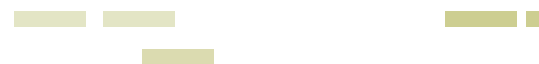


Creating SS7 Circuit-Switched Applications using Dialogic® Global Call API

Installation and Configuration



Executive Summary

This application note describes the installation and configuration of circuit-switched telephony systems that combine Dialogic® Signaling Boards and Dialogic® DM3 Media Boards on platforms based on Linux and Windows® operating systems.

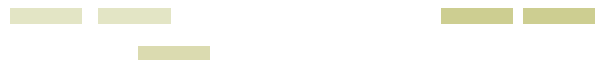


Table of Contents

Introduction.....	2
System Description.....	2
Hardware Requirements.....	2
Software Requirements	2
Step 1 – Installing the Dialogic® Software.....	2
Linux Operating System.....	2
Windows® Operating System.....	5
Locating the Appropriate Clear Channel Media Load.....	6
Step 2 – Installing the Dialogic® SS7 Development Package	7
Linux Operating System.....	7
Windows® Operating System.....	7
Step 3 – Configuring the Dialogic® Signaling Distributed Architecture	8
Software (system.txt and config.txt)	
system.txt.....	8
config.txt	9
Step 4 – Proving the SS7 Configuration	12
Step 5 – Configuring the Dialogic® Global Call API for SS7 Software (gcss7.cfg)	13
Step 6 – Starting the Software.....	14
Step 7 – Running the gc_basic_call_model Demo	15
Troubleshooting.....	15
Summary	15
References.....	15
Appendix A: Linux system.txt File	16
Appendix B: Linux config.txt File	17
Appendix C: Linux gcss7.cfg.txt File	18
Appendix D: Windows® system.txt File	21
Appendix E: Windows® config.txt File	22
Appendix F: Windows® gcss7.cfg.txt File.....	23
Acronyms.....	25
For More Information.....	25

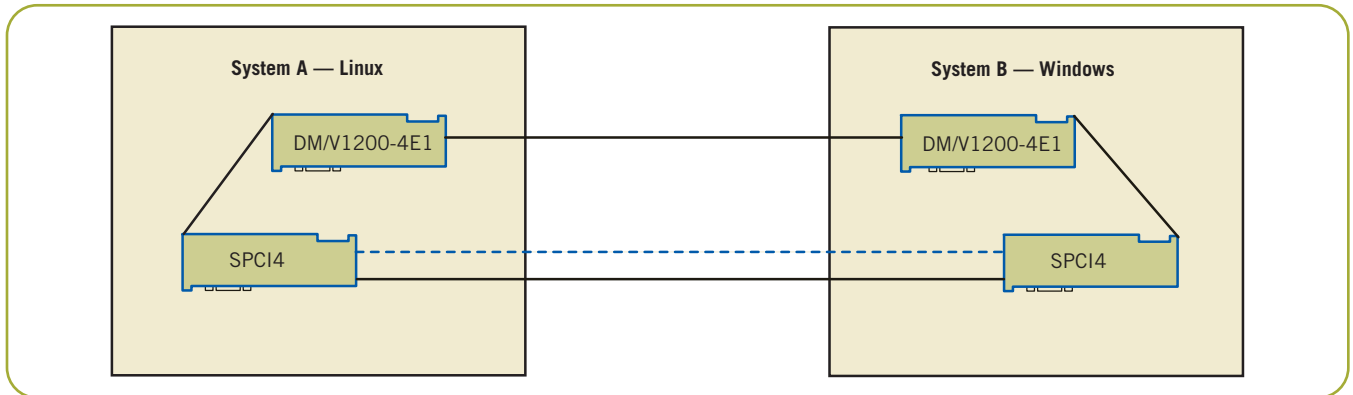


Figure 1. Sample System Configuration

Introduction

This application note describes the installation, configuration, and run-time operation of two computer telephony systems that combine a Dialogic® Signaling Board and a Dialogic® DM3 Media Board.

These systems will be connected back-to-back using crossover cables. The `gc_basic_call_model` demo will show how the Dialogic® Global Call API library facilitates the development of SS7 telephony applications running on top of the ISDN User Part (ISUP) signaling protocol.

System Description

In this application note, System A runs Red Hat Enterprise Linux v3 and System B runs Windows® XP; however, readers are free to use the same operating system on both System A and System B (see Figure 1). Since the configuration files given in the appendices are operating system independent, this should not cause any problems.

Hardware Requirements

- 2 PCs with Pentium III processors, 512 MB RAM
- 2 Dialogic® SPCI4 SS7 Interface Boards with ISUP or ISUP-L license buttons
- 2 Dialogic® DM/V1200-4E1 Voice Boards (Dialogic® Voice Boards, Dialogic® DM/V-A Media Boards or Dialogic® DM/V-B Media Boards in PCI)
- 2 H.100 CT Bus cables
- 2 crossover E-1 cables

Software Requirements

Linux Operating System

- Red Hat Enterprise Linux V3 Update 4
- Dialogic® SS7 Development Package for Linux V5.05 (or greater)

- SS7 code file for SPCI4 (`ss7.dc3`) V1.13 (or greater)
- Dialogic® System Release 6.1 for Linux Service Update 268 (or greater)

Windows® Operating System

- Microsoft® Windows XP
- Dialogic® SS7 Development Package for Windows® V5.02 (or greater)
- SS7 code file for SPCI4 (`SS7.DC3`) V1.13 (or greater)
- Dialogic® System Release 6.0 PCI for Windows® Service Update 166 (or greater)

Notes:

1. Dialogic® Signaling Distributed Architecture (SigDiA) software can be downloaded from <http://www.dialogic.com/support/helpweb/signaling/default.htm>
2. More information about Dialogic® System Releases and Service Updates can be found on <http://resource.dialogic.com/telecom/support/releases/serviceUpdates/index.htm>

Step 1 – Installing the Dialogic® Software

Linux Operating System

Refer to the *Dialogic® System Release 6.1 for Linux Software Installation Guide* (http://www.dialogic.com/manuals/sr61lin/release_install.pdf) for instructions on how to install the Dialogic® System Release software. Note that you select “Dialogic® DMV/DMN/DMT” and the Global Call SS7 Support Software packages at the installation menu.

1. When the Dialogic® System Release Software is successfully installed, start the `config.sh` script to copy and load the DM3 driver files. When this process completes, the Dialogic® DM3 Media Board should be detected and you should be prompted with this menu:

```
Dialogic(R) Configuration Manager - Main Screen
-----
1) Dialogic(R) DM3 Board Summary
2) Dialogic(R) Board Summary (NO BOARDS)
3) Dialogic(R) IPT Board Summary
4) TDM Bus Settings
-----
(s to save, x to save & quit, q to quit) the configuration
? for help and ! for navigation help
You can only configure one board at a time. Enter the number associated
with the product category of the board you want to configure :1
```

2. Select item 1 to get the list of DM3 Media Boards that are detected on your system:

```
Dialogic (R) DM3 Board Summary
-----
You must configure or disable each board shown. After a board is
configured, a valid PCD file name is displayed in the PCD File Name
column.
-----
Thumb   Board           Log   PCD
Wheel   Status          Model ID   File Name
-----
0)      C              DM/V1200-4E1  1   NOT_SET
-----
(s to save, x to save & quit, q to quit) the configuration
p to return to Dialogic(R) Configuration Manager - Main Screen
? for help and ! for navigation help
Enter the Thumb Wheel of the board to configure: 0
```

3. Enter the Thumb Wheel number of the board you want to configure (0 in this case).

```
Modify Board Settings
-----
These are the current settings for the board selected:
Physical Slot..... : 0
Model Name..... : DM/V1200-4E1
Logical ID..... : 1
Board Status..... : Configured
PCD File Name..... : NOT_SET
CONFIG File Name... : NOT_SET
-----
The following items can be modified:
1) Specify the PCD File
2) Trunk Configuration
3) Protocol Development Kit (PDK) Configuration (NOT APPLICABLE)
4) Modify NIC Configuration (NOT APPLICABLE)
5) Copy Configuration From Board
6) Advanced Board Settings
-----
(s to save, x to save & quit, q to quit) the configuration
p to return to Dialogic(R) DM3 Board Summary
? for help and ! for navigation help
Enter the number of the item to modify: 1
```

- Select item number 1 (Specify the PCD File) and you should be prompted with a list of available media loads for the board you are using.

Specify the PCD File

A Product Configuration Description (PCD) file must be selected to configure your board. The corresponding CONFIG file will be automatically selected.

The following PCD files are valid for your board:

- 1) ml1_qs_dass2.pcd
- 2) ml1_qs_dpns5.pcd
- 3) qs_isdn_net5.pcd
- 4) ml1_qs_net5.pcd
- 5) qs_isdn_qlsig1.pcd
- 6) ml1_qs_qlsig1.pcd
- 7) qs_r2mf.pcd
- 8) ml1_qs_r2mf.pcd
- 9) ml1_qs_ts16.pcd

* is current setting

(s to save, x to save & quit, q to quit) the configuration

p to return to Modify Board Settings

? for help and ! for navigation help

Enter the number corresponding to the PCD file you want to select: 1

- Select a clear channel media load, such as:

- ml1_qs_ts16.pcd for a regular Dialogic® DM/V1200-4E1 Media Board
- ml2_qsa_ts16.pcd for a Dialogic® DM/V1200A-4E1 Media Board

Note: When using a Dialogic® DM/V-B Media Board, select Trunk Configuration (option 2 in the menu below) and assign E1CC on your E1 trunks.

Modify Board Settings

These are the current settings for the board selected:

Physical Slot..... : 4
 Model Name..... : DMV1200BTEP
 Logical ID..... : 2
 Board Status..... : Configured
 PCD File Name..... : qsb_default.pcd
 CONFIG File Name... : qsb_default.config

The following items can be modified:

- 1) Specify the PCD File
- 2) Trunk Configuration
- 3) Protocol Development Kit (PDK) Configuration (NOT APPLICABLE)
- 4) Modify NIC Configuration (NOT APPLICABLE)
- 5) Copy Configuration From Board
- 6) Advanced Board Settings

(s to save, x to save & quit, q to quit) the configuration

p to return to Dialogic(R) DM3 Board Summary

? for help and ! for navigation help

Enter the number of the item to modify: 1

- From the Dialogic® Configuration Manager - Main Screen, type the number 4 and press **Enter** to select the TDM Bus Settings option.
- Select the "" menu (item number 2 in the TDM Bus Settings menu) and make sure that your Dialogic DM/V board is set as "Secondary Master Board" (item number 2), because the Dialogic® SS7 board is to be the primary CT Bus clock master.

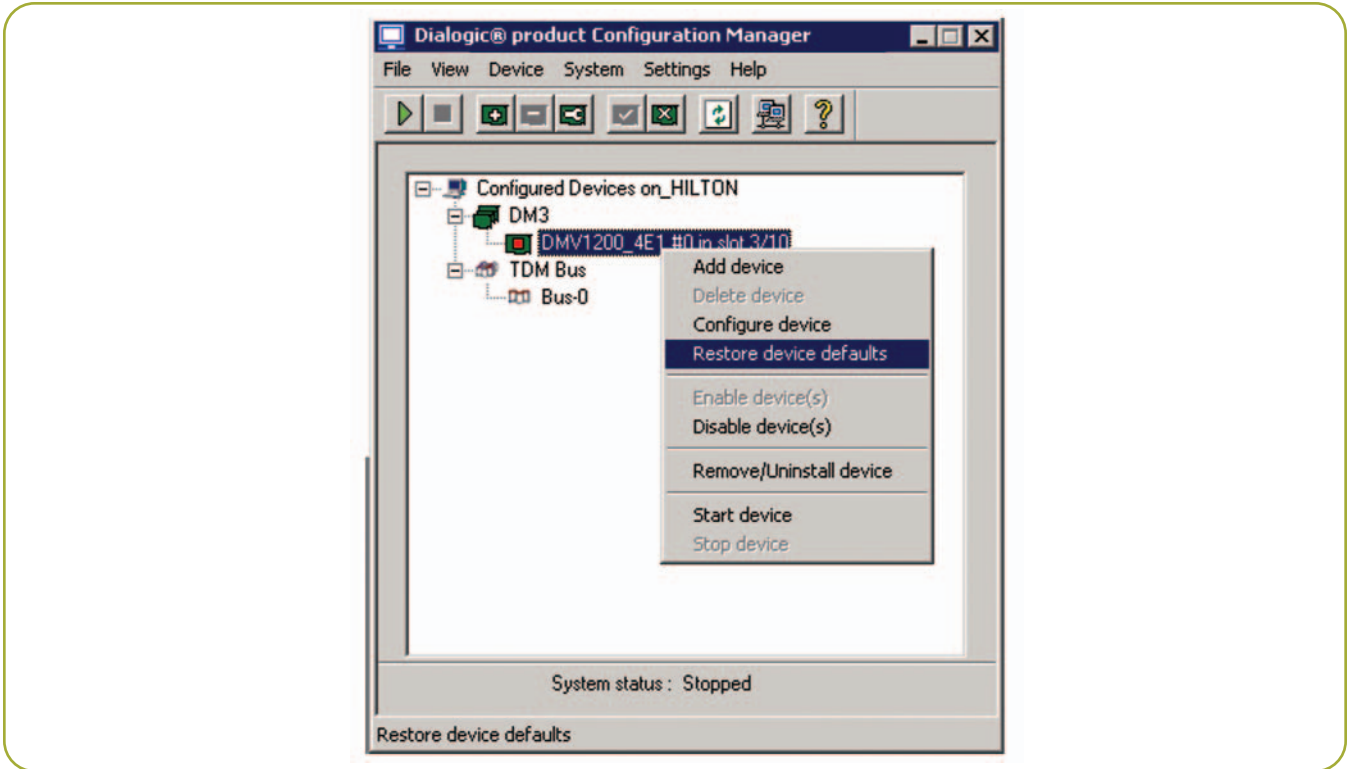


Figure 2. Forcing a Hardware Detection

Windows® Operating System

Refer to the *Dialogic® System Release 6.0 PCI for Windows Software Installation Guide*

(http://www.dialogic.com/manuals/sr60winpci/release_install.pdf) for installation instructions.

1. When the “Select Features” window appears during installation, check these options:
 - Development Package — Allows for recompiling the application, if it is needed.
 - Core Runtime Package — Installs Dialogic® board drivers, firmware, and parameter files.

- Demos — Installs the sample programs for Dialogic® boards and, among others, the `gc_basic_call_model` application, which will be used to make and receive calls between the two systems.

Once the Dialogic® System Release Software has been installed and the system has been rebooted, do the following:

2. Open up the Dialogic® Configuration Manager
3. Right click on the DM3 device and select “Restore Device Default” to force a re-detection of your Dialogic® hardware, as shown in Figure 2.

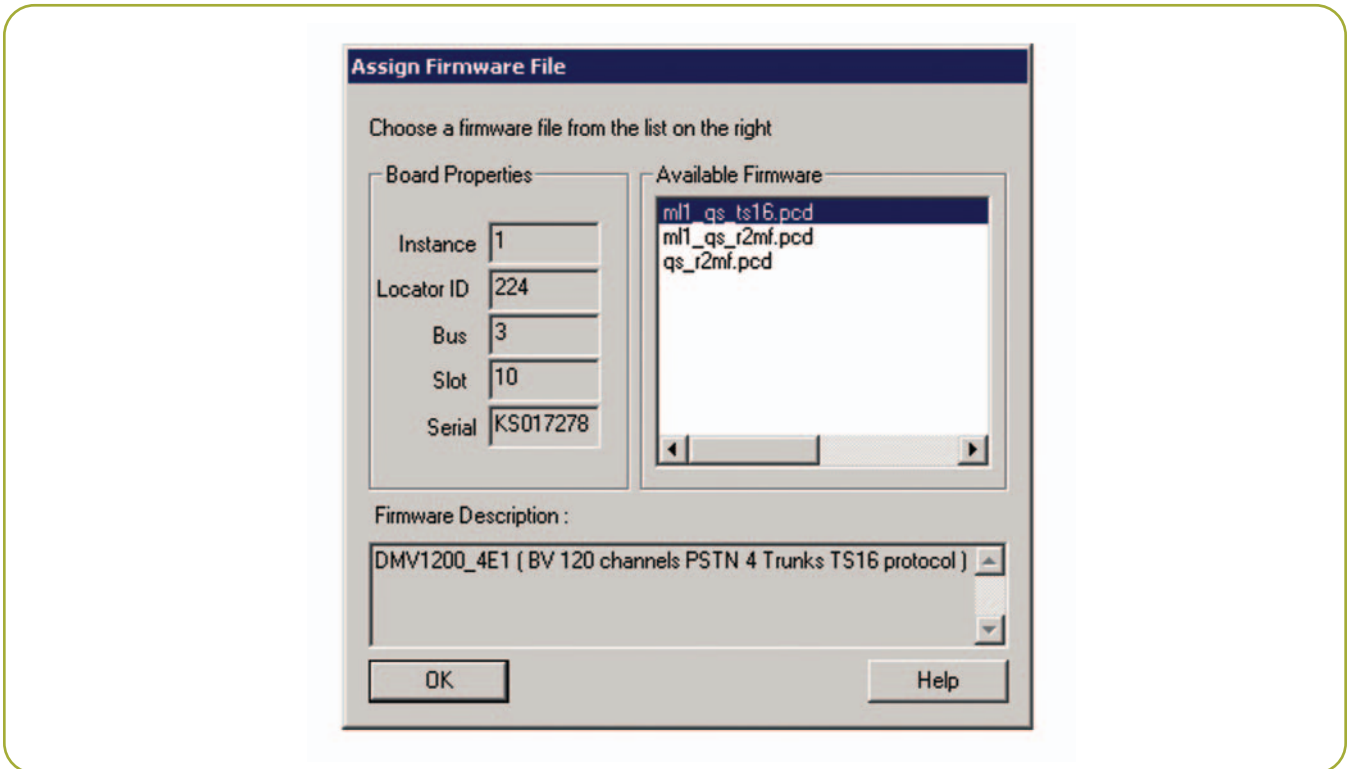


Figure 3. Selecting a Clear Channel Media Load

4. Select a clear channel media load (for example, ml1_qs_ts16.pcd for a regular DM/V1200-4E1), as shown in Figure 3.

Locating the Appropriate Clear Channel Media Load

If you cannot find an appropriate clear channel media load for your Dialogic® DM3 Media Board (for example, for a Dialogic® DM/V960-4T1 Voice Board or for a Dialogic® DM/IP Board), then manually edit the corresponding *.config file in order to configure the E-1 or T-1 trunk in clear channel mode as follows:

1. Identify the *.config file that corresponds to the media load you have chosen for your DM3 Media Board (for example, ml1_qs_cas.config) and open it in a text editor.
2. Search for the first occurrence of the 0x1602 parameter:

```
[ lineAdmin.1]
...
SetParm=0x1602,4 ! SignalingType
(CAS=4, CCS=5, Clear=6)
...
```

3. Change the parameter value to 6 to configure the corresponding trunk in clear channel mode:


```
[ lineAdmin.1]
...
SetParm=0x1602,6 ! SignalingType
(CAS=4, CCS=5, Clear=6)
...
```
4. Repeat this operation for the other trunks you want to set in clear channel mode ([lineAdmin.2], [lineAdmin.3], etc.)
5. For Windows® and Linux operating systems, the FCD file is automatically created when the PCD file and modified CONFIG file are downloaded to the board.
6. Re-start the Dialogic® Service, and verify that there are now 24 T-1 time slots or 31 E-1 time slots on the clear channel trunks, using “Devmapdump”:

```
c:\Program Files\Dialogic\bin\
Devmapdump > a.txt notepad a.txt
```


Step 2 – Installing the Dialogic® SS7 Development Package

1. Download the Dialogic® Signaling Distributed Architecture (SigDiA) software components from the Dialogic web site. The download sections discussed in this application note can be found at <http://www.dialogic.com/support/helpweb/signaling/default.htm>.
2. Go to the *Product Specific* on the web page and in Section 2.7 download the SS7 Binary (code file) for Dialogic® SPCI2S SS7 Interface Board and Dialogic® SPCI4 SS7 Interface Board. Keep the code file (SS7.DC3) in a directory on your system (usually the “Septel” directory for simplicity). On a Linux system, rename SS7.DC3 to ss7.dc3.
3. Go to the *Generic Information* section on the web page and, in Section 3.1, download the Dialogic® SS7 Development Package for Windows® or the Dialogic SS7 Development Package for Linux.

Linux Operating System

1. For Linux, the Dialogic SS7 Development Package is a compressed tar file. Manually uncompress and extract the contents of this file (using the `tar` utility) into a directory you have previously created.
2. Before attempting to load the SPCI driver module, compile the object file from the driver source code using the `build_spici_cpm.sh` script file. This build script assumes a suitable environment for building Kernel modules is available. This must include the appropriate kernel include files found at: `/usr/src/linux-`uname -r`/include` (for example, `/usr/src/linux-2.4.7-10/include`). If these are not found, the build will fail. Following that, the `install_spici_cpm.sh` script installs the device driver, automatically allocates a major device number, and creates the four appropriate device nodes.

Note: ‘`install_spici_cpm.sh`’ dynamically loads the driver module into the kernel. The module will remain loaded as long as the PC is not rebooted. After a system reboot, you will have to re-load the module using the `insmod` utility.

Windows® Operating System

The Dialogic SS7 Development Package for Windows® has an Install Shield script that copies the file and drivers to the appropriate directories. More information can be found in Section 2.1 of the *SS7 Boards Programmer’s Manual for SPCI2S, SPCI4 and CPM8* (see the *For More Information* section).

1. The SPCI device driver must be manually registered to the system using the following command:


```
c:\> net start septel
```
2. Once you have done this, open up the Windows Device Manager.
3. Make sure to turn on the Show Hidden devices option in the View menu. You should now be able to see the Septel device under the Non-Plug and Play Drivers.
4. Double-click on the Septel device, go to the Driver tab, and change the startup mode to automatic.

Step 3 – Configuring the Dialogic® Signaling Distributed Architecture Software (`system.txt` and `config.txt`)

Before configuring the Dialogic® SS7 board and protocol stack, an understanding of the underlying SS7 concepts, such as link, link set, point codes, routes, and specifically the Message Transfer Part (MTP) layers and the ISDN User Part (ISUP) is fundamental. If you are unfamiliar with those concepts, you can review the SS7 tutorial on the Dialogic web site at <http://www.dialogic.com/support/helpweb/signaling/tutorial/ss7.htm>. For more information about the SS7 architecture, read *ISDN & SS7 – Architectures for Digital Signaling Networks* [Black] (see the *References* section.)

Two configuration files are used when setting up the software for a Dialogic® Signaling Board: `system.txt` and `config.txt`. Figure 4 illustrates the protocols to be configured.

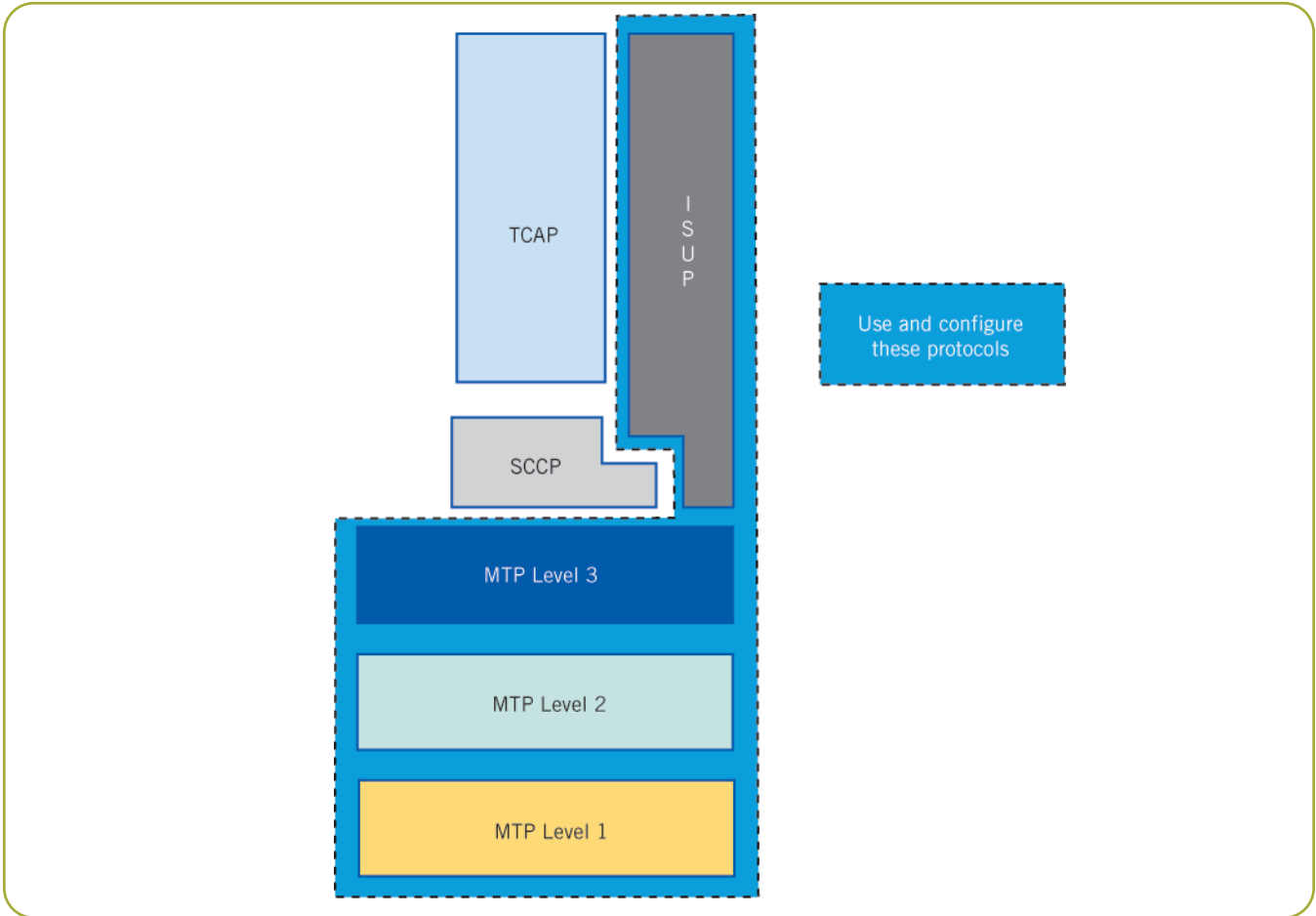


Figure 4. SS7 Protocol Stack

system.txt

The `system.txt` file specifies how the SigDiA software environment is set up. This software environment is commonly referred to as the GCTLOAD environment.

You can think of the operation of this SigDiA software as if each protocol layer is an independent module. Each protocol module can run either on the host CPU or on the Dialogic® SS7 signaling board, except MTP1 and MTP2, which must always run on the SS7 board. Protocol modules use inter-process communication mechanisms to exchange messages with each other.

GCTLOAD modules are identified with a module ID. The module ID uniquely identifies protocol modules in the GCTLOAD environment. Any piece of software (for example, the user application) that directly communicates with the SS7 protocol modules is also considered a GCTLOAD module and must also be assigned its own module ID. You can learn more about Dialogic SS7 GCTLOAD software environment in the *SS7 Protocols*

Software Environment Programmer's Manual (<http://www.dialogic.com/support/helpweb/signaling/software3.htm>).

Two working `system.txt` files (one for Linux, one for Windows®) are included in Appendices A and D, respectively, for this cross configuration. Manually copy them into the directory where you have installed the Dialogic® SS7 Development Package.

A review of the three major sections of the `system.txt` file follows:

LOCAL

The **LOCAL** commands assign module IDs for the modules that will run on the host CPU. The host machine is the computer where the application is running. Defining modules in this section also creates queues for incoming messages.

Our `system.txt` will include the following **LOCAL** commands:

```
LOCAL 0x20 * Module ID for SSD
LOCAL 0xcf * Module ID for s7_mgt
LOCAL 0xef * Module ID for s7_log
LOCAL 0x4d * Module ID for the GC/SS7 service/daemon
```

- **SSD (System Seven Dispatcher)** — Special host module that is responsible for relaying messages from the host to the board
- **S7_mgt** — Configuration utility that reads the `config.txt` file and configures the board and protocol modules (see `config.txt`; and Appendix B for Linux, Appendix E for Windows)
- **S7_log** — General-purpose logger; use to analyze system run-time behavior and troubleshoot problems
- **GC/SS7 service daemon** — Server side of the Dialogic® Global Call Software for SS7 client/server architecture, responsible for receiving and sending SS7 messages from and to the ISUP protocol module

REDIRECT

The **REDIRECT** section redirects messages sent to one module ID to an alternative module. For example, if you **REDIRECT** messages sent to ISUP (0x23) to SSD (0x20), these messages will be sent to the board via the SSD process. This would be used where protocol modules are running on the board.

Since we are running MTP2, MTP3, and ISUP on the signaling board, we **REDIRECT** the messages for these layers.

```
REDIRECT 0x71 0x20 * MTP2 messages redirected to SS7 card
REDIRECT 0x22 0x20 * MTP2 messages redirected to SS7 card
REDIRECT 0x23 0x20 * ISUP messages redirected to SS7 card
```

FORK_PROCESS

The **FORK_PROCESS** section actually executes programs, such as protocol modules running on the host or management tools (for example, `s7_mgt`). In the configuration described here, no protocols are running on the host, so only management tools will be forked (for example, `s7_log`, `s7_mgt`).

config.txt

The `config.txt` file is used for protocol configuration. It contains configuration commands for the MTP and ISUP protocols. `s7_mgt` reads the contents of this file and translates the configuration commands into management messages that are sent to the different protocol modules.

Two working `config.txt` files (one for Linux, one for Windows®) are included in Appendices B and E for this cross configuration. Manually copy them into the directory where you have installed the Dialogic SS7 Development Package.

For a description of the protocol configuration commands for the SPCI4 and SPCI2S products, refer to Appendix A of the *SS7 Programmer's Manual for SPCI2S, SPCI4 and CPM8* (see the *For More Information* section).

Physical Configuration

The SEPTELPCI_BOARD command defines the settings of the board itself. It allows defining the clock source for the board, and identifies what protocol modules are running on the board.

```
*
SEPTELPCI_BOARD    <board_id> <flags>    <code_file>    <run_mode>
                   0          0x0043    ss7.dc3       ISUP
```

<flags> - Possible values

CTbus Slave	0x00C2
CTbus Master – clock derived from one of the SS7 card line interfaces	0x0043
CTbus Master – clock derived from SS7 card internal oscillator	0x0042

In this case, set <flags> on System A to 0x0042 and on System B to 0x0043, so that System B synchronizes on the clock generated by System A.

The LIU_CONFIG command configures the operating parameters for an E-1/T-1 trunk interface. These parameters include the line encoding, the frame format, and whether Cyclical Redundancy Check (CRC) is enabled or not.

```
*
LIU_CONFIG         <board_id> <liu_id> <liu_type> <line_code> <frame_format> <crc_mode>
                   0          0          5          1          1          1
```

MTP Configuration

The three types of point codes to be configured are as follows:

- **Local Point Code (LPC)** — The SS7 address of the signaling point to be configured. Assign Point Code 1 to System A and Point Code 2 to System B.
- **Adjacent Point Code (APC)** — The SS7 address of the adjacent signaling point; this is the node to which the system has physical connection(s). For System A, the APC is Point Code 2. For System B, it is Point Code 1.
- **Destination Point Code (DPC)** — The SS7 address of the remote signaling point; this is the node the system has to exchange SS7 messages with. In this case (back-to-back configuration), the APC is also the DPC.

MTP_CONFIG is the main configuration command for the MTP module. The first two parameters are not used and can be left to 0. The <options> field allows configuring the variant being used (ITU or ANSI).

- For ITU operation, bits can be set to 0.
- For ANSI operation, bits 8, 9, 10, and 11 must be set to 1, leading to a value of 0x00000f00.

```
*
MTP_CONFIG         <reserved1> <reserved2> <options>
                   0          0          0x00000000
```

The MTP_LINKSET command is used for the configuration of a link set to an adjacent signaling point.

```
*
MTP_LINKSET        <linkset_id> <adjacent_spc> <num_links> <flags> <local_spc> <ssf>
                   0          2          1          0x0000    1          0x8
```

The <ssf> parameter is a 4-bit value made up (DCBA) with:

D C	Network type
0 0	International network
0 1	International spare
1 0	National network
1 1	National spare

B A	Priority value
0 0	0
0 1	1
1 0	2
1 1	3

In ITU, the most common SSF value for MTP3 messages is 0x8. In ANSI operation, the most common SSF value for MTP3 is 0xb.

The MTP_LINK command is used to define the links within every link set.

```
*
MTP_LINK      <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink> <stream> <timeslot> <flags>
              0         0         0         0     0         0         0         16       0x0006
```

Notes:

<blink> refers to the signaling processor resource that will handle that particular SS7 link.

<stream> can refer to the following :

- When the signaling link is on an E-1/T-1 trunk that is physically connected to the Dialogic SS7 board, <stream> is the actual E-1/T-1 port ID (ranging from 0 to 3 for an SPCI4 board) that is used for that particular trunk. In this situation, the <timeslot> refers to the actual time slot position on that trunk (for example, time slot 16).
- When the signaling link is on an E-1/T-1 trunk that is physically connected on another board in the chassis (such as a Dialogic® DM/N Digital Telephony Interface Board or Dialogic® DM3 Media Boards [DM/V boards]), <stream> must be set to 0x83. In this case, the <timeslot> refers to an internal bus slot in the SS7 board. Normally, you would start with time slot 1 for the first link and increment this value for each additional link coming from a separate board.

The MTP_ROUTE command defines the DPC for the system, and specifies which link set is used to reach it. Up to two link sets can be associated to a single DPC.

```
*
MTP_ROUTE     <dpc> <norm_ls> <user_part_mask> <flags> <second_ls>
              567   0         0x0020         0x0000 0
```

Note: The <user_part_mask> is a 16-bit value with bit n (n=0...15) set to allow the route to be used for messages with Service Indicator n. The service indicators are defined as follows:

User Part	Service Indicator	<user_part_mask>
SCCP	3	0x0008
TUP	4	0x0010
ISUP	5	0x0020
BICC	13	0x2000

ISUP Configuration

The ISUP_CONFIG command defines the ISUP characteristics: point codes, CIC codes, etc. The point codes in this section must match those in the MTP Configuration section.

Define the voice circuits, specify how many, and where they are supposed to go.

```
*
ISUP_CONFIG   <res1> <res2> <user_id> <options> <num_grps> <num_ccts>
              0     0     0x4d    0x0475    2         64
```

Note: For ANSI operation, the <options> field of ISUP_CONFIG, which is a bit field, must have bits 8 and 9 set in order to enable ANSI-compliant operations. These bits are 0 by default.

The ISUP_CFG_CCTGRP command defines the configuration parameters for a group of ISUP circuits. Usually a circuit group is made of the voice circuits on a single E-1 or T-1 trunk.

Note: For the <cic_mask> parameter, a circuit group may contain gaps in the CIC allocation. This is usually the case in Europe, where SS7 service providers usually skip every multiple of 16. This convention is used in our system set-up.

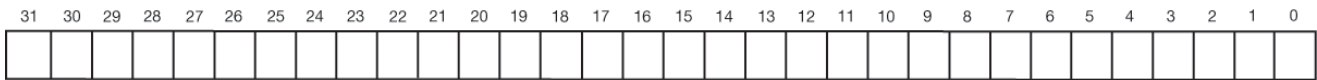
Group ID	Total Circuits	CIC Ranges	Remarks
0	30	1-15, 17-31	Time slot 16 is used to carry SS7 link
1	30	33-47, 49-63	Time slot 16 is not used

Note that multiples of 16 (0, 16, 32, 48) have been skipped in the CIC allocation scheme.

The `<cic_mask>` parameter allows for gaps in the CIC range of a particular circuit group.

To build the `cic_mask`, follow these procedures:

1. Take an empty array of 32 bits.



Bit 0 corresponds to the `<base_cic>` and to the `<base_cid>`. Bit `n` corresponds to `<base_cic+n>`, as well as to `<base_cid+n>`.

2. For each CIC belonging to the circuit group, turn the corresponding bit to 1 in the `<cic_mask>`.



Which gives the following mask: `0x7fff7fff`.

By setting the `<base_cic>` to 1 and by turning bit 15 off, CIC 16 is excluded (`<base_cic> + 15 = 16`) from the first circuit group.

By setting the `<base_cic>` to 33 and by clearing bit 15 in the `<cic_mask>`, CIC 48 is excluded from (`<base_cic> + 15 = 48`) from the second circuit group.

Step 4 – Proving the SS7 Configuration

Note: For Linux users: log in as a “super user” (root) to properly use Dialogic® SS7 OA&M commands.

1. Once the `config.txt` and `system.txt` have been properly copied in the Dialogic SS7 directory (for example, `/usr/septel` on Linux and `c:\Septel` on Windows), open a command shell. Go to this directory, and type:

```
./gctload -d (Linux) or gctload -d (Windows)
```

If a crossover cable is present and has been plugged into the correct trunk interface, the following messages display in the `gctload` window:

```
S7L:I0000 LIU Status : id=0 PCM OK
S7L:I0000 LIU Status : id=0 IN SYNC
S7L:I0000 LIU Status : id=0 AIS CLEARED
```

2. If you are not getting the above mentioned LIU status indications, verify your LIU settings (PCM encoding, CRC generation, etc.) again and fix these problems first.

Note: If you are not getting those LIU Status indications, it means that something is wrong in the physical configuration (maybe a wrong cable or a cable is connected to the wrong port).

3. If the physical connection is good, then you can try to bring the MTP link into service. To do so, open another command window, go the SS7 directory, and type:

```
./mtpsl act 0 0 (Linux) or mtpsl act 0 0 (Windows)
```

You should then get the following status indications in the `gctload` window:

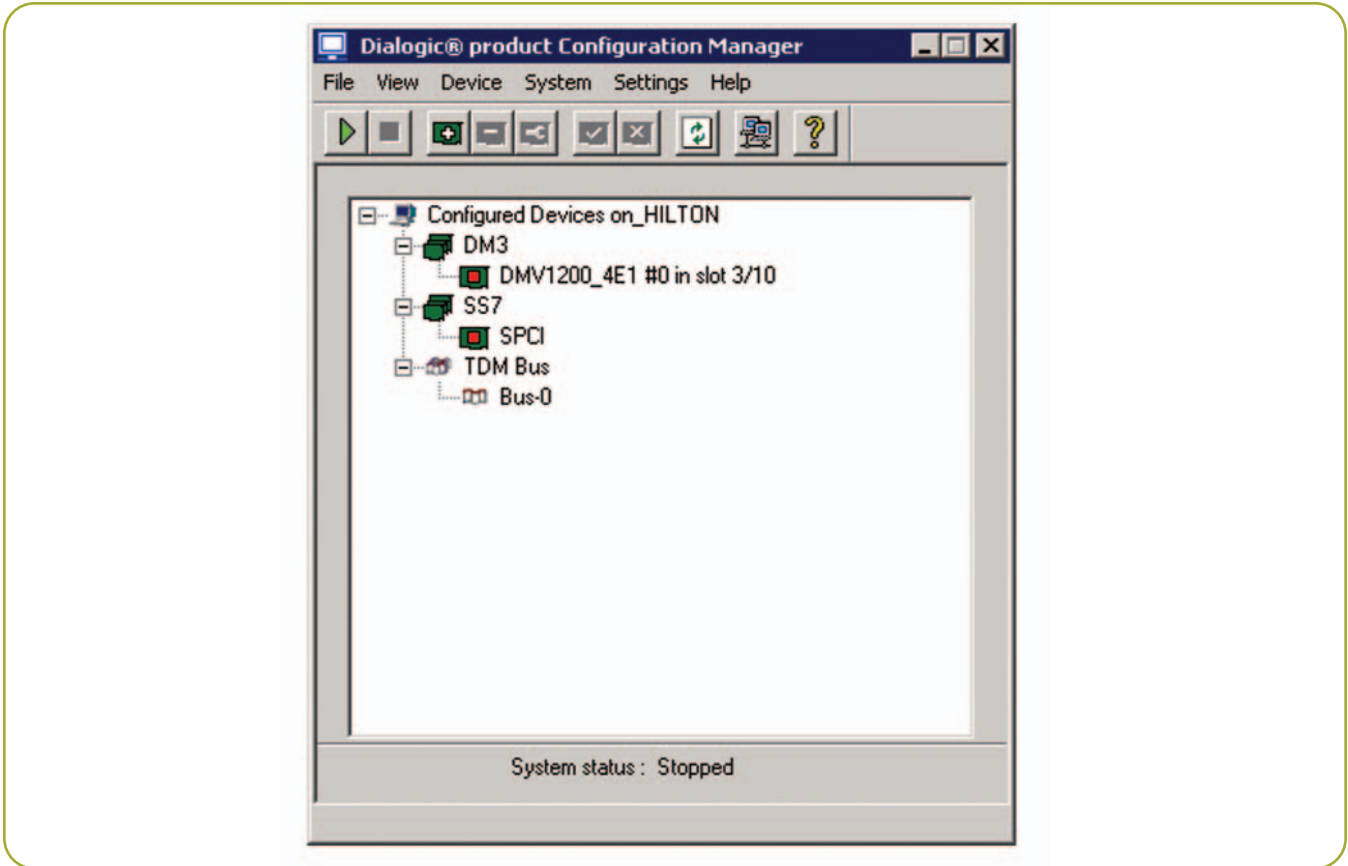


Figure 5. DCM Showing Dialogic® SS7 Board after gcss7.cfg Modifications

Windows® Operating System

```
S7L:I0000 Level 2 State : id=0 INITIAL ALIGNMENT
S7L:I0000 Level 2 State : id=0 ALIGNED READY
S7L:I0000 Level 2 State : id=0 IN SERVICE
S7L:I0000 MTP Event : linkset_id/link_ref=0000 Changeback
S7L:I0000 MTP Event : linkset_id=00 Link set recovered
S7L:I0000 MTP Event : linkset_id=00 Adjacent SP accessible
S7L:I0000 MTP Event : point_code=00000001
Destination available
```

Linux Operating System

You should get similar messages, with the only difference that the remote point code is 00000002.

Step 5 – Configuring the Dialogic® Global Call API for SS7 Software (gcss7.cfg)

The Dialogic® System Release contains a file called `gcss7.cfg`, which is used to configure the Dialogic® Global Call API for SS7 software.

- On Linux operating systems, this file is located in `/usr/diallogic/cfg`
- On Windows® operating systems, this file is located in `$DIALOGICDIR\Dialogic\cfg` (for example, `c:\Program Files\Dialogic\cfg`)

Working `gcss7.cfg` files are given in Appendices C and F. Refer to Section 3.3 of the *Global Call SS7 Technology Guide* for a description of the `gcss7.cfg` configuration commands and parameters.

1. Once the `gcss7.cfg` has been properly updated, Windows users may close the Dialogic® Configuration Manager (DCM) and re-open it. Then, the SPCI board should appear in the device tree, as shown in Figure 5.

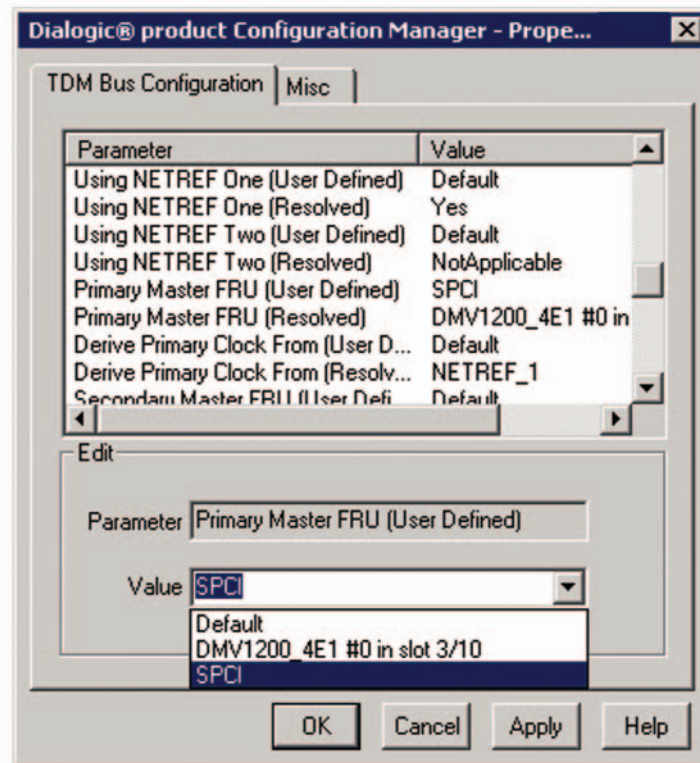


Figure 6. Configuring the SPCI Board as the TDM Bus Primary Master

2. Double-click on the 'Bus-0' device, in order to configure the SPCI board as the CT Bus Primary Master. This is illustrated in Figure 6.

Step 6 – Starting the Software

Once everything has been configured, starting the software requires two steps:

1. Start-up the Dialogic SS7 environment, using `gctlload (-d)`. Keep the `gctlload` program running at all times.
2. Start-up the Dialogic® service:
 - Linux: Invoke the `dlstart` script in `/usr/dialogic/bin`.
 - Windows®: Click on the green triangle in the DCM.

Since the Global Call API for SS7 software has been enabled to automatically activate the SS7 links, there is no need to manually do this with `mtpsl`. If everything goes well, the Dialogic service will start without any errors and the SS7 links will be activated, resulting in the following indication messages to be printed in the `gctlload` window:

```

S7L:I0000 Level 2 State : id=0 INITIAL ALIGNMENT
S7L:I0000 Level 2 State : id=0 ALIGNED READY
S7L:I0000 Level 2 State : id=0 IN SERVICE
S7L:I0000 MTP Event : linkset_id/link_ref=0000 Changeback
S7L:I0000 MTP Event : linkset_id=00 Link set recovered
S7L:I0000 MTP Event : linkset_id=00 Adjacent SP accessible
S7L:I0000 MTP Event : point code=00000001
Destination available
  
```


Step 7 – Running the gc_basic_call_model Demo

1. Edit the `gc_basic_call_model.cfg` file, uncomment the SS7 device, and rename it to `dkBIT1`.
 - Linux operating system: Set the Direction to “Out”.
 - Windows® operating system: Set the Direction to “In”.
2. Run the application binary on each system, and you should see `GCEV_UNBLOCKED` on each side and, after a short period of time, calls will start to be sent from the Linux system to the Windows system.
3. Use CTRL-C to quit the demo.

Troubleshooting

SS7-related problems can be diagnosed in different ways:

If your problem is related to Dialogic® SS7 hardware or software configuration, add the “debug mode” option to `ssds` and `s7_mgt` in your `system.txt` file:

```
FORK_PROCESS ssds.exe -d
FORK_PROCESS s7_mgt.exe -d
```

When starting up the `gctload` environment, use the `-d` option as well.

If your problem is related to the integration of Dialogic® SS7 products in the Dialogic® System Release Software, then enable the Global Call API for SS7 service traces (via `gcSS7.cfg`). This will result in a binary file called `DlgcS7.log`. Use a decoder tool, called `ss7trace`, to convert this binary log file into an ASCII file. Refer to

Section 7.2 of the *Global Call SS7 Technology Guide* (see the *For More Information* section) for information on Global Call API for SS7 service traces.

Note that it is possible to collect both `s7_log` and Global Call API for SS7 service traces in a single file. To do this, you have to redirect the `s7_log` module (0xef) to the Global Call API for SS7 service module (0x4d) in your `system.txt` file:

```
REDIRECT 0xef 0x4d
```

If your problem is related to the Global Call API for SS7 application, you can also enable Global Call API for SS7 library traces in `gcSS7.cfg`. This will result in a binary file called `ss7.log`. You can use a decoder tool, called `ss7trace`, to convert this binary log file into an ASCII file. Refer to Section 7.1 of the *Global Call SS7 Technology Guide* for more information on Global Call API for SS7 library traces.

Summary

This application note has described the steps to set up a pair of telephony systems that combine Dialogic® Signaling Boards and Dialogic® DM3 Media Boards on platforms based on Linux and Windows® operating systems.

References

[Black] Uyles Black, *ISDN & SS7: Architectures for Digital Signaling Networks*, Prentice Hall (ISBN 0-13-259193-6), 1997.

Appendix A: Linux system.txt File

```

*****
*
* Linux system.txt file
*
*****
*
* Essential modules running on host:
*
LOCAL          0x20          * ssds/ssdh - Board interface task
LOCAL          0x00          * tim_lnx - Timer task
*
* Optional modules running on the host:
*
LOCAL          0xcf          * s7_mgt - Management/config task
LOCAL          0xef          * s7_log - Display and logging utility
LOCAL          0x4d          * GC/SS7 daemon
*
* Essential modules running on the board (all redirected via ssd):
*
REDIRECT 0x71 0x20 * MTP2 module (except SS7HD boards)
* REDIRECT 0x81 0x20 * MTP2 module_id for SP 0 (SS7HD boards only)
* REDIRECT 0x91 0x20 * MTP2 module_id for SP 1 (SS7HD boards only)
* REDIRECT 0xe1 0x20 * MTP2 module_id for SP 2 (SS7HD boards only)
* REDIRECT 0xf1 0x20 * MTP2 module_id for SP 3 (SS7HD boards only)
REDIRECT 0x10 0x20 * CT bus/Clocking control module
REDIRECT 0x8e 0x20 * On-board management module
*
* Modules that optionally run on the board (all redirected via ssd):
*
REDIRECT 0x23 0x20 * ISUP module
REDIRECT 0x22 0x20 * MTP3 module
*
* Redirection of status indications:
*
REDIRECT 0xdf 0xef * LIU/MTP2 status messages -> s7_log
*
* Now start-up all local tasks:
* (For SPCI4/SPCI2S/CPM8 start-up use ssds and
* for SS7HD boards use ssdh)
*
FORK_PROCESS ./ssds
* FORK_PROCESS ./ssdh
FORK_PROCESS ./tim_lnx
FORK_PROCESS ./tick_lnx
FORK_PROCESS ./s7_mgt
FORK_PROCESS ./s7_log -fss7.log
*
*****

```

Appendix B: Linux config.txt File

```

*****
*
* Linux config.txt
*
*****
*
* Configure individual boards:
* For SPCIA4 / SPCIA2S boards:
* SEPTTELPCCI_BOARD <board_id> <flags> <code_file> <run_mode>
SEPTTELPCCI_BOARD 0 0x0042 ss7.dc3 ISUP
*
*
* Configure individual E1/T1 interfaces:
* LIU_CONFIG <board_id> <liu_id> <liu_type> <line_code> <frame_format>
* <crc_mode>
LIU_CONFIG 0 0 5 1 1 1
*
*
* MTP parameters:
*
* MTP_CONFIG <reserved> <reserved> <options>
MTP_CONFIG 0 0 0x00000000
*
* Define linksets:
* MTP_LINKSET <linkset_id> <adjacent_spc> <num_links> <flags> <local_spc> <ssf>
MTP_LINKSET 0 2 2 0x0000 1 0x08
*
* Define signaling links:
* MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink>
* <stream> <timeslot> <flags>
* For SPCIA4 / SPCIA2S / CPM8 boards:
MTP_LINK 0 0 0 0 0 0 0 16 0x0006
*
* Define a route for each remote signaling point:
* MTP_ROUTE <dpc> <linkset_id> <user_part_mask>
MTP_ROUTE 2 0 0x0020
*
*
* ISUP parameters:
*
* Configure ISUP module:
* ISUP_CONFIG <reserved> <reserved> <user_id> <options> <num_grps> <num_ctts>
ISUP_CONFIG 0 0 0x4d 0x0475 2 64
*
* Configure ISUP circuit groups:
* ISUP_CFG_CCTGRP <gid> <dpc> <base_cic> <base_cid> <cic_mask> <options>
* <user_inst> <user_id> <opc> <ssf> <variant> <options2>
ISUP_CFG_CCTGRP 0 2 0x01 0x01 0x7fff7fff 0x001c 0 0x4d 1 0x8 0 0x00
ISUP_CFG_CCTGRP 1 2 0x21 0x21 0x7fff7fff 0x001c 0 0x4d 1 0x8 0 0x00
*
*****

```

Appendix C: Linux gcss7.cfg File

```

#
# Dialogic GlobalCall SS7 Configuration File (gcss7.cfg)
#
#   for Linux
#
#####
# Type of System Configuration #
#####

# Leave commented out or set to "None" when not using Dialogic SS7.
# Depending on the value of this parameter, the sections below, that
# are specific to some configurations (SeptelCard, SIU, SIU.Dual) will be
# used or not.
# Format: String - ["None", "Card", "SIU", "DualSIU"]
System.Configuration = "Card"

#####
# Parameters for the GlobalCall SS7 Call Control Library #
#####

# If defined, this parameter will cause the library logging to be
# activated at the first gc_Open() of an SS7 circuit and the trace
# file will have the specified name.
# Format: String
Library.LogFile = "ss7.log"

# Logging Level for the library
# Format: String - ["None", "Errors", "All"]
# Default: "Errors" (and Warnings)
Library.LogLevels = "All"

# Maximum size of the library log in kilobytes
# Format: Integer, Default: 200
#Library.LogMaxSize = 200

#####
# Parameters for the Dialogic SS7 service/daemon #
#####

# Logging Level for the service (Dlgs7.log)
# Format: String - ["None", "Errors", "All"]
# Default: "Errors" (and Warnings)
Service.LogLevels = "All"

# Maximum size of the service log in kilobytes
# Format: Integer, Default: 200
#Service.LogMaxSize = 200

# Does the service need to start GCTLOAD automatically?
# Format: String - ["Yes", "No"]
Service.GCTLOAD_Control = "No"

# Path to GCTLOAD (Used only if GCTLOAD Control is set to 'Yes')
# For Setpel Cards, the parameter defaults to the same path as ConfigDir
# Format: String
#Service.GCTLOAD_Path = "/usr/septel"

# GCT-environment module id used by the service
# Format: Integer, Default: 0x4d
#Service.ModuleID = 0x4d

```

```
# Maximum timeout (in seconds) for server-application keep-alive mechanism
# Format: Integer; Default: 7; 0 means the mechanism is off
#Service.WatchDogMaxTime = 7

# Time (in ms) during which to accumulate Circuit Group Supervision Requests
# (reset, block, unblock) for a circuit group.
# Format: Integer - Default: 500
#Service.GroupCommandTimer = 500

#####
# Configuration for Septel Card Systems #
#####

# Path to the config.txt file
# Format: String
SeptelCard.ConfigDir = "/usr/septel"

# Should MTP links be automatically activated ?
# Format: String - ["None", "All"]
SeptelCard.Auto_Links_Activation = "All"

#####
# Configuration for SIU Systems #
#####

# ID of this host - Use 0 if only one host accessing the SIU(s)
# Format: Integer
SIU.HostID = 0

# SIU A - IP Address
# Format: String
#SIU.A.IP_Address = "111.112.113.114"

# SIU A - Account to use to connect to SIU when using FTP
# Format: String
#SIU.A.FTP_Account = "ftp"

# SIU A - Password for the FTP account
# Format: String
#SIU.A.FTP_Password = "ftp"

# SIU A - Directory to which to change (in FTP session) in order to get
config.txt
# Format: String
#SIU.A.RemoteConfigDir = "."

# Maximum time (in seconds) to wait at startup for an SIU to come on-line before
# considering it as being down.
# Format: Integer, Default: 10
#SIU.InitTimeout = 10

#####
# Parameters specific to Dual-Resilient SIU Configurations #
#####

# SIU B Parameters - See the same parameters for SIU.A
#SIU.B.IP_Address = "111.112.113.114"

#SIU.B.FTP_Account = "ftp"

#SIU.B.FTP_Password = "ftp"

#SIU.B.RemoteConfigDir = "."
```

```
# Max time (in seconds) to wait for group (de)activation command
# responses from SIU.
# Format: Integer, Default: 5
#SIU.Dual.SiuCommandTimeout = 5

# Debounce time (in seconds) for SIU Down indications
# Format: Integer, Default: 8
#SIU.Dual.SiuUpDebounceTime = 8

# Maximum number of retries for SIU group (de)activation commands
# Format: Integer, Default: 5
#SIU.Dual.MaxCmdRetries = 5

#####
# Parameters that are related to config.txt #
#####

# MTP Link source, link ID must match the value in config.txt.
# MtpLink <link_id> <"link_source">

# Circuit Group configuration, Group ID must match the value in config.txt.
# CGrp <gid> <"trunk_name"> [<base_TS> [<"Pref_SIU">]]
CGrp 0 "dkB1"
CGrp 1 "dtiB1"
#
# End of gcass7.cfg
#
```

Appendix D: Windows® system.txt File

```

*****
*
* Windows system.txt
*
*****
*
* Essential modules running on host:
*
LOCAL          0x20          * ssd/ssds/ssdh - Board interface task
LOCAL          0x00          * tim_nt - Timer task
*
* Optional modules running on the host:
*
LOCAL          0xcf          * s7_mgt - Management/config task
LOCAL          0xef          * s7_log - Display and logging utility
LOCAL          0x4d          * GC/SS7 service
*
* Essential modules running on the board (all redirected via ssd):
*
REDIRECT 0x71 0x20 * MTP2 module (except SS7HD boards)
* REDIRECT 0x81 0x20 * MTP2 module_id for SP 0 (SS7HD boards only)
* REDIRECT 0x91 0x20 * MTP2 module_id for SP 1 (SS7HD boards only)
* REDIRECT 0xe1 0x20 * MTP2 module_id for SP 2 (SS7HD boards only)
* REDIRECT 0xf1 0x20 * MTP2 module_id for SP 3 (SS7HD boards only)
REDIRECT 0x10 0x20 * CT bus/Clocking control module
REDIRECT 0x8e 0x20 * On-board management module
*
* Modules that optionally run on the board (all redirected via ssd):
*
REDIRECT 0x23 0x20 * ISUP module
REDIRECT 0x22 0x20 * MTP3 module
*
* Redirection of status indications:
*
REDIRECT 0xdf 0xef * LIU/MTP2 status messages -> s7_log
*
* Now start-up all local tasks:
* (For SPCI4/SPCI2S/CPM8 start-up use ssds.exe and
* for SS7HD boards use ssdh.exe)
*
FORK_PROCESS ssds.exe
* FORK_PROCESS ssdh.exe
FORK_PROCESS tim_nt.exe
FORK_PROCESS tick_nt.exe
FORK_PROCESS s7_mgt.exe
FORK_PROCESS s7_log.exe -fss7.log
*
*****

```

Appendix E: Windows® config.txt File

```

*****
*
* Windows config.txt
*
*****
*
* Configure individual boards:
* For SPCI4 / SPCI2S boards:
* SEPTELPCI_BOARD <board_id> <flags> <code_file> <run_mode>
SEPTELPCI_BOARD 0 0x0043 ss7.dc3 ISUP
*
*
* Configure individual E1/T1 interfaces:
* LIU_CONFIG <board_id> <liu_id> <liu_type> <line_code> <frame_format>
* <crc_mode>
LIU_CONFIG 0 0 5 1 1 1
*
*
* MTP parameters:
*
* MTP_CONFIG <reserved> <reserved> <options>
MTP_CONFIG 0 0 0x00000000
*
* Define linksets:
* MTP_LINKSET <linkset_id> <adjacent_spc> <num_links> <flags> <local_spc> <ssf>
MTP_LINKSET 0 1 2 0x0000 2 0x08
*
* Define signaling links:
* MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink>
* <stream> <timeslot> <flags>
* For SPCI4 / SPCI2S / CPM8 boards:
MTP_LINK 0 0 0 0 0 0 0 16 0x0006
*
* Define a route for each remote signaling point:
* MTP_ROUTE <dpc> <linkset_id> <user_part_mask>
MTP_ROUTE 1 0 0x0020
*
* Define any user provided Layer 4 protocol:
* MTP_USER_PART <service_ind> <module_id>
MTP_USER_PART 0x0a 0x2d
*
*
* ISUP parameters:
*
* Configure ISUP module:
* ISUP_CONFIG <reserved> <reserved> <user_id> <options> <num_grps> <num_ctcs>
ISUP_CONFIG 0 0 0x4d 0x0475 2 64
*
* Configure ISUP circuit groups:
* ISUP_CFG_CCTGRP <gid> <dpc> <base_cic> <base_cid> <cic_mask> <options>
* <user_inst> <user_id> <opc> <ssf> <variant> <options2>
ISUP_CFG_CCTGRP 0 1 0x01 0x01 0x7fff7fff 0x001c 0 0x4d 2 0x8 0 0x00
ISUP_CFG_CCTGRP 1 1 0x21 0x21 0x7fff7fff 0x001c 0 0x4d 2 0x8 0 0x00
*
*****

```


Appendix F: Windows® gcss7.cfg File

```

#
# Dialogic GlobalCall SS7 Configuration File (gcss7.cfg)
#
#   for Windows
#

#####
# Type of System Configuration #
#####

# Leave commented out or set to "None" when not using Dialogic SS7.
# Depending on the value of this parameter, the sections below, that
# are specific to some configurations (SeptelCard, SIU, SIU.Dual) will be
# used or not.
# Format: String - ["None", "Card", "SIU", "DualSIU"]
System.Configuration = "Card"

#####
# Parameters for the GlobalCall SS7 Call Control Library #
#####

# If defined, this parameter will cause the library logging to be
# activated at the first gc_Open() of an SS7 circuit and the trace
# file will have the specified name.
# Format: String
Library.LogFile = "ss7.log"

# Logging Level for the library
# Format: String - ["None", "Errors", "All"]
# Default: "Errors" (and Warnings)
Library.LogLevels = "All"

# Maximum size of the library log in kilobytes
# Format: Integer, Default: 200
#Library.LogMaxSize = 200

#####
# Parameters for the Dialogic SS7 service/daemon #
#####

# Logging Level for the service (Dlgs7.log)
# Format: String - ["None", "Errors", "All"]
# Default: "Errors" (and Warnings)
Service.LogLevels = "All"

# Maximum size of the service log in kilobytes
# Format: Integer, Default: 200
#Service.LogMaxSize = 200

# Does the service need to start GCTLOAD automatically?
# Format: String - ["Yes", "No"]
Service.GCTLOAD_Control = "No"

# Path to GCTLOAD (Used only if GCTLOAD_Control is set to "Yes")
# For Setpel Cards, the parameter defaults to the same path as ConfigDir
# Format: String
Service.GCTLOAD_Path = "c:\Septel"

# GCT-environment module id used by the service
# Format: Integer, Default: 0x4d
Service.ModuleID = 0x4d

# Maximum timeout (in seconds) for server-application keep-alive mechanism
# Format: Integer; Default: 7; 0 means the mechanism is off

```

```

Service.WatchDogMaxTime = 0
#####
# Configuration for Septel Card Systems #
#####

# Path to the config.txt file
# Format: String
SeptelCard.ConfigDir = "c:\Septel"

# Should MTP links be activated automatically?
# Format: String - ["None", "All"]
SeptelCard.Auto_Links_Activation = "All"

#####
# Configuration for SIU Systems #
#####

# ID of this host - Use 0 if only one host accessing the SIU(s)
# Format: Integer
#SIU.HostID = 0

# SIU A - IP Address
# Format: String
#SIU.A.IP_Address = "172.22.197.103"

# SIU A - Account to use to connect to SIU when using FTP
# Format: String
#SIU.A.FTP_Account = "ftp"

# SIU A - Password for the FTP account
# Format: String
#SIU.A.FTP_Password = ""

# SIU A - Directory to which to change (in FTP session) in order to get
# config.txt
# Format: String
#SIU.A.RemoteConfigDir = "."

#####
# Parameters specific to Dual-Resilient SIU Configurations #
#####

# SIU B Parameters - See the same parameters for SIU.A
#SIU.B.IP_Address = "111.112.113.114"

#SIU.B.FTP_Account = "ftp"

#SIU.B.FTP_Password = "ftp"

#SIU.B.RemoteConfigDir = "."

#####
# Parameters that are related to config.txt #
#####

# MTP Link source, link ID must match the value in config.txt.
# MtpLink <link_id> <"link_source">

# Circuit Group configuration, Group ID must match the value in config.txt.
# CGrp <gid> <"trunk_name"> [<base_TS> [<"Pref_SIU">]]
CGrp 0 "dkB1"
CGrp 1 "dtiB1"

#
# End of gcss7.cfg
#

```

Acronyms

APC	Adjacent Point code
BICC	Bearer Independent Call Control
CIC	Circuit Identification Code
CRC	Cyclical Redundancy Check
DPC	Destination Point Code
ISUP	ISDN User Part
LPC	Local Point Code
MTP	Message Transfer Part
SigDiA	Dialogic® Signaling Distributed Architecture

For More Information

Dialogic® Signaling Distributed Architecture (SigDiA) software can be downloaded at <http://www.dialogic.com/support/helpweb/signaling/default.htm>

SS7 Boards Programmer's Manual for SPCI2S, SPCI4 and CPM8 – Issue 2

<http://www.dialogic.com/support/helpweb/signaling/software2.htm>

SS7 Protocols Software Environment Programmer's Manual – Issue 5

<http://www.dialogic.com/support/helpweb/signaling/software3.htm>

Global Call SS7 Technology Guide – August 2005

http://www.dialogic.com/manuals/docs/globalcall_for_ss7_v4.pdf

Dialogic® System Release 6.0 PCI for Windows Software Installation Guide —

http://www.dialogic.com/manuals/sr60winpci/release_install.pdf

Dialogic® System Release 6.1 for Linux Software Installation Guide —

http://www.dialogic.com/manuals/sr61lin/release_install.pdf

Dialogic® System Releases and Service Updates —

<http://resource.dialogic.com/telecom/support/releases/serviceUpdates/index.htm>

To learn more, visit our site on the World Wide Web at <http://www.dialogic.com>.

Dialogic Corporation
9800 Cavendish Blvd., 5th floor
Montreal, Quebec
CANADA H4M 2V9

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH PRODUCTS OF DIALOGIC CORPORATION OR ITS SUBSIDIARIES ("DIALOGIC"). 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.

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

Dialogic is a registered trademark of Dialogic Corporation. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. Other names of actual companies and products mentioned herein are the trademarks of their respective owners. Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement their concepts or applications, which licenses may vary from country to country.

Copyright © 2007 Dialogic Corporation All rights reserved.

09/07 9443-02