Dialogic® TX Series SS7 Boards

TX Utilities Manual

July 2009 64-0459-01

www.dialogic.com

Copyright and legal notices

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
9000-6437-10	June, 1998	GG
9000-6437-11	November, 1998	MVH
9000-6437-12	September, 2000	MCM; Fusion 3.1
9000-6437-13	January, 2002	MVH; SS7 3.8
9000-6437-14	November, 2003	MCM; SS7 4.0 Beta
9000-6437-15	April, 2004	MCM; SS7 4.0
9000-6437-16	August, 2004	SRR, SS7 4.1
9000-6437-17	April 2005	LBG, SS7 4.2
9000-6437-18	July 2008	SRG, SS7 5.0 Beta
9000-6437-19	September 2008	SRG, SS7 5.0
64-0459-01	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

1 Introduction

The *Dialogic® TX Series SS7 Boards TX Utilities Manual* explains how to use the TX basic operation utilities to load and configure a TX board and to diagnose hardware and software issues.

Before using the TX utilities, ensure that the TX board, the Natural Access software, and the NaturalAccess Signaling Software are properly installed. Refer to the appropriate installation manual for operating system specific details regarding installation.

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 TX utilities overview

TX basic operation utilities

NaturalAccess Signaling Software provides the following TX basic operation utilities. Run these utilities from the *\Program Files\dialogic\tx\bin* directory in Windows or from the */opt/dialogic/tx/bin* directory in UNIX.

Utility	Description	For more information, refer to...
cpcon	Uses Natural Access to manage communication with the TX board. This utility is an operator console run from the command line. *cpcon* supports Hot Swap. If you want to bypass the Natural Access environment, use the *cpcon_* utility to manage communication with the TX board.	*Running cpcon* on page 33
cplot	Loads communications processor tasks to TX boards.	• *Loading tasks to TX boards: cplot* on page 25 • *TX communications processor tasks* on page 11
cpmodel	Displays the board type for each installed TX board.	*Viewing a list of installed TX boards: cpmodel* on page 14
pcigetcfg	Obtains bus and slot information during software installation (UNIX only).	*Obtaining bus and slot information during software installation (UNIX only)* on page 14
ss7trace	Used when monitoring or troubleshooting SS7 signaling. Displays any debug data tracing that is turned on in MTP or ISUP and any Ethernet trace data that has been activated.	*Receiving data traces from TX based tasks: ss7trace* on page 22
txalarm	Uses Natural Access to display and optionally log alarm messages generated by SS7 tasks running on all TX boards. *txalarm* supports Hot Swap. If you want to bypass the Natural Access environment, use the *txalarm_* utility to display and optionally log alarm messages.	*Receiving alarms from TX based tasks: txalarm* on page 19
txccode	Displays a text description of a completion code (error code) reported by a TX board.	*Viewing completion code descriptions: txccode* on page 30
txcpcfg	Assigns CP numbers to TX boards based on bus and slot.	*Assigning CP numbers to TX boards: txcpcfg* on page 13
txdiag	Provides diagnostic information about TX boards.	*Diagnosing TX boards: txdiag* on page 27
txdump	Dumps the contents of the shared memory used for communication between the TX board and the host driver.	*Analyzing host-CP shared memory: txdump* on page 29
txeeprom	Displays information stored in the EEPROM.	*Viewing EEPROM information: txeeprom* on page 15
txflash	Updates the TX board firmware.	*Updating the board flash memory: txflash* on page 17

Utility	**Description**	**For more information, refer to...**
txinfo	Obtains board system-level information.	*Obtaining board system-level information: txinfo* on page 31
txlocate	Blinks the end bracket LEDs of a TX board to physically locate the board in the chassis.	*Locating a TX board in a chassis: txlocate* on page 18
txreset	Resets a TX board's firmware while showing the boot progress through the board's internal bank of diagnostic LEDs.	*Resetting and rebooting from flash: txreset* on page 18
txsnap	Generates a core dump of a TX board, creating a snapshot file.	*Generating a snapshot file: txsnap* on page 28
txstats	Displays statistics maintained by the TX device driver.	*Viewing host driver statistics: txstats* on page 26

Utilities and demonstration programs

NaturalAccess Signaling Software provides the *txconfig* utility to configure TDM-based and IP-based interfaces on TX boards. *ss7load* calls *txconfig* at board boot time. Refer to the *Dialogic® NaturalAccess™ Signaling Software Configuration Manual* for information about *txconfig*.

NaturalAccess Signaling Software provides the following programs in compiled and uncompiled form to demonstrate the usage of the TDM libraries:

Program	**Demonstrates how to...**
t1demo	Use the T1/E1 and H.100/H.110 library functions and test the effect of the functions on other TX boards in a system.
t1stat	Receive unsolicited T1/E1 status messages and performance reports.
txdynamic	Dynamically switch SS7 links across TDM channels without rebooting the TX boards.
txsdemo	Use the TX SWI library. Use this program as a starting point to control switching on a TX board.

Refer to the *Dialogic® TX Series SS7 Boards TDM for SS7 Developer's Reference Manual* for information about these demonstration programs.

TX communications processor tasks

The following table lists the TX board task executables and the priority at which the task executes:

File	Name	Priority	Description
debug.elf	debug	11	Low-level debugger (firmware-resident task for certain TX board types).
isup.elf	isup	21	SS7 ISUP task. Stack size: 40960.
mtp.elf	mtp	20	SS7 MTP layers 1 through 3. Stack size: 40960.
mtp12.elf	mtp	20	SS7 MTP layers 1 and 2. Stack size: 12000
sigtran.elf	sigtran	20	SS7 M3UA and SCTP layers Stack size: 40960
sccp.elf	sccp	21	SS7 SCCP task. Stack size: 16384
tcap.elf	tcap	23	SS7 TCAP task. Stack size: 16384
tup.elf	tup	22	SS7 TUP task. Stack size: 40960
txmon.elf	txmon	19	TX health monitor task.

3 TX board management utilities

Assigning CP numbers to TX boards: txcpcfg

Use *txcpcfg* to view all detected TX boards in a system and to assign a CP (communications processor) number to a board.

Viewing all detected boards

To view all detected TX boards, run *txcpcfg* with no parameters:

```
txcpcfg
```

The following example shows a system with three TX boards where only two boards have assigned CP numbers:

```
txcpcfg

Bus    Slot    CP Number    CP Model
---    ----    ---------    --------
1      10      1            TX 4000
1      11      2            TX 4000
2      5       UNDEFINED    TX 5500E
```

Note: TX 4000 also refers to a TX 4000C board.

Assigning a CP number

To assign a CP number to a TX board at a given PCI bus and slot, enter the following command:

```
txcpcfg bus slot boardnum
```

where:

Argument	Description
bus	Bus number of the board that you are configuring.
slot	Slot number of the board that you are configuring.
boardnum	CP number to assign to the board. Valid CP numbers start at 1 and must be unique. They do not have to be consecutive.

In a Windows system, any changes to the configuration information are automatically saved. In a UNIX system, you must save the configuration changes by editing the *cpcfg* file, located in */opt/dialogic/tx/bin*.

Obtaining bus and slot information during software installation (UNIX only)

pcigetcfg is used under UNIX to display information for all TX boards detected on the PCI bus. *pcigetcfg* assigns a unique number to each detected TX board, starting from 0, and displays the following information for each device:

Field	**Description**
NUMCNTLS	Number of TX boards detected.
HWTYPE***n***	Type of TX board (for board ***n***).
SERIAL***n***	Serial number of the TX board (for board ***n***).
CPID***n***	CP number assigned to the TX board (for board ***n***).
BUS***n***	Bus number of the board (for board ***n***).
SLOT***n***	Slot number of the board (for board ***n***).

When the NaturalAccess Signaling Software is installed, an installation script uses the bus and slot information provided by *pcigetcfg* to determine the entries to place in the *cpcfg* script executed at board boot time.

Note: The output of *pcigetcfg* is designed to assist the installation script rather than to provide well-formatted screen output. Use *txcpcfg* instead of *pcigetcfg* to view the PCI bus and slot information for all detected TX boards.

Viewing a list of installed TX boards: cpmodel

Use *cpmodel* to display a list of TX boards that are currently known to the system. Each board is listed with a CP (communications processor) number that is used by all other utilities to address a particular board.

Procedure

Run *cpmodel* according to the following syntax:

```
cpmodel -b boardnum
```

where ***boardnum*** is a valid CP number. The default is 0, which lists all TX boards.

Description

The following example shows a system with four TX boards installed:

```
cpmodel
CPMODEL V3.0 ©Copyright 1998-2009, Dialogic Corporation. All rights Reserved
Board #1 is a TX 4000
Board #2 is a TX 4000
Board #3 is a TX 5500E
Board #4 is a TX 5020E
```

Note: TX 4000 also refers to a TX 4000C board.

Viewing EEPROM information: txeeprom

Use *txeeprom* to display the information stored in a TX board EEPROM. The information is determined at TX board manufacturing time and describes the hardware options available on the board. The board serial number is also recorded in the EEPROM.

Procedure

Run *txeeprom* according to the following syntax:

```
txeeprom -c channelnum -b boardnum -t
```

where:

Argument	Description
-c ***channelnum***	DPR channel to use. Default is 246.
-b ***boardnum***	Board number to which to attach. Default is 1.
-t	Trace program. Default is no tracing.

Description

Most fields in the EEPROM display are static; they do not change from one TX board to another. The following table presents the configurable parameters set at TX board manufacturing time:

Field	Description
SerialNum	Unique serial number assigned to the TX board. Use this number to track the board.
Licensing	Shows the various licensing options that have been activated for the specified TX board. All boards are licensed for signaling at the MTP layer. The following additional licensing options are available: • Full Stack - TX board is licensed for entire set of signaling protocol stacks • HSL – TX board is licensed for High Speed Links Any currently unused licensing bits are reserved for future options.
NumSS7Links	Maximum number of SS7 communication links that can operate on the given TX board.
NIC Addr ***n***	Ethernet MAC address assigned to each LAN adapter on TX boards.

The following example shows the *txeeprom* display for a TX 5500E board:

```
       Family:  TX500E
    TestLevel:  0xC81D
 TestLevelRev:  0x01
 SoftwareComp:  0x0
      MFGYear:  2008
      MFGWeek:  37
   ATETestBit:  0x00
    SerialNum:  1108371946
AssemblyLevel:  0xC7EF
  AssemblyRev:  515
 AssemblyYear:  2008
 AssemblyWeek:  39
    Licensing:  Full Stack
    FlashSize:  32 MB
  NumEthernet:  3
     CPUSpeed:  800 MHz
   L1ClockDiv:  0
    NumTrunks:  4
  NumSS7Links:  128
  MemBusSpeed:  400 MHz
   NIC Addr 1:  00:20:22:32:D6:F6
   NIC Addr 2:  00:20:22:32:D6:F7
   NIC Addr 3:  00:20:22:32:D6:F8
```

Updating the board flash memory: txflash

Use *txflash* to update the TX board firmware. *txflash* is called as part of a standard load of a TX board using the -s option to synchronize flash image versions. If the version of the kernel image stored on the TX board flash memory is not identical to the version currently installed on the host system, *txflash* upgrades the flash on the board before performing a reset of the board.

Procedure

Run *txflash* according to the following syntax:

```
txflash -c channelnum -b boardnum -w writefile -t -v -f filename -o -s
```

where:

Argument	Description
-c ***channelnum***	DPR channel to use. Default is 245.
-b ***boardnum***	Board number to which to attach. Default is 1.
-w ***writefile***	File name from which to write flash information. Default is not to write information to flash.
-t	Trace progress of *txflash*. Default is no tracing.
-v	Show version of kernel on flash. Default is not to show version.
-f ***filename***	Show version of kernel file. Default is not to show version.
-o	Override version checking. Default is no override.
-s	Synchronize flash version. Default is no synchronization.

Description

When a flash update is triggered for a TX board, *txflash* displays the following information:

```
Updating FLASH from cpk4000.fls to board 1...
Sending image #1 to the board...
Sending image #2 to the board...
FLASH image successfully transferred to the CP.
Triggering Flash BURN...
<Erasing Flash>
<Erasing Flash>
<Erasing Flash>
<Erasing Flash>
<Erasing Flash>
<Erasing Flash>
<Erasing Flash>
<Programming Flash>
<Programming Flash>
<Programming Flash>
<Programming Flash>
<Programming Flash>
<Programming Flash>
<Programming Flash>
<Verifing Flash>
FLASH image successfully burned.
```

After *txflash* copies a new flash image to the TX board, use *txreset* to reset the board to boot to the new flash image. Refer to *Resetting and rebooting from flash: txreset* on page 18 for information.

Resetting and rebooting from flash: txreset

Use *txreset* to reboot a TX board's firmware while showing the boot progress through the board's internal bank of diagnostic LEDs. After the board is reset, it is ready to accept downloads of TX-based tasks.

Procedure

Run *txreset* according to the following syntax:

```
txreset -c channelnum -b boardnum
```

where:

Argument	Description
-c ***channelnum***	DPR channel to use. Default is 244.
-b ***boardnum***	Board number to which to attach. Default is 1.

Note: You can also reset the board using *ss7load*. Refer to the *Dialogic® NaturalAccess™ Signaling Software Configuration Manual* for information.

Locating a TX board in a chassis: txlocate

Use *txlocate* to physically locate a TX board in a chassis. Use this utility when:

- Making board cabling changes
- Debugging
- Hot swapping a CompactPCI TX board

txlocate blinks the end bracket LEDs of the specified board.

Procedure

Run *txlocate* according to the following syntax:

```
txlocate -b boardnum -h holdtime -p passes
```

where:

Argument	Description
-b ***boardnum***	Number of the board for which to blink the LEDs. Default is 1.
-h ***holdtime***	Number of milliseconds to hold the LEDs on and off for each toggle. Default is 1000 ms.
-p ***passes***	Number of times to toggle the LEDs on and off. Default is 3.

Description

When you run *txlocate* on a PCI or PCI Express TX board, all end bracket LEDs blink for the specified ***holdtime*** and ***passes***. When you run *txlocate* on a CompactPCI TX board, all end bracket LEDs with the exception of green, yellow, and red status LEDs blink for the specified ***holdtime*** and ***passes***.

For example, to blink the LEDs for TX board 1 for one minute, enter the following command:

```
txlocate -b 1 -p 30
```

4 Communications processor utilities

Receiving alarms from TX based tasks: txalarm

Use *txalarm* under Natural Access to display and optionally log alarm messages generated by NaturalAccess Signaling Software tasks running on all TX boards. *txalarm* supports Hot Swap.

If you want to bypass the Natural Access environment, use the *txalarm_* utility to display and optionally log alarm messages.

Procedure

Run *txalarm* according to the following syntax:

```
txalarm [-f filename]
```

where ***filename*** specifies the file to which alarms are copied.

Description

The following example is a sample *txalarm* message:

```
<01/07/2004 16:17:04> mtp  1  18180 MTP3 Link 1 Down
```

All TX-based tasks that need to log their board load activity or send asynchronous change-in-status indications pass messages to the *txalarm* channel. Alarms are broken into two major divisions:

- Alarms issued by the TXBASE tasks (alarm numbers below 2048)
- Alarms issued by higher layer Natural Access Signaling tasks.

All alarm numbers are defined in the include file “txlog.h”.

The following table describes all TXBASE alarms as well as alarms that are issued by the common core libraries in use by the Natural Access Signaling tasks. For a description of other alarms that may be issued by the given higher layer task, refer to the specific NaturalAccess Signaling Layer’s manual.

Number	Name	Severity	Message
296	OS NO REARIO	Info	CPCI Rear IO board not present
304	OS SYSMON FANSTOP	Critical	CPU Fan has stopped! Board Temp dC [dF]. CPU Temp dC [dF] **Note:** TX 4000 Series alarms do not include temperatures.
305	OS SYSMON FANSTART	Critical	CPU Fan has Re-Started. Board Temp dC [dF]. CPU Temp dC [dF] **Note:** TX 4000 Series alarms do not include temperatures.

Number	Name	Severity	Message
306	OS SYSMON TEMP	Info Critical	On Board Temp <tempName>: dC [dF]. CPU Temp dC [dF]. Fan State:<fanState> tempName: Normal, Warm, High, Very High or Critical fanState: Running, Stopped, Restarted **Note:** TX 4000 Series alarms do not CPU or Fan information.
307	OS SYSMON DISABLE	Critical	Disabling <resource> resource: Ethernets, TDMs, 8260, etc. (board type-specific) Final Text: Board will be Disabled if temp continues to rise
320	OS CLK ROLE	Info	CT Bus Clock Role: Slaved to CT Bus Clocks CT Bus Clock Role: Secondary CT Bus Master CT Bus Clock Role: Primary CT Bus Master
320	OS CLK ROLE	Error	CT Bus Clock Role: Invalid Fallback Clock Config
321	OS CLK FAILURE	Warning	Primary Clock Failure, fallback to Secondary clock source Secondary Clock Failure, fallback to Secondary CT Clock Secondary Clock Failure, fallback to Secondary CT Clock Ref Clock Fallback, unknown State, fallback to internal OSC Invalid Configuration, fallback to Internal OSC Secondary CT Bus Clock Failure, fallback to Internal OSC Clock Fallback Timer, Unexpected State, stopping timer Clock Fallback, unknown State, Event=CT_CLOCK_RECOVER Primary Clock Failure, Mastering secondary clock Secondary Clock Failure, fallback to internal OSC Clock Fallback, unknown State, fallback to internal OSC Primary Clock Failure, fallback to secondary clock
321	OS CLK FAILURE	Error	Error in Clock Fallback
322	OS CLK RECOVERED	Info	Primary Clock Recovery, Returning to Primary clock source Secondary CT Clock Recovery, return to secondary CT clock Primary Clock Master SEM, Unknown Event Secondary Clock Master SEM, Unknown Event Secondary Clock Recovery, return to secondary clock Clock Slave SEM, Unexpected Event
336	OS DHCP ERR	Error	DHCP [<iface>]: TIMEOUT while trying to … DHCP [<iface>]: Invalid DHCP OFFER received (..)
338	OS DHCP ACQ	Info	DHCP [<iface>]: Acquired IP Address <address>
352	OS LICENSE HSL	Error	Cannot create port <portNum>: board not licensed for High Speed Links
16384	SS7 INIT	Error	<message indicating specific initialization step that failed> Example: Cannot run task ..: board not licensed for full stack operation **Note:** Each time a TX board initializes all active protocol layers will perform license verification.
16385	SS7 SERROR	Error	<SS7 higher-layer error message>

Number	Name	Severity	Message
16386	SS7 SLOGERROR	Error	<SS7 higher-layer error message>
16387	SS7 SPRINT	Error	<SS7 higher-layer log (print) message> Example: Invalid TDM port number (17) configured **Note:** An alarm with a similar description to this example is issued any time a link number is configured that is beyond the number of links the given board is licensed for. An error message will also be displayed by the MTP3CFG utility whenever this alarm is generated: Specified link exceeds maximum.
16388	SS7 INTERNAL	Error	Internal Error <SS7 higher-layer error description>
16389	SS7 HOST	Error	hbus_notif:<error description> **Note:** hbus errors indicate problems communication with host-based application(s).
16390	SS7 MBOX	Error	mbox_notif: <error description> **Note:** mbox errors indicate problems using mailbox messaging to communicate with other board-based tasks.
16391	SS7 IBC	Error	ibc_notif: <error description> **Note:** ibc errors indicate problems with Inter-Board Communication (redundant mate communication)
16640	MTP1 IFACE	Error	<errors reported from MTP level-1 driver> Example: SS7 level-1 driver [MCC <mccNum>] Global Reset
19200	SIGTRAN INTERNAL	Error	<SIGTRAN-related internal error message> Example: UNKNOWN packet type received
19201	SIGTRAN IFACE	Warning Error	<virtual socket interface communication error message> Example: vsockif_notif error: [CHECKSUM] receive error

The following example shows a sample *txalarm* message from the mtp task that is executing on TX board number 1:

```
<07/20/20049 16:17:04> mtp  1  18180 MTP3 Link 1 Up
```

Receiving data traces from TX based tasks: ss7trace

Use *ss7trace* when monitoring or troubleshooting SS7 signaling.

Purpose

Displays any debug data tracing that is turned on (through the layer's manager program or initial configuration) in the MTP, ISUP, M3UA, or SCTP tasks or in the TX operating system [Ethernet tracing]. It does not accept commands; it only displays the tracing.

Using ss7trace

Enter the following command to start *ss7trace*:

```
ss7trace
```

The utility returns the following information:

```
Trace Monitor SS7 V5.1: Hit Enter to exit
©Copyright 1998-2009, Dialogic Corporation
```

Enabling packet tracing

Tracing must be enabled to trace all data packets sent and received to the *ss7trace* utility. Use the following commands to enable and disable tracing:

Utility	Command	Description
mtpmgr	trace on	Enable MTP layer packet tracing on all configured links.
	trace off	Disable MTP layer packet tracing on all links.
	link <***n***> tre	Enable MTP layer packet tracing on a specific link.
	link <***n***> trd	Disable MTP layer packet tracing on a specific link.
isupmgr	trace data on	Enable ISUP layer packet tracing on all ISUP packets sent and received.
	trace data off	Disable ISUP layer tracing.
sctpmgr	trace ena	Enable packet tracing at the lower interface of SCTP.
	trace dis	Disable SCTP layer packet tracing.
m3uamgr	trace ena	Enable packet tracing at the lower interface of M3UA.
	trace dis	Disable M3UA layer packet tracing.
txconfig *cpcon*	ethtrace <***i***> on	Enable tracing of Ethernet packets sent and received over Ethernet <***i***>.
	ethtrace <***i***> off	Disable Ethernet packet tracing over Ethernet interface <***i***>.

MTP/ISUP sample trace output

The following sample shows the output generated for a single packet received followed by a single packet transmitted when both MTP and ISUP layer tracing is enabled. Note that these are hexadecimal dumps of the actual packets sent and received, so familiarity with the detailed encodings of a given layer's packets is required to decode the trace data. The packets in this example were collected from an ANSI configuration; therefore, the packet contents (for example, the point code length in the routing label) for ITU or Japan protocol variants will be different.

```
14:46:36.0 MTP3.1 <-- : Link # 1
85 01 00 00 02 00 00 05 06 00 01 00 20 01 0A 03 ............ ...
06 0B 03 C0 90 A2 05 03 10 01 01 01 0A 05 03 10 ................
04 22 04 00 00                                  ."...
14:46:36.0 ISUP.1 <-- 0.0.2:
06 00 01 00 20 01 0A 03 06 0B 03 C0 90 A2 05 03 .... ...........
10 01 01 01 0A 05 03 10 04 22 04 00             .........".. 
14:46:36.0 ISUP.1 --> 0.0.2:
06 00 06 14 14 00                               ......
14:46:36.0 MTP3.1 --> : Link # 1
85 02 00 00 01 00 00 09 06 00 06 14 14 00 00    ...............
```

The heading for each message shows the time the packet is sent or received, the layer generating the trace, and the direction of the message (--> transmitted packets, <-- received packets). The MTP heading also indicates the link number the packet is sent or received on. The ISUP header indicates the destination point code that the packet is sent to or received from.

The MTP packet trace contains the content of the packet starting with the service information octet (SIO), followed by the routing label (DPC, OPC, and SLS) and packet data.

The ISUP packet trace contains the ISUP portion of the packet being sent or received starting with the circuit identification code (CIC), followed by the message type and parameters.

SCTP/M3UA sample trace output

The following sample shows the trace output generated when SCTP and M3UA are enabled:

```
16:00:58.0 SCTP.1 <-- Sap ID: 0

0B 59 0B 59 0D A0 03 8A 8E C2 BF 46 03 00 00 10 .Y.Y.......F....
0D A0 03 B2 00 00 7F E7 00 00 00 00             ............

16:01:00.0 M3UA.1 --ASPSM--> : Service User: 0

01 00 03 01 00 00 00 18 00 04 00 0F 03 6B 8C 80 .............k..
03 6C 28 9C 06 AF 2C 00                         .l(...,.

16:01:00.0 SCTP.1 --> Sap ID: 0

0B 59 0B 59 0A D9 33 C9 CE E6 2B 8F 00 03 00 28 .Y.Y..3...+....(
0D A0 03 B3 00 00 00 29 00 00 00 03 01 00 03 01 .......)........
00 00 00 18 00 04 00 0F 03 6B 8C 80 03 6C 28 9C .........k...l(.
06 AF 2C 00                                     ..,.
```

The heading for each SCTP message shows the:

- Time the message was sent or received
- SCTP layer and the board number (SCTP.1)
- Direction of the message (--> = transmitted and <-- = received)
- SAP ID

The hex dump of the message begins with the 12 byte SCTP common header (source port – 2 bytes, destination port – 2 bytes, verification tag – 4 bytes, checksum – 4 bytes). The chunk type is the 13th byte. The chunk type in the received message is 03 (SACK). The chunk type in the sent message is 00 (DATA).

The heading for each M3UA message shows the:

- Time the message was sent or received
- M3UA layer and the board number (M3UA.1)
- Direction and class of message (--ASPSM--> = transmitted ASPSM message)
- Service user ID

The hex dump of the message starts with the M3UA common header (version – 1 byte, reserved - 1 byte (always 0), message class – 1 byte, and message type – 1 byte). The third and fourth bytes (message class/type) define the M3UA message type. Class = 03 (ASPSM) and type = 01 (ASPUP).

Ethernet sample trace output

The following sample shows the trace output generated when ethtrace is enabled:

```
14:07:34.0 ETH[1].1 --> (98 bytes):

00 20 22 31 98 12 00 20 22 31 7B BA 08 00 45 00 . "1... "1{...E.
00 54 00 56 00 00 3C 01 F4 4F 42 01 01 01 42 01 .T.V..<..OB...B.
05 01 08 00 BC 04 00 03 00 01 54 58 20 50 49 4E ....+.....TX PIN
47 3A 40 41 42 43 44 45 46 47 54 58 20 50 49 4E G:@ABCDEFGTX PIN
47 3A 40 41 42 43 44 45 46 47 54 58 20 50 49 4E G:@ABCDEFGTX PIN
47 3A 40 41 42 43 44 45 46 47 54 58 20 50 49 4E G:@ABCDEFGTX PIN
47 3A                                           G:

14:07:34.0 ETH[1].1 <-- (98 bytes):

00 20 22 31 7B BA 00 20 22 31 98 12 08 00 45 00 . "1{.. "1....E.
00 54 00 56 00 00 3C 01 F4 4F 42 01 05 01 42 01 .T.V..<..OB...B.
01 01 00 00 C4 04 00 03 00 01 54 58 20 50 49 4E ..........TX PIN
47 3A 40 41 42 43 44 45 46 47 54 58 20 50 49 4E G:@ABCDEFGTX PIN
47 3A 40 41 42 43 44 45 46 47 54 58 20 50 49 4E G:@ABCDEFGTX PIN
47 3A 40 41 42 43 44 45 46 47 54 58 20 50 49 4E G:@ABCDEFGTX PIN
47 3A                                           G:
```

Each message is a complete Ethernet packet. The heading for each Ethernet message shows the:

- Time the message was sent or received
- Ethernet interface number that the packet was sent or received over
- Direction of the message (--> = transmitted and <-- = received)
- Total byte size of the message

The hex dump of the message begins with the Ethernet header and contains the full payload of the message.

Loading tasks to TX boards: cplot

Use *cplot* to dynamically load communications processor tasks to TX boards. Tasks have been compiled into **.elf* object files. The *cplot* utility loads ELF files onto the communications processor. Refer to *TX communications processor tasks* on page 11 for more information.

Procedure

Run *cplot* according to the following syntax:

```
cplot argument argument
```

where:

Argument	Description
-b ***boardnum***	TX board number to which to load the task. The default is 1.
-f ***filename***	ELF Object file for *cplot* to load. This argument is required.
-l ***cmd_line***	Command line string enclosed in double quotes.
-n ***taskname***	Communications processor task to load. When used with the -f option, *cplot* requests a communications processor task to be created with the given task name. The program object file is then loaded as the program for the new communications processor task. This argument is required. The task name can have a maximum of eight characters.
-o ***options***	Options to use when executing the communications processor task. ***options*** is task specific and can use hexadecimal notation by prepending the entry with 0x. Default is 0.
-p ***priority***	Priority of the communications processor task. Valid priority levels are 0 through 31, with 0 being the highest priority and 31 being the lowest priority. By default, the created communications processor task has a priority of 30.
-s ***size***	Stack size of the communications processor task. By default, the communications processor task stack size is 4096 bytes. Hexadecimal notation can be used by prepending a 0x.
-t ***tracelevel***	Level of tracing information displayed by *cplot* while downloading a task. The default is 0 (no tracing).
-v	Version and build date of a downloadable task (.*elf* file).

Viewing host driver statistics: txstats

Use *txstats* to view statistics maintained by the TX driver for a particular board or for all known boards. This information is used when analyzing data flow problems between the host and the TX board.

Procedure

Run *txstats* according to the following syntax:

```
txstats -b boardnum -a appchan -d drvchan -p period -m statname -x maxvalue -o options -l
-z -q
```

where:

Argument	Description
-b ***boardnum***	TX board number or 0 to view all board statistics.
-a ***appchan***	CPI channel held by an application. Default is 0 (not applicable).
-d ***drvchan***	CPI channel used to access the driver. Default is 254.
-p ***period***	Interval (in ms) between periodic statistics requests.
-m ***statname***	Monitors statistics in histogram format. Use –m ? to view a list of available statistics.
-x ***maxvalue***	Maximum value for -m display range.
-o ***options***	Specifies options directly. Refer to the *txstats.h* file for more information.
-l	Displays layer-specific statistics extension. Default is view common statistics.
-z	Zeros statistics after displaying current values.
-q	Does not display statistics as part of operations.

Diagnosing TX boards: txdiag

Use *txdiag* to perform board level diagnostic tests on a TX board and to test communications between the host driver and the board.

Caution:	Running *txdiag* terminates any functions that are still executing. After running *txdiag*, reload the TX board to return it to a normal state.

Procedure

Run *txdiag* according to the following syntax:

```
txdiag -b boardnum -c channelnum -a -f filename -h testname
```

where:

Argument	Description
-b ***boardnum***	TX board number to diagnose. Default is 1.
-c ***channelnum***	DPR channel number. Default is 249.
-a	Performs tests on all boards.
-f ***filename***	Reads parameter values from this text file.
-h	Shows usage.
testname	Diagnostic test to perform. Default is ALL. Refer to *txdiag diagnostic tests* on page 27 for more information.

txdiag diagnostic tests

You can run the following *txdiag* diagnostic tests:

Run this test...	To...
NMI	Verify that the host is able to assert the non-maskable interrupt.
MEMSWEEP	Verify that the SDRAM memory space can be accessed.
INTR	Verify that the host and CP can generate and receive interrupts.
ALL	Execute all tests in the order shown in this table. When you specify ALL, *txdiag* performs each test regardless of the results of the previous test. Examine the results of each test individually to verify its success or failure before assuming that all tests successfully executed.

Description

The following example shows output from *txdiag* when the -b argument is used:

```
> txdiag -b 1
TXDIAG V2.0: Copyright 1998-2009, Dialogic Corporation
CP 1: NMI - Resetting...
CP 1: NMI - Starting test...
CP 1: NMI - SUCCESS
CP 1: MEMSWEEP - Resetting...
CP 1: MEMSWEEP - Starting test...
CP 1: MEMSWEEP - SUCCESS
CP 1: INTR - Resetting...
CP 1: INTR - Starting test...
CP 1: INTR - Entering Polling Loop...
CP 1: INTR - SUCCESS
```

Generating a snapshot file: txsnap

Use *txsnap* to take a snapshot of the board to diagnose TX-based problems. A snapshot file contains an image of the board memory space. Use a snapshot to view the system as it was at the time the snapshot was taken. By default, the snapshot file is created in the current directory and is named *cpsnap.dmp*.

To generate a snapshot file for TX boards, run *txsnap* according to the following syntax:

```
txsnap -b boardnum -p -r -f filename
```

where:

Argument	**Description**
-b ***boardnum***	TX board number for which to generate a snapshot file. Default is 1.
-p	Snaps only PCI-mapped memory.
-r	Creates a raw memory snapshot file without headers.
-f ***filename***	File to which to write the TX board information. Default is *cpsnap.dmp*.

txsnap is generally run with the -b argument only. Because the resulting snapshot file is greater than 128 Mb, zip the snapshot file before transferring it.

Analyzing host-CP shared memory: txdump

Use *txdump* to dump a hexadecimal/ASCII display of the shared memory used by the host driver to communicate with the TX board.

Procedure

Run *txdump* according to the following syntax:

```
txdump-c channelnum -b boardnum -o offset -l bytelength -s
```

where:

Argument	Description
-c ***channelnum***	DPR channel number. Default is 247.
-b ***boardnum***	Board number to dump. Default is 1.
-o ***offset***	Offset from which to dump. Default is 0.
-l ***bytelength***	Byte length to dump. Default is 2048.
-s	Dump status registers (only valid for certain TX board types).

Description

The following example shows *txdump* when the -s argument is used:

```
>txdump -s

TX Memory Dump Utility V4.0
Copyright 1997-2009, Dialogic Corporation

CP #1 Status / Control Register Set:

Serial Number: 104426053

Raw Status Registers:
  [0]: $06396A45
  [1]: $00000000
  [2]: $00000000
  [3]: $00000000
  [4]: $00000000
  [5]: $00000000
  [6]: $00000000
  [7]: $00000000
```

Viewing completion code descriptions: txccode

Use *txccode* for a description of a completion code (error code) reported by a TX board.

Run *txccode* according to the following syntax:

```
txccode ccode
```

where:

Argument	Description
ccode	Completion code to be described. Specify the letter a to indicate that it displays all current error codes. When specifying a particular error code, begin the code with 0x to indicate a hexadecimal error code (otherwise decimal is assumed).

Description

The following example shows output of *txccode*:

```
> txccode 0xE000010
Error Code: 0xE000010 (SCCSWI_INVALID_TIMESLOT) - invalid timeslot provided
```

Obtaining board system-level information: txinfo

Use *txinfo* to obtain board system-level information.

Run *txinfo* according to the following syntax:

```
txinfo -b boardnum
```

where:

Argument	Description
-b ***boardnum***	TX board number from which to obtain information. Default is 1.

Description

txinfo issues a request for system information using **txinfoSystemInfo**. This request is then serviced by the $info task, which executes as part of the TX firmware. The $info task responds to the request by providing a copy of the relevant system-level information. Upon receipt of the response message containing this system-level information, *txinfo* displays select information. Refer to the *TX INFO Library Reference Manual* for more information.

The system-level information structure is defined in the *txinfomsg.h* include file.

The following example shows sample output from *txinfo*:

```
---------------------------------
TX board 1 System Information:
---------------------------------

    ID: cpNum: 1, BoardType: TX 5500E, serialNum: 104426053, kernel:15.28

  UTIL: idleCount: 40498654, idlePeak: 73111823

  TIME: bootMsecs: 13554 (bmsecWraps: 0), epochSecs: 1246162728
        6/28/2009 04:21:07(+554 msecs)


--------------------------
TX board 1 CPU Status:
--------------------------

CPU Util_____ 0...10...20...30...40...50...60...70...80...90...100
 Current:  26 =============
 Average:  20 ==========

CPU Temp: 30.2C [8603F] degrees
CPU  Fan: Normal
```

System-level information consists of the following sections:

Section	Description	Fields

ID	TX board and operating system identification.	cpNum = CP number.
		BoardType = Type of TX board.
		serialNum = Board serial number.
		kernel = Operating system version/revision.

UTIL	System utilization area. Used for computing TX processor load.	idleCount = Counter maintained by $idle task (for computing CPU utilization). idlePeak = Peak value that idleCount has ever reached.
TIME	Time handing area (the time since the TX board was booted and the current time of day).	bootMsecs = Number of milliseconds since boot. bmsecWraps = Number of times bootMsecs has wrapped. epochSecs = Number of seconds since epoch. sysTime = System time in timestamp format.
CPU Status	TX board processor (CPU) current status.	Util = Average and current processor utilization (shown as numeric percentage and as a bar graph). CPU Temp = Current CPU temperature. Brd Temp = Overall board temperature. Fan = Current state of the on-board fan or blower.

Note: *txinfo* is also provided as a sample application to easily obtain information such as the specific TX board type, the TX board serial number, and the current state of the CPU.

5 Communications processor console utility: cpcon

Running cpcon

Use *cpcon* under Natural Access to manage communication with the TX board. This utility is an operator console run from the command line. *cpcon* supports Hot Swap*.*

To bypass the Natural Access environment, use *cpcon_* to manage communication with the TX board.

cpcon command sets enables you to monitor and control different aspects of a TX board. Refer to *Command set overview* on page 35 for more information.

Using cpcon

Complete the following steps to run *cpcon*:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Start <i>cpcon</i> by entering the following command at the prompt:
cpcon -b <b><i>boardnum</i></b> -c <b><i>channelnum</i></b>
where:
<table>
<tr><th>Argument</th><th>Description</th></tr>
<tr><td>-b <b><i>boardnum</i></b></td><td>TX board number. Valid values are 1 through 16. The default is 1.</td></tr>
<tr><td>-c <b><i>channelnum</i></b></td><td>DPR channel number. The default is 144.</td></tr>
</table></td></tr>
<tr><td>2</td><td>Specify a command set by entering the command set name enclosed in square brackets. Use the [?] command to view all command sets. Refer to <i>Command set overview</i> on page 35 for more information.
The <i>cpcon</i> prompt returns the TX board number (or communications processor number) and the currently selected command set. For example:
(1) [hbus] &gt;</td></tr>
<tr><td>3</td><td>Use the ? command to view all commands in the current command set.</td></tr>
<tr><td>4</td><td>To change to a different board number, exit the program and run <i>cpcon</i> again with a different board number.</td></tr>
</table>

Using cpcon_

Complete the following steps to run *cpcon_*:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Start <i>cpcon_</i> by entering the following command at the prompt:
<code>cpcon_ -b <b><i>boardnum</i></b> -c <b><i>channelnum</i></b> -p <b><i>portnum</i></b> -i <b><i>inscript</i></b> -I <b><i>inscript</i></b> -o <b><i>outfile</i></b> -O <b><i>outfile</i></b> -v</code>
where:
<table>
<tr><th>Argument</th><th>Description</th></tr>
<tr><td>-b <b><i>boardnum</i></b></td><td>TX board number. Valid values are 1 through 16. The default is 1.</td></tr>
<tr><td>-c <b><i>channelnum</i></b></td><td>DPR channel number. The default is 144.</td></tr>
<tr><td>-p <b><i>portnum</i></b></td><td>Sets the board port number to pass requests to. The default is 6.</td></tr>
<tr><td>-i inscript</td><td>Processes input script and then exit <i>cpcon_</i>.</td></tr>
<tr><td>-I inscript</td><td>Processes input script and then remain in <i>cpcon_</i>.</td></tr>
<tr><td>-o outfile</td><td>Forks all output to file name provided.</td></tr>
<tr><td>-O outfile</td><td>Redirects output to file name (none to screen).</td></tr>
<tr><td>-v</td><td>Activates verbose tracing during board initialization.</td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Specify a command set by entering the command set name enclosed in square brackets. Use the [?] command to view all command sets. Refer to <i>Command set overview</i> on page 35 for more information.
The <i>cpcon_</i> prompt returns the TX board number (or communications processor number) and the currently selected command set. For example:
<code>(1) [hbus] ></code></td></tr>
<tr><td>3</td><td>Use the ? command to view all commands in the current command set.</td></tr>
<tr><td>4</td><td>To change to a different board number, use the command cp <b><i>boardnum</i></b>, where <b><i>boardnum</i></b> is a valid TX board number. Use the cp ? command to determine the board number currently in use.</td></tr>
<tr><td>5</td><td>To cause <i>cpcon_</i> to process an input script from the <i>cpcon_</i> prompt, use the < command:
(1) [con] > <b>< myscript</b>
This causes <i>cpcon_</i> to read the file that is named myscript in the current working directory (where <i>cpcon_</i> was executed from) and to perform each line-read from the script as if those commands were typed in manually.
Other input script control commands are:
<b>< ?</b> = Displays name of the current input script.
<b>< !</b> = Cancels the current input script.
pause = Temporarily pauses input script processing.
resume = Ready for input script processing to continue.</td></tr>
<tr><td>6</td><td>To cause <i>cpcon_</i> to begin forking output to an output file on the local system, use the > command:
(1) [con] > <b>> myoutfile</b>
Other commands are:
<b>> !</b> = Stops outputting to a file.
<b>> ?</b> = Determines whether currently outputting to a file.</td></tr>
</table>

Command set overview

The *cpcon* command sets monitor and control different aspects of a TX board. After starting *cpcon*, select a command set by entering the command set name enclosed in square brackets.

Use the [?] command (with brackets) to view all command sets. Use the ? command (without brackets) to view all commands in the current command set.

The following table describes the *cpcon* command sets:

Command set	**Description**	**For more information, refer to...**
[blst] 5000 Series	Blast (test packet generator) control commands.	*[blst] Blast (test packet generator) control commands* on page 37
[brdg] 5000 Series	Bridge (test packet receiver) control commands.	*[brdg] Bridge (test packet receiver) control commands* on page 38
[cmd]	Console task commands for TX boards.	*[cmd] Console task commands* on page 38
[con]	Standard operator console commands.	*[con] Standard commands* on page 39
[cpu]	PowerPC CPU command set.	*[cpu] PowerPC CPU commands* on page 40
[debug]	System debugger for TX boards. To use [debug], on TX 4000 boards, you must download the debug task (*debug.elf*) to the board. By default, *debug.elf* is commented out in the *ss7load* script. Uncomment the line and run *ss7load* to boot the board and download the task.	*[debug] System debugger commands* on page 41
[eeprom] 5000 Series	EEPROM access commands.	*[eeprom] EEPROM access commands* on page 42
[etsec] 5000 Series	Enhanced Three-Speed Ethernet Controller commands.	*[etsec] Enhanced three-speed Ethernet controller commands* on page 42
[framer]	T1/E1/J1 $framer task information commands for TX boards.	*[framer] T1/E1/J1 $framer task information commands* on page 42
[gpmem]	GPLIB memory diagnostic commands.	*[gpmem] GPLIB memory diagnostic commands* on page 43
[hbus] 4000 Series	Host bus interface commands.	*[hbus] Host bus interface commands* on page 44
[hcomm]5000 Series	Host communication interface commands.	*[hcomm] Host communication interface commands* on page 45
[hweth]	Ethernet hardware-level commands.	*[hweth] Ethernet hardware level commands* on page 45
[ip]	IP (internet protocol) commands for TX boards.	*[ip] IP commands* on page 46
[ipeth]	Low-level information related to the bridge between IP and Ethernet handling.	*[ipeth] IP/Ethernet commands* on page 55

Command set	**Description**	**For more information, refer to...**
[led]	LED control commands.	*[led] LED control commands* on page 55
[m826] 4000 Series	Motorola 8260 commands.	*[m826] Motorola 8260 commands* on page 56
[mail]	Mailbox communication commands for TX boards.	*[mail] Mailbox control commands* on page 56
[mpc] 5000 Series	MPC8568E System-On-Chip register access commands.	*[mpc] MPC8568E system-on-chip commands* on page 57
[mrg]	Memory region commands for TX boards.	*[mrg] Memory region commands* on page 58
[mtp1]	MTP 1 (SS7 level 1) commands for TX boards.	*[mtp1] MTP level 1 commands* on page 59
[nobj]	Named object commands for TX boards.	*[nobj] Named object commands* on page 59
[pcmon]	Program counter monitoring commands.	*[pcmon] program counter monitoring commands* on page 60
[quadfalc]	Infineon QuadFalc T1/E1/J1 framer control for TX boards.	*[quadfalc] Infineon QuadFALC T1/E1/J1 framer control* on page 60
[si]	Serial interface commands.	*[si] Serial interface commands* on page 61
[slog]	System log commands for TX boards.	*[slog] System log commands* on page 61
[swi]	H.100/H.110 and local stream switching control for TX boards.	*[swi] H.100/H.110 and local stream switching control* on page 62
[sys]	System information commands.	*[sys] System information commands* on page 63
[t1e1]	T1/E1/J1 control commands.	*[t1e1] T1|E1|J1 control commands* on page 64
[t81] 4000 Series	Diagnostic interface to the T8100 (H.100/H.110) chip.	*[t81] Diagnostic interface to T8100 chip* on page 65
[t81-low] 4000 Series	Low-level interface to the T8100 (H.100/H.110) chip.	*[t81-low] Low-level interface to T8100 chip* on page 67
[task]	Task control commands.	*[task] Task control commands* on page 68
[timer]	Timer control commands for TX boards.	*[timer] Timer control commands* on page 69
[tsa]	Timeslot assigner.	*[tsa] Timeslot assigner commands* on page 69
[tsi] 5000 Series	H.100/H.110 TSI (time slot interchanger) commands.	*[tsi] H.100/H.110 TSI (time slot interchanger) commands* on page 70
[uart] 5000 Series	UART (Serial Port) commands.	*[uart] UART (serial port) commands* on page 70

Command set	Description	For more information, refer to...
[vsock]	Virtual socket commands for TX boards.	*[vsock] Virtual socket commands* on page 71

[blst] Blast (test packet generator) control commands

The [blst] command set is available on TX 5000 Series boards.

The [blst] commands are used during TX board evaluation to establish a test message flow over the full set of external communication interfaces including host bus (application communication), UDP and SCTP (IP protocols) and MTP1 (over TDM).

The following table lists the [blst] commands:

[blst] commandsv (TX 5000 Series boards)	Description
usage	Display blast command line options.
config	Display blast configuration.
stats	Display blast statistics.
start	Start traffic load test (out VSOCK <ADDR> provided).
size	Set the test packet size.
cycle	Set disable/re-enable cycle time.
reflect	Set reflect wait time (in msecs).
numout	Set number of TXs to keep outstanding.
total	Set total number of packets to send.
bandwidth	Set outbound data bandwidth limit (in kilobits per second).
rxtrigger	Set whether receive triggers next send.
rxignore	Set whether RXs that are ignored are alarmed.
rxlimited	Set whether Limited Pool attached to VSOCK RX.
txcomplete	Set whether TX complete notifications desired.
ipopt	Control use of IP options.
rxipopts	Display last received IP options.

[brdg] Bridge (test packet receiver) control commands

The [brdg] command set is available on TX 5000 Series boards.

The [brdg] commands are used during TX board evaluation to receive and forward a test message flow over the full set of external communication interfaces including host bus (application communication), UDP and SCTP (IP protocols) and MTP1 (over TDM).

The following table lists the [brdg] commands:

[brdg] commands (TX 5000 Series boards)	**Description**
config	Display bridge configuration.
stats	Display bridge statistics (optionally zeroing after display).
side	Configure one side of the bridge.
routereg	Register VSOCK for receipt of route change notifications.
routelog	Log all current routes as alarms.
icmpreg	Register VSOCK for receipt of ICMP error notifications.
start	Start the bridge (optionally acting as a sink).
rxlimited	Set whether Limited Pool attached to VSOCK RX.

[cmd] Console task commands

Use [cmd] commands to perform basic TX board operations through the operator console task. The following table lists the [cmd] commands:

[cmd] commands	**Description**
change	Changes the value of a byte of memory.
dump	Displays a hex/ASCII dump of the specified address (256 bytes shown). This command is also present in the *[con] Standard commands* on page 39.
findascii	Displays all addresses holding ASCII string.
log	Displays all entries in the kernel error log. This command is also present in the *[con] Standard commands* on page 39.
outlines	Displays the number of output lines specified.
setinfo	Displays command set information.
snap	Toggles between snapshot mode and normal mode.

[con] Standard commands

The following table describes the [con] commands for TX boards and the command's associated command set:

[con] commands	Description	Associated command set
arp	Shows an ARP summary or detailed ARP information.	*[ip] IP commands* on page 46
avail	Shows the total amount of memory in use and available.	*[mrg] Memory region commands* on page 58
chan	Lists SS7 channel-specific information.	*[mtp1] MTP level 1 commands* on page 59
channels	Displays all defined channel connections.	*[swi] H.100/H.110 and local stream switching control* on page 62
comm	Displays common communications statistics for all IP layers.	*[ip] IP commands* on page 46
dhcp	Shows (or zero stats, show log, pause, resume) DHCP interfaces.	*[ip] IP commands* on page 46
dump	Displays a hex/ASCII dump of the specified memory address (256 bytes display).	*[cmd] Console task commands* on page 38
eth	Displays Ethernet summary information or detailed information about a specific Ethernet.	*[ip] IP commands* on page 46
ethlog	Manages the Ethernet packet log.	*[ip] IP commands* on page 46
ethtrace	Controls Ethernet data tracing.	*[ip] IP commands* on page 46
frcfg	Displays the configuration of each framer trunk (T1/E1/J1).	*[t1e1] T1|E1|J1 control commands* on page 64
frstats	Displays statistics for each framer trunk (T1/E1/J1).	*[t1e1] T1|E1|J1 control commands* on page 64
frstatus	Displays the status of each framer trunk (T1/E1/J1).	*[t1e1] T1|E1|J1 control commands* on page 64
gateway	Defines the gateway route.	*[ip] IP commands* on page 46
iface	Displays a list of IP interfaces or details for a specific interface.	*[ip] IP commands* on page 46
ifcreate	Defines the IP interface.	*[ip] IP commands* on page 46
ifopt	Specifies an IP interface configuration option.	*[ip] IP commands* on page 46
info	Displays top-level system information.	*[sys] System information commands* on page 63
log	Displays all entries in the kernel error log.	*[cmd] Console task commands* on page 38
mate	Shows the IP address of the TX board's redundant mate board.	*[ip] IP commands* on page 46
mem	Displays total amount of memory in use and available.	*[gpmem] GPLIB memory diagnostic commands* on page 43
ping	Manages pings.	*[ip] IP commands* on page 46

[con] commands	**Description**	**Associated command set**
pmem	Displays percentage of memory assigned to each task.	*[mrg] Memory region commands* on page 58
pools	Displays all memory pools on the system.	*[gpmem] GPLIB memory diagnostic commands* on page 43
proto	Displays a list of IP protocols or details of a specific protocol.	*[ip] IP commands* on page 46
route	Manages the IP routing table.	*[ip] IP commands* on page 46
task	Displays detailed information about the given task.	*[task] Task control commands* on page 68
tasks	Shows a list of currently defined tasks.	*[task] Task control commands* on page 68
time	Provides current date and time (as viewed by the communications processor).	*[sys] System information commands* on page 63
tmem	Displays all memory regions assigned to the specified task.	*[mrg] Memory region commands* on page 58
total	Displays total dynamic memory usage (MEM_INFO_TOTAL).	*[gpmem] GPLIB memory diagnostic commands* on page 43
udp	Displays a list of UDP ports or details for a specific port.	*[ip] IP commands* on page 46
usage	Shows limited pool memory usage.	*[gpmem] GPLIB memory diagnostic commands* on page 43
util	Provides current and average system utilization.	*[sys] System information commands* on page 63

[cpu] PowerPC CPU commands

Use [cpu] commands to see PowerPC CPU information. The following table lists the [cpu] commands:

[cpu] command	**Description**
exc	Displays information about any unexpected exception.

[debug] System debugger commands

Use [debug] commands to debug a TX board-based task. To use [debug], download the debug task (*debug.elf*) to the board. By default, *debug.elf* is commented out in the *ss7load* script. Uncomment the line and run the modified *ss7load* to boot the board and download the task.

The following table lists the [debug] commands:

[debug] commands	Description
asm	Displays area indicated as PowerPC ASM instructions.
break	Sets a breakpoint at the indicated symbol location or address.
breaks	Lists all of the currently defined breakpoints for the selected task.
clear	Removes a breakpoint at the indicated symbol location or address.
clearall	Removes all breakpoints for a particular task.
clrmon	Clears the selected area for monitoring.
dstack	Dumps the stack of the currently selected task as raw hexadecimal data.
info	Displays information about the debug session.
kdstack	Dumps the kernel stack as raw hexadecimal data.
kregs	Displays the kernel registers.
kstack	Dumps the kernel call stack.
ksym	Displays kernel symbol information.
mon	Monitors the indicated area for changes.
regs	Displays registers for the currently selected task or live registers.
regset	Sets the given register to the specified value.
run	Runs the currently selected task.
stack	Dumps the selected task call stack.
s	Steps through the currently selected task (default is step over).
si	Steps into every instruction.
stop	Stops the currently selected task.
sym	Displays symbol information (name, address).
symset	Sets the value of memory at the symbol location indicated.
symshow	Displays the value of memory at the symbol location indicated.
task	Sets the current debug task.

[eeprom] EEPROM access commands

The [eeprom] command set is available on TX 5000 Series boards.

The [eeprom] commands can be used to view the EEPROM contents directly from the TX operating system. Commands used to verify EEPROM integrity across various host related events are also part of the command set.

The following table lists the [eeprom] commands:

[eeprom] commands (TX 5000 Series boards)	**Description**
read	Read a block of data from an EEPROM.
reset	Reset the board.
msi	Generate message signaled interrupt.

[etsec] Enhanced three-speed Ethernet controller commands

The [etsec] command set is available on TX 5000 Series boards.

TSEC refers to any three-speed Ethernet controller (with the three speeds being 10 Megabit/s, 100 or 1000 [= 1 Gigabit]). eTSEC indicates enhanced TSEC which provides features like performance enhancements.

Use the [etsec] command to display information about the registers used to perform low-level control of the three-speed Ethernet controllers. The following table lists the [etsec] command:

[etsec] commands (TX 5000 Series boards)	**Description**
reg	Show all registers related to the given eTSEC. **Syntax** `regs <`***`Ethernet number`***` (1 or 2)>` **Note:** TX 5000 Series boards contain two eTSECs and one 10/100 Ethernet (dual speed).

[framer] T1/E1/J1 $framer task information commands

Use [framer] commands on TX boards to display control information managed by the internal kernel task $framer. The $framer task services all interrupts generated by the framer chips on the board. The following table lists the [framer] commands:

[framer] commands	**Description**
15min	Displays 15-minute performance statistics.
24hr	Displays 24-hour performance statistics.
ctx	Displays $framer task top-level control context.
ring	Displays entries from event information ring.
stats	Displays statistics maintained by $framer task.
status	Displays line status as seen by $framer task.
zstats	Displays zero statistics maintained by $framer task.

[gpmem] GPLIB memory diagnostic commands

GPLIB (general purpose library) memory diagnostic commands provide a detailed breakdown of all memory currently allocated on the TX board. The TX kernel GPLIB manages all memory control on the board. Use the following GPLIB memory diagnostic commands to identify memory corruption issues:

[gpmem] commands	Description
cfg	Displays pool configuration settings.
listchk	Traverses a list verifying list consistency. [v] = verbose.
chkpools	Finds all GPLIB pools. Checks the following: [l]: Lists [P]: BKs [z]: Zero-fill
mem	Shows memory usage. This command is also present in the *[con] Standard commands* on page 39.
memwalk	Walks a memory chain (following link pointers).
pbk	Displays memory area as a !pbk (pooled memory block).
pool	Displays memory area as a !poo (pooled memory descriptor).
pools	Displays all memory pools on the system, including: • Pool Name: Name describing the purpose of the memory pool. • Free: Number of buffers assigned to the pool but not currently allocated. • State: Memory depletion state (for limited pools). • Curr: Number of buffers currently allocated from the pool. • Max: Maximum number of buffers ever allocated (maximum value of Curr). • Total: Total number of buffer allocations made using the pool. • Failed: Number of times an allocation attempt from the pool failed. This command is also present in the *[con] Standard commands* on page 39.
prg	Displays memory area as a !prg (pooled memory region).
total	Shows total dynamic memory usage (MEM_INFO_TOTAL). This command is also present in the *[con] Standard commands* on page 39.
usage	Displays limited pool memory usage. This command is also present in the *[con] Standard commands* on page 39.

[hbus] Host bus interface commands

The [hbus] command set is available on TX 4000 Series boards.

The following host bus interface commands provide detailed information about packet flows between the TX board and the host system. Use this information when debugging communication problems between host-based applications and TX-based tasks.

[hbus] commands (TX 4000 Series boards)	**Description**
dec	Displays DEC 21555 information.
info	Displays Hbus-specific control information.
lev1ctx	Displays level 1 driver context information.
pcidev	Shows all local PCI devices.
pciread	Reads a U32 from PCI configuration space.
pciwrite	Writes a U32 to PCI configuration space.
pldread	Reads a byte from PLD (programmable logic device) global registers.
pldwrite	Writes a byte to PLD global registers.
res	Displays resource information.
restats	Displays Hbus resource statistics.
rxchan	Shows receive routing information (per de-multiplexing channel).
stats	Displays Hbus statistics.
vsock	Displays vsock information.

[hcomm] Host communication interface commands

The [hcomm] command set is available on TX 5000 Series boards.

The following host communication commands provide detailed information about packet flows between the TX board and the host system. Use this information when debugging communication problems between host-based applications and TX-based tasks.

[hcomm] commands (TX 5000 Series boards)	Description
info	Show HCOMM control information.
config	Show HCOMM configuration information.
status	Show current HCOMM status.
stats	Show HCOMM statistics [& zero].
errors	Show errors reported by level-1 driver [& zero].
rxchan	Show receive channel statistics [& zero].
wdctl	Show Work Descriptor control information
wdtx	Show transmit Work Descriptor(s).
wdrx	Show receive Work Descriptor(s).
res	Show HCOMM resource information.
vsock	Show HCOMM VSOCK information.

[hweth] Ethernet hardware level commands

Use [hweth] commands to see Ethernet hardware level information.

TX 4000 Series boards

The following table lists the [hweth] commands for TX 4000 Series boards:

[hweth] commands (TX 4000 Series boards)	Description
i82551	Displays Ethernet control information for the Intel 82551 chip.
i82551bds	Displays Ethernet (type Intel 82551) RX and TX BD tables.
ethctx	Displays an Ethernet control context.

TX 5000 Series boards

The following table lists the [hweth] commands for TX 5000 Series boards:

[hweth] commands (TX 5000 Series boards)	Description
ethctx	Displays an Ethernet control context.
mpc8568	Display Ethernet Control Info specific to MPC8568e chip.
mpc8568bds	Display Ethernet (type MPC8568e) RX and TX BD tables.
regs	Display Ethernet registers.

[IP] IP commands

This topic describes the IP (Internet protocol) commands. The [ip] commands, with the exception of vsock, are also present in the *[con] Standard commands* on page 39. Refer to the *Dialogic® NaturalAccess" Signaling Software Configuration Manual* for information about IP control.

mate

Sets the IP address of the TX board's redundant mate board using the following syntax:

```
mate [ IP address | NONE ]
```

where:

Value	Description
IP address	IPv4 address in dot notation, for example 1.2.3.4.
NONE	No mate IP address (default).

The following example indicates that the TX board's redundant mate is at address 10.1.1.2

```
mate 10.1.1.2
```

Use the mate command with no parameter to view the currently assigned mate address:

```
mate
```

Refer to the *Dialogic® TX Series SS7 Boards Health Management Developer's Reference Manual* for information about SS7 redundancy.

dhcp

Shows information (optionally zeroing statistics) related to DHCP protocol handling. A DHCP interface is enabled for each Ethernet interface that is created as type DHCP [ifcreate <ethNum> DHCP].

```
dhcp [<interface number>|* [ZERO|LOG|PAUSE|RESUME]]
```

The following table describes the DHCP command options:

Option	Description
dhcp	Shows a summary display for each DHCP interface.
dhcp 1	Shows a full display for Ethernet number 1s DHCP.
dhcp 1 zero	Shows Ethernet 1 full display then zero Ethernet 1s statistics.
dhcp * zero	Shows a summary of each DHCP, then zero all DHCP statistics.
dhcp 1 log	Shows a full display, followed by the DHCP State/Event log.
dhcp 1|* pause	Pauses all DHCP processing for a specific Ethernet or all Ethernets.
dhcp 1|* resume	Resumes all DHCP processing for a specific Ethernet or all Ethernets.

udp

Shows a list of UDP ports or details for a specific port (optionally zeroing statistics immediately after they have been displayed) using the following syntax:

```
udp [<UDP port>|* [ZERO]]
```

proto

Shows a list of IP protocols or details for a specific protocol (optionally zeroing statistics immediately after they have been displayed) using the following syntax:

```
proto [<protocol number>|* [ZERO]]
```

ifopt

Specifies an IP interface configuration option using the following syntax:

```
ifopt [!]<optName>[ = <optValue>]
```

where bit options are specified as:

Value	Description
optName	***optName*** indicates that the option is enabled. ***!optName*** indicates that the option is disabled.
optValue	Value options are specified as: `optName = optValue`

The following table describes the options that can be specified using the ifopt command:

Option	Type	Description
PINGBLOCK	BIT	Indicates if automatic responses to PING messages are blocked.
MTU	VALUE	Maximum transmission unit (in bytes).
FRAGDROP	SECS	Partial fragment drop timeout.
HOPLIMIT	VALUE	Not applicable in this release.
ICMPRATE	VALUE	Not applicable in this release.
IPV6	BIT	Not applicable in this release.
IPSEC	BIT	Not applicable in this release.
NDATTEMPTS	VALUE	Specifies the neighbor discovery attempt (NDA) limit for the specified Ethernet interface.
NDRETRAN	SECS	Neighbor discovery retransmission timeout.
NDREACH	SECS	Not applicable in this release.
ARPRETRY	MSECS	Amount of time to wait after issuing an ARP request before re-transmitting the same ARP request.
ARPMAX	VALUE	Maximum number of times to retry an ARP request before terminating retry attempts, considering the remote entity is unreachable.
ARPREACH	MSECS	Maximum amount of time without receiving any message from the remote IP address before probing to verify connectivity.
ARPUNUSED	MSECS	Amount of time that an unused ARP entry remains in the ARP table. An ARP entry is used each time an outbound IP packet is sent to the remote entity for the given ARP entry.

Examples

Enter the following command to block responses to received PING requests:

```
ifopt pingblock
```

Enter the following command to drop all fragments if incomplete after 10 seconds:

```
ifopt fragdrop 10
```

ifcreate

Defines an IP interface that uses either:

- DHCP to determine IP address and mask, or
- A fixed IP address and mask

An interface is created with default options unless options were set using a previous ifopt command.

Using DHCP

The ifcreate command syntax using DHCP is:

```
ifcreate intfNum DHCP
```

where:

Value	Description
intfNum	1-based Ethernet interface number.
DHCP	Obtains the IP address, network mask, and default gateway.

The following example shows an interface that uses DHCP:

```
ifcreate 1 dhcp
```

Not using DHCP

The ifcreate command syntax when not using DHCP is:

```
ifcreate iftfNum IPaddr mask
```

where:

Value	Description
intfNum	1-based Ethernet interface number.
IPaddr	IP address to assign to the interface.
mask	IP subnet mask.

The following example shows an interface with a specific address and mask:

```
ifcreate 2 10.3.9.15 255.255.255.0
```

iface

Shows a list of IP interfaces or details for a specific interface (optionally zeroing statistics immediately after they have been displayed) using the following syntax:

```
iface [<IP address>|<ifNum>|* [ZERO]]
```

gateway

Defines a gateway route using the following syntax:

```
gateway IPaddress mask gatewayAddress
```

where:

Value	Description
IPaddress	IPv4 address in dot notation. For example, 1.2.3.4.
mask	Mask associated with IP address, in dot notation.
gatewayAddress	IPv4 address, in dot notation, of routing gateway.

The following example sends all traffic not covered by a more specific route to the gateway at 10.1.0.1:

```
gateway 0.0.0.0 0.0.0.0 10.1.0.1
```

route

Manages the IP routing table using the following syntax:

```
route [<command> [<IP addr> <mask> <iface>|<GW addr>]]
```

where:

Value	Description
command	Executes one of the following commands: add, delete, get, or table. The default value is table (or print), which shows the route table.
IP addr	IP address in dot notation. For example, 10.3.9.15.
mask	IP subnet mask associated with IP address, in dot notation.
iface	Interface number, for example (1..n).
GW addr	IP address, in dot notation, of routing gateway. For example, 10.3.9.1.

The following example shows how to add an IP address to interface 1:

```
route add  10.3.9.15  255.255.255.0   1
```

The following example adds a gateway router:

```
route add  11.0.0.0   255.0.0.0      10.3.9.1
```

The following example displays the routing table:

```
route table
```

arp

Shows information (optionally zeroing statistics) related to ARP protocol handling using the following syntax:

```
arp [<interface number (1-based)|*> [<remote IP>|*] [ZERO|LOG|DELETE]]
```

The ARP protocol translates from a destination IP address to a physical Ethernet address. An ARP entry is automatically created for each IP-to-Ethernet address pair being tracked by the ARP layer.

The following table describes the arp command options:

Option	Description
arp	Shows top-level ARP statistics, followed by a summary for each ARP entry that exists in the ARP table.
arp 1	Shows top-level statistics with a summary for Ethernet 1 ARP entries only.
arp 1 1.2.3.4	Shows information about the ARP entry tracking remote IP address 1.2.3.4 over Ethernet 1.
arp 1 1.2.3.4 log	Show ARP entry information followed by a dump of the ARP entries state/event log.
arp 1 1.2.3.4 zero	Shows ARP entry information followed by zeroing the ARP entry statistics.
arp 1 1.2.3.4 delete	Immediately removes the ARP entry from the table.
arp * * zero	Shows top-level statistics with a summary for each ARP entry (zeroing all ARP entry statistics after display).
arp * * delete	Purges the ARP table (deletes current ARP entries).

eth

Shows summary information for each Ethernet or detailed information for a specific Ethernet (optionally zeroing statistics after display). The command can also be used to pause an Ethernet (treated as if a cable was pulled), and later resume that Ethernet as a diagnostic tool.

```
eth [<interface number (1-based)|*> [ZERO|PAUSE|RESUME]]
```

The following table describes the eth command options:

Option	Description
eth	Displays a summary of each Ethernet interface.
eth 1	Displays information about Ethernet 1.
eth 1 zero	Displays information about Ethernet 1 and then immediately zeroes Ethernet 1 statistics.
eth 1 pause	Pauses Ethernet 1 as if a cable was removed. Use for diagnostic purposes only.
eth 1 resume	Resumes a previous pause on Ethernet 1. Simulates replacing a cable.
eth * zero	Displays Ethernet summary information, and then immediately zeroes all Ethernet statistics.
eth * pause	Pauses all Ethernets as if a cable was removed. Use for diagnostic purposes only.
eth * resume	Resumes a previous pause on any Ethernet. Simulates replacing a cable.

ethcfg

Ethernet interfaces can be configured using the following syntax:

```
ethcfg <Ethernet number> <speed> <duplex> [<mdix control>]
```

where:

Value	Description
Ethernet number	1-based number of Ethernet interface.
speed	Interface speed in megabits/sec: 10 | 100 Deafult: Auto-negotiate up to maximum speed supported by all devices.
duplex	HALF | FULL
mdix control	NOMDIX | MDIX (Optional parameter) Controls swapping of transmit and receive pairs to avoid use of a crossover cable. Default: NOMDIX, No swapping of TX and RX pairs

The default Ethernet configuration is to use auto-negotiation, negotiate for the fastest speed possible, and to perform pair swap detection. Certain TX board Ethernet interfaces support a maximum speed of 100 Mb/s while other Ethernet interfaces support up to 1 Gigabit/s. Do not specify the ETHCFG command for any interface where Gigabit speed is desired, allowing auto-negotiation to select the speed.

ethlog

Manages the Ethernet packet log using the following syntax:

```
ethlog [<command> [<param(s)>]]
```

where <***command***> is one of the following:

Value	Description
Not specified	Dumps the log entry header for each packet in the log. Enter the following command to display information and display log header lines: `ethlog`
start	Starts logging Ethernet packets. For example: `ethlog start`
stop	Stops logging Ethernet packets.

info	Displays top-level information [<***param***>=ZERO to clear stats].
	Enter the following command to display information about the Ethernet log:
	`ethlog info`
	Enter the following command to display information and then clear the statistics:
	`ethlog info zero`

dump	Dumps each log entry. Use ***param*** to limit the dump size.
	Enter the following command to dump the entire log:
	`ethlog dump`
	Enter the following command to dump the entire log but limit each entry to 34 bytes:
	`ethlog dump 34`

ethtrace

Controls tracing of Ethernet packets sent and received to tracing applications (such as *ss7trace* or *txalarm_*) using the following syntax:

```
ethtrace [<interface number (1-based)|*> [ON [<host chan>]|OFF|INFO]]
```

The following table shows examples of ethtrace:

Command	Description
ethtrace ethtrace * info	Shows how each Ethernet interface is configured for tracing.
ethtrace 1 on	Activates tracing on Ethernet interface number 1. Packets are traced to the default host channel [47 = *ss7trace* channel].
ethtrace * on	Activates tracing on all Ethernet interfaces (tracing to default host channel).
ethtrace 2 on 12	Activates tracing on Ethernet interface number 2 with all trace packets sent to host channel 12 (txalarm_).
ethtrace * off	Deactivates tracing on all Ethernet interfaces.

Refer to *Receiving data traces from TX based tasks: ss7trace* on page 22 and *Receiving alarms from TX based tasks: txalarm* on page 19 for more information.

You can also use the *txconfig* host-based utility to control Ethernet tracing. Refer to the *Dialogic® NaturalAccess™ Signaling Software Configuration Manual* for more information.

ping

Manages PINGs using the following syntax:

```
ping [<remote IP>|* [INFO|CLEAR|STOP|START] [-<option> [<param>]]
```

where:

remote IP is the remote IP address that uniquely identifies each ping.

The following table describes the ping commands:

Command	Description
INFO	Show details about a specific ping or a summary indicated with an asterisk (*).
CLEAR	Clears a previous PING from memory, which implies STOP.
STOP	Stops the specific ping. An asterisk (*) stops all pings.
START	Starts sending pings to a given remote IP address.

If you use ping without specifying any command, the resulting action depends on whether or not there is already a ping in progress for the specified ***remote IP*** address. If a ping for the specified ***remote IP*** address already exists, the ping is treated as an INFO command. If no ping exists for the specified ***remote IP*** address, the PING is treated as a START command.

The following table describes the START options:

Option	Description
-n ***count***	Stops issuing additional requests after sending and receiving ***count*** pings. The default value is an infinite count (send until stopped).
-i ***interval***	Issues the next ping every ***interval*** ms. The default value is 1000 ms (one ping request per second).
-w ***wait***	Expects ping reply within ***wait*** ms (else late). The default value is 30000 ms (wait 30 seconds).
-z ***size***	Sets the number of data bytes in ping. The default value is 56.
-p ***pattern***	Uses the provided pattern as a fill pattern. The default value is to fill with ASCII text message.
-l ***ttl***	Sets the time-to-live. The default value is 60.
-s ***tos***	Sets the type-of-service. The default value is 0.

-m ***type***	Sends ICMP requests of a given message where ***type*** can be one of the following values:
	ECHO = Sends echo requests (expect ECHO replies). This is the default value.
	TIME = Sends timestamp requests (expect timestamp replies).
	INFO = Sends information requests (expect information replies).

-d	Sets the don't fragment flag.
-r	Use the IP option of RECORD ROUTE.
-t	Use the IP option of RECORD TIMESTAMPS.

vsock

Displays control information for an IP VSOCK or set of VSOCKs using the following syntax:

```
vsock UDP|PROTO|IFACE [<ID number>]
```

vsock is the only [ip] command that is not present in the [con] -Standard commands since this command provides information that is targeted for low-level diagnostics.

comm

Displays common communications statistics for IP layers (DHCP, UDP, PROTO, IFACE, ARP, and ETH) using the following syntax:

```
comm [DHCP|UDP|PROTO|IFACE|ARP|ETH|* ZERO]
```

Displaying a statistical summary for all IP layers

Use the comm command with no parameters to display a summary of all statistics for each IP layer as shown in the following example:

```
===============================================================================
Layer     Rx Frames  Tx Frames  | Rx Errors  Rx Last    | Tx Errors  Tx Last
========= ========== ========== | ========== ========== | ========== ==========
     DHCP          0       5128 |          0 0x00000000 |          0 0x00000000
-------------------------------------------------------------------------------
      UDP          0       5128 |       5128 0x1C00001C |          0 0x00000000
-------------------------------------------------------------------------------
 PROTOCOL       5128       5128 |          0 0x00000000 |          0 0x00000000
-------------------------------------------------------------------------------
INTERFACE       5128       5128 |          0 0x00000000 |          0 0x00000000
-------------------------------------------------------------------------------
      ARP          0          0 |          0 0x00000000 |          0 0x00000000
-------------------------------------------------------------------------------
 ETHERNET       5128       5128 |          0 0x00000000 |          0 0x00000000
-------------------------------------------------------------------------------
```

The example shows that 5128 DHCP packets were transmitted with no DHCP packets received. There were 5128 UDP packets transmitted (each DHCP packet is transmitted as a UDP packet). This example also shows that there were 5128 UDP receive errors. The last error code associated with a UDP receive error is listed in the RX Last column as 0x1C00001C. The RX Last and TX Last columns hold the last error code encountered by the given layer (either for receive or transmit handling). Use the *txccode* utility to obtain a text description of the error code.

In this case, *txccode* outputs the following information:

```
txccode 0x1c00001c
SCCIP_NO_PORT_RX_ROUTE - no route for received protocol-specific port number
```

Refer to *Viewing completion code descriptions: txccode* on page 30 for more information.

Displaying a statistical summary for a specific IP layer

The following comm command shows information about common communications statistics at the UDP layer:

```
comm udp
```

The following output displays:

```
===============================================================================
UDP:

  COMM: Rx Frames           :          0  |  Tx Frames           :       5128
        Rx Bytes            :          0  |  Tx Bytes            :    1436400
        Rx Fails            :       5128  |  Tx Fails            :          0
        Rx LastReason       : 0x1C00001C  |  Tx LastReason       : 0x00000000
        Rx LastInfo         : 0x00000043  |  Tx LastInfo         : 0x00000000
```

Information similar to the statistical summary displays with the addition of RX LastInfo and TX LastInfo. These additional values provide information that is specific to the LastReason error code. For the SCCIP_NO_PORT_RX_ROUTE error code, RX LastInfo holds the specific UDP port number where the last received UDP packet was destined.

LastInfo is a hexadecimal value regardless of the meaning of the field. Since UDP port numbers are presented in decimal format, you must convert the LastInfo fields to decimal, for example, 0x43 = 67. Therefore, the last received unroutable UDP packet had a destination UDP port number of 67.

Displaying and zeroing the layer statistics

The following comm command displays UDP layer common communication statistics and then immediately zeroes the statistics:

```
comm udp zero
```

The following comm command displays a summary of all statistics for each IP layer and then immediately zeroes the communications statistics for all layers:

```
comm * zero
```

[ipeth] IP/Ethernet commands

Use [ipeth] commands to see low-level information related to the bridge between IP and Ethernet handling. The following table lists the [ipeth] commands:

[ipeth] commands	Description
ethinfo	Shows Ethernet summary information.
netinfo	Shows network level information.
res	Displays a list of resources in use for IP-over-Ethernet.
vsock	Displays a list of vsocks (virtual sockets) in use for Ethernet.

[led] LED control commands

Use [led] commands to test the function of the LED indicators maintained by the TX operating system. The following table lists the [led] commands:

[led] commands	Description
ledlog	Show last value written as led log (or log specified code/loc).
all	Set value of all LEDs.
mode	Show (or set) LED operating mode.
map	Map internal LEDs to face plate.
set	Show (or change) current LED setting.
names	Show names of LEDs.

[m826] Motorola 8260 commands

The [m826] command set is available on TX 4000 Series boards.

Use [m826] commands to obtain more information about the Motorola 8260 commands.

The following table lists the [m826] commands:

[m826] commands (TX 4000 Series boards)	Description
info	Shows MPC8260 top-level information.
si	Shows the current SI RAM tables (receive and transmit).
shadow	Shows the shadow SI RAM tables (receive and transmit).
siregs	Shows all registers related to the serial interface.
si1ram	Modifies the half of the SI 1 RAM that is not active.
si2ram	Modifies the half of the SI 2 RAM that is not active.
si1reg	Modifies the SI 1 register.
si2reg	Modifies the SI 2 register.
siu	Displays the system interface unit configuration.
tsaenab	Enables the timeslot assigner.
bus	Shows the bus transfer control information.
dma	Shows the serial DMA (SDMA) and independent DMA (IDMA).

[mail] Mailbox control commands

TX-based tasks use mailbox messages to perform all inter-process communications (IPC). Use [mail] commands to monitor all TX board mailbox communications. The following table lists the [mail] commands:

[mail] commands	Description
res	Displays a list of resources.
vsock	Displays a list of vsocks.

[mpc] MPC8568E system-on-chip commands

The [mpc] command set is available on TX 5000 Series boards.

Use [mpc] commands to display register sets that are used to control specific functions within the main SoC features. The following table lists the [mpc] commands:

[mpc] commands **(TX 5000 Series boards)**	**Description**
info	Show top-level Information for the MPC8568E SoC.
law	Show local access windows (all or window specified).
ecm	Show e500 coherency module.
dram	Show overall DRAM controller or show the array value or values.
i2c	Show the I2C device controller.
uart	Show the UART device controller.
lbus	Show overall local bus controller or show the array value or values.
ocean	Show the ocean controller.
pci	Show overall PCI controller or show the array value or values.
pex	Show overall PCI Express controller or show the array value or values.
l2cache	Show overall level 2 cache or show the array value or values.
dma	Show overall DMA or show specified DMA channel.
etsec	Show overall eTSEC device or show the array value or values.
tlu	Show overall table lookup unit or show the array value or values.
sec	Show security engine.
pic	Show overall programmable Int. Ctlr or the value or values.
srio	Show serial rapid IO.
devutils	Show overall device-specific utilities or value or values.
general	Dump the entire MPC8568E general utilities space.
ccsr	Dump the entire MPC8568E CCSR memory space.

[mrg] Memory region commands

Use [mrg] commands to display information about memory usage on TX boards. All memory is managed using memory region descriptors (RGDs). The following table lists the [mrg] commands:

[mrg] commands	**Description**
allmem	Displays assignments for all memory regions.
avail	Displays total amount of memory in use and available. This command is also present in the *[con] Standard commands* on page 39.
full	Displays control information at the time of the first full indication.
info	Displays top-level region information.
pmem	Displays the percentage of memory assigned to each task. This command is also present in the *[con] Standard commands* on page 39.
rgd	Displays a memory area as an RGD (memory region descriptor).
rgdwalk	Traverses an RGD list and displays each region descriptor (RGD).
tmem	Provides all memory regions assigned to the specified task, including: ● Owner: Name of task. *KERNEL if no task is specified. ● Name: Name of memory region or address if not named. ● Start: Starting memory address of region. ● End: Address of last byte in memory region. ● Size: Total number of bytes in region. ● Access: Memory access flag. This command is also present in the *[con] Standard commands* on page 39.

[mtp1] MTP level 1 commands

Use [mtp1] commands to monitor the SS7 level 1 interface provided by the TX kernel on TX boards. The following table lists the [mtp1] commands:

[mtp1] commands	Description
cfg	Displays SS7 channel configuration.
chan	Displays MCC channel-specific information. This command is also present in the *[con] Standard commands* on page 39.
info	Displays MTP1Info data structure contents.
lvl1	Displays MTP1 level 1 layer information.
mcc	Displays MCC-specific information.
mcode	Displays microcode revision information.
mtp1log	Displays $mtp1 log entries. This command is also present in the *[con] Standard commands* on page 39.
pstats	Displays SS7 protocol-specific statistics. This command is also present in the *[con] Standard commands* on page 39.
res	Displays a list of resources.
rxring	Displays SS7 channel receive ring.
ticker	Alters the periodic ticker configuration.
trigger	Controls $mtp1 log triggers. This command is also present in the *[con] Standard commands* on page 39.
txring	Displays SS7 channel transmit ring.
vsock	Displays vsock information.

[nobj] Named object commands

Use [nobj] commands on TX boards to display named objects. The following table lists the [nobj] commands:

[nobj] commands	Description
info	Shows top-level named object information.

names	Displays all named objects defined in the system, including:
	RES = Resources
	VSOCK = Virtual sockets
	MEM = Memory areas
	POOL = Memory pools

[pcmon] program counter monitoring commands

[pcmon] commands allow developers to activate and view the results of program counter monitoring. The TX operating system can track the value of the program counter throughout the execution code space as a means of identifying bottlenecks during performance review. The following table lists the [pcmon] commands:

[pcmon] commands	Description
zoomaddr	Set zoom region (or turn off zoom counting).
start	Start [or restart] PC mon (clearing all buckets).
stop	Stop PC monitoring.
show	Show PC mon buckets [limit by percent of samples].
zoom	Show zoom region [limit by percent of zoom samples].

[quadfalc] Infineon QuadFALC T1/E1/J1 framer control

Use [quadfalc] commands to communicate directly with the T1/E1/J1 framer chips on TX boards. The following table lists the [quadfalc] commands:

[quadfalc] commands	Description
get	Retrieves the contents of a register.
names	Displays the names of registers for a given framer type.
pulse	Decode pulse shape from register values.
regs	Displays all framer registers.
rres	Resets the receiver.
set	Sets the contents of a register.
sres	Resets the signaling transmitter.
xres	Resets the transmitter.

[si] Serial interface commands

Use [si] commands to view the current configuration of the serial interface. The following table lists the [si] commands:

[si] commands	Description
siregs	Show all registers related to serial interface.
sitbl	Show the current SI RAM tables (Rx and Tx).
shadow	Show the shadow SI RAM tables (Rx and Tx).
si1ram	Modify the SI 1 RAM (whichever half is not active).
si2ram	Modify the SI 2 RAM (whichever half is not active).
si1reg	Modify the SI 1 register.
si2reg	Modify the SI 2 register.

[slog] System log commands

The TX system log provides the following types of logging: alarms, printf, and trace. Each type of logging is managed separately. Log records are stored on the TX board until they are forwarded to a host-based collection utility. The *txalarm* utility collects alarm logs. The printf and trace log entries are used only during development.

The following table lists the [slog] commands:

[slog] commands	Description
alarm	Displays all alarm logs.
info	Displays logging control information.
printf	Displays all printf logs.
ring	Displays ring information.
stalog	Displays ring state/event log.
stats	Displays statistics and optionally zeros them.
trace	Displays all trace logs.

[swi] H.100/H.110 and local stream switching control

Use [swi] commands to manually control TX switching. Many of the commands correspond to functions in the host-based TX SWI library. The following table lists the [swi] commands and the corresponding TX SWI functions:

[swi] commands	Description	TX SWI library functions
bdinfo	Displays TX board information.	No equivalent.
break	Breaks a half-duplex connection.	**txswiDisableOutput**
beakall	Breaks all connections.	**txswiDisableOutput**
caps	Queries the H.100/H.110 switch capabilities.	**txswiGetSwitchCaps**
channels	Displays all defined channel connections. This command is also present in the *[con] Standard commands* on page 39.	**txswiGetOutputState**
clock	Sets the H.100/H.110 clock registers.	**txswiConfigBoardClock**
connect	Makes a half-duplex connection.	**txswiMakeConnection**
disable	Disables the H.100/H.110 switch.	No equivalent.
enable	Enables the H.100/H.110 switch.	No equivalent.
getclock	Retrieves the current clocking configuration.	**txswiGetBoardClock**
netref	Sets the H.100/H.110 NETREF signal.	**txswiConfigNetrefClock**
pattern	Issues a repeating character pattern.	**txswiSendPattern**
pmatch	Displays all connections outputting a pattern.	**txswiGetOutputState**
port	Makes a full-duplex connection.	**txswiConfigLocalTimeslot**
ports	Displays all defined communications ports.	**txswiGetLocalStreamInfo**
query	Queries the output connection configuration.	**txswiGetOutputState**
reset	Resets the H.100/H.110 switch.	**txswiResetSwitch**
sample	Displays a data sample from the connection source.	**txswiSampleInput**
speed	Adjusts the speed of the local communications port.	**txswiConfigLocalTimeslot**
status	Displays the current status of the switch.	No equivalent.

You can also use the host-based utility *txconfig* to control switching. Refer to the *Dialogic® NaturalAccess™ Signaling Software Configuration Manual* for more information. For more information on the TX SWI library, refer to the *Dialogic® TX Series SS7 Boards TDM for SS7 Developer's Reference Manual.*

[sys] System information commands

The following table describes the system information commands:

[sys] commands	Description
clock	Shows clock information.
cpkctx	Shows CPK control context.
dip	Shows the current DIP switch settings.
eeprom	Shows the EEPROM contents.
flash	Shows flash information.
general	Shows general system information.
info	Displays system information (SysInfo) block. This command is also present in the *[con] Standard commands* on page 39.
lietempc	Sets CPU temperature in C (state 0..4) [TEST].
lietempb	Sets Board temperature in C (state 0..4) [TEST].
liefan	Sets fan state (START|STOP) [TEST].
mon	Shows monitored system status (optionally clearing history).
msecs	Sets milliseconds-since-boot [TEST].
time	Shows the date and time. This command is also present in the *[con] Standard commands* on page 39.
util	Shows CPU utilization. This command is also present in the *[con] Standard commands* on page 39.

[t1e1] T1|E1|J1 control commands

The [t1e1] command set provides full control of all T1/E1/J1 capabilities on the board. Use the following commands to diagnose TDM communication problems on the T1, E1, or J1 interfaces:

[t1e1] commands	Description
ais	Transmits an alarm indication signal.
chanloop	Places a single framer channel in loop mode.
ctx	Displays the T1/E1/J1 top-level control context.
defect	Inserts single bit defects.
disable	Disables the framer.
e1cfg	Configures the framer as E1 mode.
e1opt	Sets the E1 configuration option.
enable	Enables the framer.
frcfg	Shows the current framer configuration. This command is also present in the *[con] Standard commands* on page 39.
frstats	Shows statistics for the framer. This command is also present in the *[con] Standard commands* on page 39.
frstatus	Shows the current framer status. This command is also present in the *[con] Standard commands* on page 39.
info	Displays top-level information for each framer.
j1cfg	Configures the framer as J1 mode.
j1opt	Sets the J1 configuration option.
llbdown	Transmits a line loopback deactivate code.
llbup	Transmits a line loopback activate code.
loop	Places the entire framer in loop mode.
prbs	Transmits a pseudo-random bit sequence.
ra	Transmits a remote alarm.
resync	Forces framer resynchronization.
sim	Initiates an alarm simulation or advances to the next test.
t1cfg	Configures the framer as T1 mode.
t1opt	Sets the T1 configuration option.

[t81] Diagnostic interface to T8100 chip

The [t81] command set is available on TX 4000 Series boards.

T8100 commands provide a direct interface to the H.100/H.110 switching hardware, which is controlled through the T8100 chip on the TX board. These commands help you analyze TDM connections established with the *tdmcfg* utility, the *txconfig* utility, the TX SWI library, and the TDM libraries, including commands entered through the *t1demo* application.

This topic presents:

- [t81] commands
- TDM pattern test commands

[t81] commands

[t81] commands (TX 4000 Series boards)	**Description**
check	Checks received patterns across the stream range. Used to test H.100/H.110 or T1/E1 connectivity to another TX board. Refer to *TDM pattern test commands* on page 66.
ctldb	Displays the Ctrl db.
fallback	Forces into clock fallback mode or returns to normal.
info	Displays top-level T8100 control information.
local	Displays all currently defined local connections.
looptrunks	Loops all T1/E1 trunks back on themselves.
reflect	Reflects timeslots across a given stream range. Used to test H.100/H.110 or T1/E1 connectivity to another TX board. Refer to *TDM pattern test commands* on page 66.
spread	Spreads patterns across a stream range. Used to test H.100/H.110 or T1/E1 connectivity to another TX board. Refer to *TDM pattern test commands* on page 66.
stats	Displays statistics on the T8100.
status	Displays the current status of the T8100.
vreflect	Verifies CAM (content addressable memory) definitions (reflecting timeslots). Used to test H.100/H.110 or T1/E1 connectivity to another TX board. Refer to *TDM pattern test commands* on page 66.
vspread	Verifies CAM definitions (spreading patterns). Used to test H.100/H.110 or T1/E1 connectivity to another TX board. Refer to *TDM pattern test commands* on page 66.

TDM pattern test commands

Use the following [t81] commands to verify TDM (H.100/H.110 or T1/E1 connectivity) between two TX boards:

Command	Description
check	Checks received patterns across a stream range using the following syntax: `check` ***bus min stream*** [***max stream*** [***ts delta*** [***pattern***]]]
reflect	Reflects timeslots across a stream range using the following syntax: `reflect` ***bus min stream*** [***max stream*** [***ts delta***]]
spread	Spreads patterns across a stream range using the following syntax: `spread` ***bus min stream*** [***max stream*** [***ts delta*** [***pattern***]]]
vreflect	Verifies CAM definitions (reflecting timeslots) using the following syntax: `vreflect` ***bus min stream*** [***max stream*** [***ts delta***]]
vspread	Verifies CAM definitions (spreading patterns) using the following syntax: `vspread` ***bus min stream*** [***max stream*** [***ts delta*** [***pattern***]]]

The following parameters apply to each command:

Parameter	Description

bus	Valid values:
	H = H.100/H.110 bus (resource direction).
	N = H.100/H.110 bus (network direction).
	L = LOCAL bus.

min stream	Minimum stream number (paired with ***min***+1 for H.100/H.110 bus).
max stream	Maximum stream number (paired with ***max***-1 for H.100/H.110 bus). Default: ***min stream***+1.
ts delta	Delta used to reach next timeslot. Default: 1. **Note:** Each time the stream is incremented (by 2), the next timeslot is reset to 0.
pattern	Constant pattern to send or expect. If not specified, then a range of patterns is used.

Follow this procedure to use the pattern test commands:

Step	Action
1	Connect two TX boards with either an H.100/H.110 cable or a T1/E1 crossover cable.
2	Download a TDM configuration file to each board specifying TDM clocking configuration (and T1 or E1 configuration if testing T1/E1).
3	On one TX board, use the reflect command to define a set of connections that reflect patterns received over H.100/H.110 (or T1/E1) and transmitted over H.100/H.110 (or T1/E1).
4	On the other board, use the spread command to spread a set of outbound patterns over the interface being tested.
5	Use the check command on the same TX board that the spread command was entered on to verify that the proper reflected patterns are received.
6	Use the vreflect and vspread commands to verify that all connections made using the corresponding command (reflect or spread) are actually in the TDM switching control database.

[t81-low] Low-level interface to T8100 chip

The [t81-low] command set is available on TX 4000 Series boards.

T8100 low-level interface commands perform low-level manipulations to the T8100 chip, which is used for H.100 and H.110 switching control. Use the following low-level commands to test and analyze the internal behavior of the T8100 interface:

[t81-low] commands TX 4000 Series boards)	Description
ramr	Reads the value in the AMR (address mode register).
ridr	Reads the value in the IDR (indirect data register).
rlar	Reads the value in the LAR (lower address register).
rmcr	Reads the value in the MCR (master control register).
showregs	Displays the status of all registers.
wamr	Writes a value to the AMR register.
wctlreg	Writes a value to the given control register.
widr	Writes a value to the IDR register.
wlar	Writes a value to the LAR register.
wmcr	Writes a value to the MCR register.

[task] Task control commands

The following table describes the task control commands:

[task] commands	Description
info	Shows top-level task control information.
rings	Lists tasks (head and tail) on each task priority ring.
sqe	Displays a memory area as a task SQE (service queue element).
task	Shows detailed information about the named task. This command is also present in the *[con] Standard commands* on page 39.
tasks	Displays all created tasks. Information includes: • Name: Name of the given task. A leading \$\ indicates kernel task. • Prior: Priority of task. 1 = highest priority. 31 = lowest. • State: Current state of task. Valid states include: • Created: Task was created but never executed. • Stalled: Kernel detected invalid instruction (usually breakpoint during debug session). • Waiting: Task is asleep and waiting for work. • Blocked: Higher priority task currently blocking. • Running: Task is executing. • Holding: Invalid kernel service requested. • Msgs: Number of messages queued to task. • Peak: Maximum number of messages ever queued to task. • Ovfl: Messages dropped due to queue overflow. • Total: Total number of messages processed by task. • TCB Addr: Task control block address. • Stk Depth: Maximum stack depth used by task. • Cmd Opts: Command options provided to task. This command is also present in the *[con] Standard commands* on page 39.
tcb	Displays a memory area as a TCB (task control block).

[timer] Timer control commands

Use [timer] commands to view all timers used on TX boards. The following table lists the [timer] commands:

[timer] commands	**Description**
active	Shows all active timers.
gpt	Shows detailed information for the specified timer.
info	Shows top-level timer control information.
timers	Provides the following information for each currently defined timer: • Owner: Name of task that created the timer. • Name: Address of timer control block. • Period: Number of milliseconds between timer ticks. • Init: Initial number of milliseconds waited. • Value: Number of milliseconds remaining before timer fires. • User Key: Key provided by the application. • User Pkt: Timer control packet provided by user. • Status: Current status of timer (running or stopped). • Granularity: The tick interval used to advance timers.

[tsa] Timeslot assigner commands

Use [tsa] commands to see timeslot information. The following table lists the [tsa] commands:

[tsa] command	**Description**
ports	Shows all defined ports as defined by the *txconfig* utility. Refer to the *Dialogic® NaturalAccess™ Signaling Software Configuration Manual* for information about *txconfig*.

[tsi] H.100/H.110 TSI (time slot interchanger) commands

The [tsi] command set is available on TX 5000 Series boards.

Use [tsi] commands to display information maintained by the Time Slot Interchanger used by the TX board to perform all TDM switching. The following table lists the [tsi] commands:

[tsi] commands TX 5000 Series boards)	**Description**
clock	Show or set DSP/GT/Framer/MPC clock and frame enable.
control	Show or set the master control register.
h1clock	Show or set H.100 clocking control.
h1stream	Show or set H.100 stream control.
path0	Show or set clock path [0].
path1	Show or set clock path [1].
pattrerninc	Send incrementing pattern on a range of streams[sendPatternInc].
sampleall	Display data sample from connection source [SampleInput].
tsiclock	Show all registers related to clocking control.
tsimem	Dump the connection memory.
tsiregs	Show the TSI register set.

[uart] UART (serial port) commands

The [uart] command set is available on TX 5000 Series boards.

Use [uart] commands to display information related to the control of the on-board serial port. The following table lists the [uart] commands:

[uart] commands TX 5000 Series boards)	**Description**
info	Displays UARTInfo data structure contents.
res	Displays list of resources.
vsock	Displays vsock information.

[vsock] Virtual socket commands

The TX boards use virtual sockets (VSOCKs) as the standard communication interface between protocol layers. Use the [vsock] command to view the vsocks currently defined on the board. The following table lists the [vsock] commands:

[vsock] command	Description
info	Displays VSOCK top-level information.
resctx	Displays a memory area as a resource context (RESCTX).
ress	Displays summary information for all resources, including: • Res Type: Type of resource identifier. • Name: Name of the resource. • TX Frames: Total number of packets transmitted over the resource. • TX Fails: Total number of failed transmit attempts over the resource. • RX Frames: Total number of packets received over the resource. • RX Drops: Total number of receive indications that the resource dropped. • Last Error: Error code identifying the reason for the last error the resource detected. • MaxTX Time: Maximum number of milliseconds that the resource held a transmit packet. • MaxRX Time: Maximum number of milliseconds that the resource held a receive packet.
rxtxd	Displays a memory area as an RX/TX descriptor (RXTXD).
txpend	Displays all transmits currently pending for the given VSOCK.
vsockctx	Displays a memory area as a VSOCK control context (VSOCKCTX).
vsocks	Displays summary information for all VSOCKs, including: • VSOCK Address: Memory address for the VSOCK control context. • Owner: Name of the task that owns the VSOCK. • Res Type: Type of resource to which VSOCK is attached. • TX Pnd: Count of the currently pending transmits over the VSOCK. • TX Frames: Total number of packets transmitted over the VSOCK. • TX Fails: Total number of failed transmit attempts over the VSOCK. • RX Frames: Total number of packets received over the VSOCK. • RX Fails: Total number of receive indications with errors detected. • Last Error: Error code identifying the reason for the last error that VSOCK detected. • MaxTX Time: Maximum number of milliseconds to TX over the VSOCK. • MaxRX Time: Maximum number of milliseconds for any RX over the VSOCK.

Index