



Dialogic® TX Series SS7 Boards

Loader Library Developer's Reference Manual

July 2009

64-0457-01

www.dialogic.com

Copyright and legal notices

Copyright © 1998-2009 Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation at the address provided below.

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 or its subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic 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.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at www.dialogic.com.

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 may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic 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 and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Dialogic 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.

Any use case(s) shown and/or described herein represent one or more examples of the various ways, scenarios or environments in which Dialogic® products can be used. Such use case(s) are non-limiting and do not represent recommendations of Dialogic as to whether or how to use Dialogic products.

Dialogic, Dialogic Pro, Brooktrout, Diva, Cantata, SnowShore, Eicon, Eicon Networks, NMS Communications, NMS (stylized), Eiconcard, SIPcontrol, Diva ISDN, TruFax, Exnet, EXS, SwitchKit, N20, Making Innovation Thrive, Connecting to Growth, Video is the New Voice, Fusion, Vision, PacketMedia, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation or its subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. The names of actual companies and product mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Revision history

Revision	Release date	Notes
9000-6430-10	June 1998	GG, loader.doc
9000-6430-11	January 1999	GG/MVH
9000-6430-12	September 2000	MCM, Fusion 3.1
9000-6430-13	February 2001	GJG, SS7 3.6
9000-6430-14	August 2001	GJG, SS7 3.8 Beta
9000-6430-15	November 2003	MCM, SS7 4.0
9000-6430-16	September 2008	SRG, SS7 5.0
64-0457-01 Rev A	July 2009	LBG, SS7 5.1
Last modified: July 7, 2009		

Refer to www.dialogic.com for product updates and for information about support policies, warranty information, and service offerings.

Table Of Contents

Chapter 1: Introduction	7
Chapter 2: Overview of the Loader library	9
Development environment.....	9
Task executable list	10
Chapter 3: Function reference	11
Using the function reference	11
TxLoader.....	12
Return values	14

1 Introduction

The *Dialogic® TX Series SS7 Boards Loader Library Developer's Reference Manual* explains how to use the Loader library function to develop applications that perform task loading.

Note: The product(s) to which this document pertains is/are among those sold by NMS Communications Corporation ("NMS") to Dialogic Corporation ("Dialogic") in December 2008. Certain terminology relating to the product(s) has been changed, whereas other terminology has been retained for consistency and ease of reference. For the changed terminology relating to the product(s), below is a table indicating the "New Terminology" and the "Former Terminology". The respective terminologies can be equated to each other to the extent that either/both appear within this document.

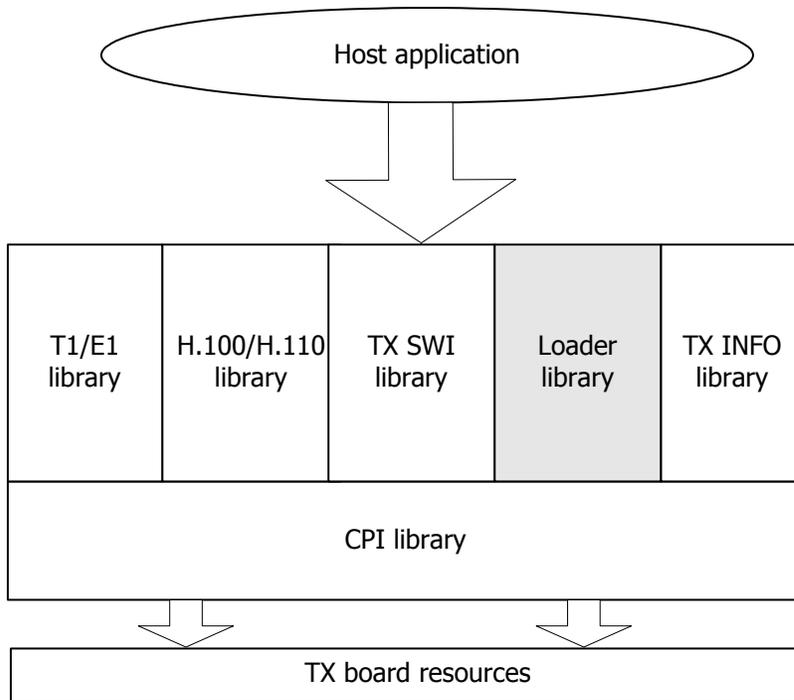
Former terminology	Current terminology
NMS SS7	Dialogic® NaturalAccess™ Signaling Software
Natural Access	Dialogic® NaturalAccess™ Software

2

Overview of the Loader library

Development environment

The TX host application development environment consists of libraries that enable you to configure and control the protocol engines loaded on the TX board. This manual describes the Loader library.



After a system reset, the TX boards are left in a reset state. TX-based system and protocol tasks must be downloaded to the TX board memory. Refer to the *Task executable list* on page 10 for more information. TX-based tasks are usually downloaded with the *cplot* utility, which uses the Loader library to load tasks to the TX board. The Loader library enables you to develop custom applications that perform task loading with an interface rather than by calling the *cplot* utility.

After loading all desired protocol tasks, a TX board is ready for configuration. To download a configuration file to TX boards, use the *txconfig* utility. For more information on creating a TX board configuration file, refer to the *NaturalAccess™ Signaling Software Configuration Manual*. As an alternative to the *txconfig* utility, use the TX SWI library to perform TX board configuration directly from a host application. Refer to the *Dialogic® TX Series SS7 Boards TDM for SS7 Developer's Reference Manual* for more information about the TX SWI library.

The Loader function prototypes and structure definitions are located in the *cplot.h* include file and are provided by the *cplot* library module (*cplot.lib* for Windows systems, *libtxld.so* for UNIX systems).

Task executable list

The following table lists the TX board task executable files and the priority at which each task executes:

File	Name	Priority	Description
<i>debug.elf</i>	debug	11	Low level debugger
<i>isup.elf</i>	isup	20	SS7 ISUP task Stack size: 40960
<i>mtp.elf</i>	mtp	19	SS7 MTP layers 1 through 3 Stack size: 40960
<i>sigtran.elf</i>	sigtran	20	SS7 M3UA and SCTP layers Stack size: 40960
<i>sccp.elf</i>	sccp	21	SS7 SCCP task
<i>tcap.elf</i>	tcap	23	SS7 TCAP task
<i>tup.elf</i>	tup	22	SS7 TUP task
<i>txmon.elf</i>	txmon	18	TX health monitor task

3

Function reference

Using the function reference

This section provides a reference to the Loader library function. A typical function includes:

Prototype	The prototype is followed by a list of the function's arguments. Dialogic data types include: <ul data-bbox="422 598 714 787" style="list-style-type: none">• U8 (8-bit unsigned)• S8 (8-bit signed)• U16 (16-bit unsigned)• S16 (16-bit signed)• U32 (32-bit unsigned)• S32 (32-bit signed) If a function argument is a data structure, the complete data structure is defined.
------------------	---

TxLoader

Downloads a task to the TX board.

Prototype

U32 TxLoader (TXLOADER_DESC **desc*)

Argument	Description
<i>desc</i>	<p>Pointer to the following parameter structure that controls the loading and execution of the task:</p> <pre> /* Loader description structure */ typedef struct { U8 debug; /* Debug level */ U8 priority; /* Task priority */ U8 cp; /* Communications processor (TX board) number */ U8 avail[5]; /* For future use */ U32 exopts; /* Task execution options */ U32 stack_size; /* Task stack size */ S8 taskname[10]; /* Name of task */ S8 loadfile[128]; /* Filename containing binary image of the task */ S8 cmdline[128]; /* Optional command line string */ S8 *descrip; /* Pointer to 80 byte array for description */ U32 status; /* Completion status */ U8 reserved[128]; /* Reserved for future use */ } TXLOADER_DESC; </pre> <p>Refer to the Details section for field descriptions.</p>

Details

The following table describes the TXLOADER_DESC fields:

Name	Type	Description
debug	U8	Level of debug messages returned. 0 = No debug output 1 = Limited debug output 2 = Standard debug output 3 = Verbose debug output
priority	U8	Priority at which the task executes. Allowed values are from 1 through 31, with 1 being the highest priority. Refer to the <i>Task executable list</i> on page 10 to determine the priority to use.
cp	U8	Communications processor (TX board) number. Specifies which TX board to load the task on in multi-board systems. Valid values are 1 through 16.
avail	U8[5]	Not currently used. Zero-fill for future backwards compatibility.
exopts	U32	Execution options. Internal flags used for debugging. Set to 0. Note: Some downloadable tasks use execution options to control task-specific behavior; other tasks use command line strings to control behavior. Set execution options only if specifically indicated for the given downloaded task.
stack_size	U32	Size of the stack used for task execution. Minimum value is 4096 (default). The default is sufficient for most tasks. Tasks requiring additional stack space are noted in the <i>Task executable list</i> on page 10.

Name	Type	Description
taskname	S8[16]	Name of the task. Refer to the <i>Task executable list</i> on page 10 to determine the name to use. Must be null terminated. Note: The TX operating system currently limits task names to 8 characters.
loadfile	S8[128]	Filename containing the binary image of the task. This is a pointer to a null-terminated string containing the filename. If the file to be loaded is not in the path variable, the filename must be fully qualified.
cmdline	S8[128]	Optional command line string. Normally, set [0] = 0 to indicate no command line string. Note: Certain downloadable tasks use execution options to control task-specific behavior; other tasks use command line strings to control behavior. Set command line options only if specifically indicated for the given downloaded task.
descrip	S8[128]	An area where a task information string is stored upon successful task download or where an error description string is stored upon failed task download.
status	U32	Completion status (0 if task loaded successfully; otherwise an error code). Refer to <i>Return values</i> on page 14.
reserved	U8[128]	Reserved for future use.

Return values

The following table lists the values that can be returned from **TxLoader**. These return values are defined in the *cplot.h* include file.

In addition to the return values listed in the table, it is also possible to receive a CPI error code that is transparently passed back to the calling function. CPI error codes are defined in *txcpi.h* and can be differentiated from loader errors as follows:

```
if (error >= CPI_ERRCODE_BASE)
<handle as CPI error code>
else
<handle as loader error>
```

Return value	Description
TXLDERR_CPI	CPI library returned an error that was not transparently passed back to the calling function.
TXLDERR_OPEN	Unable to open the task load file specified in loadfile.
TXLDERR_TRUNC	Truncated load file detected. The task load file header indicates more data than the file actually includes.
TXLDERR_INVALID	Invalid load file detected. The task load file either is not a loadable object file or is corrupt.
TXLDERR_XFER	Error occurred while transferring a code block to the TX board.
TXLDERR_EXISTS	Task of the name specified in task_name already exists on the TX board.
TXLDERR_CREATE	Unable to create the task on the TX board.
TXLDERR_NOMEM	Insufficient memory on the host system to complete the load request.
TXLDERR_CPMEM	Unable to obtain the required TX-board based memory.
TXLDERR_CLOSING	Error while closing the load file.
TXLDERR_TASKPC	Error while setting the task's program counter.
TXLDERR_BADFUNC	Unknown kernel function requested.
TXLDERR_DRVINIT	Unable to access the TX device driver.
TXLDERR_CHANNEL	Unable to obtain the communications channel used for task loading.
TXLDERR_BADTYPE	Unknown type of TX board reported.
TXLDERR_TASKNAME	Invalid task name specified in taskname.
TXLDERR_TSKSTART	Error when attempting to start the task.
TXLDERR_MAXFRAG	Number of code fragments in the load file exceeds the maximum allowed.
TXLDERR_FRAGHDR	Bad fragment header detected in the load file.
TXLDERR_DPR	Anomaly detected in the dual-port-ram communication.

Index

C

configuration 9
CPI error codes 14
cplot.h 9, 14
cplot.lib 9

D

debug.elf 10
development environment 9

E

error codes 14

I

isup.elf 10

L

libtxld.so 9
Loader library definition 9

M

mtp.elf 10

P

priority of tasks 10

R

return values 14

S

sccp.elf 10
sigtran.elf 10

T

task executables 10
tcap.elf 10
tup.elf 10
txconfig 9
txcpi.h 14
TXLDERR_XXXX 14
TxLoader function 12
TXLOADER_DESC 12
txmon.elf 10