



**Dialogic<sup>®</sup> SS7 Protocols  
INAP Test Utility (INTU) User Guide**

Document Reference U29SSS

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: U29SSS, Issue 3

**Revision History**

<b>Issue</b>	<b>Date</b>	<b>Description</b>
1	22-Mar-02	Initial Issue
2	04-Aug-03	Naming changed to reference SPC14/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

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
1.1	Software requirements.....	5
<b>2</b>	<b>INTU Application .....</b>	<b>6</b>
2.1	SSF .....	7
2.2	Message Sequence Chart .....	7
2.2.1	Successful Call – Connect Operation .....	8
2.2.2	Invalid Service Key – User Error .....	9
2.2.3	Invalid Parameter - ReleaseCall Operation.....	10
2.3	Customizing the example applications .....	11
2.4	INTU source code.....	11
<b>3</b>	<b>Building the INTU application .....</b>	<b>12</b>
3.1	Host software directory structure.....	13
3.2	Building INTU.....	13
<b>4</b>	<b>Configuration.....</b>	<b>15</b>
4.1	System Configuration .....	15
4.1.1	SS7G2x SIU .....	15
4.1.2	SPCI2S, SPCI4 and SS7HD .....	15
4.1.3	SIGTRAN M2PA.....	15
4.2	Protocol Configuration .....	16
4.2.1	SCCP.....	16
4.2.2	TCAP .....	16
4.2.3	INAP .....	16
<b>5</b>	<b>Running the INTU application.....</b>	<b>17</b>
5.1	INTU Command Line Arguments .....	17
5.1.1	INTU Display Options.....	18
<b>6</b>	<b>References .....</b>	<b>19</b>
<b>7</b>	<b>Abbreviations.....</b>	<b>20</b>
	<b>Appendix A - Example Configuration Files using SS7HD and SS7G21SIU.....</b>	<b>21</b>
	<b>Appendix B - Example Configuration Files using SIGTRAN M2PA .....</b>	<b>32</b>
	<b>Appendix C - INAP User - point code 2 .....</b>	<b>39</b>
	<b>Appendix D - Sample Output.....</b>	<b>44</b>
	<b>Appendix E - SDL Diagrams .....</b>	<b>45</b>

## 1 Introduction

The INTU application is suited to work with the Dialogic<sup>®</sup> INAP module. This application, in conjunction with the Service Switching Function (SSF), will simulate the generation and acknowledgement of simple INAP messages between the two nodes. This user guide will describe the design, build and usage of this application for developers.

This user guide is for users developing their own applications that will interface with and use the functionality provided by the INAP module.

### 1.1 Software requirements

The INTU application requires the following software:

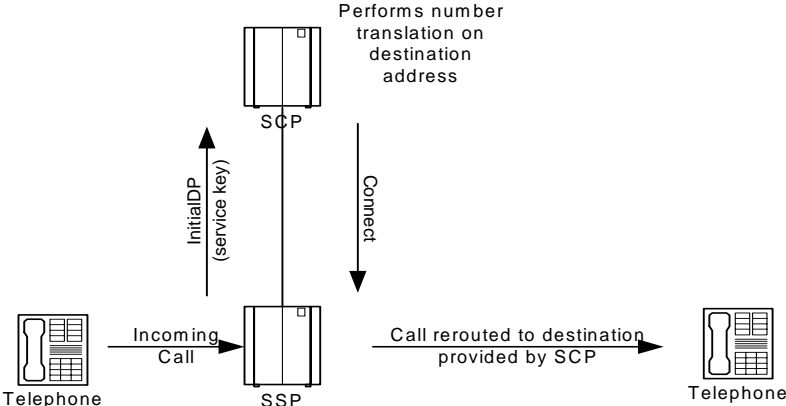
1. Dialogic<sup>®</sup> SS7Development Package
2. Dialogic<sup>®</sup> User Part Development Package
3. For TDM-based configurations:
  - ss7.dc3 or ss7.dc4 codefile
  - Dialogic<sup>®</sup> MTP3, SCCP, TCAP, INAP host binaries, as required
4. For SIGTRAN-based configurations:
  - Dialogic<sup>®</sup> M2PA, MTP3, SCCP, TCAP, INAP host binaries, as required

Software can be downloaded from

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

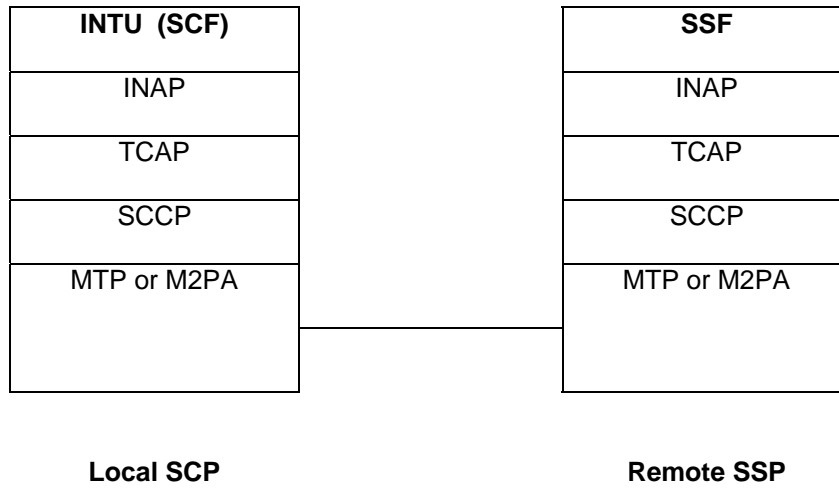
## 2 INTU Application

The simple service implemented by INTU is a number translation from a fixed calling party number to a fixed destination routing address. The translation is triggered by the reception of a pre-defined service key contained in the InitialDP operation received from the SSP.



**Figure 1.** InitialDP Operation

INTU offers a simple SCF service implementation, which sits above the INAP layer in the SS7 stack. Figure 2 below shows INTU at the local node behaving as an SCP connected to a remote SSP.



**Figure 2.** INTU behaving as a SCF

### 2.1 SSF

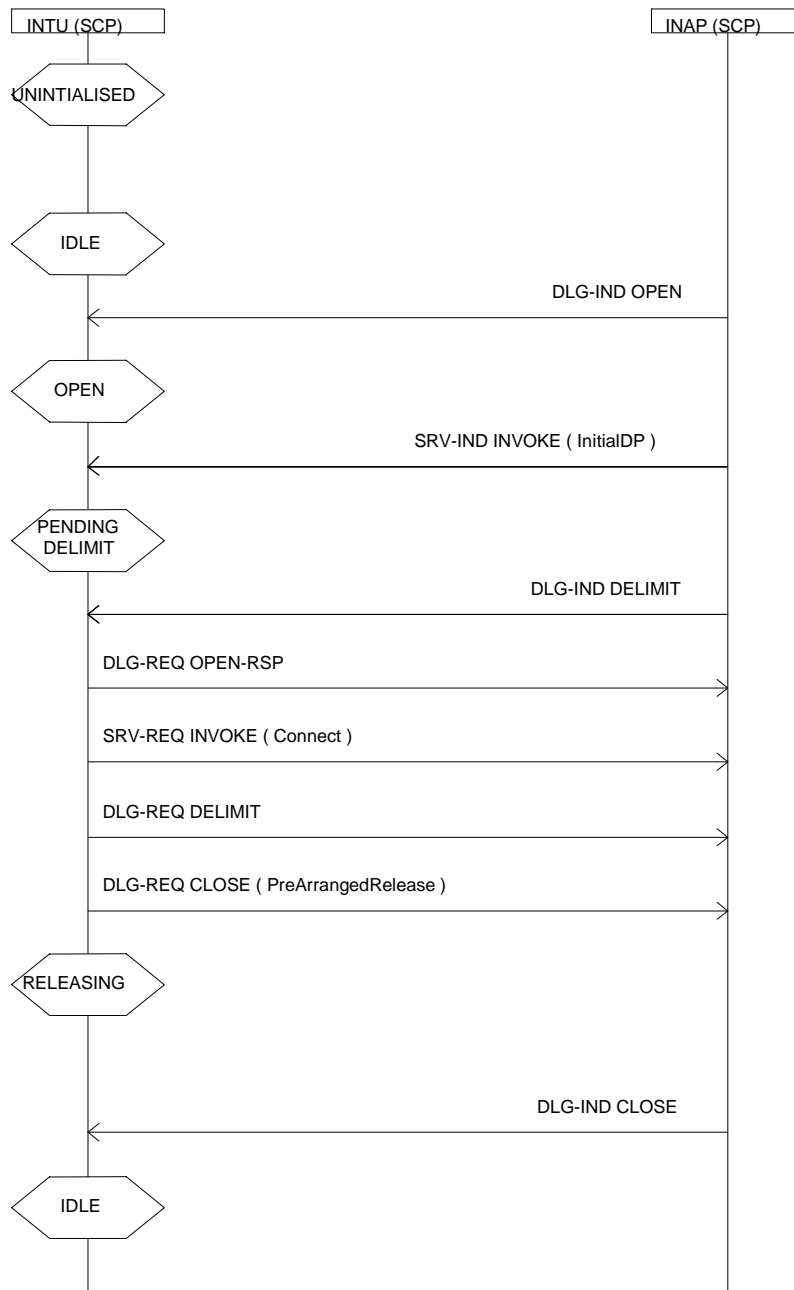
For the purpose of this user guide, SSF is simply a file whose contents are sent to INTU using s7\_play.

### 2.2 Message Sequence Chart

The following message sequence charts show three call flows through INTU. The charts show the messages flowing between INTU and its local INAP.

### 2.2.1 Successful Call – Connect Operation

This is the normal success path for the call. The InitialDP operation is received correctly and a Connect containing the destination address is sent back to the SSP enabling the call to be rerouted to that address.

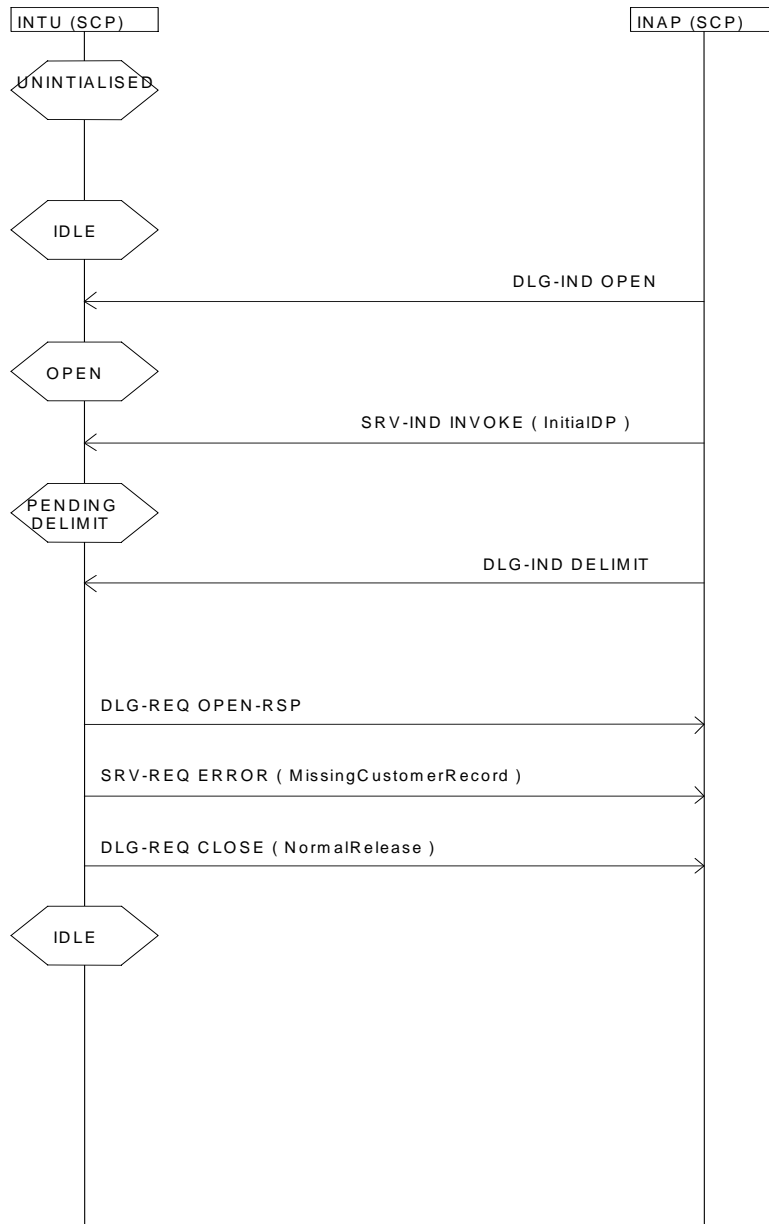


**Figure 3.** Successful Call - Connect Operation



### 2.2.2 Invalid Service Key – User Error

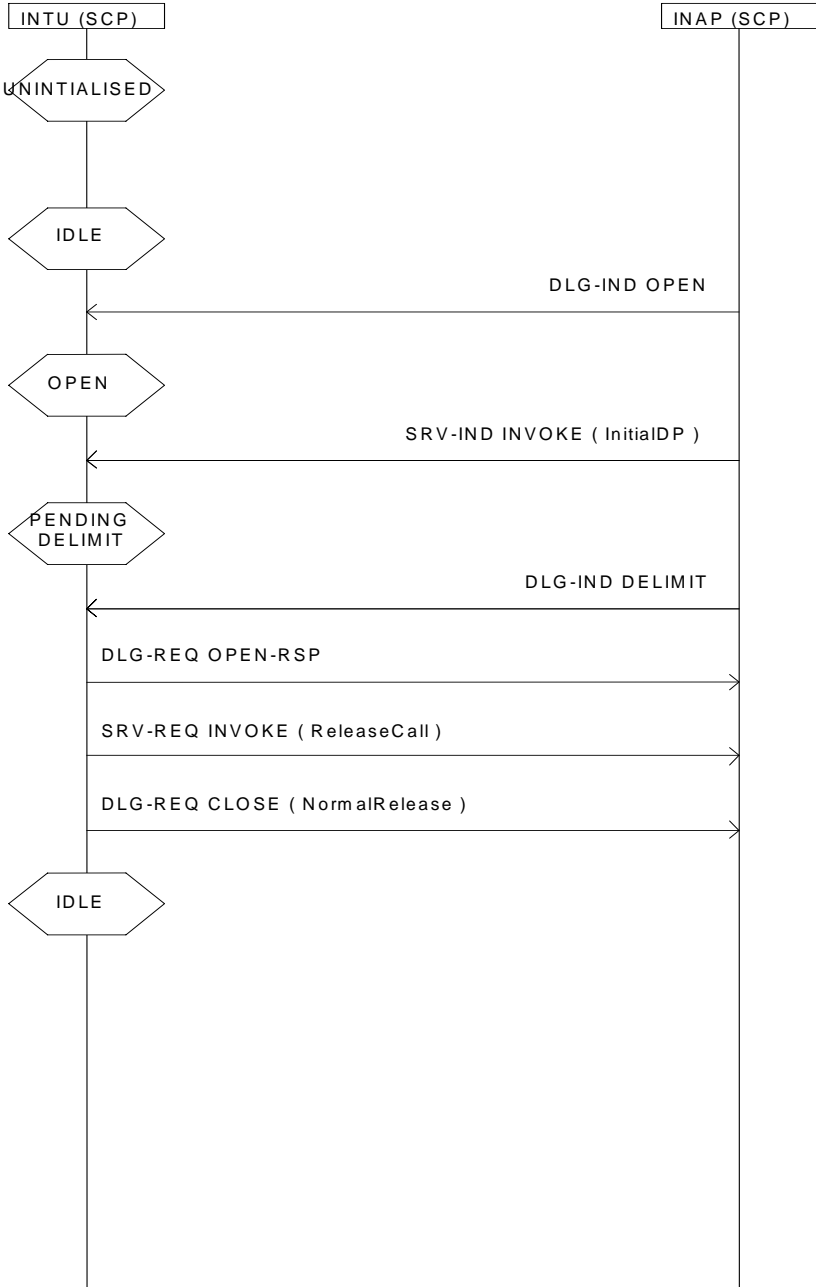
In the case of the reception of an invalid service key, an error is returned to the SSP that then releases the call.



**Figure 4.** Invalid Service Key – User Error

### 2.2.3 Invalid Parameter - ReleaseCall Operation

In this case, the InitialDP is received with the correct service key but the called party number is incorrect so a ReleaseCall operation is returned to the SCP to direct it to release the call.



**Figure 5.** Invalid Parameter - ReleaseCall

## 2.3 Customizing the example applications

INTU is example code for development using the INAP module and API. As such, a number of simplifications and limitations are implemented.

The service supported by INTU is a simple number translation from a single fixed number to a single destination routing address. The called party number, destination routing address and expected service key all have their values fixed at compile time. These may need to be changed before building the application for any particular network.

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

## 2.4 INTU source code

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

File	Notes
intu.c	The main loop for INTU is in <b>intu_ent()</b> . It waits for a message to be received. If it is a dialogue, <b>INTU_dlg_ind()</b> is called, but if the message is a invoke then <b>INTU_srv_ind()</b> is called. When an invoke has been received, the number translation is performed. INTU responds with the appropriate message sequence (section 2.2.1).
intu_trc.c	Display procedures for tracing INTU.
intu_sys.c	Message sending and formatting procedures for INTU.
intumain.c	This file contains the <b>main()</b> function. This reads the command line arguments and passes them to <b>intu_ent()</b> .
intu_def.h	Contains function prototypes and #defines used by INTU.

## 3 Building the INTU 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 the Windows® operating system, 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.

The INAP API library (the dynamically linked GCT library or the GCT shared object) should be copied to the *Dialogic® SS7 Development Package* directory.

For UNIX, re-configure the dynamic linker's run time configuration by running the following command (as root):

```
ldconfig -v
```

A series of symbolic links to the INAP API shared object must also be created in the *Dialogic® SS7 Development Package* directory

e.g. ln -s libin_api.so.1.0.0 libin_api.so.1
ln -s libin_api.so.1.0.0 libin_api.so

### 3.1 Host software directory structure

To build the INTU 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 INTU 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 INTU program.

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

Root directory			
Septel			
INC	UPD		
	BIN	SRC	
asciibin.h bit2byte.h in_inc.h in_ain_inc.h inap_inc.h msg.h pack.h ss7_inc.h strtonum.h sysgct.h system.h	<b>BACKUP_WIN</b>  <b>BACKUP_LNX</b>  <b>BACKUP_SOL</b>	<b>INTU</b>  intu.mnt intu.mak intu_sol.mak intu_iss.txt intu_def.h intu.c intumain.c intu_sys.c intu_trc.c	makdefs.mnt makdefs_sol.mak makdefs.mak makeall.bat makeall makeall_sol

### 3.2 Building INTU

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 INTU application, for each operating system, can be located within the backup subdirectories in the BIN directory.

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

```
make -f intu.mak
```

```
make -f intu_sol.mak
```

```
nmake /f intu.mnt
```

## 4 Configuration

The local and remote ends of the system need to be configured before the INTU 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:

<b>Root directory</b>	
<b>RUN</b>	
<b>INTU</b>	
<b>“CONFIG1”</b> config.txt system.txt [uis.ms7]	<b>“CONFIG2”</b> config.txt system.txt [uis.ms7] [idp.ms7]

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 - Example Configuration Files for further information.

### 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 [3].

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

Example configuration files for INTU are contained in Appendix A - Example Configuration Files using SS7HD and SS7G21SIU.

#### 4.1.3 SIGTRAN M2PA

It is also possible to run the INTU application from 2 hosts connected in back-to-back with SIGTRAN M2PA links. Configuration files for such systems are contained in Appendix B - Example Configuration Files using SIGTRAN M2PA. When running INTU on a Windows® host system using Dialogic® M2PA, MTP3, SCCP, TCAP and INAP host binaries 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, note the following sections.

Before configuring the protocol modules, 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 option) are contained in the main SCCP configuration message and should be set to the appropriate value as described in the programmer's manual [1]. In addition, configuration messages 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 option of the configuration message. The dialogue id ranges should be set to allow the appropriate number of ids split between incoming and outgoing dialogues (further details in the programmer's manual [5]). Some applications may require initiation of dialogues in one direction only.

#### 4.2.3 INAP

INAP should be configured for ANSI or ITU-T depending on the configuration of SCCP and TCAP. Also similar to TCAP, the ids should be split for incoming and outgoing dialogues. See programmer's manual for further details [6].



## 5 Running the INTU application

Before running the INTU application, the GCT environment must first be initialized and the signaling links brought into service. This is achieved by running the gctload program, and activating the links using the MTPSL utility. SCCP sub-systems should be brought into service using 'User In Service' messages detailed in Section A.3 Local Sub-system 'User In Service' messages.

Refer to manuals [1], [5] or [6] for details as appropriate.

### 5.1 INTU 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
<b>-m</b>	0x3d	INTU module Id
<b>-l</b>	0x35	INAP module Id
<b>-n</b>	2048	Number of incoming INAP dialogues to use
<b>-b</b>	0	Base incoming dialogue id
<b>-o</b>	0x0300	Display options Add together required values for tracing options given in table below (section 5.1.1). The default option displays state changes and a dialogue status count.
<b>-h, -H -v</b>	-	Displays help message

Example:

```
intu -b0x8000 -o0xffff
```

The above example would assume default values for the INAP module id, INTU module id and the number of incoming dialogues. Also all the trace options would be enabled and the base incoming dialogue id would be 0x8000.

## 5.1.1 INTU Display Options

Tracing option	Value	Notes
INTU_OPT_TR_DLG_REQ	0x0001	Trace transmitted dialogue msgs
INTU_OPT_TR_DLG_IND	0x0002	Trace received dialogue msgs
INTU_OPT_TR_SRV_REQ	0x0004	Trace transmitted service req msgs
INTU_OPT_TR_SRV_IND	0x0008	Trace received service ind msgs
INTU_OPT_TR_DLG_PARAM	0x0010	Include dialogue parameters in trace
INTU_OPT_TR_SRV_PARAM	0x0020	Include service req params in trace
INTU_OPT_TR_STATE	0x0100	Trace state changes
INTU_OPT_TR_ACTV_DLG	0x0200	Trace the dialogue status count
INTU_OPT_TR_NUM_TRANS	0x0400	Show the numbers translated to/from

## 6 References

- [1] U05SSS, Dialogic<sup>®</sup> SCCP Programmer's Manual
- [2] U10SSS, Dialogic<sup>®</sup> Software Environment Programmer's Manual
- [3] 05-2302, Dialogic<sup>®</sup> SS7G2x SIU Mode User Manual
- [4] U03HSP, Dialogic<sup>®</sup> SS7 Programmer's Manual for SPCI2S, SPC14 and CPM8
- [5] U06SSS, Dialogic<sup>®</sup> TCAP Programmer's Manual
- [6] U16SSS, Dialogic<sup>®</sup> INAP Programmer's Manual
- [7] 05-2063, Dialogic<sup>®</sup> SS7HD Programmer's Manual
- [8] U04STN, Dialogic<sup>®</sup> Programmer's Manual for SIGTRAN Host Software

The latest 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
INAP	Intelligent Network Application Part
INTU	INAP Test Utility
ITU-T	International Telecommunications Union
SCCP	Signaling Connection Control Part
SCF	Service Control Function
SCP	Service Control Point
SIU	Signaling Interface Unit
SSF	Service Switching Function
SSP	Switched Signaling Point

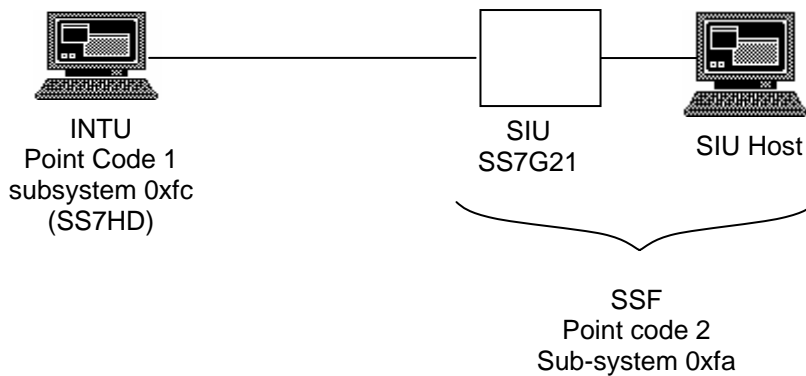
## Appendix A - Example Configuration Files using SS7HD and SS7G21SIU

This section provides example configuration files for use with INTU on Windows<sup>®</sup> host systems for SS7HD and SS7G21 SIU. Configuration of the Dialogic<sup>®</sup> SS7HD board host shows Dialogic<sup>®</sup> SCCP, TCAP and INAP modules are running on the host and Dialogic<sup>®</sup> MTP host binary running on the board. SIU configuration shows all protocol modules are running on the Dialogic<sup>®</sup> SS7G21 SIU. INTU is running as module ID 0x3d.

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 (SSF) point code is 2 and the remote point code is 1. In the provided example configuration files, INTU is running on local point code 1 (0).

### Example configuration

Operating system:	Windows <sup>®</sup>
Local point code (INTU) / SS7 hardware type:	1 / SS7HD (PCI)
Remote point code (SSF) / SS7 hardware type:	2 / SS7G21 SIU
INTU module ID (point code 2):	0x3d
Modules running on board or on host:	SCCP/TCAP/INAP
Modules running SIU or on host	SCCP/TCAP/INAP



**Figure 6.** Example configuration

## A.1 system.txt

This section provides two example system.txt files for a host with a Dialogic® SS7HD (PCI) board, and an SS7G21 SIU host. Both systems are running under Windows®.

The following example system.txt files are for point codes 1 and 2. All comments are denoted by '\*'.

### A.1.1 system.txt for point code 1 (SS7HD PCI)

```
*****
* Example system.txt.
* Edit this file to reflect your configuration.
*****
*
* Essential modules running on host:
*
LOCAL      0x20      * ssd/ssds - Board interface task
LOCAL      0x00      * tim_nt - Timer task
LOCAL      0xef      * s7_log - SS7 message display task
*
* Optional modules running on the host:
*
LOCAL      0xcf      * s7_mgt - Management/config task
LOCAL      0x3d      * INTU module
LOCAL      0x35      * INAP module
LOCAL      0x14      * TCAP module
LOCAL      0x33      * SCCP module
*
REDIRECT   0x81      0x20      * MTP2 for SP 0 (SS7HD boards only)
REDIRECT   0x91      0x20      * MTP2 for SP 1 (SS7HD boards only)
REDIRECT   0xe1      0x20      * MTP2 for SP 2 (SS7HD boards only)
REDIRECT   0xf1      0x20      * MTP2 for SP 3 (SS7HD boards only)
REDIRECT   0x22      0x20      * MTP3 module
REDIRECT   0x10      0x20      * CT bus/Clocking control module
REDIRECT   0x8e      0x20      * On-board management module
*
* Redirection of status indications:
*
REDIRECT   0xdf      0xef      * LIU/MTP2 status messages -> s7_log
*REDIRECT  0xef      0x2d      * Other indications -> upe
*
* Now start-up all local tasks:
* (for Septel cP / PCI start-up ssds)
*
FORK_PROCESS ..\..\..\..\ssdh.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 ..\..\..\..\inap_nt.exe -t
FORK_PROCESS ..\..\..\..\tcp_nt.exe -t
FORK_PROCESS ..\..\..\..\sccp.exe -t
```

## A.1.2 system.txt for point code 2 (SS7G21 SIU)

```

*****
* Example system.txt.
* Edit this file to reflect your configuration.
*****
* Module Id's running locally on the host machine:
*
LOCAL          0x00          * tim_xxx - Timer task
LOCAL          0xb0          * rsi Module Id
LOCAL          0xef          * REM_API_ID Module Id (s7_log)
LOCAL          0xfd          * rsicmd Module Id
*LOCAL        0x14          * TCAP
*LOCAL        0x33          * SCCP
*LOCAL        0x35          * INAP
*
* Redirect modules running on the SIU to RSI:
*
REDIRECT       0xdf         0xb0    * SIU_MGT module Id
REDIRECT       0x22         0xb0    * MTP3 module Id
REDIRECT       0x14         0xb0    * TCAP module Id
REDIRECT       0x33         0xb0    * SCCP module Id
REDIRECT       0x32         0xb0    * RMM module Id
REDIRECT       0x71         0xb0    * MTP2 module Id
REDIRECT       0x35         0xb0    * INAP messages
REDIRECT       0x4d         0x35    * redirect incoming messages to INAP
*
* Now start-up the Host tasks ....
*
FORK_PROCESS   tim_nt.exe
FORK_PROCESS   tick_nt.exe
FORK_PROCESS   s7_log.exe -fs7log.lst
FORK_PROCESS   rsi.exe -r./rsi_lnk.exe -ll
*FORK_PROCESS   inap_nt.exe -t
*FORK_PROCESS   tcp_nt.exe -t
*FORK_PROCESS   sccp_nt.exe -t
*
* Start the Host-SIU link:
*
FORK_PROCESS   rsicmd.exe 0 0xef 0 192.168.0.1 9000

```

## A.1.3 Using different operating systems and board configurations

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

### A.1.3.1 Running INTU with SPCI2S or SPC14

If using SPCI2S or SPC14 the following lines:

```
REDIRECT      0x81    0x20    * MTP2 for SP 0 (SS7HD boards only)
REDIRECT      0x91    0x20    * MTP2 for SP 1 (SS7HD boards only)
REDIRECT      0xe1    0x20    * MTP2 for SP 2 (SS7HD boards only)
REDIRECT      0xf1    0x20    * MTP2 for SP 3 (SS7HD boards only)
```

```
FORK_PROCESS  SSDH.EXE -d
```

should be replaced by:

```
REDIRECT      0x71    0x20    * MTP2 module
```

```
FORK_PROCESS  SSDS.EXE -d
```

Refer to [3] for further information.

### A.1.3.2 Running INTU 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.



## A.2 config.txt

This section provides two example config.txt files for a host with a Dialogic<sup>®</sup> SS7HD (PCI) board, and a Dialogic<sup>®</sup> SS7G21 SIU. Both systems (SS7HD host and SIU host) are running under Windows<sup>®</sup>.

All comments are denoted by '\*'.

Use of the following two example config.txt files (one for the SS7HD host and one for the SS7G21 SIU) will allow the INAP / CAMEL transaction handling system to be configured as detailed in figure 6.

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:

- Hardware type, e.g. which board type? Which SIU?
- point codes (OPC, DPC)
- signaling timeslot

## A.2.1 config.txt for point code 1 (SS7HD PCI)

```
*****
* Example config.txt.
* Edit this file to reflect your configuration.
*****
*
* For SS7HD PCI boards:
* SS7_BOARD <board_id> <board_type> <flags> <code_file> <run_mode>
*
SS7_BOARD 0 SS7HDP 0x0043 ../DC4/ss7.dc4 mtp
*
* 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
LIU_CONFIG 0 1 5 1 1 1
LIU_CONFIG 0 2 5 1 1 1
LIU_CONFIG 0 3 5 1 1 1
*
* MTP parameters:
*
* MTP_CONFIG <reserved> <reserved> <options>
MTP_CONFIG 0 0 0x0000
*
* 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>
* Note 1: For PCCS6 boards the first LIU port is stream=16 whilst for other
* boards the first LIU port is stream=0.
* Note 2: The SS7HD board requires a compound parameter for blink of the
* form sp_id-sp_channel.
*
* For SS7HD boards:
*
MTP_LINK 0 0 0 0 0 0-0 2 16 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_pc> <ssf> <options>
*
SCCP_CONFIG 1 0x08 0x322
*
*SCCP_SSR [<nc_id>] <ssr_id> RSP <remote_spc> <rsp_flags> [<pc_mask>]
*
SCCP_SSR 1 RSP 2 0 0
*
*SCCP_SSR [<nc_id>] <ssr_id> LSS <local_ssn> <module_id> <lss_flags>
<protocol>
*
SCCP_SSR 2 LSS 0xfc 0x3d 0 INAP
*
*SCCP_SSR [<nc_id>] <ssr_id> RSS <remote_spc> <remote_ssn> <rss_flags>
*
SCCP_SSR 3 RSS 2 0xfa 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
*
* INAP_CONFIG <options>
*
INAP_CONFIG 0
*
*     INAP_FE <fe_ref> <options> <sccp_address>
*
*
INAP_FE 0x00000001 0x00000000 0x4302000a
INAP_FE 0x00000002 0x00000000 0x4302000b
INAP_FE 0x00000003 0x00000001 0x4301000c
INAP_FE 0x00000004 0x00000001 0x4301000d
*
* INAP_AC <ac_ref> <ac>
*
INAP_AC 0x00 0xa109060704000101010000
INAP_AC 0x01 0xa109060704000101010100
INAP_AC 0x02 0xa109060704000101010200
INAP_AC 0x03 0xa109060704000101010300
INAP_AC 0x04 0xa109060704000101010400
INAP_AC 0x05 0xa109060704000101010500
INAP_AC 0x06 0xa109060704000101010600
INAP_AC 0x10 0xa109060700118942010000
INAP_AC 0x20 0xa109060704000001013200
INAP_AC 0x21 0xa109060704000001013201
INAP_AC 0x34 0xa109060704000101140304
*
```

## A.2.2 config.txt for point code 2 (SS7G21 SIU)

```
* SS7G2x Protocol Configuration File (config.txt)
* Define the network address of the partner SIU (dual operation only) :
* SIU_REM_ADDR <remote_address>
*
*SIU_REM_ADDR xxx.xxx.xxx.xxx
*
* Define the number of hosts that this SIU will connect to :
* SIU_HOSTS <num_hosts>
*
SIU_HOSTS 1
*
* Set physical Interface Parameters :
* SS7_BOARD <bpos> <board_type> <flags>
*
SS7_BOARD 1 SPCI4 0x0041
*
* LIU_CONFIG <port_id> <pcm> <liu_type> <line_code> <frame_format>
<crc_mode>
*
LIU_CONFIG 0 1-1 5 1 1 1 1
*LIU_CONFIG 1 1-2 5 1 1 1 1
*LIU_CONFIG 2 1-3 5 1 1 1 1
*LIU_CONFIG 3 1-4 5 1 1 1 1
*
* MTP Parameters :
* MTP_CONFIG <local_spc> <ssf> <options>
MTP_CONFIG 0 0 0x0000
*
* Define linksets :
* MTP_LINKSET <linkset_id> <adjacent_spc> <num_links> <flags> <local_spc>
<ssf>
*
MTP_LINKSET 0 1 1 0x0000 2 0x8
*MTP_LINKSET 1 1 1 0x8000 1 0x8 * inter-siu
*
* Define signalling links :
* MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <bpos> <blink>
<stream> <timeslot> <flags>
*
MTP_LINK 0 0 0 0 1 0 0 0x10 0x0006
*
* Define a route for each remote signalling point :
* MTP_ROUTE [<nc_id>] <route_id> <dpc> <linkset_id> <user_part_mask>
<flags>
* <second_ls> <reserved>
*
MTP_ROUTE 1 1 0 0x0008 0x0000 0 0
*
* SCCP Parameters :
* SCCP_CONFIG <local_pc> <ssf> <options>
*
SCCP_CONFIG 2 0x08 0x0322
*SCCP_RSP 200 0
*sccp_LSS 8 0x0d 1 tcap
*sccp_rss 200 8 0
*SCCP_SSR [<nc_id>] <ssr_id> RSP <remote_spc> <rsp_flags> [<pc_mask>]
*
SCCP_SSR 1 RSP 1 0
*
*SCCP_SSR [<nc_id>] <ssr_id> LSS <local_ssn> <module_id> <lss_flags>
<protocol>
*
```

```
SCCP_SSR 2 LSS 0xfa 0xef 0 INAP
*
*SCCP_SSR [<nc_id>] <ssr_id> RSS <remote_spc> <remote_ssn> <rss_flags>
*
SCCP_SSR 3 RSS 1 0xfc 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
*
INAP_CONFIG 0
*
*       INAP_FE <fe_ref> <options> <sccp_address>
*
*
INAP_FE 0x00000001 0x00000001 0x4302000a
INAP_FE 0x00000002 0x00000001 0x4302000b
INAP_FE 0x00000003 0x00000000 0x4301000c
INAP_FE 0x00000004 0x00000000 0x4301000d
*
* INAP_AC <ac_ref> <ac>
*
INAP_AC 0x00 0xa109060704000101010000
INAP_AC 0x01 0xa109060704000101010100
INAP_AC 0x02 0xa109060704000101010200
INAP_AC 0x03 0xa109060704000101010300
INAP_AC 0x04 0xa109060704000101010400
INAP_AC 0x05 0xa109060704000101010500
INAP_AC 0x06 0xa109060704000101010600
INAP_AC 0x10 0xa109060700118942010000
INAP_AC 0x20 0xa109060704000001013200
INAP_AC 0x21 0xa109060704000001013201
INAP_AC 0x34 0xa109060704000101140304
*
INAP_TRACE 0xf 0xf 0x7f
*
*
* End of file
```

## A.2.3 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.2.3.1 Running INTU with PCI2S or SPC14

If using Dialogic® SPC12S or SPC14S PCI boards, the 'SS7\_BOARD 0 SS7HDP' command (from config.txt for point code 1) should be replaced with the following:

```
SEPTELPCI_BOARD 0 0x0043 ss7.dc3 MTP  
  
LIU_CONFIG 0 0 5 1 1 1
```

The MTP\_LINK 'blink' parameter will need to be modified for use with SPC12S or SPC14S.

Refer to [3] for further information.

### A.2.3.2 Running INTU with the SIU

If using the SIU, additional commands required by the SIU will need to be included. The example system.txt provided in this appendix is for information. Refer to [3] for further information.

### A.2.3.3 Running INAP/TCAP/SCCP on the board

If using a board license button so that the INAP, TCAP, SCCP and MTP3 modules are run on the board instead of the host, remove the FORK\_PROCESS command for the INAP host binary, the LOCAL declaration should be removed for the INAP 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 [4] or [7] and as appropriate.

### A.2.3.4 Running INTU with other operating systems

There are no additional commands specific to various operating systems.

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

### A.3 Local Sub-system 'User In Service' messages

This section provides two 'User In Service' (UIS) messages, one for INTU and another for SSF. These messages bring the local SCCP sub-systems into service, enabling the complete system to begin handling INAP / CAMEL traffic.

For this example INTU is point code 1 and SSF is point code 2 (14-bit point codes assumed). The local sub-system at point code 1 is 0xfc and the sub-system at point code 2 is 0xfa.

#### A.3.1 INTU UIS message – point code 1

```
*                               --Format id - Sub-system Allowed SSA (1)
*                               --Primitive type - N-STATE-REQ (1)
*                               -- Local sub-system number
*
M-tc744-i00fc-fef-d33-r8000-p0101000000000000
*
```

#### A.3.2 SSF UIS message – point code 2

```
*                               --Format id - Sub-system Allowed SSA (1)
*                               --Primitive type - N-STATE-REQ (1)
*                               -- Local sub-system number
*
M-tc744-i00fa-fef-d33-r8000-p0101000000000000
*
```

## Appendix B - Example Configuration Files using SIGTRAN M2PA

This section provides example configuration files for use with INTU on Windows® host systems using SIGTRAN M2PA links. Dialogic® INAP, TCAP, SCCP, MTP3 and M2PA modules are running on the hosts. The INTU sample program is running as module ID 0x3d.

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 (SSF) point code is 2 and the remote point code (running INTU) is 1 as shown in figure 7.

### Example configuration

Operating system:	Windows®
Link type:	SIGTRAN M2PA
Remote point code (INTU)	1
Local point code (SSF)	2
INTU module ID (point code 2):	0x3d
Modules running on host :	INAP, TCAP, SCCP, MTP3, M2PA



**Figure 7.** Example configuration



## B.1 system.txt

This section provides an 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 file is valid for SIGTRAN hosts acting as point codes 1 and 2. All comments are denoted by '\*'.

```
*****
* Example system.txt.
* Edit this file to reflect your configuration.
*****
*
LOCAL          0x00          * Timer Task
*
* Optional modules running on the host:
*
LOCAL          0xcf          * s7_mgt - Management/config task
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          0x35          * INAP module*
LOCAL          0xEF          * S7_LOG
LOCAL          0x3d          * intu
* 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   ..\..\..\..\inap_nt.exe -t
FORK_PROCESS   ..\..\..\..\tcp_nt.exe -t
FORK_PROCESS   ..\..\..\..\sccp.exe -t
FORK_PROCESS   ..\..\..\..\s7_mgt.exe -d
FORK_PROCESS   ..\..\..\..\s7_log.exe -fss7.log
```

## 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 will allow a INAP / SIGTRAN transaction handling system to be configured as detailed in figure 7.

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
- Host IP addresses

Refer to [3] for SIGTRAN related parameter changes (such as port numbers, IP addresses, etc.)

### B.2.1 config.txt for point code 1

```
*****
* 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=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 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_CONFIG 1 0x08 0x322
*
*SCCP_SSR [<nc_id>] <ssr_id> RSP <remote_spc> <rsp_flags> [<pc_mask>]
*
SCCP_SSR 1 RSP 2 0 0
*
*SCCP_SSR [<nc_id>] <ssr_id> LSS <local_ssn> <module_id> <lss_flags>
<protocol>
*
SCCP_SSR 2 LSS 0xfc 0x3d 0 INAP
*
*SCCP_SSR [<nc_id>] <ssr_id> RSS <remote_spc> <remote_ssn> <rss_flags>
*
SCCP_SSR 3 RSS 2 0xfa 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
*
* INAP_CONFIG <options>
*
INAP_CONFIG 0
*
*     INAP_FE <fe_ref> <options> <sccp_address>
*
*
INAP_FE 0x00000001 0x00000000 0x4302000a
INAP_FE 0x00000002 0x00000000 0x4302000b
INAP_FE 0x00000003 0x00000001 0x4301000c
INAP_FE 0x00000004 0x00000001 0x4301000d
*
* INAP_AC <ac_ref> <ac>
*
INAP_AC 0x00 0xa109060704000101010000
INAP_AC 0x01 0xa109060704000101010100
INAP_AC 0x02 0xa109060704000101010200
INAP_AC 0x03 0xa109060704000101010300
INAP_AC 0x04 0xa109060704000101010400
INAP_AC 0x05 0xa109060704000101010500
INAP_AC 0x06 0xa109060704000101010600
INAP_AC 0x10 0xa109060700118942010000
INAP_AC 0x20 0xa109060704000001013200
INAP_AC 0x21 0xa109060704000001013201
INAP_AC 0x34 0xa109060704000101140304
*
* End of file
```

## B.2.2 config.txt for point code 2

```
*****
* 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=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 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_CONFIG          2          0x08          0x0322
*
*SCCP_SSR [<nc_id>] <ssr_id> RSP <remote_spc> <rsp_flags> [<pc_mask>]
*
SCCP_SSR 1 RSP 1 0 0
*
*SCCP_SSR [<nc_id>] <ssr_id> LSS <local_ssn> <module_id> <lss_flags>
<protocol>
*
SCCP_SSR 2 LSS 0xfa 0xef 0 INAP
*
*SCCP_SSR [<nc_id>] <ssr_id> RSS <remote_spc> <remote_ssn> <rss_flags>
*
SCCP_SSR 3 RSS 1 0xfc 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
*
INAP_CONFIG 0
*
*   INAP_FE <fe_ref> <options> <sccp_address>
*
*
INAP_FE 0x00000001 0x00000001 0x430200a
INAP_FE 0x00000002 0x00000001 0x430200b
INAP_FE 0x00000003 0x00000000 0x430100c
INAP_FE 0x00000004 0x00000000 0x430100d
*
* INAP_AC <ac_ref> <ac>
```

```
*
INAP_AC 0x00 0xa109060704000101010000
INAP_AC 0x01 0xa109060704000101010100
INAP_AC 0x02 0xa109060704000101010200
INAP_AC 0x03 0xa109060704000101010300
INAP_AC 0x04 0xa109060704000101010400
INAP_AC 0x05 0xa109060704000101010500
INAP_AC 0x06 0xa109060704000101010600
INAP_AC 0x10 0xa109060700118942010000
INAP_AC 0x20 0xa109060704000001013200
INAP_AC 0x21 0xa109060704000001013201
INAP_AC 0x34 0xa109060704000101140304
*
INAP_TRACE 0xf 0xf 0x7f
*
* End of file
```

## B.3 Local Sub-system 'User In Service' messages

This section provides two 'User In Service' (UIS) messages, one for INTU and another for SSF. These messages bring the local SCCP sub-systems into service, enabling the complete system to begin handling INAP / CAMEL traffic.

For this example, INTU is point code 1 and SSF is point code 2 (14-bit point codes assumed). The local subsystem at point code 1 is 0xfc and at point code 2 is 0xfa.

### B.3.1 INTU UIS message – point code 1

```
*           --Format id - Sub-system Allowed SSA (1)
*           --Primitive type - N-STATE-REQ (1)
*           -- Local sub-system number
*
M-tc744-i00fc-fef-d33-r8000-p0101000000000000
*
```

### B.3.2 SSF UIS message – point code 2

```
*           --Format id - Sub-system Allowed SSA (1)
*           --Primitive type - N-STATE-REQ (1)
*           -- Local sub-system number
*
M-tc744-i00fa-fef-d33-r8000-p0101000000000000
*
```

## Appendix C - INAP User - point code 2

This section provides an example script file, which may be used to represent the INAP user (SSF) at point code 2. The file can be used with `s7_play` and is supplied in the *Dialogic® User Part Development Package*. The 3 dialogues started by this file represent the message sequences shown in section 2.2.

**Please note: The final 3 dialogue sequences in this example script file are commented out.**

These can optionally be uncommented to show dialogue handling for ITU-T CS-1, CAMEL v1 and CAMEL v2 application contexts.

### C.1 idp.ms7

```

*
* This file contains 3 ETSI dialogues which will reflect the message sequences shown in the
INTU
* User Guide. The first dialogue is a successful dialogue. The second dialogue shows an
error.
* The last dialogue will cause a different message to be sent from INTU due to a deliberate
mismatch
* in the InitialDP message.
*
* Each dialogue sends an open request, an InitialDP operation, a delimit and
* finally a close with a pre-arranged end.
*
*
* ETSI CS-1
*
* This is an example of a successful dialogue as mentioned in the INTU user guide
*
*
* -----Orig Address
* -----Dest Address
* --Open Req -----AC ref
M-tc7f2-i0000-fef-d35-p010104430100fc0304430200falb010000
*
* InitialDP with optional parameters, parameter data is dummy
M-tc7f0-i0000-fef-d35-
p011001011501001301021402000a1225a123020101020100301b80040a0b0c0d820701108000214365830a0104020
50605050a050a00
* --invokeId param --component name --
context prim 2 tag
* InvokeREQ-- ^^len ^^len ^^len
* ==InvokeID --component type tag
=====CalledPartyNumber
* --Opcode Name ^^len
--context prim 3 tag
* ^^len --invoke id tag
^^len
* ==opCode ^^len
=====CallingPartyNumber
* --class param --invoke id
* ^^len ==op code tag
* ==class ^^len
* --timeout name --op code
* ^^len ==sequence
* ==timeout ^^len
tag --context prim 0
*
* ^^len
* ==service
key
*
*
* --delimit

```

```

M-tc7f2-i0000-fef-d35-p0300
*
*
D-m1000
*
*           -----pre-arranged
*           --close
M-tc7f2-i0000-fef-d35-p0207010100
*
D-m1000
* ETSI CS-1
*
* This dialogue is an example of a a dialogue which will result in an error. The error will
* be
* MissingCustomerRecord and is generated because the service key provided in the InitialDP
* message
* will not match the service key provided inside INTU.
*
*
*           -----Orig Address
*           -----Dest Address
*           --Open Req           -----AC ref
M-tc7f2-i0001-fef-d35-p010104430100fc0304430200falb010000
*
*
* InitialDP with optional parameters, parameter data is dummy
M-tc7f0-i0001-fef-d35-
p011001011501001301021402000a1225a123020101020100301b80040a0b0c0e820701108000214365830a0104020
50605050a0050a00
*
*           --invokeId param           --component name           --
context prim 2 tag
*           InvokeREQ--  ^^len           ^^len           ^^len
*           ==InvokeID           --component type tag
=====CalledPartyNumber
*           --Opcode Name           ^^len
--context prim 3 tag
*           ^^len           --invoke id tag
^^len
*           ==opCode           ^^len
=====CallingPartyNumber
*           --class param           --invoke id
*           ^^len           ==op code tag
*           ==class           ^^len
*           --timeout name           --op code
*           ^^len           ==sequence
*           ====timeout           ^^len
tag           --context prim 0
*
*           ^^len
*           =====service
key--bad
*
*
*           --delimit
M-tc7f2-i0001-fef-d35-p0300
*
*
D-m1000
*
*           -----pre-arranged
*           --close
M-tc7f2-i0001-fef-d35-p0207010100
*
D-m1000
* ETSI CS-1
*
* This dialogue will result in a ReleaseCall being generated by INTU due to the
* calledpartynumber
* provided in the InitialDP message not matching the 0800 number in INTU which is meant to be
* translated.
*
*
*           -----Orig Address
*           -----Dest Address
*           --Open Req           -----AC ref
M-tc7f2-i0002-fef-d35-p010104430100fc0304430200falb010000
*
*
* InitialDP with optional parameters, parameter data is dummy

```



```

M-tc7f0-i0002-fef-d35-
p011001011501001301021402000a1225a123020101020100301b80040a0b0c0d820701108999214365830a0104020
50605050a050a00
*                               --invokeId param                --component name                --
context prim 2 tag
*           InvokeREQ--  ^^len                ^^len                ^^len
*                               ==InvokeID                --component type tag
* =====CalledPartyNumber--mismatch
*                               --Opcode Name                ^^len
--context prim 3 tag
*                               ^^len                --invoke id tag
^^len
*                               ==opCode                ^^len
* =====CallingPartyNumber
*                               --class param                --invoke id
*                               ^^len                ==op code tag
*                               ==class                ^^len
*                               --timeout name                --op code
*                               ^^len                ==sequence
*                               ====timeout                ^^len
*                               --context prim 0
tag
*                               ^^len
*                               =====service
key
*
*
*                               --delimit
M-tc7f2-i0002-fef-d35-p0300
*
*
D-m1000
*
*                               -----pre-arranged
*                               --close
M-tc7f2-i0002-fef-d35-p0207010100
*
D-m1000
*
* ITU-T CS-1
*
* Successful dialogue showing that INTU supports the ITU-T application context
*
*
*                               -----Orig Address
*                               -----Dest Address
*                               --Open Req                -----AC ref
*M-tc7f2-i0001-fef-d35-p010104430100fc0304430200fa1b011000
*
*
* InitialDP with optional parameters, parameter data is dummy
*M-tc7f0-i0001-fef-d35-
p011001011501001301021402000a1225a123020101020100301b80040a0b0c0d820701108000214365830a0104020
50605050a050a00
*                               --invokeId param                --component name                --
context prim 2 tag
*           InvokeREQ--  ^^len                ^^len                ^^len
*                               ==InvokeID                --component type tag
* =====CalledPartyNumber
*                               --Opcode Name                ^^len
--context prim 3 tag
*                               ^^len                --invoke id tag
^^len
*                               ==opCode                ^^len
* =====CallingPartyNumber
*                               --class param                --invoke id
*                               ^^len                ==op code tag
*                               ==class                ^^len
*                               --timeout name                --op code
*                               ^^len                ==sequence
*                               ====timeout                ^^len
*                               --context prim 0
tag
*                               ^^len
*                               =====service
key
*
*
*                               --delimit

```

```

*M-tc7f2-i0001-fef-d35-p0300
*
*
*D-m1000
*
*          -----pre-arranged
*          --close
*M-tc7f2-i0001-fef-d35-p0207010100
*
*D-m1000
*
* CAMEL v1
*
* Successful dialogue showing that INTU supports the Camel v1 application context
*
*
*          -----Orig Address
*          -----Dest Address
*          --Open Req          -----AC ref
*M-tc7f2-i0002-fef-d35-p010104430100fc0304430200falb012000
*
*
* InitialDP with optional parameters, parameter data is dummy
*M-tc7f0-i0002-fef-d35-
p011001011501001301021402000a1225a123020101020100301b80040a0b0c0d820701108000214365830a0104020
50605050a050a00
*
*          --invokeId param          --component name          --
context prim 2 tag
*          InvokeREQ--  ^^len          ^^len          ^^len
*          ==InvokeID          --component type tag
=====CalledPartyNumber
*          --Opcode Name          ^^len
--context prim 3 tag
*          ^^len          --invoke id tag
^^len
*          ==opCode          ^^len
=====CallingPartyNumber
*          --class param          --invoke id
*          ^^len          ==op code tag
*          ==class          ^^len
*          --timeout name          --op code
*          ^^len          ==sequence
*          ====timeout          ^^len
*          --context prim 0
tag
*          ^^len
*          =====service
key
*
*
*          --delimit
*M-tc7f2-i0002-fef-d35-p0300
*
*
*D-m1000
*
*          -----pre-arranged
*          --close
*M-tc7f2-i0002-fef-d35-p0207010100
*
*D-m1000
*
* CAMEL v2
*
* Successful dialogue showing that INTU supports the Camel v2 application context
*
*
*          -----Orig Address
*          -----Dest Address
*          --Open Req          -----AC ref
*M-tc7f2-i0003-fef-d35-p010104430100fc0304430200falb012100
*
*
* InitialDP with optional parameters, parameter data is dummy
*M-tc7f0-i0003-fef-d35-
p011001011501001301021402000a1225a123020101020100301b80040a0b0c0d820701108000214365830a0104020
50605050a050a00
*
*          --invokeId param          --component name          --
context prim 2 tag
*          InvokeREQ--  ^^len          ^^len          ^^len

```

```

*                               ==InvokeID                --component type tag
=====CalledPartyNumber
*                               --Opcode Name              ^^len
--context prim 3 tag
*                               ^^len                    --invoke id tag
^^len
*                               ==opCode                ^^len
=====CallingPartyNumber
*                               --class param            --invoke id
*                               ^^len                    ==op code tag
*                               ==class                  ^^len
*                               --timeout name          --op code
*                               ^^len                    ==sequence
*                               ====timeout            ^^len
*                                               --context prim 0
tag
*                                               ^^len
*                                               =====service
key
*
*                               --delimit
*M-tc7f2-i0003-fef-d35-p0300
*
*
*D-m1000
*
*                               -----pre-arranged
*                               --close
*M-tc7f2-i0003-fef-d35-p0207010100
*
*D-m1000
*
* ETSI CS-2
*
* Successful dialogue showing that INTU supports the ETSI CS-2 application context
*
*
*                               -----Orig Address
*                               -----Dest Address
*                               --Open Req              -----AC ref
*M-tc7f2-i0004-fef-d35-p010104430100fc0304430200fa1b013400
*
*
* InitialDP with optional parameters, parameter data is dummy
*M-tc7f0-i0004-fef-d35-
p011001011501001301021402000a1225a123020101020100301b80040a0b0c0d820701108000214365830a0104020
50605050a050a00
*                               --invokeId param        --component name        --
context prim 2 tag
*                               InvokeREQ-- ^^len        ^^len                    ^^len
*                               ==InvokeID            --component type tag
=====CalledPartyNumber
*                               --Opcode Name              ^^len
--context prim 3 tag
*                               ^^len                    --invoke id tag
^^len
*                               ==opCode                ^^len
=====CallingPartyNumber
*                               --class param            --invoke id
*                               ^^len                    ==op code tag
*                               ==class                  ^^len
*                               --timeout name          --op code
*                               ^^len                    ==sequence
*                               ====timeout            ^^len
*                                               --context prim 0
tag
*                                               ^^len
*                                               =====service
key
*
*                               --delimit
*M-tc7f2-i0004-fef-d35-p0300
*
*
*D-m1000
*
*                               -----pre-arranged

```

```
*                               --close
*M-tc7f2-i0004-fef-d35-p0207010100
*
*D-m1000
```

## Appendix D - Sample Output

The following is sample output from INTU at point code 1.

INTU was started using:

```
intu -b0x8000 -o0xffff
```

```
INTU: Example INAP application (C) Dialogic Corporation 1998-2007. All  
Rights Reserved.
```

```
=====
```

```
SS7-INAP-API Version 202.2
```

```
INTU module ID - 0x3d
```

```
INAP module ID - 0x35
```

```
Number of dialogues - 0x0800 (2048)
```

```
Base dialogue ID - 0x8000
```

```
Options set - 0xffff
```

```
INTU: DLG-IND OPEN, dialogue 0x8000
```

```
Param : dest_address          (0x0001) Len : 0x04  
        43 01 00 fc
```

```
Param : orig_address          (0x0003) Len : 0x04  
        43 02 00 fa
```

```
Param : applic_context_index  (0x001b) Len : 0x01  
        00
```

```
INTU: State change for dialogue 0x8000 from IDLE to OPEN
```

```
INTU: Dialogues: Active [1], Completed [0], Successful [0], Failed [0]
```

```
INTU: SRV-IND Invoke - dialogue 0x8000 in OPEN state
```

```
Param : ServiceKey            (0x0003) Len : 0x04  
        0a 0b 0c 0d
```

```
INTU: State change for dialogue 0x8000 from OPEN to PENDING_DELIMIT
```

```
INTU: DLG-IND DELIMIT, dialogue 0x8000
```

```
INTU: DLG-REQ OPEN_RSP, dialogue 0x8000
```

```
Param : Result                (0x0005) Len : 0x01  
        00
```

```
INTU: Called Party [0800123456], Dest Routing Addr [01425651300]
```

```
INTU: SRV-REQ Invoke - dialogue 0x8000 in PENDING_DELIMIT state
```

```
INTU: DLG-REQ DELIMIT, dialogue 0x8000
```

```
INTU: DLG-REQ CLOSE, dialogue 0x8000
```

```
Param : release_method        (0x0007) Len : 0x01  
        01
```

```
INTU: State change for dialogue 0x8000 from PENDING_DELIMIT to CLOSING
```

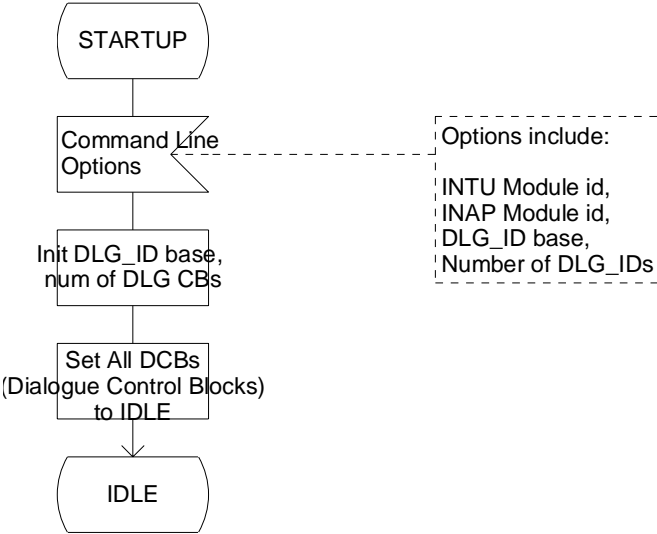
```
INTU: DLG-IND CLOSE, dialogue 0x8000
```

```
INTU: State change for dialogue 0x8000 from CLOSING to IDLE
```

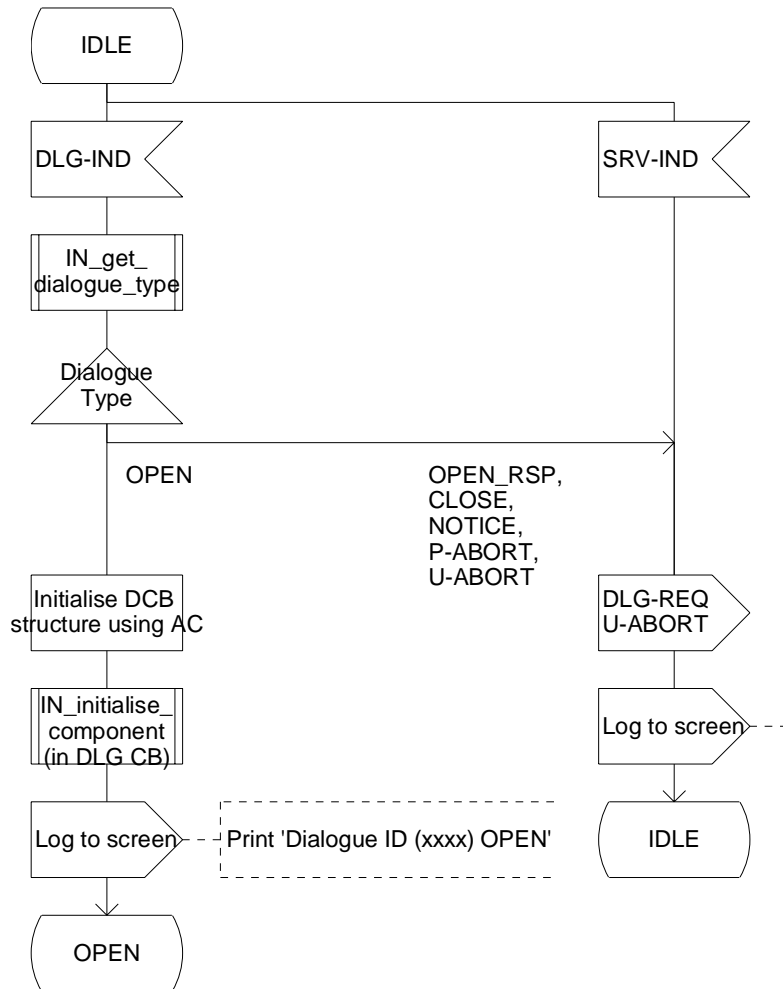
```
INTU: Dialogues: Active [0], Completed [1], Successful [1], Failed [0]
```

**Appendix E - SDL Diagrams**

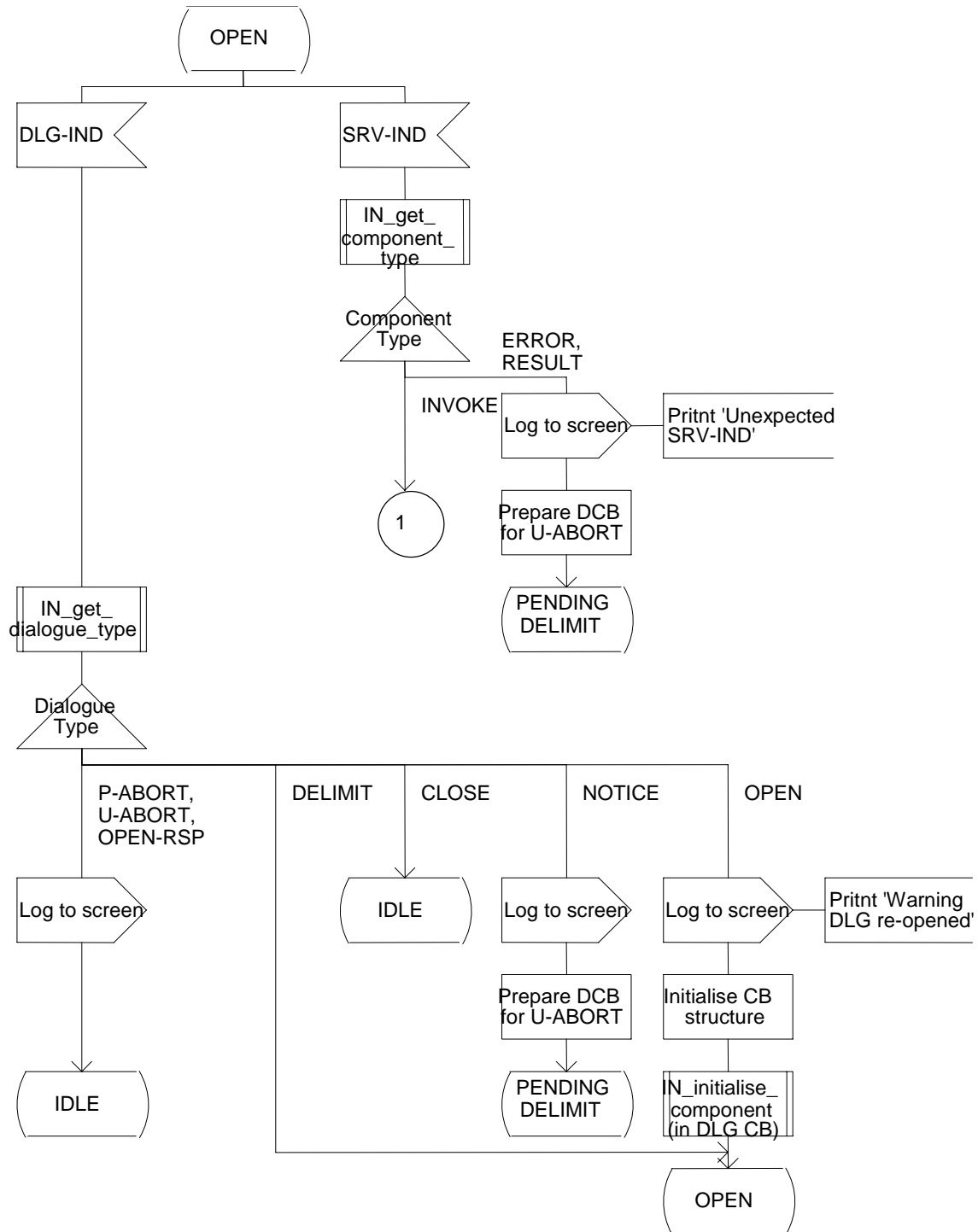
E1



E2

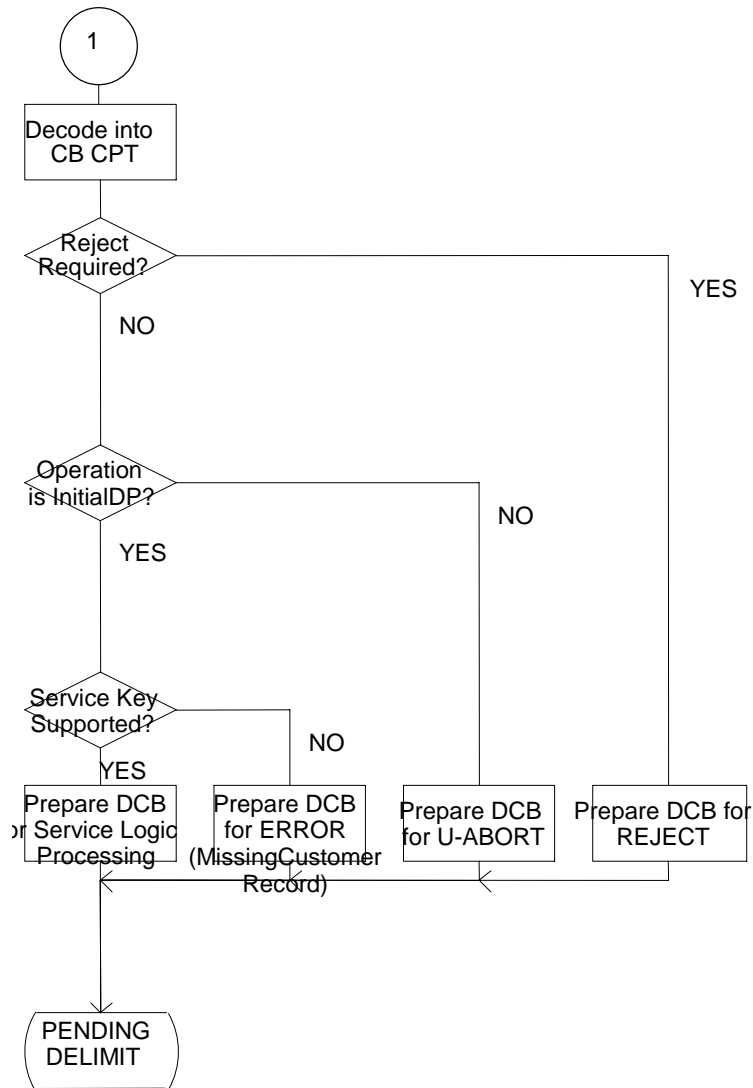


E3

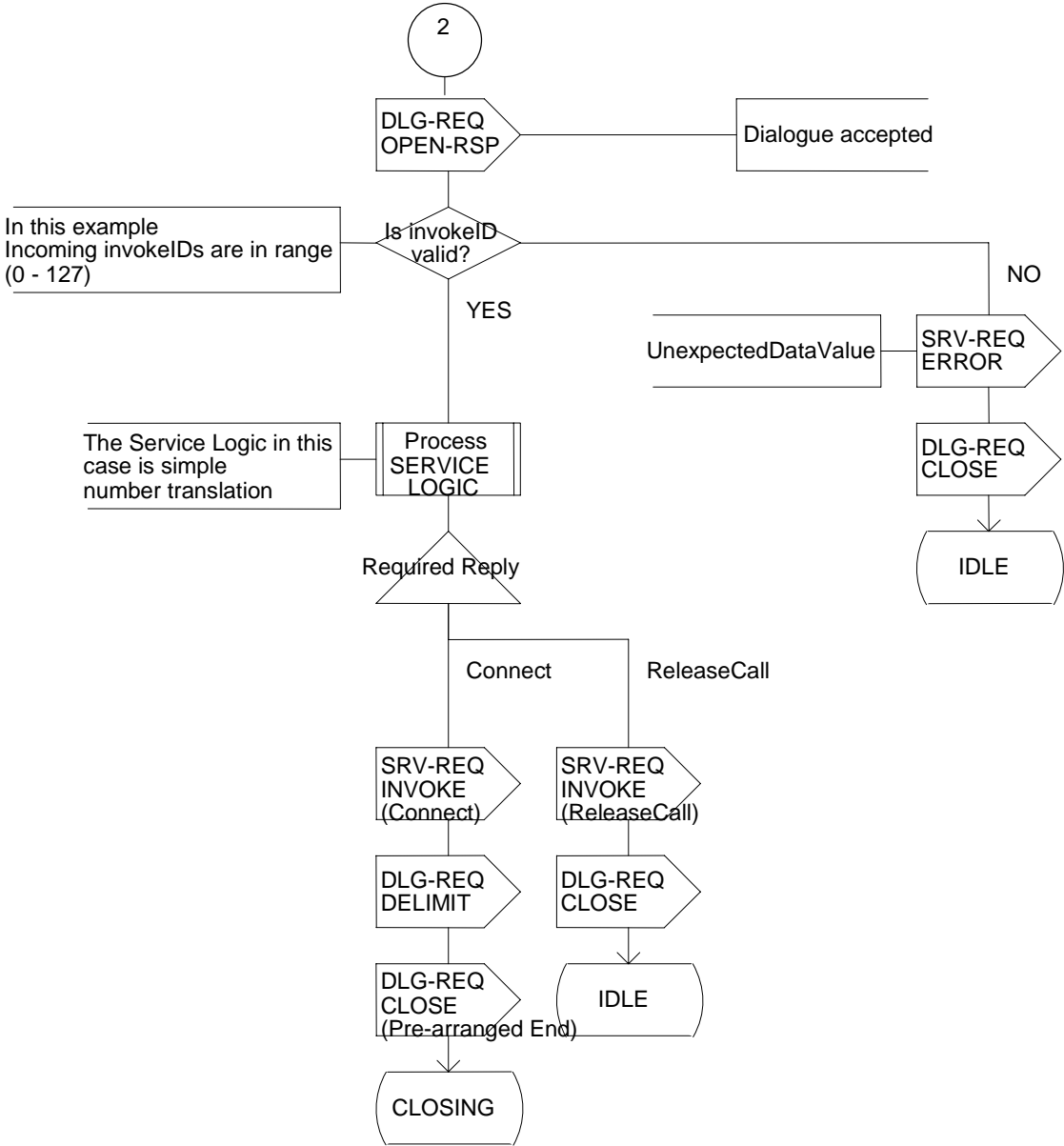




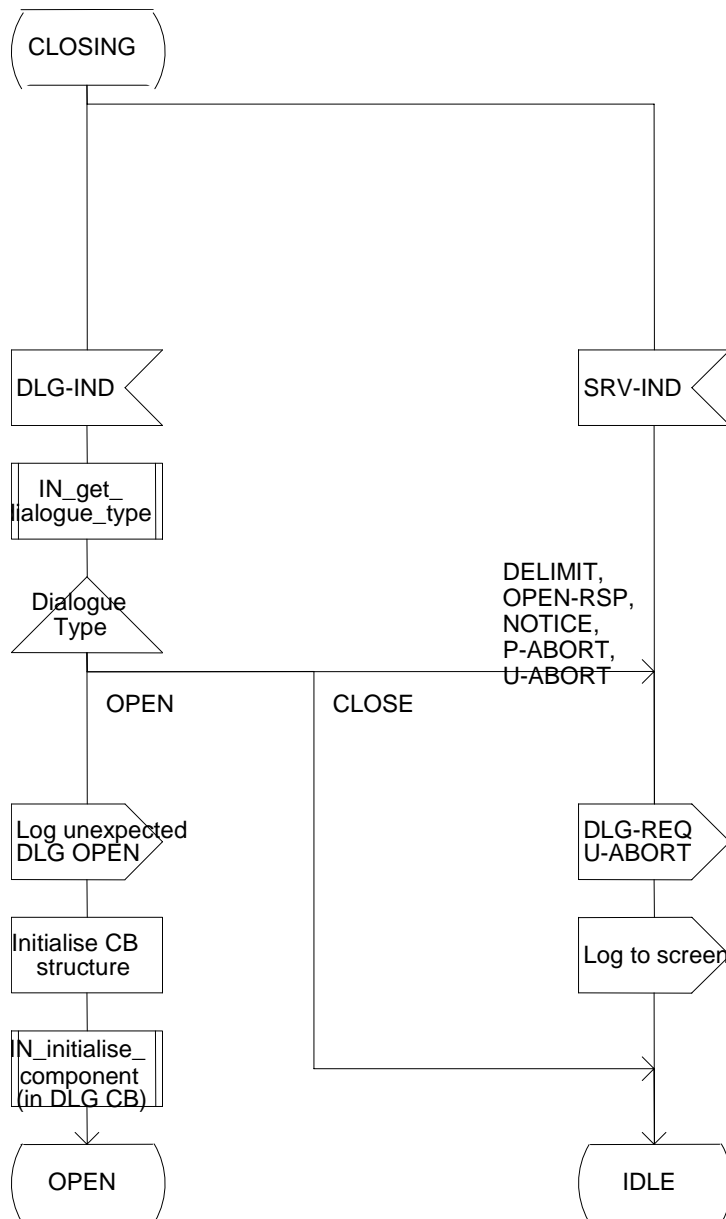
E4



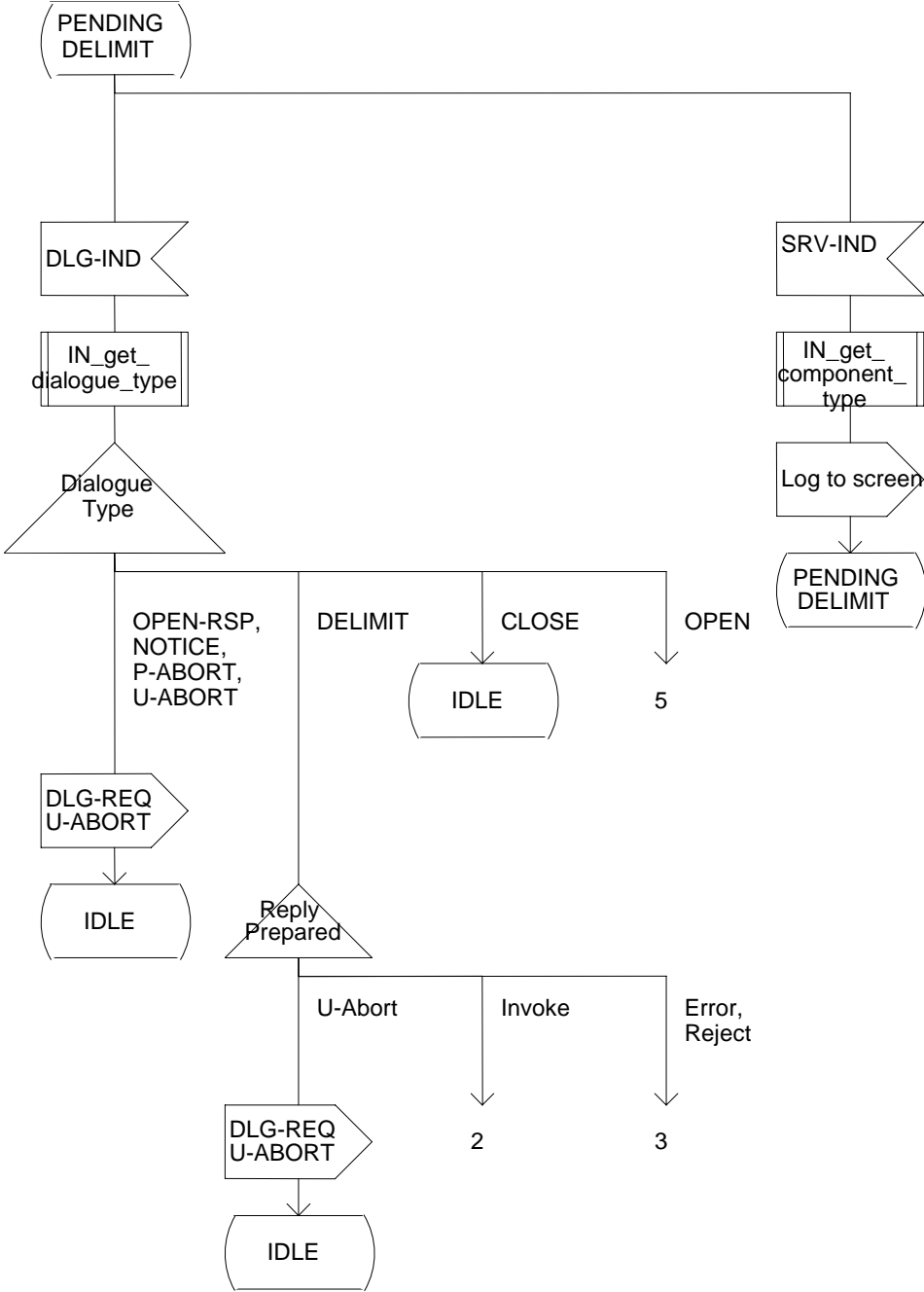
E5



E6



E7



E8

