

GlobalCall™ DPNSS ISDN Protocol Reference for Linux and Windows

Copyright © 2001 Dialogic Corporation

05-1317-003

COPYRIGHT NOTICE

All contents of this document are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation. Every effort is made to ensure the accuracy of this information. However, due to ongoing product improvements and revisions, Dialogic Corporation cannot guarantee the accuracy of this material, nor can it accept responsibility for errors or omissions. No warranties of any nature are extended by the information contained in these copyrighted materials. Use or implementation of any one of the concepts, applications, or ideas described in this document 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 condone or encourage such infringement. Dialogic makes no warranty with respect to such infringement, nor does Dialogic waive any of its own intellectual property rights which may cover systems implementing one or more of the ideas contained herein. Procurement of appropriate intellectual property rights and licenses is solely the responsibility of the system implementer. The software referred to in this document is provided under a Software License Agreement. Refer to the Software License Agreement for complete details governing the use of the software.

All names, products, and services mentioned herein are the trademarks or registered trademarks of their respective organizations and are the sole property of their respective owners. DIALOGIC (including the Dialogic logo), DTI/124, and SpringBoard are registered trademarks of Dialogic Corporation. A detailed trademark listing can be found at: <http://www.dialogic.com/legal.htm>.

Publication Date: October, 2001

Part Number: 05-1317-003

Dialogic, an Intel Company
1515 Route 10
Parsippany NJ 07054
U.S.A.

For **Technical Support**, visit the Dialogic support website at:
<http://support.dialogic.com>

For **Sales Offices** and other contact information, visit the main Dialogic website at:
<http://www.dialogic.com>

Table of Contents

1. How To Use This Reference	1
2. IEs and ISDN Message Types for the DPNSS Protocol	3
2.1. gc_GetCallInfo() and gc_GetSigInfo() Information Elements	3
2.2. gc_SetInfoElem() Information Elements	8
2.3. gc_SndMsg() DPNSS Message Types	12
3. DPNSS Call Scenarios	17
3.1. Executive Intrusion - Normal	18
3.2. Executive Intrusion - with Prior Validation	18
3.3. Hold and Retrieve - Locally Initiated	19
3.4. Hold and Retrieve - Remotely Initiated	20
3.5. Local Diversion - Outbound	21
3.6. Local Diversion - Inbound	21
3.7. Remote Diversion - Outbound	22
3.8. Remote Diversion - Inbound	23
3.9. Transfer	24
3.10. Virtual Call -Outbound	26
3.11. Virtual Call -Inbound	27
Index	29

GlobalCall™ DPNSS ISDN Protocol Reference for Linux and Windows

List of Tables

Table 1. gc_GetCallInfo() and gc_GetSigInfo() Information Elements	4
Table 2. gc_SetInfoElem() Information Elements	8
Table 3. gc_SndMsg() DPNSS Message Types	13

GlobalCall™ DPNSS ISDN Protocol Reference for Linux and Windows

1. How To Use This Reference

This ISDN (Integrated Services Digital Network) protocol reference is for users of Dialogic's GlobalCall Application Programming Interface (API) who develop Linux or Windows applications in an E-1 ISDN DPNSS protocol environment. The DPNSS (Digital Private Network Signaling System) is an ISDN protocol commonly used in a network of PBXs (Private Branch eXchanges). This protocol reference also includes call control scenarios for the DPNSS protocol.

Use this protocol reference along with the *GlobalCall API Software Reference* and the *GlobalCall ISDN Technology User's Guide* to develop ISDN DPNSS applications using Dialogic's ISDN Primary Rate Software as an ISDN call control library running under the GlobalCall API.

The DPNSS ISDN protocol reference is organized as follows:

Chapter 2 presents the information elements (IEs) and ISDN message types that support the DPNSS protocol.

Chapter 3 describes call control scenarios that are specific to the DPNSS protocol.

An **Index** is presented at the end of this reference.

See the *GlobalCall API Software Reference* and the *GlobalCall ISDN Technology User's Guide* for a list of products supported. For a *Glossary* of terms, see the *GlobalCall API Software Reference*.

For additional information on ISDN technology or Dialogic products, refer to the *Related Publications* appendix in the *GlobalCall API Software Reference*.

GlobalCall™ DPNSS ISDN Protocol Reference

2. IEs and ISDN Message Types for the DPNSS Protocol

The information elements (IEs) and ISDN message types that support the DPNSS protocol are listed and described in this chapter. Each IE or ISDN message type is retrieved or set by one or more of the following functions:

- **gc_GetCallInfo()**, see 2.1. *gc_GetCallInfo() and gc_GetSigInfo() Information Elements*
- **gc_GetSigInfo()**, see 2.1. *gc_GetCallInfo() and gc_GetSigInfo() Information Elements*
- **gc_SetInfoElem()**, see 2.2. *gc_SetInfoElem() Information Elements*
- **gc_SndMsg()**, see 2.3. *gc_SndMsg() DPNSS Message Types*.

2.1. gc_GetCallInfo() and gc_GetSigInfo() Information Elements

The different IEs that can be retrieved for the DPNSS protocol using the **gc_GetCallInfo()** and **gc_GetSigInfo()** functions are described in *Table 1. gc_GetCallInfo() and gc_GetSigInfo() Information Elements*.

Table 1. gc_GetCallInfo() and gc_GetSigInfo() Information Elements

Field	Description	Field Selection	Definition
Intrusion IE:			
1. IE ID	Busy IE ID	BUSY_IE	BUSY_IE value for the GCEV_PROCEEDING event indicates that the called party is busy
Diversion IE:			
1. IE ID	Diversion IE ID	DIVERSION_IE	1. DIVERSION_IE value in a GCEV_OFFERED event provides information about "diverted from" party. 2. DIVERSION_IE value in a GCEV_PROCEEDING event provides information about "divert to" party.
2. Data	Diversion IE Length	2 + length of Diversion Number	Number of data bytes in this IE
3. Data	Diversion Type	DIVERT_IMMEDIATE DIVERT_ON_BUSY DIVERT_NO_REPLY	Diverted immediately Diverted when called party was busy Diverted when called party did not answer
4. Data	Diversion Location	DIVERT_LOCAL DIVERT_REMOTE	Local diversion Remote diversion

2. IEs and ISDN Message Types for the DPNSS Protocol

Field	Description	Field Selection	Definition
5. Data	Diversion Number	ASCII string	Diverted number
Diversion Validation IE:			
1. IE ID	Diversion Validation IE ID	DIVERSION_VALIDATION_IE	This IE when part of a GCEV_OFFERED event indicates that the diversion number needs to be validated.
Transit IE:			
1. IE ID	Transit IE ID	TRANSIT_IE	This IE is received with a GCEV_TRANSIT event.
2. Data	Transit IE Length	Length of Transit data	Number of data bytes in this IE
3. Data	Transit Data	data	Transit data that needs to be sent to the other transfer party.
Text Display IE:			
1. IE ID	Text Display IE ID	TEXT_DISPLAY_IE	This IE can be part of a GCEV_OFFERED event.
2. Data	Text Display IE Length	1 + length of Text Display string	Number of data bytes for this IE

GlobalCall™ DPNSS ISDN Protocol Reference

Field	Description	Field Selection	Definition
3. Data	Text Display Message Type	TEXT_TYPE_NOT_PRESENT TEXT_TYPE_NAME TEXT_TYPE_MESSAGE TEXT_TYPE_REASON	Associated text is of no particular type Associated text is a name Associated text is a message Associated text is a reason
4. Data	Text Display String	ASCII string	Text Display string. The '*' and '#' symbols cannot be used directly; 0x01 and 0x02 values are substituted, respectively.
Network Specific Indications (NSI) IE:			
1. IE ID	NSI IE ID	NSI_IE	This IE can be part of any event including the GCEV_NSI event.
2. Data	NSI IE Length	2 + Length of Network Specific Indications (NSI) string	Number of data bytes for this IE
3. Data	NSI Message Type	NSI_EEM NSI_LLM	End-to-End Message Link-to-Link Message
4. Data *	NSI String Length	Length of NSI string	Length of next NSI string
5. Data *	NSI String	ASCII string	Network Specific Indications string
* NSI IE fields 4 and 5 can be repeated multiple times, as needed.			

2. IEs and ISDN Message Types for the DPNSS Protocol

Field	Description	Field Selection	Definition
Extension Status IE:			
1. IE ID	Extension Status IE ID	EXTENSION_STATUS_IE	This IE is used in conjunction with the Virtual Call IE to inquire about the current status of an extension.
Virtual Call IE:			
1. IE ID	Virtual Call IE ID	VIRTUALCALL_IE	This IE, when part of a GCEV_OFFERED event, indicates a virtual call.

2.2. gc_SetInfoElem() Information Elements

The information elements that can be set for the DPNSS protocol using the **gc_SetInfoElem()** function are described in *Table 2. gc_SetInfoElem() Information Elements*.

Table 2. gc_SetInfoElem() Information Elements

Field	Description	Field Selection	Definition
Intrusion IE:			
1. Length	Total number of bytes of the following data field	4	Required value
2. IE ID	Intrusion IE ID	INTRUSION_IE	Use with the gc_MakeCall() function to indicate intrusion privilege.
3. Data	Intrusion IE Length	2	Number of data bytes for this IE
4. Data	Intrusion Type	INTRUDE_PRIOR_VALIDATION INTRUDE_NORMAL	Validate Intrusion Level prior to intrude Intrude (without validation)
5. Data	Intrusion Level	INTRUSION_LEVEL_1 INTRUSION_LEVEL_2 INTRUSION_LEVEL_3	Intrusion protection level 1 Intrusion protection level 2 Intrusion protection level 3

2. IEs and ISDN Message Types for the DPNSS Protocol

Field	Description	Field Selection	Definition
Diversion IE:			
1. Length	Total bytes of the following data field	4 + length of Diversion Number	Required value of 4 bytes plus the number of bytes used for the Diversion data fields.
2. Data	Diversion IE ID	DIVERSION_IE	Use with the gc_MakeCall() function to indicate why the call was diverted and from where the call was diverted.
3. Data	Diversion IE Length	2 + length of Diversion Number	Number of data bytes for this IE
4. Data	Diversion Type	DIVERT_IMMEDIATE DIVERT_ON_BUSY DIVERT_NO_REPLY	Diverted immediately Diverted when called party was busy Diverted when called party did not answer
5. Data	Diversion Location	DIVERT_LOCAL DIVERT_REMOTE	Local diversion Remote diversion
6. Data	Diversion Number	ASCII string	Diverted number
Diversion Bypass IE:			
1. Length	Total bytes of the following data field	1	Required value

GlobalCall™ DPNSS ISDN Protocol Reference

Field	Description	Field Selection	Definition
2. Data	Diversion Bypass IE ID	DIVERSION_BYPASS_IE	Use with the gc_MakeCall() function to indicate that no diversion is allowed.
Inquiry IE:			
1. Length	Total bytes of the following data field	1	Required value
2. Data	Inquiry IE ID	INQUIRY_IE	Use with the gc_MakeCall() function to indicate a three-party call.
Extension Status IE:			
1. Length	Total bytes of the following data field	1	Required value.
2. Data	Extension Status IE ID	EXTENSION_STATUS_IE	Use in conjunction with the Virtual Call IE to inquire about the current status of an extension.
Virtual Call IE:			
1. Length	Total bytes of the following data field	1	Required value

2. IEs and ISDN Message Types for the DPNSS Protocol

Field	Description	Field Selection	Definition
2. Data	Virtual Call IE ID	VIRTUALCALL_IE	Use with the gc_MakeCall() function to indicate a virtual call.
Text Display IE:			
1. Length	Total bytes of the following data field	3 + length of Text Display String	Required value of 3 bytes plus the number of bytes used for the Text Display String.
2. Data	Text Display IE ID	TEXT_DISPLAY_IE	This IE can be part of a GCEV_OFFERED event.
3. Data	Text Display IE Length	1 + length of Text Display string	Number of data bytes for this IE
4. Data	Text Display Message Type	TEXT_TYPE_NOT_PRESENT TEXT_TYPE_NAME TEXT_TYPE_MESSAGE TEXT_TYPE_REASON	Associated text is of no particular type Associated text is a name Associated text is a message Associated text is a reason
5. Data	Text DISPLAY String	ASCII string	Text Display string. The '*' and '#' symbols cannot be used directly; 0x01 and 0x02 values are substituted, respectively

Field	Description	Field Selection	Definition
Network Specific Indications (NSI) IE:			
1. Length	Total bytes of the following data field	4 + length of NSI String	Required value of 4 bytes plus the number of bytes used for the NSI String.
2. Data	NSI IE ID	NSI_IE	
3. Data	NSI IE Length	2 + length of NSI string	Number of data bytes for this IE
4. Data	NSI Message Type	NSI_EEM NSI_LLM	End-to-End Message Link-to-Link Message
5. Data *	NSI String Length	Length of NSI string	Length of next NSI string
6. Data *	NSI String	ASCII string	Network Specific Indications string
* NSI IE fields 5 and 6 can be repeated multiple times, as needed.			

2.3. gc_SndMsg() DPNSS Message Types

The ISDN DPNSS protocol message types that the **gc_SndMsg()** function support are described in *Table 3. gc_SndMsg() DPNSS Message Types*.

2. IEs and ISDN Message Types for the DPNSS Protocol

Table 3. gc_SndMsg() DPNSS Message Types

Field	Description	Field Selection	Definition
SndMsg Divert:			
1. Length	Total number of bytes of the following data field	4 + length of Diversion Number	Required value of 4 bytes plus the number of bytes used for the Diversion data fields.
2. Data	Diversion IE ID	DIVERSION_IE	
3. Data	Diversion IE Length	2 + length of Diversion Number	Number of data bytes for this IE
4. Data	Diversion Type	DIVERT_IMMEDIATE DIVERT_ON_BUSY DIVERT_NO_REPLY	Diverted immediately Diverted when called party was busy Diverted when called party did not answer
5. Data	Diversion Location	DIVERT_LOCAL DIVERT_REMOTE	Local diversion Remote diversion
6. Data	Diversion Number	ASCII string	Diverted number
SndMsg Intrude:			
1. Length	Total number of bytes of the following data field	3	Required value.
2. Data	Intrude IE ID	INTRUDE_IE	

Field	Description	Field Selection	Definition
3. Data	Intrude IE Length	1	Number of data bytes for this IE
4. Data	Intrude Type	INTRUDE INTRUDE_WITHDRAW	
SndMsg_NSI:			
1. Length	Total number of bytes of the following data field	4 + length of NSI String	Required value of 4 bytes plus the number of bytes used for the NSI String.
2. Data	NSI IE ID	NSI_IE	
3. Data	NSI IE Length	2 + length of NSI string	Number of data bytes for this IE
4. Data	NSI Message Type	NSI_EEM NSI_LLM	End-to-End Message Link-to-Link Message
5. Data *	NSI String Length	Length of NSI string	Length of next NSI string
6. Data *	NSI String	ASCII string	Network Specific Indications string
* NSI IE fields 5 and 6 can be repeated multiple times as needed.			
SndMsg_Transfer:			
1. Length	Total number of bytes of the following data field	3	Required value.
2. Data	Transfer IE ID	TRANSFER_IE	

2. IEs and ISDN Message Types for the DPNSS Protocol

Field	Description	Field Selection	Definition
3. Data	Transfer IE Length	1	Number of data bytes for this IE
4. Data	Transfer Direction	TRANSFER_ORIG TRANSFER_TERM	Originating end Terminating end
SndMsg_Transit:			
1. Length	Total number of bytes of the following data field	2 + Length of Transit Data	Required value of 2 bytes plus the number of bytes used for the Transit data fields.
2. Data	Transit IE ID	TRANSIT_IE	
3. Data	Transit IE Length	Length of Transit Data	Number of data bytes for this IE
4. Data	Transit Data	data	Transit data received from a GCEV_TRANSIT event

3. DPNSS Call Scenarios

Call control scenarios that are specific to the DPNSS protocol are described in this chapter. Each scenario includes:

- a table that illustrates the GlobalCall functions issued by the application to either initiate a transaction or to respond to an external action, and the resulting Dialogic event that is returned to the application.
- a step-by-step description of the scenario following the table.

The following call control scenarios describe call setup and tear down, network and application initiated call termination, requests for various ISDN services, etc., using both asynchronous and synchronous mode programming. See the section indicated for a description of each scenario:

- *3.1. Executive Intrusion - Normal*
- *3.2. Executive Intrusion - with Prior Validation*
- *3.3. Hold and Retrieve - Locally Initiated*
- *3.4. Hold and Retrieve - Remotely Initiated*
- *3.5. Local Diversion - Outbound*
- *3.6. Local Diversion - Inbound*
- *3.7. Remote Diversion - Outbound*
- *3.7. Remote Diversion - Outbound*
- *3.9. Transfer*
- *3.10. Virtual Call -Outbound*
- *3.11. Virtual Call -Inbound*

3.1. Executive Intrusion - Normal

Step	Dialogic API	Action/Result	Dialogic Event
1	<code>gc_MakeCall()</code> (with Intrusion IE)	-->	
2		<--	GCEV_PROCEEDING
3	---	Intrusion succeeded <--	--- GCEV_CONNECTED
4	---	Intrusion failed <--	--- GCEV_DISCONNECT

1. Application places an outgoing call using the `gc_MakeCall()` function to a busy extension with Intrusion Type set to INTRUDE_NORMAL value. See *Table 2. gc_SetInfoElem() Information Elements* for the format of the Intrusion IE.
2. Receives call proceeding (GCEV_PROCEEDING) event.
3. Receives call connected (GCEV_CONNECTED) event. Call successfully intruded.
4. Receives call disconnect (GCEV_DISCONNECT) event. Call was not intruded.

3.2. Executive Intrusion - with Prior Validation

Step	Dialogic API	Action/Result	Dialogic Event
1	<code>gc_MakeCall()</code> (with Intrusion IE)	-->	
2		<--	GCEV_PROCEEDING (with Busy IE)
3	<code>gc_SndMsg()</code> (SndMsg Intrude)	-->	
4	---	Intrusion succeeded <--	--- GCEV_CONNECTED
5	---	Intrusion failed <--	--- GCEV_DISCONNECT

1. Application places an outgoing call using the `gc_MakeCall()` function to a busy extension with Intrusion Type set to INTRUDE_PRIOR_VALIDATION value. See *Table 2. gc_SetInfoElem() Information Elements* for the format of Intrusion IE.

3. DPNSS Call Scenarios

2. Receives call proceeding (GCEV_PROCEEDING) event with indication that remote party was busy. Use the **gc_GetSigInfo()** function to retrieve the BUSY_IE value. See *Table 1. gc_GetCallInfo() and gc_GetSigInfo() Information Elements* for the Intrusion IE format.
3. Application sends intrude request using the **gc_SndMsg()** function. See the **gc_SndMsg()** function description in your *GlobalCall API Software Reference* and/or *GlobalCall ISDN Technology User's Guide* for details.
4. Receives call connected (GCEV_CONNECTED) event. Call successfully intruded.
5. Receives call disconnect (GCEV_DISCONNECT) event. Call was not intruded.

3.3. Hold and Retrieve - Locally Initiated

Step	Dialogic API	Action/Result	Dialogic Event
1	--- gc_HoldCall()	Call connected -->	---
2	---	Call held <--	--- GCEV_HOLDACK
3	Unroute SCbus time slot for held call	:	
4	gc_RetrieveCall()		
5		<--	GCEV_RETRIEVEACK
6	Reroute SCbus time slot for retrieved call		
7	---	Call not held <--	--- GCEV_HOLDREJ
8	Take no action		

1. Application places a connected call on hold using the **gc_HoldCall()** function.
2. When call is held, application will receive a hold acknowledge (GCEV_HOLDACK) event.
3. Application should unroute SCbus time slot for held call.

4. Application retrieves a held call using the **gc_RetrieveCall()** function.
5. When call is retrieved, application will receive a retrieve acknowledge (GCEV_RETRIEVEACK) event.
6. Application should reroute SCbus time slot for retrieved call.
7. When a call is not held, application will receive a hold reject (GCEV_HOLDREJ) event.
8. Application should take no action on call's SCbus time slot.

NOTE: The retrieval of a held call cannot be rejected when using the DPNSS protocol.

3.4. Hold and Retrieve - Remotely Initiated

Step	Dialogic API	Action/Result	Dialogic Event
1	---	Call connected <--	--- GCEV_HOLDCALL
2	Unroute SCbus time slot for held call	Call held	---
3	gc_HoldAck()	--> :	
4		<--	GCEV_RETRIEVECALL
5	Reroute SCbus time slot for retrieved call		
6	Take no action	Call not held	---
7	gc_HoldRej()	-->	

1. Receives a request (GCEV_HOLDCALL event) to place a connected call on hold.
2. Application accepts hold request; should unroute SCbus time slot for requested call.
3. Application accepts hold request using the **gc_HoldAck()** function.
4. Receives request (GCEV_RETRIEVECALL event) to retrieve a held call.

3. DPNSS Call Scenarios

5. Application receives retrieve request; should reroute SCbus time slot for requested call.
6. Application rejects hold request; should take no action on call's SCbus time slot.
7. Application rejects hold request using the **gc_HoldRej()** function.

NOTE: The retrieval of a held call cannot be rejected when using the DPNSS protocol.

3.5. Local Diversion - Outbound

Step	Dialogic API	Action/Result	Dialogic Event
1	gc_MakeCall()	-->	
2		<--	GCEV_PROCEEDING (with Diversion IE, diversion location: DIVERT_LOCAL)
3		<--	GCEV_CONNECTED

1. Application places an outgoing call using the **gc_MakeCall()** function.
2. Receives call proceeding (GCEV_PROCEEDING) event with indication that call was diverted to another location. Use the **gc_GetSigInfo()** function to retrieve the Diversion IE. See *Table 1. gc_GetCallInfo() and gc_GetSigInfo() Information Elements* for the Diversion IE format.
3. Receives call connected (GCEV_CONNECTED) event. Call established.

3.6. Local Diversion - Inbound

Step	Dialogic API	Action/Result	Dialogic Event
1		<--	GCEV_OFFERED
2	gc_SndMsg() (SndMsg_Divert, Diversion Location: DIVERT_LOCAL)	-->	
3	gc_AnswerCall()	-->	
4		<--	GCEV_ANSWERED

1. Receives incoming call (GCEV_OFFERED) event.

2. Application diverts incoming call to a different extension using the **gc_SndMsg()** function. See the **gc_SndMsg()** function description in the *GlobalCall API Software Reference* and/or *GlobalCall ISDN Technology User's Guide* for details.
3. Application answers incoming call using the **gc_AnswerCall()** function.
4. Receives call connected (GCEV_ANSWERED) event.

3.7. Remote Diversion - Outbound

Step	Dialogic API	Action/Result	Dialogic Event
1	gc_MakeCall()	-->	
2	gc_SndMsg() (SndMsg_Divert, Diversion Location: DIVERT_REMOTE)	<--	GCEV_PROCEEDING (with Diversion IE, Diversion Location: DIVERT_REMOTE)
3	gc_DropCall()	-->	
4		<--	GCEV_DROPCALL
5	gc_ReleaseCall()	-->	
6	gc_MakeCall() (with Diversion IE)	-->	
	---	Divert succeeded	---
7		<--	GCEV_PROCEEDING
8		<--	GCEV_DIVERTED
9		<--	GCEV_CONNECTED
	---	Divert failed	---
10		<--	GCEV_DISCONNECT

1. Party 1 calls Party 2 by issuing a **gc_MakeCall()** function.
2. Party 1 receives a GCEV_PROCEEDING event from Party 2 with an indication that the call needs to be diverted to Party 3. Diversion IE will contain the telephone number of Party 3. See *Table 1. gc_GetCallInfo() and gc_GetSigInfo() Information Elements* for the Diversion IE format.
3. Party 1 disconnects original call to Party 2 using a **gc_DropCall()** function.
4. Party 1 receives a call disconnect (GCEV_DROPCALL) event from Party 2.
5. Releases first call using a **gc_ReleaseCall()** function.
6. Party 1 diverts call to Party 3 by issuing a **gc_MakeCall()** function. Calling party number IE should contain Party 3's telephone number. Diversion IE

3. DPNSS Call Scenarios

should contain Party 2's telephone number. See the **gc_SetInfoElem()** function description in your *GlobalCall API Software Reference* and/or *GlobalCall ISDN Technology User's Guide* for details on sending a Diversion IE.

7. Party 1 receives proceeding (GCEV_PROCEEDING) event from Party 3.
8. Party 1 receives divert successful (GCEV_DIVERTED) event from Party 3.
9. Party 1 receives call connected (GCEV_CONNECTED) event from Party 3. Call successfully diverted.
10. Party 1 receives divert failed (GCEV_DISCONNECT) event from Party 3. Call was not diverted.

3.8. Remote Diversion - Inbound

Step	Dialogic API	Action/Result	Dialogic Event
1		<--	GCEV_OFFERED
2	gc_SndMsg() (SndMsg_Divert, Diversion Location: DIVERT_REMOTE)	-->	
3		<--	GCEV_DISCONNECT
4	gc_DropCall()	-->	
5		<--	GCEV_DROPCALL
6	gc_ReleaseCall()	-->	

1. Party 2 receives incoming call (GCEV_OFFERED) event from Party 1.
2. Party 2 diverts incoming call to Party 3. Send Party 3's telephone number as Diversion number. See *Table 3. gc_SndMsg() DPNSS Message Types* for the format of the SndMsg_Divert message.
3. Party 1 disconnects call to Party 2.
4. Party 2 drops call using the **gc_DropCall()** function.
5. Party 2 receives drop call (GCEV_DROPCALL) event from Party 1.
6. Party 2 releases call using the **gc_ReleaseCall()** function.

3.9. Transfer

Step	Dialogic API	Action/Result	Dialogic Event
1		<--	GCEV_OFFERED (CRN 1)
2	gc_AnswerCall() (CRN 1)	-->	
3	gc_HoldCall() (CRN 1)	-->	
4		<--	GCEV_HOLDACK (CRN 1)
5	gc_MakeCall() (CRN 2, with Inquiry IE)	-->	
6		<--	GCEV_PROCEEDING (CRN 2 with Inquiry IE)
7		<--	GCEV_CONNECTED (CRN 2 with Inquiry IE)
8	gc_SndMsg() (SndMsg Transfer, CRN 1)	-->	
9	gc_SndMsg() (SngMsg Transfer, CRN 2)	-->	
10		<--	GCEV_TRANSFERACK (CRN 1)
11		<--	GCEV_TRANSFERACK (CRN 2)
12	Cross connect CRN 1's and CRN 2's SBus time slots		
13		<--	GCEV_TRANSIT (CRN 1)
14	gc_SndMsg() (SndMsg Transit, CRN 2)	-->	
15		<--	GCEV_TRANSIT (CRN 2)
16	gc_SndMsg() (SndMsg Transit, CRN 1)	-->	
17		<--	GCEV_DISCONNECT (CRN 1)
18	gc_DropCall() (CRN 1)	-->	
19		<--	GCEV_DROPCALL (CRN 1)
20	gc_ReleaseCall() (CRN 1)	-->	
21		<--	GCEV_DISCONNECT (CRN 2)
22	gc_DropCall() (CRN 2)	-->	
23		<--	GCEV_DROPCALL (CRN 2)
24	gc_ReleaseCall() (CRN 2)	-->	

1. Party 2 receives incoming call (GCEV_OFFERED) event from Party 1.
2. Party 2 answers call from Party 1 using **gc_AnswerCall()** function.
3. Party 2 places call on hold using a **gc_HoldCall()** function.
NOTE: Some switches may not support holding a call.
4. Party 2 receives a call on hold acknowledge (GCEV_HOLDACK) event.

3. DPNSS Call Scenarios

5. Party 2 places an inquiry call to Party 3 using the **gc_MakeCall()** function. Application should use Party 1's telephone number as the calling party number and Party 3's telephone number as called party number. See *Table 2. gc_SetInfoElem() Information Elements* for the Inquiry IE format.
6. Party 2 receives a call proceeding (GCEV_PROCEEDING) event with an Inquiry IE from Party 3. See *Table 2. gc_SetInfoElem() Information Elements* for the Inquiry IE format.
7. Party 2 receives a call connected (GCEV_CONNECTED) event with Inquiry IE from Party 3. See *Table 2. gc_SetInfoElem() Information Elements* for the Inquiry IE format.
8. Party 2 sends a transfer request to Party 1 with a TRANSFER_ORIG value as the transfer direction using the **gc_SndMsg()** function. See *Table 3. gc_SndMsg() DPNSS Message Types* for message format.
9. Party 2 sends a transfer request to Party 3 with a TRANSFER_TERM value as the transfer direction using the **gc_SndMsg()** function. See *Table 3. gc_SndMsg() DPNSS Message Types* for message format.
10. Party 2 receives a transfer acknowledge (GCEV_TRANSFERACK) event from Party 1.
11. Party 2 receives a transfer acknowledge (GCEV_TRANSFERACK) event from Party 3. Transfer completed. At this time, Party 2 loses control of the call.
12. Application should cause Party 1 to listen to Party 2's SCbus transmit time slot and Party 2 to listen to Party 1's SCbus transmit time slot.
13. Party 2 receives transit (GCEV_TRANSIT) event from Party 1. Party 2 should retrieve the content of the Transit IE using the **gc_GetSigInfo()** function.
14. Party 2 sends content of the Transit IE (unchanged) from Party 1 to Party 3 using the **gc_SndMsg()** function. See *Table 3. gc_SndMsg() DPNSS Message Types* for message format.
15. Party 2 receives transit (GCEV_TRANSIT) event from Party 3. Party 2 should retrieve the content of the Transit IEe using the **gc_GetSigInfo()** function.

16. Party 2 sends content of Transit IE (unchanged) from Party 3 to Party 1 using the **gc_SndMsg()** function. See *Table 3. gc_SndMsg() DPNSS Message Types* for message format.
17. Party 2 receives a disconnect all (GCEV_DISCONNECT) event from Party 1.
18. Party 2 drops call to Party 1 using the **gc_DropCall()** function.
19. Party 2 receives a drop call (GCEV_DROPCALL) event from Party 1.
20. Party 2 releases call to Party 1 using the **gc_ReleaseCall()** function.
21. Party 2 receives a disconnect call (GCEV_DISCONNECT) event from Party 3.
22. Party 2 drops call to Party 3 using the **gc_DropCall()** function.
23. Party 2 receives a drop call (GCEV_DROPCALL) event from Party 3.
24. Party 2 releases call to Party 3 using the **gc_ReleaseCall()** function.

- NOTES:**
1. Steps 3 and 4 are optional and need not be carried out on most PBXs.
 2. Steps 12 through 16 may be repeated multiple times depending on when or whether the distant PBX supports Route Optimization. When Route Optimization occurs, or if either end of the transferred call is terminated, the call flow proceeds to step 17.

3.10. Virtual Call -Outbound

Step	Dialogic API	Action/Result	Dialogic Event
1	gc_MakeCall() (with Virtual Call IE)	-->	
2		<--	GCEV_DISCONNECT
3	gc_DropCall()	-->	
4		<--	GCEV_DROPCALL
5	gc_ReleaseCall()	-->	

1. Application places an outgoing call with Virtual Call IE and any other information set, such as NSI strings or Extension Status using the **gc_MakeCall()** function. See *Table 2. gc_SetInfoElem() Information Elements* for the format of the Virtual Call IE.

3. DPNSS Call Scenarios

2. Receives call disconnected (GCEV_DISCONNECT) event. Use the **gc_ResultValue()** function to retrieve the clearing cause. A RESP_TO_STAT_ENQ value means that the call was Acknowledged and a FACILITY_REJECT value means that the call was Rejected.
3. Application issues a **gc_DropCall()** function.
4. Receives a drop call (GCEV_DROPCALL) event.
5. Application issues a **gc_ReleaseCall()** function.

3.11. Virtual Call -Inbound

Step	Dialogic API	Action/Result	Dialogic Event
1		<--	GCEV_OFFERED (with Virtual Call IE)
2	gc_DropCall()	-->	
3		<--	GCEV_DROPCALL
4	gc_ReleaseCall()	-->	

1. Receives a call offered (GCEV_OFFERED) event with an indication that this is a virtual call. Use the **gc_GetSigInfo()** function to retrieve the Virtual Call IE and any other information, such as NSI strings.
2. Application issues a **gc_DropCall()** function with clearing cause set to the RESP_TO_STAT_ENQ value to acknowledge the call or set to the FACILITY_REJECT value to reject the call.
3. Receives a drop call (GCEV_DROPCALL) event.
4. Application issues a **gc_ReleaseCall()** function.

Index

B

BUSY_IE, 4, 19

C

call control scenario, 17

Call held, 19, 20

Call not held, 19, 20

D

Digital Private Network Signaling
System
DPNSS, 1

Diversion Bypass IE, 9

Diversion IE, 4, 9, 21, 22

Diversion Number, 4, 5, 9, 13

Diversion Validation IE, 5

DIVERSION_BYPASS_IE, 10

DIVERSION_IE, 4, 9, 13

DIVERSION_VALIDATION_IE, 5

Divert failed, 22

Divert succeeded, 22

DIVERT_IMMEDIATE, 4, 9, 13

DIVERT_LOCAL, 4, 9, 13, 21

DIVERT_NO_REPLY, 4, 9, 13

DIVERT_ON_BUSY, 4, 9, 13

DIVERT_REMOTE, 4, 9, 13, 22, 23

DPNSS protocol
Scenarios, 17

E

Executive Intrusion - Normal, 18

Executive Intrusion - with Prior
Validation, 18

Extension Status IE, 7, 10

EXTENSION_STATUS_IE, 7, 10

F

FACILITY_REJECT, 27

G

GC API
GlobalCall Application
Programming Interface, 1

gc_AnswerCall(), 21, 22, 24

gc_DropCall(), 22, 23, 24, 26, 27

gc_GetCallInfo(), 3

gc_GetSigInfo(), 3, 19, 21, 25, 27

gc_HoldAck(), 20

gc_HoldCall(), 19, 24

gc_HoldRej(), 20, 21

gc_MakeCall(), 8, 9, 10, 11, 18, 21, 22,
24, 25, 26

gc_ReleaseCall(), 22, 23, 24, 26, 27

gc_ResultValue(), 27

gc_RetrieveCall(), 19, 20

gc_SetInfoElem(), 3, 8, 23

gc_SndMsg(), 3, 12, 19, 21, 22, 23, 24,
25, 26

GCEV_ANSWERED
event, 21, 22

GCEV_CONNECTED
event, 18, 19, 21, 22, 23, 24, 25

GCEV_DISCONNECT
event, 18, 19, 22, 23, 24, 26, 27

GCEV_DIVERTED
event, 22, 23

GCEV_DROPCALL
event, 22, 23, 24, 26, 27

GCEV_HOLDACK
event, 19, 24

GCEV_HOLDCALL
event, 20

GCEV_HOLDREJ
event, 19, 20

GCEV_NSI
event, 6

GCEV_OFFERED
event, 4, 5, 7, 11, 21, 23, 24, 27

GCEV_PROCEEDING
event, 4, 18, 19, 21, 22, 23, 24, 25

GCEV_RETRIEVEACK
event, 19, 20

GCEV_RETRIEVECALL
event, 20

GCEV_TRANSFERACK
event, 24, 25

GCEV_TRANSIT
event, 5, 15, 24, 25

GlobalCall Application Programming
Interface
GC API, 1

H

Hold and Retrieve - Locally Initiated,
19

Hold and Retrieve - Remotely Initiated,
20

I

IE
information element, 3

information element
IE, 3

Inquiry IE, 10, 24, 25

INQUIRY_IE, 10

Integrated Services Digital Network
ISDN, 1

INTRUDE, 14

INTRUDE_IE, 13

INTRUDE_NORMAL, 8, 18

INTRUDE_PRIOR_VALIDATION, 8,
18

INTRUDE_WITHDRAW, 14

Intrusion failed, 18

Intrusion IE, 4, 8, 18, 19

Intrusion succeeded, 18

INTRUSION_IE, 8

INTRUSION_LEVEL_1, 8

INTRUSION_LEVEL_2, 8

INTRUSION_LEVEL_3, 8

ISDN
Integrated Services Digital Network,
1

ISDN message type, 3

L

Local Diversion - Inbound, 21
Local Diversion - Outbound, 21

N

Network Specific Indications
 NSI, 6, 12
Network Specific Indications IE
 NSI IE, 6
NSI String, 6, 12, 14
NSI_EEM, 6, 12, 14
NSI_IE, 6, 12, 14
NSI_LLM, 6, 12, 14

P

PBX
 Private Branch eXchange, 1
Private Branch eXchange
 PBX, 1

Q

Q.SIG protocol
 Scenarios, 17

R

Remote Diversion - Inbound, 23
Remote Diversion - Outbound, 22
RESP_TO_STAT_ENQ, 27
retrieval
 held call, 20, 21

S

SndMsg_Divert, 13, 21, 22, 23
SndMsg_Intrude, 13, 18
SndMsg_NSI, 14
SndMsg_Transfer, 14, 24
symbols, '*' and '#', 6, 11

T

Text Display IE, 5, 11
Text Display String, 6, 11
TEXT_DISPLAY_IE, 5, 11
TEXT_TYPE_MESSAGE, 6, 11
TEXT_TYPE_NAME, 6, 11
TEXT_TYPE_NOT_PRESENT, 6, 11
TEXT_TYPE_REASON, 6, 11
Transfer
 call, 24

TRANSFER_IE, 14
TRANSFER_ORIG, 15, 25
TRANSFER_TERM, 15, 25
Transit IE, 5, 25, 26
TRANSIT_IE, 5, 15

V

Virtual Call IE, 7, 10, 11, 26, 27
Virtual Call -Inbound, 27
Virtual Call -Outbound, 26

