Dialogic® SS7 Protocols
MAP Test Utility (MTU) and Responder (MTR) User Guide

Document Reference  U30SSS
## Revision History

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>03-Oct-01</td>
<td>Initial Issue</td>
</tr>
<tr>
<td>1</td>
<td>15-Feb-02</td>
<td>Updated after review</td>
</tr>
<tr>
<td>2</td>
<td>31-Jan-03</td>
<td>Branding changed: Septel PCI now SPCI4/SPCI2S and Septel cP now CPM8.</td>
</tr>
<tr>
<td>3</td>
<td>01-Oct-07</td>
<td>Changed copyright to Dialogic Corporation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remove reference to PCCS6, include non-circuit protocol configuration commands in config.txt and add support for SIGTRAN M2PA links</td>
</tr>
</tbody>
</table>
B.2.2 config.txt for point code 2 (MTR).................................................................................................. 37
Appendix C - Sample Output.................................................................................................................. 39
1 Introduction

The Dialogic® MAP Test Utility (MTU) and the responder (MTR) are example applications for demonstrating the use of the Dialogic® MAP module, using the direct interface to the protocol module. This user guide describes the design, building and usage of this application for developers.

The MTU and MTR demonstrate the behavior of the Dialogic® MAP module by offering simple functionality while showing real examples of the interface to the module. The MTU and MTR applications can also be used to provide simple verification of system configuration and set-up.

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 MAP module.

1.1 Software requirements

The MTU/MTR 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, TCAP, MAP host binaries, as required

4. For SIGTRAN-based configurations:
   - Dialogic® M2PA, MTP3, SCCP and TCAP, MAP host binaries, as required

Software can be downloaded from http://www.dialogic.com/support/helpweb/signaling/software3.htm
2 MTU Application

The Dialogic® MTU and MTR applications may be used to demonstrate sending MAP services in a GSM network. MTU is able to generate the following services:

- MAP-SEND-IMSI
- MAP-SEND-ROUTING-INFO-FOR-GPRS
- MAP-FORWARD-SHORT-MESSAGE

The following diagram shows the network entities involved in the MAP-FORWARD-SHORT-MESSAGE service:

![GSM Network Architecture Diagram]

Figure 1. GSM Network Architecture
The SS7 layers involved and the two applications are shown below in Figure 2.

![Diagram showing SS7 layers and MTU, MTR applications]

**Figure 2.** Typical configuration

### 2.1 MTU

MTU is the Dialogic® MAP Test Utility, which generates the dialogues to which MTR responds. It first opens a dialogue to the remote node then sends the service request and finally waits for the result. Using command line options, you may set the dialogue addresses, data fields and dialogue ids to be used. Further options allow the utility to generate a single dialogue or multiple dialogues. Full tracing of sent and received messages is also included.

### 2.2 MTR

The MTR utility responds to the MTU generated requests and displays the message data. It also allows the tracing of sent and received messages. As it replies to dialogues from the remote node it need not allocate dialogue ids itself as the Dialogic® MAP module will take that responsibility.
2.3 Message Sequence Chart

The following pages contain message sequence charts showing a typical message flow for a single dialogue between MTU and MTR. Each chart shows the message flow between the application and the MAP[7], TCAP[6] and SCCP[5] modules.

**Figure 3.** Single dialogue at MTU node
Figure 4. Single dialogue at MTR node
2.4 Customizing the MTU and MTR applications

MTU/MTR are example programs for development using the MAP module and as such, a number of simplifications and limitations have been imposed for this purpose.

MTU does not try to prevent congestion on the link, so it is possible to generate more traffic than the links can handle. The lower levels of the stack provide some protection from the effect of link congestion by simply discarding messages. Without changes, MTU should be run at lower traffic loading levels.

MTU does not provide support for segmentation of messages that are too long for the 'short' message service.

MTR assumes the use of MTU as the dialogue generating application and supports only the functionality required to reply to dialogues generated by MTU.

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

The MTU program can be found in the Dialogic® User Part Development Package. The following table describes the files that make up the MTU application:

<table>
<thead>
<tr>
<th>File</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mtu_main.c</td>
<td>This file contains the main() function. This reads the command line arguments and passes them to mtu_ent().</td>
</tr>
<tr>
<td>mtu.c</td>
<td>mtu_ent() calls the function MTU_open_dlg() to open a dialogue, send a service request message and send a delimiter to indicate that currently no further requests will be made. It waits for indications to be received, and these are handled by the main state machine MTU_smac(). Depending on the trace options the message is displayed by MTU_display_recvd_msg().</td>
</tr>
<tr>
<td>mtu_fmt.c</td>
<td>Contains functions to format dialogue and service requests destined for MAP and also to convert indications from MAP into a structured format.</td>
</tr>
<tr>
<td>mtu.h</td>
<td>Contains macro and structure definitions used in MTU.</td>
</tr>
</tbody>
</table>

2.6 MTR source code

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

<table>
<thead>
<tr>
<th>File</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mtr_main.c</td>
<td>This file contains the main() function. This reads the command line arguments and passes them to mtr_ent().</td>
</tr>
<tr>
<td>mtr.c</td>
<td>This file contains mtr_ent() which sends an open response using MTR_send_OpenResponse() when an open dialogue indication is received and then waits for a service indication and a delimiter. When both have been received, a service response is sent, followed by a close dialogue request using MTR_send_MapClose().</td>
</tr>
<tr>
<td>mtr.h</td>
<td>Contains macro and structure definitions used in MTR.</td>
</tr>
</tbody>
</table>
3 Building the MTU and MTR applications

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

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic UNIX (Solaris, Linux)</td>
<td>.mak</td>
</tr>
<tr>
<td>Windows®</td>
<td>.mnt</td>
</tr>
</tbody>
</table>

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 GCT shared object 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 MTU 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 MTU and MTR programs only.

2. The C header files in the INC directory shown in the table below lists the header files required by the MTU and MTR programs.

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

<table>
<thead>
<tr>
<th>Root directory</th>
<th>Septel</th>
</tr>
</thead>
<tbody>
<tr>
<td>INC</td>
<td>BIN</td>
</tr>
<tr>
<td>asciibin.h</td>
<td>mtu.mnt</td>
</tr>
<tr>
<td>bit2byte.h</td>
<td>mtu.mak</td>
</tr>
<tr>
<td>map_inc.h</td>
<td>mtu_sol.mak</td>
</tr>
<tr>
<td>msg.h</td>
<td>mtu.c</td>
</tr>
<tr>
<td>pack.h</td>
<td>mtu_fmt.c</td>
</tr>
<tr>
<td>ss7_inc.h</td>
<td>mtu_main.c</td>
</tr>
<tr>
<td>strtonum.h</td>
<td>mtu.h</td>
</tr>
<tr>
<td>sysgct.h</td>
<td>mtu.iss.txt</td>
</tr>
<tr>
<td>system.h</td>
<td></td>
</tr>
</tbody>
</table>

3.2 Building MTU and MTR

It is assumed that the UPD is extracted in the Dialogic® SS7 Development Package directory i.e. for Windows® C:\Septel as shown above. Similarly, for Linux, a symbolic link should be created it is assumed that the UPD is unpacked and stored within the DPK directory.

A script is provided in the BIN\SRC directory to build and copy all of the example programs into the UPD/BIN directory. To run this script, change to the BIN\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 MTU and MTR applications, for each operating system, can be located within the backup subdirectories in the BIN directory.
To build the MTU program, change to the SRC\MTU directory and type one of the following commands depending on the operating system:

make –f mtu.mak
make –f mtu_sol.mak
nmake /f mtu.mnt

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

make –f mtr.mak
make –f mtr_sol.mak
nmake /f mtr.mnt
4 Configuration

The local and remote ends of the system need to be configured before the Dialogic® MTU and MTR applications 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:

<table>
<thead>
<tr>
<th>Root</th>
<th>MTU</th>
<th>MTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>“CONFIG”</td>
<td>config.txt</td>
<td>config.txt</td>
</tr>
<tr>
<td></td>
<td>system.txt</td>
<td>system.txt</td>
</tr>
</tbody>
</table>

Example configurations for TDM and SIGTRAN are provided. The configuration files in the MTU and MTR directories should be copied to the appropriate node. Refer to Appendix A - Example configuration files for further information.

4.1 System Configuration

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, TCAP and MAP modules is achieved using the config.txt file.

4.1.1 SPCI2S, SPCI4 and SS7HD

For TDM based configurations, example board based configuration files are provided in Appendix A for MTU and MTR.

When running MTU and MTR on a Windows® host system using SPCI4 with MTP2 running on the board and Dialogic® MTP3, SCCP, TCAP, and MAP running on the host, the provided configuration files may be used without any modification.

Configuration details for other board types are also provided for reference.

4.1.2 SIGTRAN M2PA

It is also possible to run a similar configuration as described in Section 4.1.1 but with SIGTRAN M2PA links. Configuration files for such systems are contained in Appendix B. When running MTU and MTR on a Windows® host system using Dialogic® M2PA, MTP3, SCCP, TCAP and MAP running on the host, the provided configuration files may be used without any modification.

4.1.3 SS7G2x

System and protocol information is configured using the management module and commands in the config.txt and system.txt files for the SS7G2x. Further information on this can be obtained from the user manual [2].

*Note: These files are not contained in the User Part Development Package.*
4.2 Protocol Configuration

All protocol modules are configured using commands in the config.txt file. The example configuration 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 is contained in the main SCCP configuration command and this should be set to the appropriate value. In addition, configuration commands are required for the local subsystem, remote point code and remote sub-system.

4.2.2 TCAP

TCAP should be configured for ITU operation in the flags parameter of the TCAP configuration command (refer to the programmer's manual [6] for details). 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.

4.2.3 MAP

The TCAP dialogue base id and number values should be set to match those given in the TCAP configuration module. The user dialogue values are a separate independent range from the TCAP dialogue ids but will need to be similarly dimensioned, e.g. if 16 incoming dialogue ids are configured in MAP then TCAP must also support at least 16 incoming dialogue ids.
5 Running the MTU and MTR applications

Copy the host binaries to the Dialogic® SS7 Development Package directory.

Before running the Dialogic® MTU and MTR applications, 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 [1], [3] or [4] for details as appropriate.

5.1 MTU Command Line Arguments

The module takes a number of command line arguments, which are summarized below:

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>-</td>
<td>destination address (encoded in accordance with Q.713 definition of called party address). This parameter is mandatory.</td>
</tr>
<tr>
<td>-b</td>
<td>0x0000</td>
<td>base MAP dialogue ID</td>
</tr>
<tr>
<td>-d</td>
<td>0</td>
<td>Mode of operation (see section 5.1.1 for details)</td>
</tr>
<tr>
<td>-g</td>
<td>-</td>
<td>originating address (encoded in accordance with Q.713 definition of called party address) This parameter is mandatory.</td>
</tr>
<tr>
<td>-h, -H -v</td>
<td>-</td>
<td>Displays help message</td>
</tr>
<tr>
<td>-m</td>
<td>0x2d</td>
<td>MTU module ID</td>
</tr>
<tr>
<td>-n</td>
<td>0x400</td>
<td>number of dialogues to use</td>
</tr>
<tr>
<td>-o</td>
<td>0x00f</td>
<td>Output display options (add together required values for tracing options (section 5.1.2)</td>
</tr>
<tr>
<td>-u</td>
<td>0x15</td>
<td>MAP module ID</td>
</tr>
<tr>
<td>-x</td>
<td>0</td>
<td>number of active dialogues to maintain</td>
</tr>
</tbody>
</table>

Example:

```
mtu -d0 -a43020008 -g43010008 -s"good morning" -i987654321
```

The above example will open a single dialogue and send a short message from point code 1, SSN 0x8, to point code 2, SSN 0x8. The short message sent will be ‘good morning’, and the international mobile subscriber ID will be 987654321. If option –x were added, MTU would open multiple dialogues up to the value specified by option –x or up to the maximum that could be supported. If this maximum is exceeded and error tracing is activated (section 5.1.1) a message indicating failure will be displayed.
5.1.1 MTU Modes of Operation

MTU is able to accept parameters other than those shown in section 5.1. Whether the extra parameters can be used is dependent on the value provided for option –d (mode of operation). There are four acceptable values for option –d ranging from 0 to 3. The meaning of these values and other parameters which MTU can accept are detailed in the tables below.

If –d parameter has value 0 (default) – Forward short message

<table>
<thead>
<tr>
<th>Option</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-p</td>
<td>MAP phase (can also take value 1)</td>
</tr>
<tr>
<td>-i</td>
<td>international mobile subscriber ID (see programmer’s manual for format [7]). This parameter is mandatory.</td>
</tr>
<tr>
<td>-c</td>
<td>Service centre address (see programmer’s manual for format [7])</td>
</tr>
<tr>
<td>-s</td>
<td>short message (free text). This parameter is mandatory.</td>
</tr>
</tbody>
</table>

If –d parameter has value 1 – Send IMSI

<table>
<thead>
<tr>
<th>Option</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-e</td>
<td>MSISDN (see programmer’s manual for format [7])</td>
</tr>
</tbody>
</table>

If –d parameter has value 2 – Send routing info for GPRS

<table>
<thead>
<tr>
<th>Option</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f</td>
<td>GGSN number (see definition of ISDN-AddressString [9])</td>
</tr>
<tr>
<td>-i</td>
<td>international mobile subscriber ID (see programmer’s manual for format [7]). This parameter is mandatory.</td>
</tr>
</tbody>
</table>

If –d parameter has value 3 – Send IMSI & Send routing info for GPRS

<table>
<thead>
<tr>
<th>Option</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-e</td>
<td>MSISDN (see programmer’s manual for format [7])</td>
</tr>
<tr>
<td>-f</td>
<td>GGSN number (see definition of ISDN-AddressString [9])</td>
</tr>
</tbody>
</table>

5.1.2 MTU Display Options

<table>
<thead>
<tr>
<th>Tracing option</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTU_TRACE_TX</td>
<td>0x0001</td>
<td>Trace transmitted dialogue messages</td>
</tr>
<tr>
<td>MTU_TRACE_RX</td>
<td>0x0002</td>
<td>Trace received dialogue requests</td>
</tr>
<tr>
<td>MTU_TRACE_ERROR</td>
<td>0x0004</td>
<td>Trace error messages</td>
</tr>
<tr>
<td>MTU_TRACE_STATS</td>
<td>0x0008</td>
<td>Trace the number of opened dialogues</td>
</tr>
</tbody>
</table>
5.2 MTR Command Line Arguments

The module takes a number of command line arguments, which are summarized below:

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-m</td>
<td>0x2d</td>
<td>MTR module id</td>
</tr>
<tr>
<td>-u</td>
<td>0x15</td>
<td>MAP module id</td>
</tr>
<tr>
<td>-t</td>
<td>Enabled</td>
<td>Disable tracing. If -t parameter is provided, tracing is disabled.</td>
</tr>
<tr>
<td>-h, -H, -?, -v</td>
<td>-</td>
<td>Displays help information</td>
</tr>
</tbody>
</table>

Example:

mtr -m0xd

The above example will run MTR with a module id of 0x0d and MTR will assume the MAP module below has a module id of 0x15. MTR will also trace any messages received or transmitted.
6 References

[9] ETSI TS 100 974v7.5.1, Digital Cellular Telecommunications System (Phase 2+)

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

The following lists acronyms alphabetically used in this user guide.

GGSN Gateway GPRS Support Node
GPRS General Packet Radio Service
GSM Global System for Mobile Communications
HLR Home Location Register
IMSI International Mobile Subscriber Identity
MAP Mobile Application Part
MSC Mobile Switching Centre
MTP Message transfer part
MTR MAP Test Responder
MTU MAP Test Utility
SCCP Signaling Connection Control Part
SIU Signaling Interface Unit
SMS Short Message Service
TCAP Transaction Capabilities Application Part
VLR Visitor Location Register
Appendix A - Example configuration files using SPCI4 boards

This section provides example configuration files for use with Dialogic® MTU and MTR applications on a Windows® host system for SPCI4 boards. The Dialogic® MAP, TCAP, SCCP and MTP3 modules are running on the host and both MTU and MTR are running as module ID 0x2d.

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 MTU point code is 1 and the MTR point code is 2 (Figure 5).

Example configuration

<table>
<thead>
<tr>
<th>Operating system:</th>
<th>Windows®</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board type:</td>
<td>SPCI4</td>
</tr>
<tr>
<td>Local point code:</td>
<td>1 (MTU)</td>
</tr>
<tr>
<td>Remote point code:</td>
<td>2 (MTR)</td>
</tr>
<tr>
<td>MTU module ID (Local point code):</td>
<td>0x2d</td>
</tr>
<tr>
<td>MTR module ID (Remote point code):</td>
<td>0x2d</td>
</tr>
<tr>
<td>Modules running on the host:</td>
<td>MAP/TCAP/SCCP/MTP3</td>
</tr>
</tbody>
</table>

Figure 5. Example configuration
Appendix A - Example configuration files using SPCI4 boards

A.1 system.txt

This section provides one example system.txt file for an SPCI4 board running under Windows® using the example configuration described earlier 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(MTU) and point code 2(MTR)

*************************************************************************
* 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           0x2d            * MTR or MTU module
LOCAL           0x15            * MAP module
LOCAL           0x14            * TCAP module
LOCAL           0x33            * SCCP module
LOCAL           0x22            * MTP3 module
*
* 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
REDIRECT        0x9e    0xef
*
* Now start-up all local tasks:
*
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    ..\..\..\..\map_nt.exe -t
FORK_PROCESS    ..\..\..\..\tcap_nt.exe -t
FORK_PROCESS    ..\..\..\..\sccp_nt.exe -t
FORK_PROCESS    ..\..\..\..\mtp_nt.exe -t
A.2 Using different operating systems and board configurations

The following subsections provide information regarding the system.txt file if using different operating systems or board based configurations.

A.2.1 Running MTU/MTR with SS7HD

If using SSHD, the following lines:

```
REDIRECT        0x71    0x20    * MTP2 module
FORK_PROCESS    SSDS.EXE -d
```

should be replaced by:

```
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
```


A.2.2 Running MTU/MTR with SS7G2x

If using the SIU, additional commands required by the SIU will need to be included. Therefore, the example system.txt provided in this appendix should not be used. Refer to [2] for further information.

A.2.3 Running MAP/TCAP/SCCP on the board

If using a board license button so that the MAP, TCAP, SCCP and MTP3 modules are run on the board instead of the host, remove the FORK_PROCESS command for the MAP host binary, the LOCAL declaration should be removed for the MAP module and a REDIRECT command (which redirects messages for that module to the board) should be added. Similar changes should also be made for the MTP3, SCCP and TCAP modules.

Refer to [2], [3] and [4] as appropriate.

A.2.4 Running MTU/MTR 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.

Refer to [3] and [4] as appropriate.
A.3 config.txt

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

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 MAP with 14-bit point codes to be demonstrated. If connecting to other equipment, all 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.3.1 config.txt for point code 1 (MTU)

*************************************************************************
* 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 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
  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>
  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>
  MTP_LINKSET 0 2 1 0x0000 1 0x08
  Define signaling links:
  MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink>
  MTP_LINK 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 0x08 0x2d 0 MAP
*
* Define Remote Sub-Systems:
* SCCP_SSR <ssr_id> RSS <remote_spc> <remote_ssn> <flags>
SCCP_SSR 3 RSS 2 0x08 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
*
* MAP Parameters:
* MAP_CONFIG <options>
MAP_CONFIG 0x0
*
* Enable MAP traces:
* MAP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
MAP_TRACE 0xf 0xf 0x15
A.3.2 config.txt for point code 2 (MTR)

*************************************************************************
* 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 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>
  * Note: The SS7HD board requires a compound parameter for blink of the 
  form
  * sp_id-sp_channel.
  * For SPCI4 / SPCI2S / CFM8 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 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 0x08 0x2d 0 MAP

* Define Remote Sub-Systems:
* SCCP_SSR <ssr_id> RSS <remote_spc> <remote_ssn> <flags>
SCCP_SSR 3 RSS 1 0x08 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

* MAP Parameters:
* MAP_CONFIG <options>
MAP_CONFIG 0x0

* Enable MAP traces:
* MAP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
MAP_TRACE 0xf 0xf 0x15
A.4 Using different operating systems and board configurations

The following subsections provide information regarding the config.txt file if using different operating systems or board based configurations.

A.4.1.1 Running MTU/MTR with SS7HDP

If using SS7HDP boards, the `SEPTELPCI_BOARD` command should be replaced with the following:

```
SS7_BOARD  0 SS7HDP 0x0003 ss7.dc4 MTP2
```

Refer to [4] for further information.

A.4.1.2 Running MAP/TCAP/SCCP on the board

When using a license button so that the MAP, TCAP, SCCP and MTP modules run on the board instead of the host:

- For SPCI2S and SPCI4:
  
  the `<run_mode>` field in the `SEPTELPCI_BOARD` command should be set to MAP

- For SS7HDP:
  
  the `<run_mode>` field in the `SS7_BOARD` command should be set to MAP

Refer to [3] and as appropriate.

A.4.1.3 Running MTU/MTR with the SIU

If using the SIU, additional commands required by the SIU will need to be included. Therefore, the example config.txt provided in this appendix should not be used.


A.4.1.4 Running MTU/MTR with other operating systems

There are no additional commands specific to various operating systems.

Refer to [3] and [4] as appropriate.
Appendix B - Example configuration files using SIGTRAN M2PA

This section provides example configuration files for use with Dialogic® MTU and MTR applications on a Windows® host system using SIGTRAN M2PA links. The Dialogic® MAP, TCAP, SCCP, MTP3 and M2PA modules are running on the host. Both MTU and MTR are running as module ID 0x2d.

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 MTU point code is 1 and the MTR point code is 2 (Figure 5).

**Example configuration**

- Operating system: Windows®
- Link type: SIGTRAN M2PA
- Local point code: 1 (MTU)
- Remote point code: 2 (MTR)
- MTU module ID (Local point code): 0x2d
- MTR module ID (Remote point code): 0x2d

![Example configuration diagram](image)

**Figure 6.** Example configuration
B.1 system.txt

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

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

```
*************************************************************************
* Example system.txt.
* Edit this file to reflect your configuration.
*************************************************************************
*
* Modules running on the host:
*
LOCAL  0x00  * Timer Task
LOCAL  0xcf  * s7_mgt - Management/config task
LOCAL  0xef  * s7_log - Display and logging utility
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  0x15  * MAP module
LOCAL  0x2d  * MTR or MTU module
*
* 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 ..\..\..\..\..\m2pa_nt.exe -t
FORK_PROCESS ..\..\..\..\..\mtp_nt.exe -t
FORK_PROCESS ..\..\..\..\..\mbm.exe -d
FORK_PROCESS ..\..\..\..\..\sccp_nt.exe -t
FORK_PROCESS ..\..\..\..\..\tcp_nt.exe -t
FORK_PROCESS ..\..\..\..\..\map_nt.exe -t
FORK_PROCESS ..\..\..\..\..\s7_mgt.exe -d
FORK_PROCESS ..\..\..\..\..\s7_log.exe -fss7.log
```

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.

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 MAP 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

If using the example config.txt files please edit the ip address of the local and remote host to match your system.

Refer to [8] for SIGTRAN related parameter changes (such as port numbers, IP addresses, etc.)
B.2.1 config.txt for point code 1 (MTU)

*************************************************************************
* Example config.txt.                                          
* Edit this file to reflect your configuration.                  
*************************************************************************

* NOTE: The ip addresses in the CNSYS and SNSLI command needs to match
* your configuration. Change the ip address in the CNSYS command to
* match your local machine and in the SNSLI command to match your remote
* machine.
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 0x102

* 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 0x08 0x2d 0 MAP

* Define Remote Sub-Systems:
* SCCP_SSR <ssr_id> RSS <remote_spc> <remote_ssn> <flags>
SCCP_SSR 3 RSS 2 0x08 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 <opEvt_mask> <ipEvt_mask> <nonPrim_mask>
TCAP_TRACE 0x7 0xf 0x0
*
* MAP Parameters:
* MAP_CONFIG <options>
MAP_CONFIG 0x0
*
* Enable MAP traces:
* MAP_TRACE <opEvt_mask> <ipEvt_mask> <nonPrim_mask>
MAP_TRACE 0xf 0xf 0x15
B.2.2 config.txt for point code 2 (MTR)

*************************************************************************
* 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 0x08 0x2d 0 MAP

* Define Remote Sub-Systems:
* SCCP_SSR <ssr_id> RSS <remote_spc> <remote_ssn> <flags>
SCCP_SSR 3 RSS 1 0x08 0

* TCAP Parameters:
* TCAP_CONFIG <base_ogdlg_id> <nog_dialogues> <base_icdlg_id> <nic_dialogues> <options> <dlg_hunt> [<addr_format>]
TCAP_CONFIG 0x0 8192 0x0000 8192 0x0000 0 0

* Enable TCAP traces:
* TCAP TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
TCAP_TRACE 0x7 0xf 0x0
* MAP Parameters:
  * MAP_CONFIG <options>
  MAP_CONFIG 0x0

  * Enable MAP traces:
  * MAP_TRACE <op_evt_mask> <ip_evt_mask> <non_prim_mask>
  MAP_TRACE 0xf 0xf 0x15
Appendix C - Sample Output

The following is a sample output from the Dialogic® MTU and MTR applications. Both MTU and MTR have module id of 0x2d.

MTU was started using:

```
mtu -d0 -g43010008 -a43020008 -i987654321 -s"Hello World"
```

MTU example output:

```
MTU MAP Test Utility  Copyright (C) Dialogic Corporation 1997-2006. All Rights Reserved
==============================================================
MTU mod ID 0x2d; MAP module Id 0x15
mode 0 - Forward Short Message

MTU Tx: sending Open Request
MTU Tx: I0000 M tc7e2 i0000 f2d d15 s00 p010b0906070400000100190201044302000803044301000800
MTU Tx: sending Forward Short Message Request
MTU Tx: I0000 M tc7e0 i0000 f2d d15 s00
p030e01011707000589674523f118080406a12436587f9191c2f090421436587f90400601002212342000bc832
9f9d0654d7f376192d223c0000
MTU Tx: sending Delimiter Request
MTU Tx: I0000 M tc7e2 i0000 f2d d15 s00 p0500
MTU Rx: received Open Confirmation
MTU Rx: I0000 M t87e3 i0000 f15 d2d s00
p8250100b0906070400000100190201044301000803044302000800
MTU Rx: received Forward Short Message Confirmation
MTU Rx: I0000 M t87e1 i0000 f15 d2d s00 p8400e010100
MTU Rx: received Close Indication
MTU Rx: I0000 M t87e3 i0000 f15 d2d s00 p0400

Corresponding MTR output:

```
MTR MAP Test Responder (C) Dialogic Corporation 1997-2006. All Rights Reserved
=================================================================
MTR mod ID - 0x2d; MAP module Id 0x15
MTR Rx: I0000 M t87e3 i8003 f15 d2d s00 p020104430200080304430100080b906070400000100190200
MTR Rx: Received Open Indication
MTR Rx: Sending Open Response
MTR Rx: I0000 M tc7e2 i8003 f2d d15 s00 p810501000b0906070400000100190200
MTR Rx: received Forward Short Message Confirmation
MTR Rx: I0000 M t87e1 i8003 f15 d2d s00
p0400e0101191c2f090421436587f90400601002212342000bc8329f9d0654d7f7261917000589674523f11808
0406a12436587f900
MTR Rx: Received Forward Short Message Indication
MTR Rx: Short Message User Information:
MTR Rx: Hello World
MTR Rx: I0000 M t87e3 i8003 f15 d2d s00 p0600
MTR Rx: Received delimiter Indication
MTR Rx: Sending Forward SM Response
MTR Rx: I0000 M tc7e0 i8003 f2d d15 s00 p8300e010100
MTR Rx: Sending Close Request
MTR Rx: I0000 M tc7e2 i8003 f2d d15 s00 p0307010000
```