host Media Processing 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 - 2005, Intel Corporation

BunnyPeople, Celeron, Chips, Dialogic, EtherExpress, ETOX, FlashFile, i386, i486, i960, iCOMP, InstantIP, Intel, Intel Centrino, Intel Centrino logo, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Xeon, Intel XScale, IPLink, Itanium, MCS, MMX, MMX logo, Optimizer logo, OverDrive, Paragon, PDCharm, Pentium, Pentium II Xeon, Pentium III Xeon, Performance at Your Command, skool, Sound Mark, The Computer Inside., The Journey Inside, 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: April 2005

Document Number: 05-2439-001

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 Telecom Products website at:
<http://www.intel.com/design/network/products/telecom>

For **Sales Offices** and other contact information, visit the Where to Buy Intel Telecom Products page at:
<http://www.intel.com/buy/wtb/wtb1028.htm>



Contents

	Revision History	7
	About This Publication	9
	Purpose	9
	Intended Audience	9
	How to Use This Publication	9
	Related Information	10
1	Demo Description	11
2	System Requirements	13
	2.1 Hardware Requirements	13
	2.2 Software Requirements	13
3	Preparing to Run the Demo	15
	3.1 Overview	15
	3.2 Connecting to External Equipment	15
	3.3 Editing Configuration Files	15
	3.3.1 Editing gc_basic_call_model.cfg	16
	3.4 Compiling and Linking	18
4	Running the Demo	19
	4.1 Starting the Demo	19
	4.2 Using the Demo	20
	4.3 Stopping the Demo	24
5	Demo Details	25
	Glossary	27
	Index	29

Tables

1	Files Used by the Global Call API Demo	25
---	--	----



Revision History

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

Document No.	Publication Date	Description of Revisions
05-2439-001	April 2005	Initial version of document. Much of the information contained in this document was previously published in the <i>Global Call API Demo Guide</i> , document number 05-1818-002.



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 on the Global Call API demonstration program available with the Intel® NetStructure™ Host Media Processing (HMP) software. This guide describes the demo, its requirements and details how it works.

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 HMP software, which 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 IP Technology Guide* - provides technology-specific information when using the Global Call software with IP interfaces
- *HMP Release Guide* - provides information on the HMP software, system requirements, software and hardware features, supported hardware, and release documentation
- *HMP 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 virtual boards and channels specified by the user. The program demonstrates call control functionality only and 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 (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 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](#) 13
- [Software Requirements](#) 13

2.1 Hardware Requirements

The 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, 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 Global Call API demo, you need the HMP software which includes the Global Call software.

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

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



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

- Overview 15
- Connecting to External Equipment 15
- Editing Configuration Files 15
- Compiling and Linking 18

3.1 Overview

The 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 HMP system.

Non-Loopback Mode

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

3.2 Connecting to External Equipment

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

When running the Global Call API demo in Non-Loopback mode, use a standard Ethernet cable to connect the NIC of the system hosting the HMP software to the same IP network as the system hosting the other endpoint (for example, 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 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.
#
# ipt dev:   Prot: Dir: ipm dev:  Dest Address:   Inter-Call   TX CODEC   RX CODEC
#              Delay   (secs)   (See NOTE below)   (See NOTE below)
#              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 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 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)
#          (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, 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 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 directory:

Linux:

```
$(INTEL_DIALOGIC_DIR)/demos/gc_basic_call_model
```

Windows:

```
$(INTEL_DIALOGIC_DIR)\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.

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

- Starting the Demo 19
- Using the Demo..... 20
- Stopping the Demo 24

4.1 Starting the Demo

Start the Global Call API demo program by moving to the directory that contains the demo program executable and typing the program name at the command line. The program name is:

```
gc_basic_call_model
```

The 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\demos\gc_basic_call_model>gc_basic_call_model
[MISC]:          SRL Model Set to SR_STASYNCR | SR_POLLMODE
gc_basic_call_model_B1T1.log successfully opened
[MISC]:          ***** GC DEMO - BASIC CALL MODEL *****

[GC_APICALL]: gc_Start(startup = 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_ipmB1T1: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 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 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(starttp = 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_ipmB1T1: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=0x80000001) Success - called party =
04/05 13:53:41.207 [GC_APICALL]: gc_GetCallInfo(crn=0x80000001) Success - calling party =
TA:146.152.86.110:3595
04/05 13:53:41.217 [GC_APICALL]: gc_AcceptCall(crn=0x80000001, 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=0x80000001, 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=0x80000001, 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_DROP_CALL
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_RELEASE_CALL
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(starttp = 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_ipB1t2: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_DROP_CALL
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_RELEASE_CALL
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_DROP_CALL
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_RELEASE_CALL
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 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
\$(INTEL_DIALOGIC_DIR)/demos/gc_basic_call_model
- In a Windows environment, the files are located in
\$(INTEL_DIALOGIC_DIR)\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 (not applicable to 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

A

- activity log
 - file 20
 - inbound 21
 - outbound 23

C

- compiling the demo 18
- connecting
 - external equipment 15

D

- demo
 - description 11
 - files used 25
 - makefiles 18
 - preparation to run 15
 - prerequisites 18
 - running 19
 - starting 19
 - stopping 24
 - system requirements 13

E

- external equipment
 - connecting 15

F

- files used by the demo 25

G

- gc_basic_call_model.cfg
 - configuration file 16, 17

H

- hardware requirements 13

L

- log file
 - inbound channel activity 21
 - naming convention 20
 - outbound channel activity 23

R

- recompiling the demo 18
- related information 10
- requirements
 - hardware 13
 - software 13

S

- sample
 - demo output 19
- software requirements 13
- stopping the demo 24
- system requirements 13
 - hardware 13
 - software 13

