



**Dialogic[®] SS7 Protocols
TCAP Test Utility (TTU) User Guide**

Document Reference U27SSS

Copyright © 2002-2007 Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation.

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation. Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic Corporation does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC[®] PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic Corporation or its subsidiaries may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic Corporation does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic Corporation or its subsidiaries and no such licenses are provided except pursuant to a signed agreement with Dialogic Corporation. More detailed information about such intellectual property is available from Dialogic Corporation's legal department at 9800 Cavendish Blvd., Montreal, Quebec, Canada H4M 2V9. **Dialogic Corporation encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Dialogic 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 and Visual C++ are registered trademarks of Microsoft Corporation in the United States and/or other countries. Other names of actual companies and products mentioned herein are the trademarks of their respective owners.

Publication Date: October 2007

Document Number: U27SSS, Issue 3

Revision History

Issue	Date	Description
1	22-Mar-02	Initial Issue
2	04-Aug-03	Naming changed to reference SPCI4/SPCI2S and CPM8.
3	01-Oct-07	Remove reference to PCCS6 and CPM8, include non-circuit protocol configuration commands in config.txt. Add SS7HD and SS7G21 SIU configuration and add support for SIGTRAN M2PA links

Contents

1	Introduction.....	5
1.1	Software requirements.....	5
2	TTU Application.....	6
2.1	TC-USER.....	6
2.2	Message Sequence Chart.....	7
2.3	Customizing the TTU Application.....	8
2.4	TTU source code.....	8
3	Building the TTU application.....	9
3.1	Host software directory structure.....	10
3.2	Building TTU.....	10
4	Configuration.....	12
4.1	System Configuration.....	12
4.1.1	SS7G2x SIU.....	12
4.1.2	SPCI2S, SPCI4 and SS7HD.....	12
4.1.3	SIGTRAN M2PA.....	12
4.2	Protocol Configuration.....	13
4.2.1	SCCP.....	13
4.2.2	TCAP.....	13
5	Running the TTU application.....	14
5.1	TTU Command Line Arguments.....	14
5.1.1	TTU Options.....	14
6	References.....	15
7	Abbreviations.....	16
Appendix A - Example configuration files with TDM MTP2 links.....		17
Appendix B - Example configuration with SIGTRAN M2PA links.....		27
Appendix C - TC-USER example script file.....		34
Appendix D - Sample Output.....		35

1 Introduction

The Dialogic[®] TTU application is usable with the Dialogic[®] TCAP module. This application, in conjunction with the Service Switching Function (SSF), will simulate the generation and acknowledgement of simple TCAP messages between the two nodes. This user guide will describe the design, build and usage of this application in order for developers.

This user guide is intended for users who choose to develop their own applications that will interface with and use the functionality provided by the TCAP module.

1.1 Software requirements

The TTU application requires the following software:

1. Dialogic[®] SS7 Development Package
2. Dialogic[®] User Part Development Package
3. For TDM-based configurations:
 - ss7.dc3 or ss7.dc4 codefile
 - Dialogic[®] MTP3, SCCP and TCAP host binaries, as required
4. For SIGTRAN-based configurations:
 - Dialogic[®] M2PA, MTP3, SCCP and TCAP host binaries, as required

Software can be downloaded from

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

2 TTU Application

With the system shown below, an outgoing dialogue is started by the TC-USER and Dialogic® TTU responds to the incoming dialogue. When TTU receives the BEGIN, it responds with a TC-INVOKE and a TC-CONTINUE. When TTU receives the TC-INVOKE from the TC-USER, TTU responds with an outgoing TC-RESULT-L and TC_CONTINUE. Using command line arguments, you may set the module id and sub-system number of TTU and also a number of options affecting the behavior of TTU.

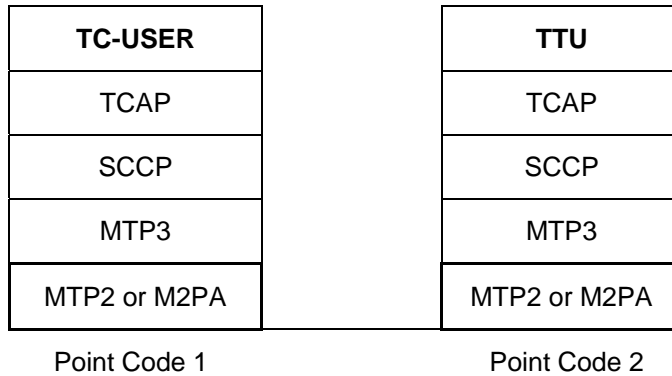


Figure 1. SS7 Layers

2.1 TC-USER

For the purpose of this user guide, the TC-USER is simply a file whose contents are sent to TTU using s7_play.

2.2 Message Sequence Chart

Shown below is a message sequence chart showing the flow of a single dialogue between the TC-USER and TTU.

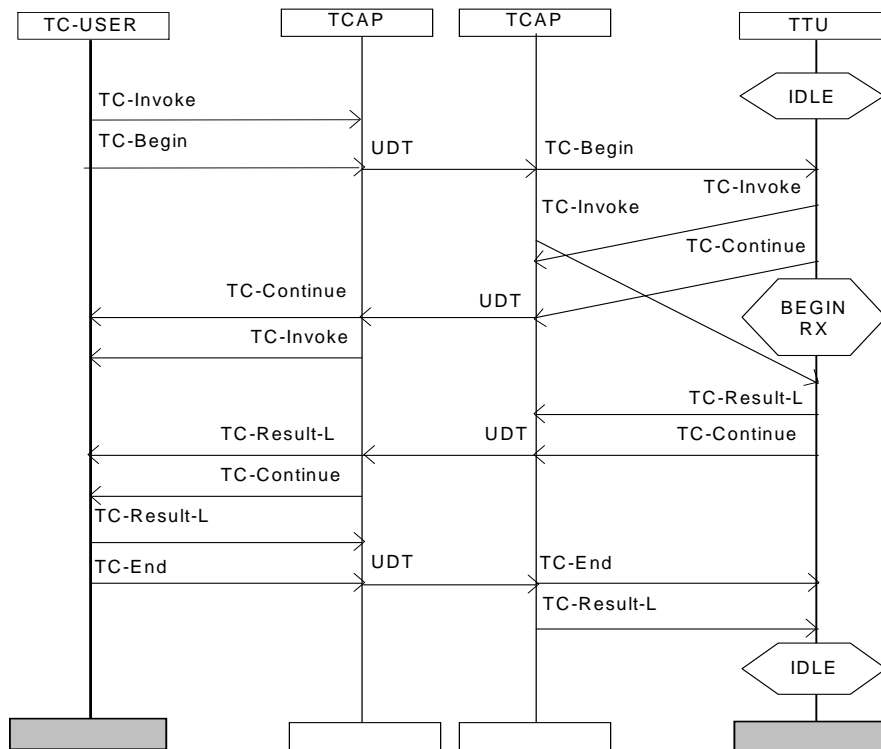


Figure 2. Single dialogue between TC-USER and TTU

2.3 Customizing the TTU Application

TTU is example code for development using the Dialogic® TCAP module and as such, a number of simplifications and limitations have been imposed for this purpose.

TTU will only allow up to 16 incoming dialogues at one time.

The developer should be aware of the limited nature of the example applications when making use of the source code for building their own applications.

2.4 TTU source code

The TTU program can be found in the *Dialogic® User Part Development Package*. The following table describes the files required by the TTU application:

File	Notes
ttu_main.c	This file contains the main() function. This reads the command line arguments and passes them to ttu_ent() .
ttu.c	TTU first of all will configure itself as a subsystem using TTU_user_in_service() . TTU then enters a loop monitoring for dialogue/component indications. TTU_dlg_ind() will receive the Begin and depending on the TTU options provided may immediately respond with a Invoke and Continue (TTU_send_invoke() & TTU_send_continue()). TTU_cpt_ind() receives an Invoke and responds with a Continue and Result-L (TTU_send_continue() & TTU_send_result_l()).
Tcap.c	Library functions for interfacing with the TCAP module
Tcap.h	Contains structures and function prototypes used in tcap.c

3 Building the TTU application

Example make-files for the following operating systems are provided and identified by a unique suffix:

Operating system	Suffix
Generic UNIX (Solaris, Linux)	.mak
Windows [®]	.mnt

A single definitions file (one for each operating system) which contains the definitions relating to the user's own development environment is supplied in the *Dialogic[®] User Part Development Package*. The definitions files are as follows and the appropriate file should be used depending on the operating system:

makdefs.mak	(Linux)
makdefs_sol.mak	(Solaris)
makdefs.mnt	(Windows [®])

For Windows[®], a dynamically linked GCT library that allows the application to link to the GCT functions is supplied in the *Dialogic[®] SS7 Development Package* as follows:

gctlib.dll	(Visual C++ [®] compiler)
------------	------------------------------------

For 'UNIX', a shared object to the GCT functions is supplied in the *Dialogic[®] SS7 Development Package*

e.g. libgctlib.so.1.0.0	(Linux & Solaris)
-------------------------	-------------------

The source code for the example program should be compiled and linked with the appropriate library for the operating system in use.

3.1 Host software directory structure

To build the TTU application, the user should first ensure that the required files are copied into the correct directories as follows:

1. Copy either the zip or tar file from the *Dialogic® User Part Development Package* to the *Dialogic® SS7 Development Package* directory and decompress using the appropriate tool. The choice of the zip or tar file is up to the user; both will create the UPD directory structure shown in the table below. The table below shows files required by the TTU program only.
2. The C header files in the INC directory shown in the table below. The C header files in the INC directory shown in the table below lists the header files required by the TTU program.

The following table lists the directory structure and files required to build the TTU programs supplied on the *Dialogic® User Part Development Package*.

Root directory			
Septel			
INC	UPD		
tcp_inc.h scp_inc.h system.h msg.h sysgct.h ss7_inc.h pack.h strtonum.h	BIN	SRC	
	BACKUP_WIN BACKUP_LNX BACKUP_SOL	TTU ttu.mnt ttu.mak ttu_sol.mak ttu_iss.txt ttu_def.h ttu.c intumain.c ttu_sys.c ttu_trc.c	makdefs.mnt makdefs_sol.mak makdefs.mak makeall.bat makeall makeall_sol

3.2 Building TTU

It is assumed that the UPD is extracted in the *Dialogic® SS7 Development Package* directory i.e. for Windows® C:\Septel as shown above.

A script is provided in the SRC directory to build and copy all of the example programs into the UPD\BIN directory. To run this script, change to the SRC directory and type one of the following commands depending on the operating system:

```
makeall      (Linux)
makeall_sol  (Solaris)
makeall.bat  (Windows®)
```

A pre-built copy of the TTU application, for each operating system, can be located within the backup subdirectories in the BIN directory.

To build the TTU program, change to the SRC\TTU directory and type one of the following commands depending on the operating system:

```
make -f ttu.mak
```

```
make -f ttu_sol.mak
```

```
nmake /f ttu.mnt
```

4 Configuration

The local and remote ends of the system need to be configured before the Dialogic® TTU application may be run. Example configuration files are provided in the *Dialogic® User Part Development Package* and after installation will be stored in the directories as shown in the following table:

Root directory	
RUN	
TTU	
“CONFIG1” config.txt system.txt [tcuser.ms7]	“CONFIG2” config.txt system.txt

The configuration files in the CONFIG1 (for point code 1) and CONFIG2 (for point code 2) directories should be copied to the appropriate node. Refer to Appendix A.

4.1 System Configuration

4.1.1 SS7G2x SIU

System and protocol information is configured using the SIU management module and commands in the config.txt and system.txt files. Further information on this can be obtained from the SIU user manual [2].

Note: These files are not contained in the Dialogic® User Part Development Package but for illustration an example is given in the appendix.

4.1.2 SPCI2S, SPCI4 and SS7HD

The GCT environment is configured using the Dialogic® gctload program and the system.txt file. The basic board configuration along with the Dialogic® MTP, SCCP and TCAP modules is achieved using the config.txt file.

Example configuration files for TTU are contained in Appendix A.

4.1.3 SIGTRAN M2PA

It is also possible to run the TTU application from 2 hosts connected in back-to-back with SIGTRAN M2PA links. Configuration files for such systems are contained in Appendix B. When running TTU on a Windows® host system using Dialogic® M2PA, MTP3, SCCP and TCAP running on the host, the provided configuration files may be used without modification.

4.2 Protocol Configuration

The example application files given in the appendices will perform the appropriate protocol configuration shown below. If the user wishes to better understand or alter the configuration given, the following sections will be of interest.

Before configuring the protocol modules it is useful to determine the following information relative to each network entity:

- Local point code
- Local sub-system number
- Remote point code and
- Remote sub-system number

4.2.1 SCCP

The local point code and configuration options (e.g. for ITU-T or ANSI operation) is contained in the main SCCP configuration command and should be set to the appropriate value as described in the programmer's manual [1]. In addition, configuration commands are required for the local subsystem, remote point code and remote sub-system.

4.2.2 TCAP

TCAP may be configured for either ITU-T or ANSI operation in the flags parameter of the TCAP configuration command (further details in the programmer's manual[4]). The dialogue id ranges should be set to allow the appropriate number of ids split between incoming and outgoing dialogues. Some applications may require initiation of dialogues in one direction only.

5 Running the TTU application

Before running the Dialogic® TTU application, the GCT environment must first be initialized and the signaling links brought into service. This is achieved by running the Dialogic® gctload program, and activating the links using the Dialogic® mtpsl utility.

Refer to manuals [7] or [2] for details as appropriate.

5.1 TTU Command Line Arguments

The module takes a number of command line arguments, which are summarized below. These include options for tracing the program as it progresses.

Option	Default	Notes
-m	0x0d	TTU module id
-n	0x66	TTU subsystem number
-o	0x0000	TTU options Add together required option values (see section 5.1.1 TTU Options).
-h, -H -v	-	Displays help message

Example:

```
ttu -m0x1d -n0x66 -o0x8000
```

The above example would set up the module id and subsystem number with the values shown for **-m** and **-n**. The **-o** parameter would prevent an automatic TC-INVOKE and TC-CONTINUE from being sent by TTU in response to a dialogue indication being received.

5.1.1 TTU Options

Options	Value	Notes
OPT_NO_PRINT	0x0001	Do not print primitives on the console
OPT_ANSI	0x4000	Use ANSI primitives
OPT_NO_BGN	0x8000	Do not respond to Begin with Invoke and Continue

6 References

- [1] U05SSS, Dialogic® SCCP Programmer's Manual
- [2] U10SSS, Dialogic® Software Environment Programmer's Manual
- [3] 05-2302, Dialogic® SS7G2x SIU Mode User Manual
- [4] U03HSP, Dialogic® SS7 Programmer's Manual for SPC12S, SPC14 and CPM8
- [5] U06SSS, Dialogic® TCAP Programmer's Manual
- [6] 05-2063, Dialogic® SS7HD Programmer's Manual
- [7] U04STN, Dialogic® Programmer's Manual for Sigtran Host Software

Updates to the documentation are available on the Dialogic web site at <http://www.dialogic.com/support/helpweb/signaling/default.htm>

7 Abbreviations

ANSI	American National Standards Institute
ITU-T	International Telecommunications Union
SCCP	Signaling Connection Control Part
SIU	Signaling Interface Unit
TCAP	Transaction Capabilities Application Part
TTU	TCAP Test Utility
UDT	Unitdata message

Appendix A - Example configuration files with TDM MTP2 links

This section provides example configuration files for use with Dialogic® TTU on a Windows® host system for Dialogic® SPC14 boards using MTP links. The Dialogic® MTP3, SCCP and TCAP modules are running on the host. TTU is running as module ID 0x0d at the remote point code.

Before configuring the protocol modules it is useful to determine information such as the local point code and remote point code relative to each network entity. For this example configuration, the local point code is 1 and the remote point code is 2 (Figure 3).

One system.txt file and two config.txt files are provided.

Example configuration

Operating system:	Windows®
Board type:	SPC14
Local point code:	1 ((TC-USER))
Remote point code:	2 (TTU)
TTU module ID:	0x0d (remote point code)
Modules running on the host:	MTP3, SCCP and TCAP

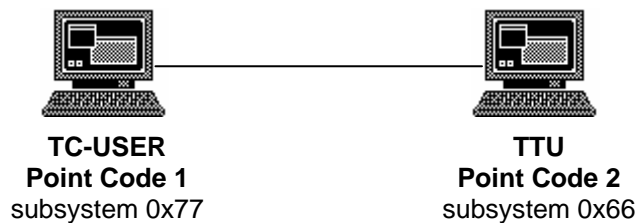


Figure 3. Example configuration

A.1 system.txt

This section provides one example system.txt file for an SPCI4 board running under the Windows® operating system using the example configuration described in this appendix.

The following example system.txt file is valid for point code 1 and point code 2. All comments are denoted by '*'.

A.1.1 system.txt (for point code 1 and point code 2)

```
*****
* Example system.txt.
* Edit this file to reflect your configuration.
*****
*
* Essential modules running on the host:
*
LOCAL          0x20          * ssd - Board Interface task
LOCAL          0x00          * Timer Task
*
* Optional modules running on the host:
*
LOCAL          0xcf          * s7_mgt - Management/config task
LOCAL          0xef          * s7_log - Display and logging utility
*
* Modules that optionally run on the host
*
LOCAL          0x14          * TCAP module
LOCAL          0x33          * SCCP module
LOCAL          0x22          * MTP3 module
LOCAL          0x0d          * TC_USER or TTU module
*
* Essential modules running on the board (all redirected via ssd):
*
REDIRECT       0x71         0x20   * MTP2 module
REDIRECT       0x10         0x20   * CTbus/Clocking control module
REDIRECT       0x8e         0x20   * On-board management module
*
* Redirection of status indications:
*
REDIRECT       0xdf         0xef    * LIU/MTP2 status messages to s7_log
*
* Now start-up all local tasks:
*   (For SPCI4/SPCI2S and CPM8 start-up ssds.exe and
*   for SS7HD boards use ssdh.exe)
*
FORK_PROCESS   ..\..\..\..\ssds.exe -d
FORK_PROCESS   ..\..\..\..\tim_nt.exe
FORK_PROCESS   ..\..\..\..\tick_nt.exe
FORK_PROCESS   ..\..\..\..\s7_mgt.exe -d
FORK_PROCESS   ..\..\..\..\s7_log.exe
FORK_PROCESS   ..\..\..\..\tcap_nt.exe -t
FORK_PROCESS   ..\..\..\..\sccp_nt.exe -t
FORK_PROCESS   ..\..\..\..\mtp_nt.exe -t
```

A.1.2 Using different operating systems and board configurations

The following subsections provide information regarding modifications required to the provided system.txt file if using different board based configurations and operating systems.

A.1.2.1 Running TTU with different board types

If using board types other than SPCI4, a number of commands may differ from the commands in the provided example config.txt file.

- For SS7HDP and SS7HDC:
Replace the following lines:

```

REDIRECT      0x71    0x20    * MTP2 module
FORK_PROCESS  ..\..\..\..\SSDS.EXE -d
with :
REDIRECT      0x81    0x20    * MTP2 module_id for SP0
REDIRECT      0x91    0x20    * MTP2 module_id for SP1
REDIRECT      0xe1    0x20    * MTP2 module_id for SP2
REDIRECT      0xf1    0x20    * MTP2 module_id for SP3
FORK_PROCESS  ..\..\..\..\SSDH.EXE -d

```

- For SPCI2S and CPM8:
The commands in the provided example system.txt file can be used without modification.

Refer to [2] and [7] for further information.

A.1.2.2 Running TTU with MTP3, SCCP, or TCAP on the board

If using a board license button, so that the MTP3, SCCP, or TCAP modules are run on the board instead of on the host, the following changes (using MTP3 as an example) will be necessary:

- Remove the LOCAL command for the MTP3 module
- Remove the FORK_PROCESS command for the MTP3 binary
- Add a REDIRECT command (to redirect messages for the MTP3 module)

Similar changes should also be made for the SCCP and TCAP modules, if these modules are to be run on the board. Refer to [2] and [5] as appropriate.

A.1.2.3 Running TTU with other operating systems

If using operating systems other than Windows®, the names of some of the executable files specified in the FORK_PROCESS commands need to be changed. If appropriate, the user must also ensure that the correct host binaries are used. Refer to [2] and [7] as appropriate.

A.1.2.4 Running TTU with SS7G2x

If using SS7G2x, additional commands will need to be included. The example system.txt provided in this appendix should not be used. Refer to [3] for further information.

A.2 config.txt

This section provides two example config.txt files for an SPCI4 board running under Windows® using the example configuration described earlier in this appendix showing how the protocol modules can be configured for use.

All comments are denoted by '*'.

Using the two example config.txt files (one at each end of the link) will allow dialogues using TCAP with 14-bit point codes to be demonstrated. If connecting to other equipment, the various parameters in the file need to be examined to determine if they are compatible with the configuration at the other end of the link, for example:

- point codes (OPC, DPC)
- signaling timeslot
- sub-system numbers (SSN)

A.2.1 config.txt for point code 1 (TC-USER)

```

*****
* Example config.txt.
* Edit this file to reflect your configuration.
*****
*
* Configure individual boards:
* For SPC14 / SPC12S boards:
* SEPTELPCI_BOARD <board_id> <flags> <code_file> <run_mode>
SEPTELPCI_BOARD 0 0x0003 ss7.dc3 MTP2
*
* For CPM8 boards:
* SEPTELCP_BOARD <board_id> <flags> <code_file> <run_mode>
*SEPTELCP_BOARD 0 0x0003 ss7.dc3 MTP2
*
* For SS7HD PCI boards:
* SS7_BOARD <board_id> <board_type> <flags> <code_file> <run_mode>
*SS7_BOARD 0 SS7HDP 0x0003 ss7.dc4 MTP2
*
* For SS7HD cP boards:
* SS7_BOARD <board_id> <board_type> <flags> <code_file> <run_mode>
*SS7_BOARD 0 SS7HDC 0x0003 ss7.dc4 MTP2
*
* 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 0x2 1 0x0000 0x1 0x08
*
* Define signaling links:
* MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink>
* <stream> <timeslot> <flags>
* For SPC14 / SPC12S / CPM8 boards:
MTP_LINK 0 0 0 0 0 0 0 1 0x0006
* For SS7HD boards:
*MTP_LINK 0 0 0 0 0 0-0 0 1 0x0006
*
* Define a route for each remote signaling point:
* MTP_ROUTE <dpc> <linkset_id> <user_part_mask>
MTP_ROUTE 2 0 0x0008
*
*
* SCCP Parameters:
* SCCP_CONFIG <local_spc> <ssf> <options>
SCCP_CONFIG 1 0x8 0x0102
*
* Enable SCCP traces:
* SCCP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
SCCP_TRACE 0x3 0x7 0x3
*
* Define Remote Signaling Points:
* SCCP_SSR <ssr_id> RSP <remote_spc> <flags> <pc_mask>
SCCP_SSR 1 RSP 2 0 0x0000
*

```

```
* Define Local Sub-Systems:
* SCCP_SSR <ssr_id> LSS <local_ssn> <module_id> <flags> <protocol>
SCCP_SSR 2 LSS 0x77 0x0d 0 TCAP
*
* Define Remote Sub-Systems:
* SCCP_SSR <ssr_id> RSS <remote_spc> <remote_ssn> <flags>
SCCP_SSR 3 RSS 2 0x66 0
*
*
* TCAP Parameters:
* TCAP_CONFIG <base_ogdlg_id> <nog_dialogues> <base_icdlg_id>
* <nic_dialogues> <options> <dlg_hunt> [<addr_format>]
TCAP_CONFIG 0x0 8192 0x8000 8192 0x0000 0 0
*
* Enable TCAP traces:
* TCAP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
TCAP_TRACE 0x7 0xf 0x0
```

A.2.2 config.txt for point code 2 (TTU)

```

*****
* Example config.txt.
* Edit this file to reflect your configuration.
*****
* Configure individual boards:
* For SPCI4 / SPCI2S boards:
* SEPTELPCI_BOARD <board_id> <flags> <code_file> <run_mode>
SEPTELPCI_BOARD 0 0x0002 ss7.dc3 MTP2
*
* For CPM8 boards:
* SEPTELCP_BOARD <board_id> <flags> <code_file> <run_mode>
*SEPTELCP_BOARD 0 0x0002 ss7.dc3 MTP2
*
* For SS7HD PCI boards:
* SS7_BOARD <board_id> <board_type> <flags> <code_file> <run_mode>
*SS7_BOARD 0 SS7HDP 0x0002 ss7.dc4 MTP2
*
* For SS7HD cP boards:
* SS7_BOARD <board_id> <board_type> <flags> <code_file> <run_mode>
*SS7_BOARD 0 SS7HDC 0x0002 ss7.dc4 MTP2
*
*
* 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 1 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 0 1 0x0006
* For SS7HD boards:
*MTP_LINK 0 0 0 0 0 0 0-0 0 1 0x0006
*
* Define a route for each remote signaling point:
* MTP_ROUTE <dpc> <linkset_id> <user_part_mask>
MTP_ROUTE 1 0 0x0008
*
*
* SCCP Parameters:
* SCCP_CONFIG <local_spc> <ssf> <options>
SCCP_CONFIG 2 0x8 0x0102
*
* Enable SCCP traces:
* SCCP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
SCCP_TRACE 0x3 0x7 0x3
*
* Define Remote Signaling Points:
* SCCP_SSR <ssr_id> RSP <remote_spc> <flags> <pc_mask>
SCCP_SSR 1 RSP 1 0 0x0000

```

```
*
* Define Local Sub-Systems:
* SCCP_SSR <ssr_id> LSS <local_ssn> <module_id> <flags> <protocol>
SCCP_SSR 2 LSS 0x66 0x0d 0 TCAP
*
* Define Remote Sub-Systems:
* SCCP_SSR <ssr_id> RSS <remote_spc> <remote_ssn> <flags>
SCCP_SSR 3 RSS 1 0x77 0
*
*
* TCAP Parameters:
* TCAP_CONFIG <base_ogdlg_id> <nog_dialogues> <base_icdlg_id>
* <nic_dialogues> <options> <dlg_hunt> [<addr_format>]
TCAP_CONFIG 0x0 8192 0x8000 8192 0x0000 0 0
*
* Enable TCAP traces:
* TCAP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
TCAP_TRACE 0x7 0xf 0x0
```


A.2.3 Using different operating systems and board configurations

The following subsections provide information regarding modifications required to the provided config.txt file if using different board based configurations and operating systems.

A.2.3.1 Running TTU with different board types

If using board types other than SPCI4, a number of commands may differ from the commands in the provided example config.txt file.

- For SS7HDP:
Replace the <SEPTELPCI_BOARD> command with the following:

```
SS7_BOARD 0 SS7HDP 0x0002 ss7.dc4 MTP2
```

Update the <blink> parameter in the MTP_LINK command as follows:

```
MTP_LINK 0 0 0 0 0 0-0 0 1 0x0006
```

- For SS7HDC:
Replace the <SEPTELPCI_BOARD> command with the following:

```
SS7_BOARD 0 SS7HDC 0x0002 ss7.dc4 MTP2
```

Update the <blink> parameter in the MTP_LINK command as above for SS7HDP.

Refer to [2] for further information.

A.2.3.2 Running TTU with MTP3, SCCP and TCAP on the board

When using a codefile, in this example the whole SS7 stack up to TCAP, may be run on the board:

- For SS7HDP and SS7HDC:
Set the <run_mode> parameter in the SS7_BOARD command to 'TCAP'
- For SPCI2S:
Set the <run_mode> parameter in the SEPTELPCI_BOARD command to 'TCAP'

If SCCP is to be run on the board, the value of the <options> parameter in the SCCP_CONFIG command should to be modified from 0x0102 to 0x0103, i.e. set bit 0 (refer to for details).

A.2.3.3 Running TTU with other operating systems

There are no operating system specific commands in the config.txt file..

A.2.3.4 Running TTU on SS7G2x

If using SS7G2x, additional commands will need to be included. The example config.txt provided in this appendix should not be used. Refer to [3] for further information.

Appendix B - Example configuration with SIGTRAN M2PA links

This section provides example configuration files for use with Dialogic® TTU on a Windows® host system using SIGTRAN M2PA links. The Dialogic® M2PA, SCCP and TCAP modules are running on the host. TTU is running as module ID, 0x0d, at the remote point code.

Like the example for MTP3 links given in Appendix A, before configuring the protocol modules it is useful to determine information such as the local point code and remote point code relative to each network entity. For this example configuration, the local point code is 1 and the remote point code is 2 (Figure 3).

One system.txt file and two config.txt files are provided.

Example configuration

Operating system:	Windows®
Link type:	SIGTRAN M2PA
Local point code:	1 (TC-USER)
Remote point code:	2 (TTU)
TTU module ID:	0x0d
Modules running on the host:	M2PA, SCCP and TCAP



Figure 4. Example configuration

B.1 system.txt

This section provides one example system.txt file for SIGTRAN M2PA hosts running under Windows® using the example configuration described earlier in this appendix.

The following example system.txt files are valid for point code 1 and point code 2. All comments are denoted by *.

```
*****
* Example system.txt.
* Edit this file to reflect your configuration.
*****
*
* Essential modules running on the host:
*
LOCAL          0x00          * Timer Task
*
* Optional modules running on the host:
*
LOCAL          0xcf          * s7_mgt - Management/config task
LOCAL          0xef          * s7_log - Display and logging utility
*
* Modules that optionally run on the host
*
LOCAL          0xc2          * mbm - Management task
LOCAL          0xd0          * SCTPD module
LOCAL          0xd1          * SCTP module
LOCAL          0xc1          * M2PA module
LOCAL          0x22          * MTP3 module
LOCAL          0x33          * SCCP module
LOCAL          0x14          * TCAP module
LOCAL          0x0d          * TTU application
*
* Now start-up all local tasks:
*
FORK_PROCESS   ..\..\..\..\tim_nt.exe
FORK_PROCESS   ..\..\..\..\tick_nt.exe
FORK_PROCESS   ..\..\..\..\sctpd.exe
FORK_PROCESS   ..\..\..\..\sctp.exe
FORK_PROCESS   ..\..\..\..\mbm.exe -d
FORK_PROCESS   ..\..\..\..\m2pa_nt.exe -t
FORK_PROCESS   ..\..\..\..\mtp_nt.exe -t
FORK_PROCESS   ..\..\..\..\sccp_nt.exe -t
FORK_PROCESS   ..\..\..\..\tcap_nt.exe -t
FORK_PROCESS   ..\..\..\..\s7_mgt.exe -d
FORK_PROCESS   ..\..\..\..\s7_log.exe -fss7.log
```

B.1.1 Using different operating systems

The following subsections provide information regarding modifications required to the provided `system.txt` file if using different operating systems.

B.1.1.1 Running TTU with other operating systems

If using operating systems other than Windows[®], the names of some of the executable files used in the `FORK_PROCESS` commands need to be changed. If appropriate, the user must also ensure that the correct host binaries are used.

Refer to the SIGTRAN Programmer's Manual [8] as appropriate.

B.2 config.txt

This section provides two example config.txt files for SIGTRAN M2PA hosts running under Windows® using the example configuration described earlier in this appendix.

The following example config.txt files are for point code 1 and point code 2. All comments are denoted by '*'.

Using the two example config.txt files (one at each end of the link) will allow dialogues using TCAP with 14-bit point codes to be demonstrated. If connecting to other equipment, the various parameters in the file need to be examined to determine if they are compatible with the configuration at the other end of the link, for example:

- point codes (OPC, DPC)
- signaling timeslot
- sub-system numbers
- Host IP addresses

Refer to [8] for SIGTRAN related parameter changes (such as port numbers, IP addresses, etc.)

B.2.1 config.txt for point code 1 (TC-USER)

```

*****
* Example config.txt.
* Edit this file to reflect your configuration.
*****
*
CNSYS:IPADDR=192.168.0.1,PER=0;
*
SNSLI:SNLINK=1,IPADDR=192.168.0.2,SNEND=C,SNTYPE=M2PA,M2PA=1,PPORT=3565;
*
* 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 1 0x0000 1 0x08
*
* Define signaling links:
* MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink>
*           <stream> <timeslot> <flags>
MTP_LINK 0 0 0 0 0 1 0 0 0x80000006
*
* Define a route for each remote signaling point:
* MTP_ROUTE <dpc> <linkset_id> <user_part_mask>
MTP_ROUTE 2 0 0x0008
*
* SCCP Parameters:
* SCCP_CONFIG <local_spc> <ssf> <options>
SCCP_CONFIG 1 0x8 0x0102
*
* Enable SCCP traces:
* SCCP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
SCCP_TRACE 0x3 0x7 0x3
*
* Define Remote Signaling Points:
* SCCP_SSR <ssr_id> RSP <remote_spc> <flags> <pc_mask>
SCCP_SSR 1 RSP 2 0 0x0000
*
* Define Local Sub-Systems:
* SCCP_SSR <ssr_id> LSS <local_ssn> <module_id> <flags> <protocol>
SCCP_SSR 2 LSS 0x77 0x0d 0 TCAP
*
* Define Remote Sub-Systems:
* SCCP_SSR <ssr_id> RSS <remote_spc> <remote_ssn> <flags>
SCCP_SSR 3 RSS 2 0x66 0
*
*
* TCAP Parameters:
* TCAP_CONFIG <base_ogdlg_id> <nog_dialogues> <base_icdlg_id>
*           <nic_dialogues> <options> <dlg_hunt> [<addr_format>]
TCAP_CONFIG 0x0 8192 0x8000 8192 0x0000 0 0
*
* Enable TCAP traces:
* TCAP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
TCAP_TRACE 0x7 0xf 0x0

```

B.2.2 config.txt for point code 2 (TTU)

```
*****
* Example config.txt.
* Edit this file to reflect your configuration.
*****
*
CNSYS:IPADDR=192.168.0.2,PER=0;
*
SNSLI:SNLINK=1,IPADDR=192.168.0.1,SNEND=S,SNTYPE=M2PA,M2PA=1,PPORT=3565;
*
* 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 1 0x0000 2 0x08
*
* Define signaling links:
* MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink>
* <stream> <timeslot> <flags>
MTP_LINK 0 0 0 0 0 1 0 0 0x80000006
*
* Define a route for each remote signaling point:
* MTP_ROUTE <dpc> <linkset_id> <user_part_mask>
MTP_ROUTE 1 0 0x0008
*
* SCCP Parameters:
* SCCP_CONFIG <local_spc> <ssf> <options>
SCCP_CONFIG 2 0x8 0x0102
*
* Enable SCCP traces:
* SCCP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
SCCP_TRACE 0x3 0x7 0x3
*
* Define Remote Signaling Points:
* SCCP_SSR <ssr_id> RSP <remote_spc> <flags> <pc_mask>
SCCP_SSR 1 RSP 1 0 0x0000
*
* Define Local Sub-Systems:
* SCCP_SSR <ssr_id> LSS <local_ssn> <module_id> <flags> <protocol>
SCCP_SSR 2 LSS 0x66 0x0d 0 TCAP
*
* Define Remote Sub-Systems:
* SCCP_SSR <ssr_id> RSS <remote_spc> <remote_ssn> <flags>
SCCP_SSR 3 RSS 1 0x77 0
*
*
* TCAP Parameters:
* TCAP_CONFIG <base_ogdlg_id> <nog_dialogues> <base_icdlg_id>
* <nic_dialogues> <options> <dlg_hunt> [<addr_format>]
TCAP_CONFIG 0x0 8192 0x8000 8192 0x0000 0 0
*
* Enable TCAP traces:
* TCAP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
TCAP_TRACE 0x7 0xf 0x0
```


B.2.3 Using different operating systems

The following subsections provide information regarding modifications required to the provided config.txt file if using different operating systems.

B.2.3.1 Running TTU with other operating systems

No modifications required. There are no operating system specific commands in the config.txt file.

Appendix C - TC-USER example script file

This section provides a example script file which may be used to represent the TCAP user at point code 1. The file can be used with the Dialogic® s7_play utility which is supplied in the *Dialogic® User Part Development Package*. This file will begin and end a new dialogue with TTU.

C.1 tcuser.ms7 – point code 1

```
*****
*
* TCAP Messages - use s7_play to send in messages to remote ttu application
*
*****
*
* TC_INVOKE service request
*
*          --data      --component
*          --length----data
*          --class --length--length          --final_byte
*          --invoke--timeout  -----data
M-tc781-i0000-fef-d14-p080301010402000a0108a10602010002010200
*
* TC_BEGIN dialogue request
*
*          -----data
*          --length      -----data
*          --dest_addr  --length
*          --begin      --orig_addr --final_byte
M-tc783-i0000-fef-d14-p0207044302006608044301007700
*
* Delay required
*
* ----delay_ms_hex
D-m1000
*
* TC_RESULT_L service request
*
*          -----data
*          --length
*          --component
*          --result_l  --final_byte
M-tc781-i0000-fef-d14-p090105a20302010100
*
* TC_END dialogue request
*
*          --final_byte
*          --end_dialogue
M-tc783-i0000-fef-d14-p0400
*
```

Appendix D - Sample Output

The following shows sample output from running the Dialogic® TTU application using the provided example configuration at point code 2. TTU has a module id of 0x0d.

TTU was started using :

```
ttu -m0x0d
```

TTU example output:

```
TTU TCAP Test Utility      Copyright (C) Dialogic Corporation 2007. All Rights
Reserved.
=====
TTU module ID - 0x0d
TTU sub-system number - 0x66
Options set - 0x0000

TTU:I00 M t8744 i0066 f33 d0d s06 p010100000000
TTU: Dialogue ind, instance 00:
  dialogue_id = 0x8000  TC-BEGIN/QUERY
  COMPONENT(S) PRESENT
  orig_addr = 43 01 00 77
  dest_addr = 43 02 00 66
TTU: Component req: 08 03 01 01 04 02 00 1e 01 08 a1 06 02 01 01 02 01 01 00
  dialogue_id = 0x8000  TC-INVOKE
  class = 1
  timeout = 30
  invoke_id = 02 01 01
  operation = 02 01 01
TTU: Dialogue req: 03 06 01 03 00
  dialogue_id = 0x8000  TC-CONTINUE/CONVERSATION
  qos = RET_OPT, SEQ_CTRL,
TTU: Component ind, instance 00:
  dialogue_id = 0x8000  (Last component)  TC-INVOKE
  invoke_id = 02 01 00
  operation = 02 01 02
TTU: Component req: 09 01 10 a2 0e 02 01 00 30 09 02 01 01 04 04 52 52 2d 4c 00
  dialogue_id = 0x8000  TC-RESULT-L
  invoke_id = 02 01 00
  operation = 02 01 01
  param = 04 04 52 52 2d 4c
TTU: Dialogue req: 03 06 01 03 00
  dialogue_id = 0x8000  TC-CONTINUE/CONVERSATION
  qos = RET_OPT, SEQ_CTRL,
TTU: Dialogue ind, instance 00:
  dialogue_id = 0x8000  TC-END/RESPONSE (Basic)
  COMPONENT(S) PRESENT
TTU: Component ind, instance 00:
  dialogue_id = 0x8000  (Last component)  TC-RESULT-L
  invoke_id = 02 01 01
```