Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards Programmer's Manual

August 2009 64-0361-03

www.dialogic.com

Copyright and Legal Notice

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.**

Dialogic, Dialogic Pro, Brooktrout, Cantata, SnowShore, Eicon, Eicon Networks, Eiconcard, Diva, SIPcontrol, Diva ISDN, TruFax, Realblocs, Realcomm 100, NetAccess, Instant ISDN, TRXStream, Exnet, Exnet Connect, EXS, ExchangePlus VSE, Switchkit, N20, Powering The Service-Ready Network, Vantage, Making Innovation Thrive, Connecting People to Information, Connecting to Growth and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic. 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.

The names of actual companies and products 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.

Dialogic Corporation License Agreement For Use of Software

This is an Agreement between you, the Company, and your Affiliates (referred to in some instances as "You" and in other instances as "Company") and all Your Authorized Users and Dialogic Corporation ("Dialogic").

YOU SHOULD CAREFULLY READ THE SOFTWARE LICENSE AGREEMENT ("AGREEMENT") ON THIS SEALED PACKAGE BEFORE OPENING THE PACKAGE. BY OPENING THE PACKAGE, YOU ACCEPT THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH OR ARE UNWILLING TO ACCEPT THESE

TERMS AND CONDITIONS, YOU MAY RETURN THE PACKAGE IN UNOPENED "AS NEW" CONDITION (INCLUDING ALL DOCUMENTATION AND BINDERS OR OTHER CONTAINERS) FOR A FULL REFUND. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING THE ENCLOSED SOFTWARE ("PROGRAM"), YOU FURTHER AGREE AND ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT AND UNDERSTAND IT, AND THAT BY TAKING ANY ONE OR MORE OF SUCH STEPS/ACTIONS YOU AGREE TO BE BOUND BY SUCH TERMS AND CONDITIONS. DIALOGIC IS UNWILLING TO LICENSE THE SOFTWARE TO YOU IF YOU DO NOT ACCEPT AND AGREE TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT.

Intellectual Property

The enclosed Software ("Program") and all accompanying documentation are individually and collectively owned by Dialogic Corporation ("Dialogic"), its subsidiaries and/or its suppliers and are protected by all applicable intellectual property laws and international treaty provisions. Therefore, You and Your Authorized Users must treat the Program and documentation like any other material so protected, except as expressly permitted in this Agreement. In particular, but without limitation, You acknowledge that the Program and its accompanying documentation constitute valuable intellectual property rights, including without limitation trade secrets and copyrights, and confidential information of Dialogic. The Program and all programs developed thereunder and all copies thereof (including without limitation translations, compilations, partial copies with modifications and updated works) are proprietary to Dialogic and title to all applicable copyrights, trade secrets, patents and other intellectual property rights therein remains in Dialogic, its subsidiaries, and/or its suppliers. Except as expressly permitted in this Agreement, You shall not sell, transfer, publish, disclose, display or otherwise make available the Program or copies thereof to others. You agree to secure and protect the Program, its accompanying documentation and copies thereof in a manner consistent with the maintenance of Dialogic's rights therein and to take appropriate action by instruction or agreement with Your employees and/or consultants who are permitted access to the Program to satisfy Your obligations hereunder. Violation of any provision of this paragraph shall be the basis for immediate termination of this Agreement. Because unauthorized use or transfer of the Software or documentation may diminish substantially the value of such materials and irrevocably harm Dialogic, if You breach the provisions of this Section of this Agreement, Dialogic shall be entitled to injunctive and/or other equitable relief, in addition to other remedies afforded by law, to prevent a breach of this Section of this Agreement.

Grant of License

Subject to the terms and conditions of this Agreement Dialogic grants to You a non-exclusive, personal, non-transferable license to use the Program in object code form only and solely in accordance with the following terms and conditions:

- You may make, install and use only one (1) copy of the Program on a single-user computer, file server, or on a workstation of a local area network, and only in conjunction with a legally acquired Dialogic® hardware or software product You may also make one copy solely for backup or archive purposes;
- The primary Authorized User on the computer on which the Program is installed may make a second copy for his/her exclusive use on either a home or portable computer;
- You may copy the Program into any machine readable or printed form for backup or modification purposes in support of Your use of one copy of the Program;
- You may distribute the Program in object code only and only as part of, or integrated by You into, a computer system that (i) contains a Dialogic hardware product, (ii) includes a substantial amount of other software and/or hardware manufactured or marketed by You and (iii) is marketed and sublicensed to an end user for the end user's own internal use in the regular course of business (a "Licensed System");
- Each end user to whom a Licensed System is distributed must agree to license terms with respect to the Program that are at least as protective of Dialogic's rights in the Program as those set forth in this Agreement;
- You shall receive one (1) Program master disk, and shall be solely responsible for copying the Program into the Licensed Systems and for warranting the physical media on which it is copied
- You may make one (1) copy of the documentation accompanying the Program, provided that all copyright notices contained within the documentation are retained;
- You may modify the Program and/or merge it into another Program for Your use in one computer; (any portion of this Program will continue to be subject to the terms and conditions of this Agreement);
- You may transfer the Program, documentation and the license to another eligible party within Your Company if the other party agrees to accept the terms and conditions of this Agreement. If You transfer the Program and documentation, You must at the same time either transfer all copies whether in printed or machine readable form to the same party or destroy any copies not transferred; this includes all modifications and portions of the Program contained in or merged into other Programs;

- You shall not remove, and each copy of the Program shall contain, the same copyright, proprietary, patent and/or other applicable intellectual property or other ownership notices, plus any restricted rights legends that appear in the Program and/or this Agreement and, if You copy the Program onto media to which a label may be attached, You shall attach a label to the media that includes all such notices and legends that appear on the Program master disk and envelope;
- You may not rent or lease the Program. You may not reverse engineer, decompile or disassemble the Program. Except as is strictly necessary for You to integrate the Program with other software and/or hardware to produce the Licensed Systems, You shall not copy, modify or reproduce the Program or documentation in any way. You shall use Your best efforts to ensure that any user of the Program does not reverse engineer, decompile or disassemble the Program to derive a source code equivalent of the Program;
- If You transfer possession of any copy, modification or merged portion of the Program or documentation to another party in any way other than as expressly permitted in this Agreement, this license is immediately and automatically terminated;
- The Program may be used only in conjunction with Dialogic hardware;
- The Program shall not be exported or re-exported in violation of any export provisions of the United States or any other applicable jurisdiction.

Upgrades

If the Program is provided as an upgrade and the upgrade is an upgrade from another product licensed to You and Your Authorized Users by Dialogic, the upgrade is governed by the license agreement earlier provided with that software product package and the present Agreement does not grant You additional license(s). If You and Your Authorized Users choose to upgrade this Program or the product used together with the Program and such upgrade requires the license of additional software (whether a charge is associated with such software or not), the license agreement associated with such additional software shall govern the license of such additional software to the exclusion of this Agreement.

Term

The Agreement is effective until terminated. You may terminate it at any time by notifying Dialogic and/or by destroying the Program and all accompanying documentation together with all copies, modifications and merged portions in any form. The Agreement will also terminate automatically upon the occurrence or lack of occurrence of certain terms and/or conditions set forth in this Agreement, or if You fail to comply with any term or condition of this Agreement. You agree that upon any such termination You shall destroy or return to Dialogic the Program and all accompanying documentation supplied by Dialogic, together with any and all copies, modifications and merged portions in any form. All provisions of this Agreement relating to disclaimers of warranties, limitation of liability, remedies, or damages, and licensor's proprietary rights shall survive termination.

Limited Warranty

Dialogic solely warrants the media on which the Program is furnished to You to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of purchase by You as evidenced by a copy of Your receipt. If such a defect appears within the warranty period, You may return the defective media to Dialogic for replacement without charge provided Dialogic, in good faith, determines that it was defective in materials or workmanship. Replacement is Your sole remedy with respect to such a defect. Dialogic offers no warranty for Your reproduction of the Program. This Limited Warranty is void if failure of the Program has resulted from accident, misuse, abuse or misapplication.

Disclaimers, Limitations of Liability and Customer Remedies

Except as set forth in the "Limited Warranty" Section of this Agreement, the Program and accompanying documentation are provided to You "as is." Neither Dialogic, its subsidiaries, its suppliers, nor its licensor(s) (if any) warrants that the Program will meet Your requirements or that its use will be uninterrupted or error-free. Except as set forth in the "Limited Warranty" Section, EACH OF DIALOGIC, ITS SUBSIDIARIES, ITS SUPPLIERS AND ITS LICENSOR(S) (IF ANY) DISCLAIMS ANY AND ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THE PROGRAM AND ACCOMPANYING DOCUMENTATION, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR AGAINST LATENT DEFECTS. Except as set forth in the "Limited Warranty" Section, neither Dialogic, its subsidiaries, its suppliers, nor its licensor(s) (if any) shall have any liability to You or any third party for any claim, loss or damage of any kind, including but not limited to lost business profits, business interruption, loss of information, or other pecuniary loss and indirect, punitive, incidental, economic, consequential or special damages, arising out of or in connection with this Agreement and/or the use, inability to use the Program and/or the Program's performance or inability to perform nor from or in connection with the Program's accompanying documentation, or any data or equipment related thereto or used in connection therewith. In no event shall Dialogic's, its subsidiaries', its suppliers' or its licensor(s)'s liability for damages, whether arising out of contract, negligence, warranty, or patent or copyright infringement, exceed the fees You paid for the Program. No representation or warranty regarding the Program may be made without Dialogic's, its subsidiaries', its suppliers', or its licensor(s)'s (if any) prior written consent, and any warranty or representation made by You or Your customers regarding the Program shall not constitute an obligation of Dialogic, its subsidiaries, its suppliers, or other licensor(s) (if any). This limited warranty gives You specific legal rights. You may have other rights, which may vary from jurisdiction to jurisdiction. Also, as some jurisdictions do not allow the exclusion or limitation for certain damages, some of the above limitations may not apply to You.

Right to Audit

If this Program is licensed for use in a Company, Your Company and You individually and collectively agree to keep all usual and proper records and books of accounts and all usual proper entries relating to each installation of the Program during the term of this Agreement and for a period of three (3) years thereafter. During this period, Dialogic may cause an audit to be made of the applicable records in order to verify Your compliance with this Agreement and prompt adjustment shall be made to compensate for any errors or omissions disclosed by such audit. Any such audit shall be conducted by an independent certified public accountant selected by Dialogic and shall be conducted during the regular business hours at Your offices and in such a manner as not to interfere with Your normal business activities. Any such audit shall be paid for by Dialogic unless material discrepancies are disclosed. For such purposes, "material discrepancies" shall mean three percent (3%) or more of the Authorized Users within the Company. If material discrepancies are disclosed,

Your Company agrees to pay Dialogic for the costs associated with the audit as well as the license fees for the additional licensed channels or additional authorized users. In no event shall audits be made more frequently than semi-annually unless the immediately preceding audit disclosed a material discrepancy.

Supplementary Software

Any Supplementary Software provided with the Program and/or referred to in this Agreement is provided "as is" with no warranty of any kind.

Miscellaneous

You acknowledge that You have read this Agreement, that You understand it, and that You agree to be bound by its terms and conditions, and You further agree that this is the complete and exclusive statement of the Agreement between the Dialogic and You ("the Parties"), which supersedes and merges all prior proposals, understandings and all other agreements, oral and written, between the Parties relating to the Program. You agree to indemnify and hold harmless Dialogic and its subsidiaries, affiliates, suppliers, officers, directors and employees from and against any claim, injury, loss or expense, including reasonable attorneys' fees, arising out of (i) Your failure to comply with the provisions of this Agreement, or (ii) any other wrongful conduct by or on behalf of You. This Agreement applies to all updates, future releases, modifications and portions of the Program contained in or merged into other programs. This Agreement may not be modified or altered except by written instrument duly executed by Dialogic. No action, regardless of form, arising out of this Agreement or the use of the Program may be brought by You more than two (2) years after the cause of action has first arisen. Except as provided herein, neither this Agreement nor any rights granted are assignable or transferable, and any assignment or transfer will be null and void. If You authorize any other person to copy the Program, You shall obligate that person in writing to comply with all conditions of this Agreement. Dialogic shall have the right to collect from You its reasonable expenses incurred in enforcing this agreement, including attorney's fees. The waiver or failure of Dialogic to exercise in any respect any right provided for herein shall not be deemed a waiver of any further right hereunder. All rights and remedies, whether conferred hereunder or by any other instrument or law, will be cumulative and may be exercised singularly or concurrently. Failure by either Dialogic or You to enforce any term or condition of the Agreement will not be deemed a waiver of future enforcement of that or any other term or conditions. The terms and conditions stated herein are declared to be severable. Should any term(s) or condition(s) of this Agreement be held to be invalid or unenforceable the validity, construction and enforceability of the remaining terms and conditions of this Agreement shall not be affected. It is expressly agreed that Dialogic and You are acting as independent contractors under this Agreement. These terms and conditions will prevail notwithstanding any different, conflicting or additional terms and conditions that may appear on any other agreement between Dialogic and You. Deviations from these terms and conditions are not valid unless agreed to in writing in advance by an authorized representative of Dialogic. Any notices sent to Dialogic under this Agreement must be sent by registered mail or courier to the attention of Dialogic's legal department at the address below or such other address as may be listed on www.dialogic.com from time to time as being Dialogic's Montreal headquarters.

U.S. Government Restricted Rights

The Program and all accompanying documentation are provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(iii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraph (c) (1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR52.227-19, both as applicable.

Governing Law

Any and all claims arising under this Agreement shall be construed and controlled by the laws in force in the Province of Quebec, Canada, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. Dialogic is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Dialogic.

Contractor/ manufacturer is:

Dialogic CORPORATION.

9800 Cavendish Blvd., Montreal, Quebec, Canada H4M 2V9

This Agreement has been drafted in English at the express wish of the parties. Ce contrat a été rédigé en anglais à la demande expresse des parties.

Contents

About this Publication

This manual is the Programmer's Manual for the Dialogic® DSI SS7 Stack running with Dialogic® Diva® PRI Media Boards. It is for system developers and integrators who choose to integrate the boards or develop applications that use the DSI SS7 Stack for signaling.

How to Use this Online Manual

- To view a section, click the corresponding blue underlined phrase in the table below.
- To view a topic that contains further information, click the corresponding blue underlined phrases in this manual.
- You may wish to print out the pages required for the configuration.

Structure of this Manual

This manual provides a detailed description of how to install the DSI SS7 Stack together with the Dialogic® Diva® System Release LIN software required for the use of Diva Media Boards and of the system and protocol configuration.

This manual is structured as follows:

Section	Contents
About Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards	Key benefits of Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards, related information
Specification	Product identifiers, supported Diva Media Boards, and license information.
Use Cases and Installation	Different use cases, software package, software installation
Configuration and Operation	System structure, system configuration, protocol configuration
Program Execution	Starting the software, executing programs
Message Reference	Individual messages sent to or received from a Diva Media Board in conjunction with the DSI SS7 Stack
Configuration Command Reference	Commands and parameters used in the config.txt protocol configuration file
Host Utilities and Host Protocols	Host utilities that can be used with the DSI SS7 Software for Diva Boards
Management Interface of the SSDD Process	Parameters of the ssdd management interface
Protocol Configuration Using Discrete Messages	Protocol configuration by building and sending messages directly to the ssdd
Glossary	Glossary of specific terms used in this manual

The manual should be used in conjunction with the appropriate Installation Guide and Reference Guide for the Diva System Release software. These and other supporting documentation, including the Programmer's Manuals for the individual protocol modules, are listed in Related Information on page 10.

Chapter 1: About Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards

The Dialogic® Diva® Media Board range contains powerful, universal communication boards from single up to four-span T1/E1 models. They may be utilized as mixed signaling, trunk and resource board. Designed to be scalable, any combination of Diva Media Boards can be mixed in your system offering an ideal selection to suit your communication needs. High-class performance and functionality is provided, as all hardware family members run the same software.

The Dialogic® Diva® System Release LIN software offers a wide range of well established APIs as CAPI, Dialogic® Diva® Software Development Kit (Diva SDK), Dialogic® Diva® SIPcontrol™ Software (SIP-PSTN Gateway software), tty, Dialogic® Host Media Processing Software (Dialogic HMP Software), and various Dialogic® DSI SS7 APIs.

The Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards is designed for "signaling only" applications or combined signaling and media usage. Therefore, the typical Dialogic® DSI SS7 Stack used in association with Diva Media Boards will consist of MTP1/2 running on the board, MTP3 and ISUP running on the host.

Diva Media Boards include PCI and PCI Express (PCIe) form factors with dedicated DSP resources. For an overview of supported boards, see Product Identifiers on page 11.

Notable benefits of Diva Media Boards and the Diva System Release LIN software include:

- Universal, multifunctional communication platform
- State-of-the-art hardware design
- High-performance media processing
- Simple and effective programming interfaces
- Superior scalability and flexibility
- Easy to install and configure

1.1 Related Information

See the following for related information:

- Dialogic® Diva® System Release LIN Reference Guide
- Dialogic® Diva® Media Board Installation Guide
- Dialogic® DSI Protocol Stacks Host Licensing Users Guide
- Dialogic® SS7 Software for Dialogic® Diva® Interfaces Reference Guide

Depending on the environment, which needs to be established, the following additional manuals may be relevant:

- Dialogic® SS7 Protocols Software Environment Programmer's Manual
- Dialogic® SS7 Protocols ISUP Programmer's Manual
- Dialogic® SS7 Protocols MTP3 Programmer’s Manual
- Dialogic® SS7 Protocols Call Test Utility (CTU) User Guide
- Dialogic® SS7 Protocols User Part Example (UPE) User Guide

Current software and documentation supporting Dialogic® Diva® Media Boards are available at:

http://www.dialogic.com/products/tdm_boards/system_release_software/Diva_for_Linux.htm

Product data sheets are available at http://www.dialogic.com/support/helpweb/signaling

For detailed descriptions of the supported features or the protocols described in this manual, refer to the programmer’s guides and release notes for the specific product by using the documentation download link at http://www.dialogic.com/products/signalingip_ss7components/SS7_Protocols.htm

Chapter 2: Specification

This chapter provides the following information:

- Product Identifiers below
- Licenses below

2.1 Product Identifiers

The Dialogic® Diva® Media Board product family includes PCI and PCI Express (PCIe) form factor boards described in the Dialogic® Diva® Media Board Installation Guide.

The following Diva Media Boards are supported by the Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards:

Dialogic® Diva® PRI Media Boards

- Diva PRI/E1/T1-CTI PCI and PCIe
- Diva V-PRI/T1-24 PCI and PCIe
- Diva V-PRI/E1-30 PCI and PCIe
- Diva UM-PRI/T1-24 PCI and PCIe
- Diva UM-PRI/E1-30 PCI and PCIe
- Diva PRI/E1/T1-8 PCI
- Diva PRI/T1-24 PCI and PCIe
- Diva PRI/E1-30 PCI and PCIe

Dialogic® Diva® V-PRI Media Boards with multiple ports

- Diva V-2PRI/T1-48 PCI
- Diva V-2PRI/E1-60 PCI
- Diva V-4PRI/T1-96 PCI
- Diva V-4PRI/E1-120 PCI

- Diva V-1PRI/E1/T1-30 PCIe HS
- Diva V-2PRI/E1/T1-60 PCIe HS
- Diva V-4PRI/E1/T1-120 PCIe HS
- Diva V-4PRI/E1/T1-120 PCIe FS
- Diva V-8PRI/E1/T1-240 PCIe FS

Note: "HS" stands for the half size and "FS" for the full size board format.

2.1.1 Protocol Resource Support

Diva Media Boards support the Message Transfer Part-Signaling Link (MTP1/2) running on the board, other protocols including ISUP and MTP3 running on the host.

All Dialogic® DSI Protocol Stacks used in conjunction with Diva Media Boards are enabled by host licenses.

2.1.2 SS7 Links

Diva Media Boards may terminate up to 4 bidirectional MTP2 links per system.

2.1.3 SS7 Protocols

MTP2 runs on the board. MTP3 and ISUP can be configured to run on the host.

2.2 Licenses

Depending on the signaling requirements, you will need the following host licenses in conjunction with the DSI SS7 Software for Diva Boards.

Product Name	Protocol
	MTP2 Protocol Layer host license
SS7SBMTP2D1	1 link
SS7SBMTP2D2	2 links
SS7SBMTP2D4	4 links
SS7SBHSTMTP3	MTP3 Protocol Layer host license
SS7SBHSTISUP	ISUP Protocol Layer host license

Note: As specified above, MTP2 host licenses are available for 1, 2, or 4 links. One license file per protocol (e.g. MTP2) is supported.

These licenses are bound to a specific host, identified by a host ID.

The host ID can be queried by calling:

```
./ssdd -v
```

in the install directory of the DSI SS7 Software for Diva Boards, the host ID will be displayed like:

```
The Licensing Host ID for this machine is: 0123456789ab.
```

For further information on the DSI SS7 host licensing, refer to the Dialogic® DSI Protocol Stacks Host Licensing Users Guide.

Note: The protocol modules which require a license may be started with `-t` to run in trial mode. Modules started in trial mode will run up to 10 hours.

Chapter 3: Use Cases and Installation

This chapter contains the following information:

- Use Cases below
- Software Package on page 16
- Software Installation on page 16

3.1 Use Cases

With the Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards, three scenarios can be realized:

- Dialogic® DSI SS7 APIs

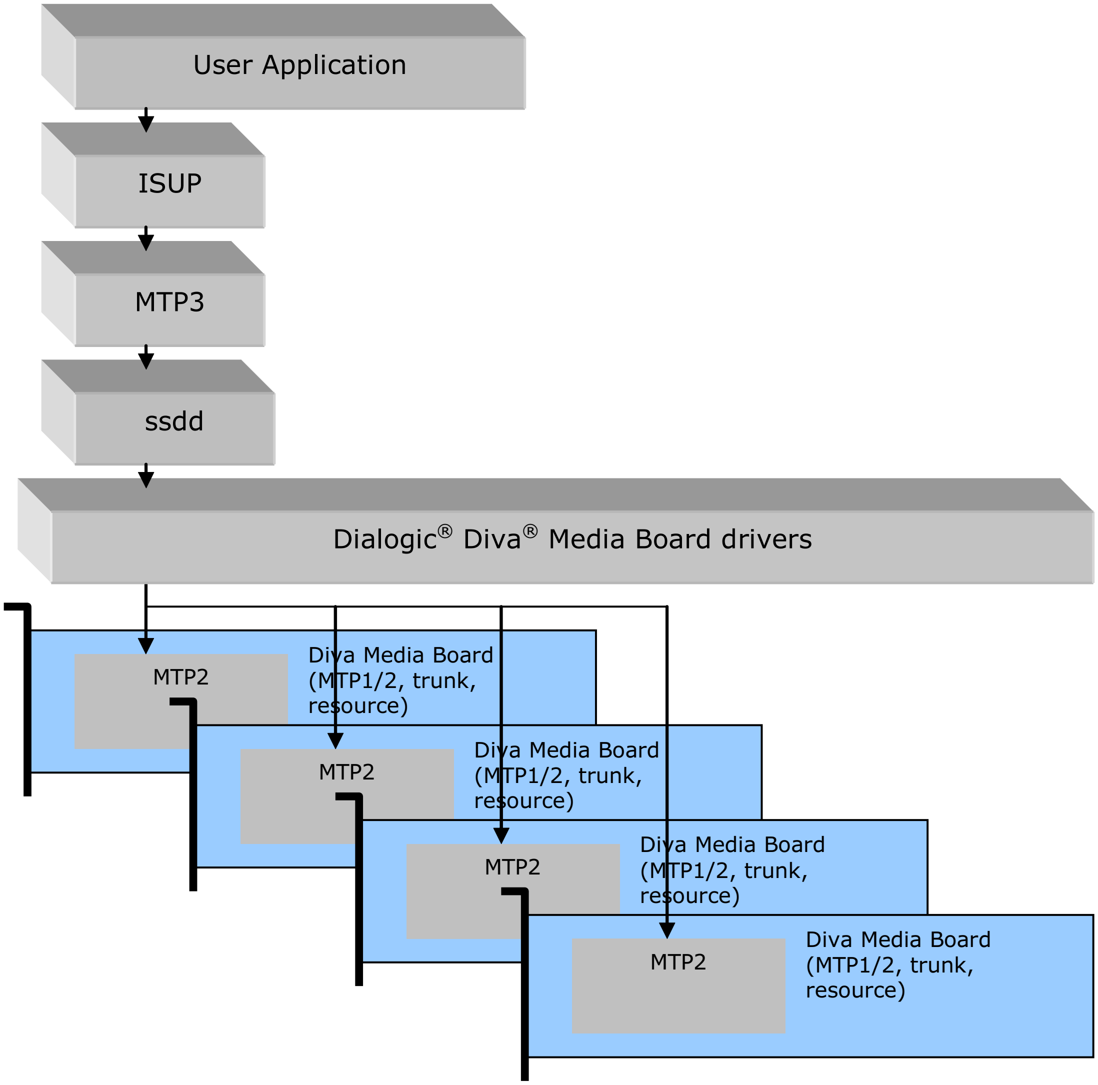
  Based on the Dialogic® DSI SS7 Stack, the ISUP interface can be directly accessed.

- Dialogic® SS7 Software for Dialogic® Diva® Interfaces

  Based on the Dialogic® SS7 Stack, the SS7 Software for Diva Interfaces adds components to use the native Diva interfaces, such as CAPI, Dialogic® Diva® Software Development Kit (SDK), IDI, and tty. If you want to replace an existing Dialogic® Diva® softSS7 software installation with the SS7 software for Diva Interfaces, the existing Diva softSS7 software installation will not be removed.

Note: The DSI SS7 Software for Diva Media Boards is bundled with a tested and released version of DSI binaries. Please note that it is possible that newer and/or different versions of the DSI binaries might be available. Please contact Dialogic Support to verify if any such newer versions have been tested and officially released.

3.1.1 Dialogic® DSI SS7 APIs


User Application
ISUP
MTP3
ssdd
Dialogic® Diva® Media Board drivers
MTP2
Diva Media Board (MTP1/2, trunk, resource)
MTP2
Diva Media Board (MTP1/2, trunk, resource)
MTP2
Diva Media Board (MTP1/2, trunk, resource)
MTP2
Diva Media Board (MTP1/2, trunk, resource)

Dialogic® DSI SS7 APIs

The Dialogic® DSI SS7 APIs combine Dialogic® DSI SS7 software modules with Dialogic® Diva® Media Boards. The MTP2 module is running on the Diva Media Boards. The MTP3 and ISUP modules will run on the host.

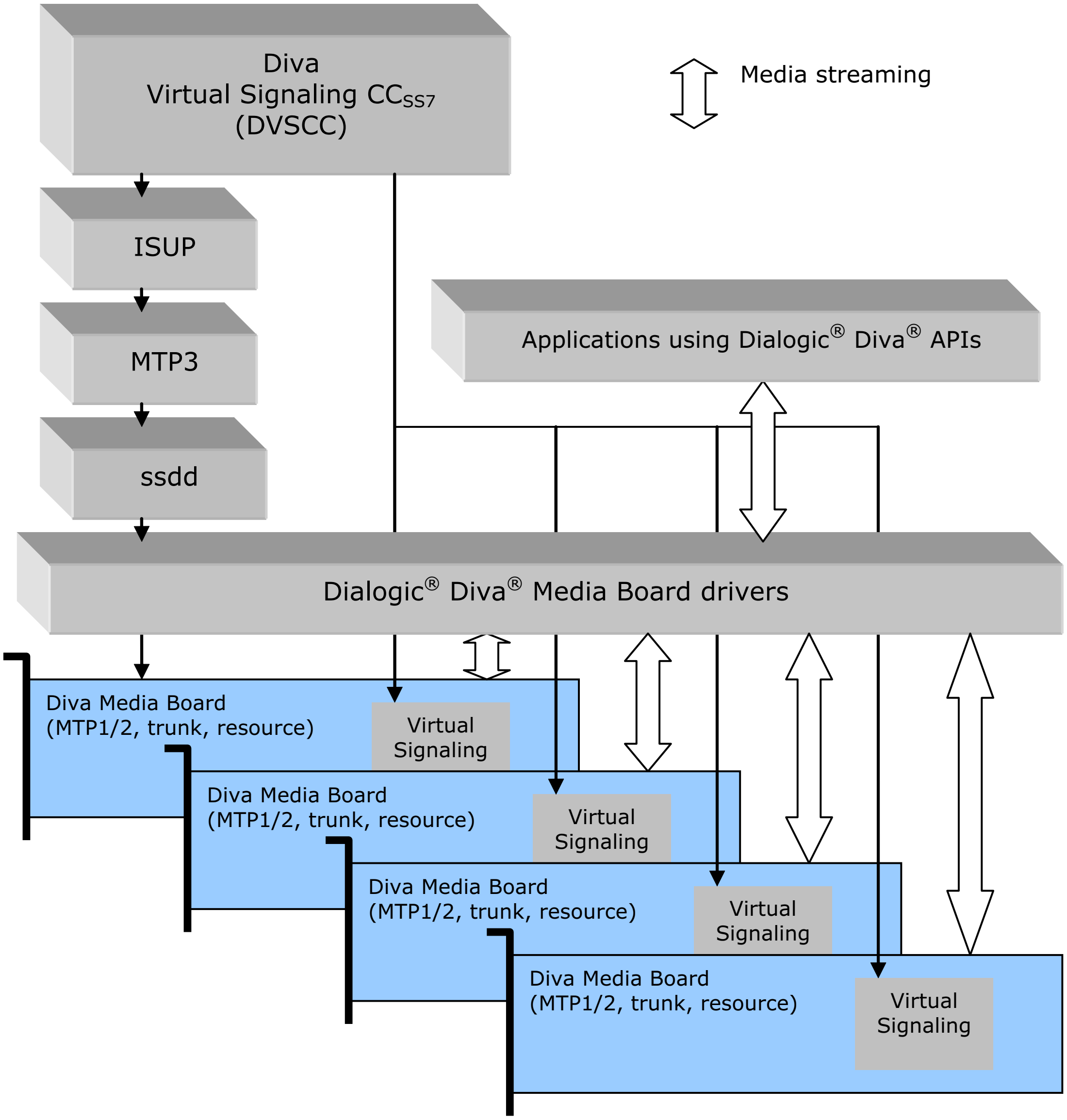
The user application is set up on top of the ISUP layer.

The DSI SS7 APIs build the base for the other scenario:

- Dialogic® SS7 Software for Dialogic® Diva® Interfaces on page 15

The DSI SS7 APIs are an add-on to the Dialogic® Diva® System Release LIN software. See the Dialogic® Diva® System Release LIN Reference Guide.

3.1.2 Dialogic® SS7 Software for Dialogic® Diva® Interfaces


Diva
Virtual Signaling CC$_{SS7}$
(DVSCC)
Media streaming
ISUP
MTP3
ssdd
Applications using Dialogic® Diva® APIs
Dialogic® Diva® Media Board drivers
Diva Media Board
(MTP1/2, trunk, resource)
Virtual Signaling
Diva Media Board
(MTP1/2, trunk, resource)
Virtual Signaling
Diva Media Board
(MTP1/2, trunk, resource)
Virtual Signaling
Diva Media Board
(MTP1/2, trunk, resource)
Virtual Signaling

Dialogic® SS7 Software for Dialogic® Diva® Interfaces

The SS7 Software for Diva Interfaces is an add-on to the Dialogic® Diva® System Release LIN software.

It enables the user of Dialogic® Diva® Media Boards to utilize SS7 signaling and still use applications with standard Dialogic® Diva® programming interfaces like CAPI, Dialogic® Diva® SDK, and tty.

Diva Virtual Signaling Call Control SS7 (DVSCC) translates on top of the Dialogic® DSI ISUP Layer the ISUP primitives into signaling primitives which are then processed by the Diva Media Board.

This scenario replaces the legacy Dialogic® Diva® softSS7 software. When upgrading from Diva softSS7 1.x to the Dialogic® SS7 Software for Dialogic® Diva® Interfaces, licenses need to be replaced by Dialogic® DSI SS7 host licenses. For further information, see the Dialogic® SS7 Software for Dialogic® Diva® Interfaces Reference Guide.

3.2 Software Package

The Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards is included in the Dialogic® Diva® System Release LIN Software, which is shipped as RPM software package. See the Dialogic® Diva® System Release LIN Software Reference Guide.

3.3 Software Installation

To install the Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards, see the following sections:

- Premises of the Dialogic® Diva® System Release LIN software below
- Requirements for the installation below
- Installing the Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards on page 17

3.3.1 Premises of the Dialogic® Diva® System Release LIN software

The physical parameters of Dialogic® Diva® Media Boards are configured in the Diva System Release LIN software while users of Dialogic® DSI SS7 board families SS7HD and SS7SPCI configure parameters of the physical layer using the Line Interface Unit (LIU) configuration.

When configuring the Diva System Release software, consider the following:

Topic	Scenario	Selection to Consider
System interfaces to activate	All scenarios	CAPI 2.0 interface and SS7 support
Interface Mode	All scenarios	Terminal Equipment (TE) mode or Network Terminator (NT) mode
D-Channel Protocol	Dialogic® SS7 Software for Dialogic® Diva® Interfaces	SS7_E1 - DSI SS7 for Diva E1 or SS7_T1 - DSI SS7 for Diva T1
	Other	E1CHAN - Channelized E1 or T1CHAN - Channelized T1
Timeslots	All scenarios	Select the timeslots used for signaling, to disable resource access
Layer 1 Framing	All scenarios	National default (default) or Doubleframing (no CRC4) or Multiframing (CRC4)
Voice Companding Options	All scenarios	If required, select: Force A-Law or Force u-Law
Diva Media Board to M-Board assigment	Dialogic® SS7 Software for Dialogic® Diva® Interfaces	Bind every Diva Media Board to own M-Board

3.3.2 Requirements for the installation

- Installed glibc version 2.3.2 and later
- gcc 3 and later

3.3.3 Installing the Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards

The DSI SS7 Software for Diva Boards is installed together with the Dialogic® Diva® System Release LIN software. Follow these steps:

1. Install the Diva System Release LIN software; see the Dialogic® Diva® System Release LIN Software Reference Guide.
2. During system configuration, in the **System Settings** dialog box (command line) or **System Configuration** page (web based configuration), select **DSI SS7 for Diva** to start the SS7 for Diva software during driver start.
3. Select the mode of operation: **SS7 Diva Interfaces** or **DSI SS7 Native Interfaces**.
4. Install the required license; see the Diva System Release LIN Software Reference Guide.

   Note: The SS7 for Diva software runs up to ten hours in trial mode.
5. Configure the Diva System Release software.

   Note: For information on layer 1 E1/T1 parameters, see Regulatory and Geographic Considerations on page 18.
6. Configure the DSI SS7 Software for Diva Boards; see the following chapters of this document.

The table below shows the DSI SS7 Software for Diva Boards files that are copied into the install directory (/usr/lib/opendiva/dlgss7).

File	Purpose
ssdd	Dialogic® DSI SS7 Stack board management and interface process for Dialogic® Diva® Media Boards
mtp_lnx6	Message Transfer Part 3 (MTP3) is the layer 3 (network layer) of an SS7 stack.
isp_lnx6	Dialogic® DSI ISUP Layer is the ISDN User Part of SS7: It abstracts ISDN signaling and adds additional functionality to it.
dvscc	Diva Virtual Signaling Call ControlSS7 translates ISUP primitives into signaling primitives forwarded to and received from the virtual signaling protocol running on the Diva Media Board. Component of Dialogic® SS7 Software for Dialogic® Diva® Interfaces
gctlib.lib	Library to be linked with user's application
system.txt	Example system configuration file
config.txt	Example protocol configuration file
versions.txt	Holds version information of components
gctload tick_lnx tim_lnx s7_mgt s7_log s7_play actlinks mtpsl	Executables for use as described later in this manual

To enable the dynamic linking of gctlib.lib (libgctlib.so), the following symbolic links are created by the installation:

```
/usr/lib/libgctlib.so -> /usr/lib/opendiva/dlgss7/libgctlib.so.1.0.0
/usr/lib/libgctlib.so.1 -> /usr/lib/opendiva/dlgss7/libgctlib.so.1.0.0
/usr/lib/libgctlib.so.1.0.0 -> /usr/lib/opendiva/dlgss7/libgctlib.so.1.0.0
```

Chapter 4: Configuration and Operation

Before you start to configure the Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards, you should get an idea of the flexibility of the protocol stack, the run-time options that exist, and the mechanisms that are used to select specific features. This chapter gives you an overview of these options. You should also read the Dialogic® SS7 Protocols Software Environment Programmer's Manual that describes the basic principles of modules and message passing.

This chapter provides the following information:

- Regulatory and Geographic Considerations below
- System Structure below
- System Configuration on page 20
- Protocol Configuration on page 22

4.1 Regulatory and Geographic Considerations

Certain functions of Dialogic® Diva® Media Boards, although implemented in the hardware, have selectable options that are configured by the Dialogic® Diva® System Release LIN software. A user or integrator must consider the requirements of the application when choosing these settings, but must also consider any local regulatory requirements for the intended deployment location to ensure a compliant overall system. As an aid to this process, the table below details some of the areas where the correct selection of configuration options may be required.

For the convenience of users of Dialogic® DSI SS7 Boards, such as Dialogic® DSI SS7HD Network Interface Boards and Dialogic® DSI SPCI Network Interface Boards, the last column shows the corresponding configuration parameters of those boards.

Configuration Area		Dialogic® Diva® Media Boards	Dialogic® DSI SS7 Boards
T1/E1 Ports	Interface type	• Selected via D-channel protocol E1/T1*) (120 ohm balanced) • 75 ohm unbalanced requires an external unbalanced to balanced transformer	liu_type (LIU_CONFIG)
	Pulse shape	Selected via D-channel protocol E1/T1	liu_type (LIU_CONFIG)
	Line code	Selected via D-channel protocol: • E1:HDB3 • T1:B8ZS	line_code (LIU_CONFIG)
	Frame format	Selected via layer 1 framing • E1:multiframe (default) or double framing, no CRC • T1:ESF	frame_format (LIU_CONFIG)
	CRC/E-bit operation	Selected via layer 1 framing • E1:CRC4 • T1:CRC6	CRC_mode (LIU_CONFIG)

* T1 includes J1

4.2 System Structure

The MTP2 module running on the board communicates with host processes running on the host computer. The physical interface to the board uses the PCI/PCIe bus. All communication with the board is handled by a device driver and the messages passing to and from the board are managed by the SS7 board management and interface process ("ssdd", sometimes generically referred to as "ssd") that runs on the host computer.

The MTP3 and ISUP modules are running on the host. In any case, the interface (ssdd) to the SS7 host processes remains identical.

The table illustrates some potential Dialogic® SS7 Software for Dialogic® Diva® Interfaces configuration scenarios for a telephony system.

Software running on the board	MTP2 Diva Virtual Signaling Media resource management
Software running on the host CPU	MTP3 ISUP DVSCC - Diva Virtual Signaling Call ControlSS7 User Application using any Dialogic® Diva® API (CAPI, SDK, tty, …)
Number of boards	1-16 ports, 1-4 links

In all cases, the SS7 board management and interface process (ssd) is required to run on the host computer. The ssd process handles the SS7 message transfer between the host and the board using CAPI.

The selection of which protocol modules to run on the host is made by editing the system.txt configuration file. The user then runs the gctload program that reads the system configuration parameters from the system.txt configuration file and starts the selected processes bringing the system into operation. For further details on the operation of the gctload program, refer to the Dialogic® DSI Protocol Stacks Host Licensing Users Guide.

The table below shows host processes and utilities, for use on the host, that are included in the package.

Note: s7_mgt , s7_log, s7_play, mtpsl, and actlinks are optional utilities. You may choose to implement the functionality provided by these utilities in your own applications.

Process or Utility	**Purpose**
gctload	Process to initialize the system environment and start all other related processes running on the host, deriving the configuration from a text file (system.txt). For more information, see gctload on page 61.
ssdd	Process to interface with the device driver for passing messages to and from the board(s) and for configuring the Dialogic® DSI SS7 part running on the board(s). This process is referred to in a generic manner as ssd, although the name of the binary for use with Dialogic® Diva® Media Boards is in fact ssdd. For more information, see ssdd on page 67
tick_lnx	Protocol timer process to send periodic tick notification to the tim_lnx process that in turn handles protocol timers. For more information, see tick (tick_lnx) on page 65.
tim_lnx	Process to receive periodic tick notification from tick_lnx and handle protocol timers for all other processes. For more information, see tim (tim_lnx) on page 64.
s7_mgt	Process to perform one time protocol configuration for all protocol modules, deriving the configuration parameters from a text file (config.txt). This process is optional. As an alternative to using it, you may select to perform protocol configuration by sending messages directly to the other modules in the system. For more information, see s7_mgt on page 66 and Protocol Configuration Using Discrete Messages on page 86.
s7_log	Utility process to allow messages received from the protocol stack to be logged to a text file. This is useful for diagnostic purposes when getting started. For more information, see s7_log on page 56.
s7_play	Utility process used to generate messages from a text file and send them into the system. This is useful for diagnostic purposes when getting started. For more information, see s7_play on page 59.
mtpsl	Utility process, which may be used to activate/deactivate signaling links. For more information, see mtpsl on page 69.
actlinks	Utility script which calls `mtpsl` to activate linksets. For more information, see actlinks on page 70.
mtp_lnx6	MTP3 host protocol module. For more information, see mtp_lnx6 on page 71.
isp_lnx6	ISUP host protocol module. For more information, see isp_lnx6 on page 72.
dvscc	Diva Virtual Signaling Call ControlSS7 translates ISUP primitives into signaling primitives forwarded to and received from the virtual signaling protocol running on the Diva Media Board. See the Dialogic® SS7 Software for Dialogic® Diva® Interfaces Reference Guide.

4.3 System Configuration

System configuration is handled by the gctload program that reads system configuration data from a file called system.txt. System initialization requires:

- First, that a pool of message buffers is created for subsequent inter-process communication.
- Second, that a message queue is created for each process that will run and that any message redirection for modules that are running remotely is initialized.
- Finally, that all processes can be started.

The gctload program handles this initialization sequence and creates the inter-process communication environment. The program reads input from the system.txt configuration file, carries out all system initialization and starts all processes.

The system.txt configuration file is a user-configurable file containing details of all the module identifiers known to the system, details of whether they are local modules or remote modules accessed by a local module (message redirection) and includes the command line for all processes to be started by the gctload program.

The gctload program creates a message queue for each of the local module identifiers. The program subsequently expects a process to service its message queue, otherwise messages written to that queue will never be read causing eventual loss of system messages.

The gctload program initializes the message queue look-up table so that messages destined for modules that do not exist locally are redirected to a message queue for a module that exists locally.

Having created the system environment, the gctload program proceeds to spawn all processes listed in the system.txt configuration file in the order listed.

Note: Prior to running the gctload program, you need to edit the system.txt configuration file to reflect the requirements of your system.

4.3.1 System Configuration File Syntax

The system.txt configuration file is a text file used by the gctload program to configure the software environment. The file syntax permits the use of comments to improve the readability of the file. Comments are inserted into the file by using an asterisk (*). All characters on the line following the asterisk are ignored.

Numbers can be entered in either decimal or hexadecimal format. Hexadecimal numbers should be prefixed with 0x. For example, the value 18 can be entered in either of the following formats:

```
0x12 *(Hexadecimal)
18 *(Decimal)
```

The system configuration file can contain the following commands:

- `LOCAL` commands to allow the gctload program to generate message queues for modules running locally
- `REDIRECT` commands to cause messages generated for modules not running locally to be redirected via a module that is running locally
- `FORK_PROCESS` commands advising the gctload program of any processes that need to be started locally

The full syntax of each command is listed in the Dialogic® SS7 Protocols Software Environment Programmer's Manual.

An example system.txt configuration file for the Dialogic® SS7 Software for Dialogic® Diva® Interfaces is shown below:

```
********************************************************************************
*
* Example System Configuration File (system.txt) for use with
* Dialogic(R) Global Call for SS7 based on Dialogic(R) Diva(R) Media Boards
*
* Edit this file to reflect your configuration.
*
********************************************************************************
*
* Essential modules running on host:
*
LOCAL  0x20    * ssdd - Board interface task
LOCAL  0x22    * MTP3 module
LOCAL  0x23    * ISUP module
LOCAL  0x4d    * application on top of ISUP - user part task
*                eg: dvscc, gcss7/s7d
LOCAL  0x00    * tim_lnx - Timer task
LOCAL  0xcf    * s7_mgt - Management/config task
*
* Optional modules running on the host:
*
LOCAL  0xef    * s7_log - Display and logging utility
*
*
* Essential modules running on the board (all redirected via ssd):
*
REDIRECT  0x71  0x20    * MTP2 module (except SS7HD boards)
REDIRECT  0x8e  0x20    * On-board management module
*
*
* Redirection of status indications:
*
REDIRECT  0xdf  0x4d  * LIU/MTP2 status messages -> ss7d
*
* Dimensioning the Message Passing Environment:
*
NUM_MSGS   10000      * Number of standard size
*                       messages in the environment
*
*
* Now start-up all local tasks:
FORK_PROCESS  ./tim_lnx
FORK_PROCESS  ./tick_lnx
FORK_PROCESS  ./s7_log
FORK_PROCESS  ./s7_mgt
FORK_PROCESS  ./ssdd
FORK_PROCESS  ./mtp_lnx6
FORK_PROCESS  ./isp_lnx6
*
******************************************************************************
```

4.3.2 Generating the System.txt Configuration File

This section describes the procedure for generating a system.txt configuration.

First, the file must contain LOCAL declarations for all modules that are to run on the host computer. At a minimum, this must include the ssd module and the timer module. Hence, the following declarations must exist:

```
LOCAL 0x20 * ssdd - Board interface task
LOCAL 0x00 * tim_lnx - Timer task
```

LOCAL declarations are also required for any optional modules running on the host. Typically, this includes the s7_mgt protocol configuration utility and the user's own application module. It may also include any host-based protocol modules and the s7_log utility. For example:

```
LOCAL 0xcf * s7_mgt - Management/config task
LOCAL 0x4d * dvscc - Diva Virtual Signaling Call Control SS7
LOCAL 0x3d * s7_log - Prints messages to screen/file
LOCAL 0x23 * ISUP module
LOCAL 0x22 * MTP3 module
```

Once all the LOCAL declarations are in place, REDIRECT commands should be added for all modules that are running on the board so that any messages destined for these modules are transported via ssd (module_id = 0x20) and the device driver to the board.

The following REDIRECT commands are always required:

```
REDIRECT 0x71 0x20 * MTP2 module_id
REDIRECT 0x10 0x20 * CT Bus/Clocking control module (to be compatible to other Dialogic DSI boards)
REDIRECT 0x8e 0x20 * On-board management module
```

Having ensured that all modules running on the board are accessible, it is then necessary to ensure that any status indications issued from the board successfully arrive at a module running on the host. If this does not happen, the system quickly runs out of available messages for inter-process communication.

Two module_id's (0xdf and 0xef) require redirection to a suitable process running on the host, initially these messages should be redirected to the s7_log utility that prints out a line for each message received. Ultimately, the user's own application will expect to receive these notifications.

```
REDIRECT 0xdf 0x3d * LIU/MTP2 status messages -> s7_log
```

It is next necessary to include FORK_PROCESS commands for all modules running on the host computer. All systems require ssd, tick, and tim binaries to be run, therefore, the mandatory FORK_PROCESS commands are:

```
FORK_PROCESS ./ssdd
FORK_PROCESS ./tim_lnx
FORK_PROCESS ./tick_lnx
```

Finally, FORK_PROCESS commands should be added for any other modules running on the host, such as, protocol modules, user's application, or diagnostic utilities. For example:

```
FORK_PROCESS ./s7_mgt
FORK_PROCESS ./s7_log
FORK_PROCESS ./isp_lnx
FORK_PROCESS ./dvscc
FORK_PROCESS ./mtp_lnx6
```

4.4 Protocol Configuration

The Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards contains the s7_mgt protocol configuration utility that performs initialization of the software modules running on Diva Media Boards. It reads the protocol configuration data from a text file, called config.txt, and provides a quick and flexible method of configuring the protocol modules without the need to write software for that purpose. For more information, see Protocol Configuration Using the s7_mgt Utility below.

Alternatively, the protocol stack may be configured by sending the individual configuration messages documented in the per-module Programmer's Manuals for each protocol module. This approach is of particular use when the application needs to reset the board and run a new configuration without stopping the application program. For more information, see Protocol Configuration Using Discrete Messages on page 86.

4.4.1 Protocol Configuration Using the s7_mgt Utility

The s7_mgt protocol configuration utility uses the config.txt protocol configuration file by default. However, the -k option allows the user to specify an alternative file name if required. For example:

```
s7_mgt -kmyconfig.txt
```

The format of each configuration command is described in Configuration Command Reference on page 44.

The s7_mgt protocol configuration utility can optionally be configured to send a message to a nominated module on completion of the configuration sequence. This option is activated using the -i option to specify the receiving module_id. For example:

```
s7_mgt -i0xef
```

To assist problem diagnosis, the s7_mgt utility can be run using the -d option that generates additional diagnostic output. For example:

```
s7_mgt -i0xef -d
```

The following is an example config.txt protocol configuration file:

```
********************************************************************************
* Example SS7 Configuration File (config.txt) for use with
* Dialogic(R) Global Call for SS7 based on Dialogic(R) Diva(R) Media Boards
*
* This file needs to be modified to suit individual circumstances.
*
********************************************************************************
*
* Board Definitions:
*
********************************************************************************
* For Diva boards:
* DIVA_BOARD <board_id> <board_type> <flags> <serial>
* board_id:   logical number to identify the board, starts at 0
* board_type: identifies the type of Diva board.
*             Valid entries:
*             Diva PRI v.3 PCI (ADSP based)
*               70 - Dialogic Diva PRI/E1-30 PCI v3
*               71 - Dialogic Diva PRI/T1-24 PCI v3
*               72 - Dialogic Diva PRI/E1/T1-8 PCI v3
*               73 - Dialogic Diva V-PRI/E1-30 PCI v3
*               74 - Dialogic Diva V-PRI/T1-24 PCI v3
*               75 - Dialogic Diva PRI/E1/T1-CTI PCI v3
*             Diva 2/4PRI Boards (BF based)
*               79 - Dialogic Diva V-2PRI/E1/T1-60 PCI v1
*               81 - Dialogic Diva V-4PRI/E1/T1-120 PCI v1
*             PCI Express version of Diva PRI v.3 boards (ADSP based)
*               94 - Dialogic Diva PRI/E1-30 PCIe v3
*               95 - Dialogic Diva PRI/T1-24 PCIe v3
*               96 - Dialogic Diva V-PRI/E1-30 PCIe v3
*               97 - Dialogic Diva V-PRI/T1-24 PCIe v3
*               98 - Dialogic Diva PRI/E1/T1-CTI PCIe v3
*             PCI Express version of Diva 2/4PRI boards (BF based)
*               110 - Dialogic Diva V-4PRI PCIe HS v1
*               111 - Dialogic Diva V-2PRI PCIe HS v1
*               112 - Dialogic Diva V-1PRI PCIe HS v1
*             PCI Express version of Diva 4/8PRI boards (BF based)
*               113 - Dialogic Diva V-4PRI PCIe FS v1
*               114 - Dialogic Diva V-8PRI PCIe FS v1
*             Unified Messaging version of Diva PRI v.3 boards (ADSP based)
*               115 - Dialogic Diva UM-PRI/E1-30 PCI v3
*               116 - Dialogic Diva UM-PRI/T1-24 PCI v3
*               117 - Dialogic Diva UM-PRI/E1-30 PCIe v3
*               118 - Dialogic Diva UM-PRI/T1-24 PCIe v3
* flags:     for future use, shall be set to 0
* serial:    serial number of that board
* Note: The LIUs are implicit declared here
*       E.g. Diva board type 81 (Dialogic Diva V-4PRI/E1/T1-120 PCI v1) consists
*       of 4 LIUs.
*       Diva 'InterfaceNr' 0 -> LIU Id 0
*                          1 -> LIU Id 1
*                          2 -> LIU Id 2
*                          3 -> LIU Id 3
*
*         <board_id> <board_type> <flags> <serial>
DIVA_BOARD     0          81        0x0     1234  * Dialogic Diva V-4PRI/E1/T1-120 PCI v1
*                                                 * serial number 1234
*
*
********************************************************************************
*
* MTP parameters:
*
********************************************************************************
* MTP_CONFIG <reserved1> <reserved2> <options>
* ============
MTP_CONFIG 0 0 0x00040004  * RPO handling according to ITU-T 1992 (and later)
*                          * 14 bit point codes
*                          * Host-based MTP3
*
********************************************************************************
* MTP_LINKSET <linkset_id> <adjacent_spc> <num_links> <flags> <local_spc> <ssf>
* ============
MTP_LINKSET 0 129 2 0x0000 130 0x08 * APC: 129 as 14 bit PC (hex: 81, 4-3-4-3:0-1-0-1)
*                                   * two links in this linkset
*                                   * OPC: 130 as 14 bit PC (hex: 82, 4-3-4-3:0-1-0-2)
*                                   * ssf: national 0
*
********************************************************************************
* MTP_ROUTE <dpc> <norm_ls> <user_part_mask> <flags> <second_ls>
* ============
MTP_ROUTE 129 0 0x0020 * DPC: 129 as 14 bit PC (hex: 81, 4-3-4-3:0-1-0-1)
*                      * linkset Id: 0
*                      * user_part_mask: Bit 5 is set -> ISUP Service indicator
*
********************************************************************************
* MTP_LINK <link_id><linkset_id><link_ref><slc><board_id><blink><stream>
* ============
MTP_LINK  0    0    0    0    0    0   0    16   0x0006 * link on board Id 0,
*                                                       * liu Id 0,timeslot 16
*                                                       * signalling link test
*                                                       * 64 kb/s
MTP_LINK  1    0    1    1    0    2   2    16   0x0006 * link on board Id 0,
*                                                       * liu Id 1,timeslot 16
*                                                       * signalling link test
*                                                       * 64 kb/s
********************************************************************************
*
********************************************************************************
*
* ISUP parameters:
*
********************************************************************************
* ISUP_CONFIG <res1><res2><user_id><options><num_grps><num_ccts>
* ============
ISUP_CONFIG  0  0  0x4d  0x0414   4  128 * gcss7 module ID: 0x4d (see system.txt)
*                                        * call release procedures are supported by gcss7
*                                        * errors are reported to gcss7
*                                        * 4 Circuit groups are defined
*                                        * Max. CID 128
*
********************************************************************************
* Configure ISUP circuit groups:
* ISUP_CFG_CCTGRP <gid><dpc><base_cic><base_cid><cic_mask><options><user_inst>
*                                           <user_id><opc><ssf><variant><options2>
* ============
ISUP_CFG_CCTGRP 0 130 0x01 0x01 0x7FFF7FFF 0x249001e 0 0x4d 129 0x8 1 0x00d0
*               * Circuit group = 0
*               * DPC = 130 (HEX:0x82, 4-3-4-3:0-1-0-2)
*               * Base CIC = 0x01 (dec:1)
*               * Base CID = 0x01 (dec:1)
*               * group consist of 30 CICs (0x1-0xF, 0x11-0x1F)
*               * options:
*               *  - outgoing call priority:
*               *    Higher point code has priority on even cic s.
*               *  - T7, T9 (ACM, ANM) timers are used
*               *  - T2 or T6 are runninng in suspended state
*               *  - abnormal circuit group blocking procedures follows
*               *    Q.767, Q.764 (03/93) and ANSI T1.113 operation
*               *  - 'Confusion' (CFN) messages are enabled
*               *  - T34 for segtmented messages will be enabled
*               *  - Q.764 (03/93) end node (i.e. exchange type A) message
*               *    and parameter compatibility handling will be
*               *    performed (passing on unrecognised information is
*               *    possible)
*               * user instance = 0
*               * gcss7 module ID: 0x4d (see system.txt)
*               * OPC = 129 (HEX:0x81, 4-3-4-3:0-1-0-1
*               * Sub Service Field = 0x8 (national 0)
*               * variant = 1 (Q.761   Q.764 (1992 and later editions)
*               * options2:
*               * - T39 (waiting for IRS after sending IDR) is used
*               * - enable the user part unavailability procedure
ISUP_CFG_CCTGRP 1 130 0x20 0x21 0x7FFFFFFF 0x249001e 0 0x4d 129 0x8 1 0x00c0 *$ conform=ETSI_V2_V3
*               * Circuit group = 1
*               * Base CIC = 0x20 (dec:32)
*               * Base CID = 0x21 (dec:33)
*               * group consist of 31 CICs (0x20-0x3E)
*               * other parameters as for group 0
ISUP_CFG_CCTGRP 2 130 0x3F 0x41 0x7FFF7FFF 0x249001e 0 0x4d 129 0x8 1 0x00c0 *$ conform=ETSI_V2_V3
*               * Circuit group = 2
*               * Base CIC = 0x3F (dec:63)
*               * Base CID = 0x41 (dec:65)
*               * group consist of 30 CICs (0x3F-0x4D, 0x4F-0x5D)
*               * other parameters as for group 0
ISUP_CFG_CCTGRP 3 130 0x5E 0x61 0x7FFFFFFF 0x249001e 0 0x4d 129 0x8 1 0x00c0 *$ conform=ETSI_V2_V3
*               * Circuit group = 3
*               * Base CIC = 0x5E (dec:94)
*               * Base CID = 0x61 (dec:97)
*               * group consist of 31 CICs (0x5E-0x7D)
*               * other parameters as for group 0
*
********************************************************************************
```

Note: When using the Dialogic® SS7 Software for Dialogic® Diva® Interfaces, additional definitions are needed to assign ISUP's CIC's to timeslots (see Dialogic® SS7 Software for Dialogic® Diva® Interfaces Reference Guide).

Chapter 5: Program Execution

This chapter describes how to start the software and execute programs. It assumes that:

- The software has already been installed. See Use Cases and Installation on page 13.
- The system.txt configuration file has been modified correctly. See System Configuration on page 20.
- The config.txt protocol configuration file has been modified correctly. See Protocol Configuration on page 22.

When running the Dialogic® SS7 Software for Dialogic® Diva® Interfaces on Dialogic® Diva® Media Boards, the required drivers and processes are started automatically if not configured otherwise.

For all other scenarios or when configured to start manually, consider the following. There are three main stages to get a new application up and running:

- Ensure that the Dialogic® Diva® System Release LIN software is installed and running (step 1, below).
- Ensure that the protocol software is running on the host (step 2 and 3, below).
- Start the application program (step 4 - 6, below).

Proceed as follows:

1. Ensure that the Diva System Release LIN software has been installed, configured, and started.
2. Ensure that the system.txt configuration file has been modified according to the system requirements to select the correct protocols.
3. If using the s7_mgt protocol configuration utility, ensure that the config.txt protocol configuration file has been edited to provide the correct protocol configuration.
4. Change to the directory containing all the SS7 binaries (default path: `/usr/lib/opendiva/dlgss7`).
5. Run the gctload program to start the software. Optionally, specify the system configuration file with the `-c` option.

   To run the system in the foreground, enter:

   ```
   gctload -csystem.txt
   ```

   To run the system in the background, enter:

   ```
   gctload -csystem.txt &
   ```

   The gctload program initializes the Dialogic® DSI SS7 system environment and starts other processes. The s7_mgt process configures all the protocol modules. A banner confirms that the system is running.
6. Activate and deactivate signaling links, if required, using the mtpsl example utility as follows:

   ```
   mtpsl {act | deact} <linkset_id> <link_ref>
   mtpsl act 0 0
   mtpsl deact 0 0
   ```

   Note: When running the Dialogic® SS7 Software for Dialogic® Diva® Interfaces this will be done by `dvscc`.
7. Shutdown the host software by running the gctload program using the `-x` option.

   ```
   gctload -x
   ```

   Any modules that have been started by the gctload program are terminated automatically.

Chapter 6: Message Reference

This chapter describes the individual messages that may be sent to or received from a Dialogic® Diva® Media Board used in conjunction with the Dialogic® DSI SS7 Stack. Some messages are sent by the user's application software, while others are sent by utility programs such as the s7_mgt protocol configuration utility.

Prior to sending any message to the board, the application should call the **GCT_set_instance( )** library function to select which board the message will be sent to. After receiving a message from the board, the application should call the **GCT_get_instance( )** library function to determine which board the message came from. These library functions are described in the Dialogic® SS7 Protocols Software Environment Programmer's Manual.

The various messages used are grouped in the following categories:

- General Configuration Messages below
- Hardware Control Messages on page 34
- MTP Interface Messages on page 34
- Event Indication Messages on page 38

The Message Summary Table on page 43 provides a summary of all messages. The message header for all messages has the same general format. See the Message Format appendix in the Dialogic® SS7 Protocols Software Environment Programmer's Manual.

6.1 Software Module IDs for Dialogic® Diva® Media Boards

The table below lists the software modules IDs (by mnemonic and value) used in conjunction with Diva Media Boards.

Mnemonic	**Value**	**Description**
MGMT_TASK_ID	0x8e	SS7 Diva Media Board Management module by ssdd
MVD_TASK_ID	0x10	dummy
SS7_MTP2_TASK_ID	0x71	MTP2 module

6.2 General Configuration Messages

General configuration messages are typically issued by the s7_mgt protocol configuration utility, in which case they need not, and should not, be generated by any user application software.

If the user elects not to use the s7_mgt protocol configuration utility, it is necessary for the application to build and send messages that:

- configure the ssd module
- reset each board
- configure each board
- optionally configure additional routes

The messages in the general configuration category include:

- SSD_MSG_RESET - SSD Reset Request on page 28
- SSD_MSG_RST_BOARD - Board Reset Request on page 29
- MGT_MSG_L1_CONFIG - Layer 1 Configuration Request on page 31
- MGT_MSG_L1_END - Layer 1 Configuration End on page 33

6.2.1 SSD_MSG_RESET - SSD Reset Request

Synopsis

Sets up ssd module run-time options at initialization time.

Note: When using the s7_mgt protocol configuration utility, this message is generated by s7_mgt and should not be generated by the user.

Format

MESSAGE HEADER	
Field Name	**Meaning**
type	SSD_MSG_RESET (0x7680)
id	0
src	Sending the module's module_id
dst	SSD_TASK_ID
rsp_req	Used to request a confirmation
hclass	0
status	0
err_info	0
len	24

PARAMETER AREA		
Offset	**Size**	**Name**
0	1	reserved. Set to 0.
1	2	reserved. Set to 0.
3	1	mgmt_id
4	18	reserved. Set to 0.
22	2	num_boards

Description

This message is used during initialization by the application to reset the ssd module and set up its run-time parameters.

The confirmation message (if requested) indicates success with a status value of 0.

Parameters

The **SSD_MSG_RESET** message includes the following parameters:

- **mgmt_id**

  The module_id of the management module to which ssd should sent board status indications.

- **num_boards**

  The maximum number of boards that ssd is required to manage. This should not exceed 16.

6.2.2 SSD_MSG_RST_BOARD - Board Reset Request

Synopsis

Checks the availability of the corresponding Dialogic® Diva® Media Board.

Note: When using the s7_mgt protocol configuration utility, this message is generated by s7_mgt and should not be generated by the user.

Format

<table>
<tr><th colspan="3">MESSAGE HEADER</th></tr>
<tr><th colspan="2">Field Name</th><th>Meaning</th></tr>
<tr><td colspan="2">type</td><td>SSD_MSG_RST_BOARD (0x7681)</td></tr>
<tr><td colspan="2">id</td><td>board_id</td></tr>
<tr><td colspan="2">src</td><td>Sending module's module_id</td></tr>
<tr><td colspan="2">dst</td><td>SSD_TASK_ID</td></tr>
<tr><td colspan="2">rsp_req</td><td>Used to request a confirmation</td></tr>
<tr><td colspan="2">hclass</td><td>0</td></tr>
<tr><td colspan="2">status</td><td>0</td></tr>
<tr><td colspan="2">err_info</td><td>0</td></tr>
<tr><td colspan="2">len</td><td>26</td></tr>
<tr><th colspan="3">PARAMETER AREA</th></tr>
<tr><th>Offset</th><th>Size</th><th>Name</th></tr>
<tr><td>0</td><td>2</td><td>board_type - 7 for Diva Media Board</td></tr>
<tr><td>4</td><td>4</td><td>reserved. Set to 0.</td></tr>
<tr><td>6</td><td>7</td><td>serial_number of Diva Media Board as ASCII value</td></tr>
<tr><td>13</td><td>11</td><td>reserved. Set to 0.</td></tr>
<tr><td>24</td><td>2</td><td>Diva Media Board type</td></tr>
</table>

Description

This message is used by the application during initialization (or reconfiguration) to detect and reset all DSI resources of the board.

The confirmation message (if requested) indicates success with a status value of 0. This implies that the reset operation has commenced, but does not imply completion. The application should then wait until a Board Status Indication message is received that indicates either successful completion of the reset and download operation or failure during the procedure.

Parameters

The **SSDD_MSG_RST_BOARD** message includes the following parameters:

- **board_type**

  The type of board to be reset. Set to 7 for a Dialogic® Diva® Media Board.

- **serial_number**

  Null terminated string giving the serial number of the Diva Media Board.

- **Dialogic Diva Media Board type**

  Identifier of the Diva Media Board type.

Dialogic® Diva® Media Board Model	**Board Type**
Diva PRI/E1/T1-CTI PCI v3	75
Diva PRI/E1/T1-CTI PCIe v3	98
Diva V-PRI/T1-24 PCI v3	74
Diva V-PRI/T1-24 PCIe v3	97
Diva V-PRI/E1-30 PCI v3	73
Diva V-PRI/E1-30 PCIe v3	96
Diva UM-PRI/T1-24 PCI v3	116
Diva UM-PRI/T1-24 PCIe v3	118
Diva UM-PRI/E1-30 PCI v3	115
Diva UM-PRI/E1-30 PCIe v3	117
Diva PRI/E1/T1-8 PCI v3	72
Diva PRI/T1-24 PCI v3	71
Diva PRI/T1-24 PCIe v3	95
Diva PRI/E1-30 PCI v3	70
Diva PRI/E1-30 PCIe v3	94
Diva V-2PRI/E1/T1-60 PCI v1	79
Diva V-4PRI/E1/T1-120 PCI v1	81
Diva V-1PRI PCIe HS v1	112
Diva V-2PRI PCIe HS v1	111
Diva V-4PRI PCIe HS v1	110
Diva V-4PRI PCIe FS v1	113
Diva V-8PRI PCIe FS v1	114

6.2.3 MGT_MSG_L1_CONFIG - Layer 1 Configuration Request

Synopsis

Message sent to a board to configure the layer 1 of each signaling (MTP2) link.

Note: When using the s7_mgt protocol configuration utility, this message is generated by s7_mgt and should not be generated by the user.

Format

<table>
<tr><th colspan="3">MESSAGE HEADER</th></tr>
<tr><th colspan="2">Field Name</th><th>Meaning</th></tr>
<tr><td colspan="2">type</td><td>MGT_MSG_L1_CONFIG (0x7f17)</td></tr>
<tr><td colspan="2">id</td><td>0</td></tr>
<tr><td colspan="2">src</td><td>Sending module's module_id</td></tr>
<tr><td colspan="2">dst</td><td>MGMT_TASK_ID</td></tr>
<tr><td colspan="2">rsp_req</td><td>Used to request a confirmation</td></tr>
<tr><td colspan="2">hclass</td><td>0</td></tr>
<tr><td colspan="2">status</td><td>0</td></tr>
<tr><td colspan="2">err_info</td><td>0</td></tr>
<tr><td colspan="2">len</td><td>40</td></tr>
<tr><th colspan="3">PARAMETER AREA</th></tr>
<tr><th>Offset</th><th>Size</th><th>Name</th></tr>
<tr><td>0</td><td>2</td><td>reserved. Set to 0.</td></tr>
<tr><td>2</td><td>2</td><td>sp_channel</td></tr>
<tr><td>4</td><td>2</td><td>data_rate</td></tr>
<tr><td>6</td><td>2</td><td>reserved. Set to 0.</td></tr>
<tr><td>8</td><td>2</td><td>reserved. Set to 0.</td></tr>
<tr><td>10</td><td>2</td><td>reserved. Set to 0.</td></tr>
<tr><td>12</td><td>2</td><td>link_stream</td></tr>
<tr><td>14</td><td>2</td><td>link_timeslot</td></tr>
<tr><td>16</td><td>2</td><td>reserved. Set to 0.</td></tr>
<tr><td>18</td><td>2</td><td>reserved. Set to 0.</td></tr>
<tr><td>20</td><td>4</td><td>options</td></tr>
<tr><td>24</td><td>4</td><td>reserved. Set to 0.</td></tr>
<tr><td>28</td><td>12</td><td>reserved. Set to 0.</td></tr>
</table>

Description

This message is used to configure physical signaling links. The message has to be sent once for each signaling link to be configured.

Parameters

The **MGT_MSG_L1_CONFIG** message includes the following parameters:

- **sp_channel**

  The logical ID (LLId) of the channel on the T1/E1 line interface unit (LIU). This value should be unique for each channel on the same board. Possible values are in the range of 0 to one less than the number of links supported per LIU.

- **data_rate**

  Used for setting the link operation. The following table shows the permitted values and their meaning.

Value	Date Rate
0	64 kbps
1	56 kbps
2	48 kbps

  All other values are reserved for future use.

- **link_stream**

  Signaling stream. It is the logical identity of the T1/E1 line interface unit (LIU) containing the signaling link. Possible values are 0 to one less than the number of LIUs. For example, the Dialogic® Diva® V-4PRI/E1/T1-120 PCI v1 Media Board provides 4 LIUs so the value for this parameter has to be set to:

  - 0 - to configure the link to run on LIU 0
  - 1 - to configure the link to run on LIU 1
  - 2 - to configure the link to run on LIU 2
  - 3 - to configure the link to run on LIU 3

  Note: In the Dialogic® Diva® System Release LIN documentation, "LIU" is called "port".
  - Port 1 corresponds to LIU 0
  - Port 2 corresponds to LIU 1
  - Port 3 corresponds to LIU 2
  - Port 4 corresponds to LIU 3

- **link_timeslot**

  Signaling timeslot. This field is used to configure conventional SS7 links when link_source is set to 0. Otherwise it must be set to 0. The value ranges for link_timeslot are:

  - For an E1 interface: 1 to 31.
  - For a T1 interface: 1 to 24.

- **options**

  A 32-bit value containing run-time options as follows:

  - Bit 0: Set to 1 to disable automatic FISU generation. This is normally required for Japanese MTP operation only.
  - All other Bits: Set to 0.

6.2.4 MGT_MSG_L1_END - Layer 1 Configuration End

Synopsis

Message sent to a board to remove an existing layer 1 link that was previously configured by sending an **MGT_MSG_L1_CONFIG** message.

Format

MESSAGE HEADER	
Field Name	**Meaning**
type	MGT_MSG_L1_END (0x7f18)
id	0
src	Sending module's module_id
dst	MGMT_TASK_ID
rsp_req	Used to request a confirmation
hclass	0
status	0
err_info	0
len	4

PARAMETER AREA		
Offset	**Size**	**Name**
0	2	reserved. Set to 0.
2	2	sp_channel

Description

This message is used to remove an existing signaling link. The message has to be sent once for each signaling link to be removed.

Parameters

The **MGT_MSG_L1_END** message includes the following parameter:

sp_channel

The logical ID (LLId) of the channel on the T1/E1 line interface unit (LIU). This value should be unique for each channel on the same board and signaling processor. Possible values are in the range of 0 to one less than the number of links supported per LIU.

6.3 Hardware Control Messages

Hardware control messages are used with the Dialogic® DSI SS7 board families SS7HD and SS7SPCI. They are not supported by Dialogic® Diva® Media Boards as the T1/E1 Line Inteface Units (LIUs) are configured by the Dialogic® Diva® System Release software (see the Dialogic® Diva® System Release LIN Reference Guide).

6.4 MTP Interface Messages

MTP interface messages allow signaling links to be activated and deactivated by the user and provide a mechanism for communication between the MTP3 module and the user part module (for example, ISUP, TUP or SCCP). In many cases, the user part module is a Dialogic® DSI SS7 protocol so the user does not need to handle the MTP primitives as they pass directly between MTP3 and the user part module.

In the case that the user application is implementing the user part functionality, the MTP primitives are applicable. See the Dialogic® SS7 Protocols MTP2 Programmer's Manual and the Dialogic® SS7 Protocols MTP3 Programmer's Manual for more information.

The messages in the MTP interface category include the SS7_MSG_CONFIG - MTP2 Link Configuration Request, see below.

6.4.1 SS7_MSG_CONFIG - MTP2 Link Configuration Request

Synopsis

Message issued by management to MTP2 to configure an individual signaling link for operation.

Format of Message Header

MESSAGE HEADER	
Field Name	**Meaning**
type	SS7_MSG_CONFIG (0x7203)
id	l2_llid
src	Sending the module's module_id
dst	MTP2 module Id (0x71)
rsp_req	Sending layers's bit set if response required
hclass	0
status	0
err_info	0
len	38, 42 or 60 (see below)

Format of Parameter Area

PARAMETER AREA		
Offset	**Size**	**Name**
0	2	options - run time options
2	1	upper_id - module Id of MTP2 user (typical MTP3's module_id)
3	1	lower_id - module Id of physical layer (0x00)
4	1	mgmt_id - module_id of management module
5	1	reserved. Set to 0.
6	2	max_SIF_len - max. length of Signaling Information Field
8	2	con_onset - congestion onset threshold
10	2	con_abate - congestion abatement threshold
12	2	pcr_n1 - Preventive Cyclic Retransmission (PCR) N1 threshold
14	2	pcr_n2 - Preventive Cyclic Retransmission (PCR) N2 threshold
16	2	rtv_attempts - max. number of retrieval attempts
18	2	t1 - timer T1 value
20	2	t2 - timer T2 value
22	2	t3 - timer T3 value
24	2	t4n - timer T4 normal value
26	2	t4e - timer T4 emergency value
28	2	t5 - timer T5 value
30	2	t6 - timer T6 value
32	2	t7 - timer T7 value
34	2	t_suerm - period between SUERM/EIM checks
36	2	t_rtv - period between retrieval attempts
38	2	cong_discard - congestion discard threshold
40	2	l3_link_id - MTP3 link Id
42	2	co1 - congestion onset threshold 1
44	2	co2 - congestion onset threshold 2
46	2	co3 - congestion onset threshold 3
48	2	ca1 - congestion abatement threshold 1
50	2	ca2 - congestion abatement threshold 2
52	2	ca3 - congestion abatement threshold 3
54	2	cd1 - congestion discard threshold 1
56	2	cd2 - congestion discard threshold 2
58	2	cd3 - congestion discard threshold 3

Description

This message is used to configure the operational parameters for an individual signaling link and to cause the power up action defined in Q.703 to be executed. One **SS7_MSG_CONFIG** message must be issued to MTP2 (after the **SS7_MSG_RESET** message has been issued, see the Dialogic® SS7 Protocols MTP2 Programmer's Manual) for each link to be used. Subsequent **SS7_MSG_CONFIG** messages may be issued to the MTP2 module to modify timer configuration parameters. However, these messages do not affect SS7 operation (that is, the power up sequence is not to be re-executed, but the parameters are modified).

For backwards compatibility, the MTP2 module accepts messages with three different parameter area lengths: 38, 42, or 60 bytes. If the length is less than 42, the **cong_discard** parameter is set to 0 so that congestion discard does not take place, and the **l3_link_id** parameter is set to the same value as the **l2_llid**. If the length is less than 60, the use of single congestion thresholds is assumed.

Note: To use multiple congestion thresholds, it is necessary to set the **S7C_MCONG** bit (bit 3) in the options field in addition to supplying a full length parameter area.

Parameters

The **SS7_MSG_CONFIG** message includes the following parameters:

- **Options**

This field is used to convey run-time options to the module as shown in the following table:

Bit	Meaning
0	Set to 1 to enable the Preventive Cyclic Retransmission of error correction or set to 0 to enable the Basic Method of error correction.
3	Set to 1 to enable the Multiple Congestion States and Multiple Message Priority option. This option should always be enabled when running in ANSI mode.
5	Set to 1 to cause generated LSSUs to have a 2 octet status field, otherwise LSSUs are generated with a single octet status field.
6	Set to 1 if it is required that MTP2 waits for a Continue Request from MTP3 prior to resuming normal operation prior to a period of processor outage.
7	Set to 1 to invoke special MTP2 operation for use in Japanese networks.
other	reserved. Set to 0.

- **upper_id**

  The module ID of the upper layer module. This is the module to which all MTP2/MTP3 indications are to be issued and is typically the module ID of the MTP3 module.

- **lower_id**

  The module ID of the on-board driver module that interfaces with the physical interface. This must always be set to 0.

- **mgmt_id**

  The module ID of the management module to which all trace messages, event indications, and state change messages are to be sent.

- **max_SIF_len**

  The maximum length of Signaling Information Field (SIF) to support. This should be set to either 62 or 272 in accordance with Q.703.

- **cong_onset**

  The congestion onset threshold for use with the single congestion threshold mode of operation. Congestion is indicated when the total number of messages in the transmit and retransmit buffers rises to this value.

- **cong_abate**

  The congestion abatement threshold for use with the single congestion threshold mode of operation. Link uncongested is indicated when the total number of messages in the transmit and retransmit buffers falls below this value.

- **pcr_n1**

  The N1 threshold for use with the Preventive Cyclic Retransmission method of error correction. For 7-bit sequence number operation, the default and maximum value is 127. This parameter may be set to a value lower than the default to limit the maximum number of messages in the retransmission buffer.

- **pcr_n2**

  The N2 threshold for use with the Preventive Cyclic Retransmission method of error correction. This should typically be set to approximately eight times the loop delay in ms for 64 kbps operation or 7 times the loop delay in ms for 56 kbps operation. If set to 0, the MTP2 module assumes a value of 12800 for an HSL link, 400 otherwise.

- **t1, t2, t3, t4n, t4e, t5, t6, t7**

  Values for the protocol timers as defined in Q.703. These should be set to the number of (tick * timer_res) intervals required for the timer. The timers are checked for expiry every timer_res number of ticks. The value given for t1, t2 etc. is the number of times that the timer is checked before indicating expiry.

- **t_suerm**

  The time interval between issuing check `SUERM/EIM` commands to the driver. Specified in the same manner as the protocol timers t1, t2 etc. This should always be set to 10.

- **t_rtv**

  The time interval between retrieval attempts specified in the same manner as the protocol timers t1, t2 etc. Retrieval can only take place once the driver has released all messages queued for transmission. This timer determines the period between successive attempts. This should always be set to 1.

- **cong_discard**

  The congestion discard threshold for use with the single message priority mode of operation. When the combined number of messages in the transmit and retransmit buffers reaches this threshold, further messages are discarded. The congestion discard threshold cannot be set to a value greater than 4160.

- **l3_link_id**

  The value to use in the ID field of all indications issued to the upper module (that is, MTP3). For single signaling processor systems, this is typically the same as the **l2_llid**. However, when a system contains more than one MTP2 processor, this may not be so.

- **co1, co2, co3, ca1, ca2, ca3, cd1, cd2, cd3**

  Congestion onset, abatement and discard thresholds for use when the Multiple Congestion Thresholds mode of operation is selected.

6.5 Event Indication Messages

Event indication messages are the mechanism by which protocol and software error events are reported to the application. These messages are generated asynchronously by different modules within the stack. They include:

- SSD_MSG_STATE_IND - Board Status Indication below
- API_MSG_CNF_IND - Configuration Completion Status Indication on page 39
- MGT_MSG_EVENT_IND - Error Indication on page 40
- MGT_MSG_SS7_EVENT - MTP2 Q.752 Event Indication on page 42

6.5.1 SSD_MSG_STATE_IND - Board Status Indication

Synopsis

SSD_MSG_STATE_IND is sent to the application on completion of the availability check of the corresponding Dialogic® Diva® Media Board or on detection of a board status event.

Note: This message is not required when using the s7_mgt protocol configuration utility.

Format

MESSAGE HEADER	
Field Name	**Meaning**
type	SSD_MSG_STATE_IND (0x06a0)
id	board_id
src	SSD_TASK_ID (0x20)
dst	mgmt_id for SSD
rsp_req	0
hclass	0
status	event_type (see below)
err_info	0
len	0

Description

This message is used to convey the status of a board reset operation (success of failure) to the user.

Parameter

The **SSD_MSG_STATE_IND** message header uses the following parameter:

- **event_type**

  The status is indicated in the status field of the message header. The following table shows the possible **event_type** values:

Value	Meaning
0x60	Reset/Detection successful
0x62	Board failure
0x64	Board removal
0x65	Board insertion
0x70	Message congestion towards the board cleared
0x71	Message congestion towards the board onset
0x72	Message discard towards the board cleared

6.5.2 API_MSG_CNF_IND - Configuration Completion Status Indication

Synopsis

Message issued by the s7_mgt protocol configuration utility on completion of initial configuration sequence.

Format

MESSAGE HEADER	
Field Name	**Meaning**
type	API_MSG_CNF_IND (0x0f09)
id	0
src	0xcf
dst	Notification module (see below)
rsp_req	0
hclass	0
status	completion_status (see below)
err_info	Reserved for future use
len	0

Description

This message is issued by the s7_mgt protocol configuration utility on completion of the initial configuration sequence and indicates either success (status=0) or an error condition that occurred during configuration. The message is only issued when s7_mgt is run with the `-i` command line option specifying the module_id of the Notification Module to which the message should be sent. For example:

```
s7_mgt -i0x2d
```

Note: It is recommended that the user invokes this option, then waits for an **API_MSG_CNF_IND** message to ensure that the application does not attempt to send messages until initial configuration is complete.

Parameters

The **API_MSG_CNF_IND** message header uses the following parameter:

- **completion_status**

  The result of initial configuration. The following table shows the possible values and their meaning.

Value	Meaning
0	Success
1	Error opening the config.txt protocol configuration file
2	Syntax or value error in the config.txt protocol configuration file
3	Error during configuration (invalid parameters)
4	Error during configuration (no response)

6.5.3 MGT_MSG_EVENT_IND - Error Indication

Synopsis

Message issued to the system management advising of errors or unexpected events occurring within the protocol software.

Format

MESSAGE HEADER	
Field Name	**Meaning**
type	MGT_MSG_EVENT_IND (0x0008)
id	0 (unless shown below)
src	sending module id
dst	Management module id
rsp_req	0
hclass	0
status	ERROR_CODE (see below)
err_info	Timestamp
len	0

Parameters

The **MGT_MSG_EVENT_IND** message header includes the following parameter:

- **ERROR_CODE**

  An error code that identifies the error. The following table shows the possible values and their meaning.

Value	Mnemonic	ID	Description
0x31	S7E_RESET_ERR		MTP2 failed to initialize
0x33	S7E_POOL_EMPTY	l2_llid	No free buffers in MTP2 transmit pool
0x34	S7E_TX_FAIL	l2_llid	Failed to send LSSU/FISU to driver
0x35	S7E_HDR_ERR	l2_llid	No room to add MTP2 header, SU not transmitted
0x36	S7E_LEN_ERR	l2_llid	Length error, SU not transmitted.
0x37	S7E_MSU_SEND	l2_llid	Failed to send SU to lower layer, protocol should handle retransmission
0x39	S7E_BAD_PRIM	l2_llid	MTP2 unable to accept primitive
0x3a	S7E_BAD_LLID	l2_llid	Invalid l2_llid in HDR structure
0x3b	S7E_MEM_ERR	l2_llid	MTP2 memory allocation error
0x3c	S7E_RTVL_ERR	l2_llid	MTP2 failure to perform retrieval
0x51	MTP_BAD_PRIM	0	MTP3 unable to accept primitive
0x52	MTP_POOL_EMPTY	0	No free frames in MTP3 transmit pool
0x53	MTP_TX_FAIL	0	MTP3 failed to send MSU to lower layer
0x54	MTP_LEN_ERR	0	MSU too long for buffer
0x55	MTP_SLT_FAIL	link_id	Signaling link test failure
0x57	MTP_TALLOC_ERR	0	MTP3 failed to allocate T_FRAME
0x58	MTP_BAD_ID	0	Invalid ID in message HDR
0x59	MTP_MALLOC_ERR	0	MTP3 unable to allocate MSG
0x5a	MTP_BSNT_FAIL	link_id	Failure to retrieve BSNT
0x5b	MTP_RTV_FAIL	link_id	Retrieval failure

0x5c	MTP_BAD_FSN	link_id	Erroneous FSN in COA
0x5d	MTP_BAD_COO	link_id	COO received after changeover complete
0x5e	MTP_SNMM_ERR	0	Internal software error
0x5f	MTP_SLTM_ERR	0	Internal software error
0x60	MTP_NO_COA	link_id	Failed to receive COA
0x61	MTP_NO_CBA	link_id	Failed to receive CBA
0x66	MTP_TIM_ERR	timer ref	MTP3 attempt to reuse active timer resource
0x67	MTP_RRT_OVRFLW		Messages discarded due to overflow of Rerouting buffer
0x68	MTP_FLUSH_FAIL	link_id	MTP3 failed to receive Flush Ack from MTP2
0x69	MTP_FLUSH_L2	link_id	MTP2 transmission buffers flushed (due to RPO)

6.5.4 MGT_MSG_SS7_EVENT - MTP2 Q.752 Event Indication

Synopsis

Message issued by the MTP2 module to advise management of protocol events in accordance with Q.752.

Format

MESSAGE HEADER	
Field Name	**Meaning**
type	MGT_MSG_SS7_EVENT (0x0202)
id	l2_llid
src	MTP2 module id
dst	Management module id
rsp_req	0
hclass	0
status	EVENT CODE (see below)
err_info	Timestamp
Next	0
len	0

Description

This primitive is used by MTP2 to advise system management of the occurrence of protocol related events in accordance with Q.752. Currently, these events relate to the following:

- the reason for a signaling link (previously in service) going out of service (events prefixed S7F_)
- the occurrence of congestion related events (prefixed S7G_)
- timer expired (prefixed S7T_)
- a proving failure (prefixed S7P_)

Parameters

The **MGT_MSG_SS7_EVENT** message header includes the following parameter:

- **EVENT CODE**

  The event that has just occurred. The following table indicates the possible values and their meaning.

Value	Mnemonic	Description
0	S7F_STOP	Stop request received
1	S7F_FIBR_BSNR	Abnormal FIBR/BSNR
2	S7F_EDA	Excessive delay of acknowledgement
3	S7F_SUERM	Excessive error rate (SUERM or EIM)
4	S7F_ECONG	Excessive congestion
5	S7F_SIO_RXD	Unexpected SIO received
6	S7F_SIN_RXD	Unexpected SIN received
7	S7F_SIE_RXD	Unexpected SIE received
8	S7F_SIOS_RXD	SIOS received
16	S7G_CONG	Onset of signaling link congestion
17	S7G_CONG_CLR	Abatement of signaling link congestion
18	S7G_CONG_DIS	Congestion event caused MSU discard
32	S7T_T1_EXP	Timer T1 expiry
33	S7T_T2_EXP	Timer T2 expiry
34	S7T_T3_EXP	Timer T3 expiry
48	S7P_AERM	Failed proving attempt

6.6 Message Summary Table

The following table lists all the messages described in this manual by message type.

Value	Mnemonic	Description
0x0008	MGT_MSG_EVENT_IND	See MGT_MSG_EVENT_IND - Error Indication
0x0202	MGT_MSG_SS7_EVENT	MTP2 Q.752 Event Indication, see MGT_MSG_SS7_EVENT - MTP2 Q.752 Event Indication
0x06a0	SSD_MSG_STATE_IND	Board Status Indication, see SSD_MSG_STATE_IND - Board Status Indication
0x0f09	API_MSG_CNF_IND	Configuration Completion Status Indication, see API_MSG_CNF_IND - Configuration Completion Status Indication
0x3680		Confirmation of SSD_MSG_RESET, see SSD_MSG_RESET - SSD Reset Request
0x3681		Confirmation of SSD_MSG_RST_BOARD, see SSD_MSG_RST_BOARD - Board Reset Request
0x7203	SS7_MSG_CONFIG	MTP2 Link Configuration Request, see SS7_MSG_CONFIG - MTP2 Link Configuration Request
0x7680	SSD_MSG_RESET	SSD Reset Request, see SSD_MSG_RESET - SSD Reset Request
0x7681	SSD_MSG_RST_BOARD	Board Reset Request, see SSD_MSG_RST_BOARD - Board Reset Request
0x7f17	MGT_MSG_L1_CONFIG	Layer 1 Configuration Request, see MGT_MSG_L1_CONFIG - Layer 1 Configuration Request
0x7f18	MGT_MSG_L1_END	Layer 1 Configuration End, see MGT_MSG_L1_END - Layer 1 Configuration End

Chapter 7: Configuration Command Reference

This chapter describes the commands and parameters used in the config.txt protocol configuration file. These commands are used by the s7_mgt protocol configuration utility to perform one time configuration of the protocol stack at startup.

The commands are logically grouped in the following categories:

- Physical Interface Configuration Command below
- MTP Configuration Commands on page 46
- ISUP Configuration Commands on page 52

Configuration of other SS7 protocols are not described in this document, please refer to the appropriate programmer’s manual.

Note: When using the Dialogic® SS7 Software for Dialogic® Diva® Interfaces, additional definitions are needed to assign ISUP's CIC's to "real" timeslots (see the Dialogic® SS7 Software for Dialogic® Diva® Interfaces Reference Guide).

7.1 Physical Interface Configuration Command

The physical interface configuration command is:

```
DIVA_BOARD - Configure Dialogic® Diva® Media Board
```

Synopsis

Command to configure a Diva Media Board for use in the DSI SS7 environment.

Syntax

```
DIVA_BOARD <board_id> <board_type> <flags> <serial>
```

Example

```
DIVA_BOARD 0 81 0x0 1092
```

Parameters

The `DIVA_BOARD` command includes the following parameters:

- **<board_id>**

  The logical identity of the board in the range of 0 to one less than the number of boards supported (staring at 0, range 0 to 15).

- **<board_type>**

  Diva Media Board type.

Dialogic® Diva® Media Board Model	Board Type
Diva PRI/E1/T1-CTI PCI v3	75
Diva PRI/E1/T1-CTI PCIe v3	98
Diva V-PRI/T1-24 PCI v3	74
Diva V-PRI/T1-24 PCIe v3	97
Diva V-PRI/E1-30 PCI v3	73
Diva V-PRI/E1-30 PCIe v3	96
Diva UM-PRI/T1-24 PCI v3	116
Diva UM-PRI/T1-24 PCIe v3	118
Diva UM-PRI/E1-30 PCI v3	115
Diva UM-PRI/E1-30 PCIe v3	117
Diva PRI/E1/T1-8 PCI v3	72

Diva PRI/T1-24 PCI v3	71
Diva PRI/T1-24 PCIe v3	95
Diva PRI/E1-30 PCI v3	70
Diva PRI/E1-30 PCIe v3	94
Diva V-2PRI/E1/T1-60 PCI v1	79
Diva V-4PRI/E1/T1-120 PCI v1	81
Diva V-1PRI PCIe HS v1	112
Diva V-2PRI PCIe HS v1	111
Diva V-4PRI PCIe HS v1	110
Diva V-4PRI PCIe FS v1	113
Diva V-8PRI PCIe FS v1	114

- **<flags>**

  Reserved for future use, set to 0x0

- **<serial>**

  Serial number of the Diva Media Board.

7.2 MTP Configuration Commands

The Message Transfer Part (MTP) configuration commands are:

7.2.1 MTP_CONFIG - Configure MTP

Synopsis

The global configuration parameters for the Message Transfer Part (MTP).

Syntax

```
MTP_CONFIG <reserved1> <reserved2> <options>
```

Example

```
MTP_CONFIG 0 0 0x00040000
```

Parameters

The MTP_CONFIG command includes the following parameters:

- **<reserved1> , <reserved2>**

  These parameters are reserved for backwards compatibility only. For applications conforming to this release, these parameters should always be set to 0.

- **<options>**

  A 32-bit value containing run-time options for the operation of MTP as follows:

  - Bit 0 is set to 1 to disable the MTP3 message discrimination function (allowing the signaling point to receive all messages irrespective of the destination point code contained in the message) or 0 to allow the discrimination function to operate normally.
  - Bit 1 is set to 1 to disable sub-service field (SSF) discrimination. If this bit is set to 0, each received MSU with an ssf value that does not match the configured ssf value for that link set is discarded.
  - Bit 3 is set to 1 to cause MTP3 to generate a User Part Unavailable (UPU) message to the network on receipt of a message containing a Service Indicator (SI) value that has not been configured. If set to 0, the message will be discarded without sending the UPU message.
  - Bit 8 is set to 1 to select ANSI operation, otherwise it should be set to 0.
  - Bit 9 is set to 1 to select 24-bit point codes in the MTP routing label or 0 to select 14-bit or 16-bit point codes. This bit should always be set to 1 when ANSI operation is required.
  - Bit 10 is set to 1 for ANSI operation, otherwise it should be set to 0.
  - Bit 11 is set to 1 for ANSI operation, otherwise it should be set to 0.
  - Bit 18 is used to control MTP functionality in the event of detection of Remote Processor Outage (RPO). If set to 1, RPO is handled in accordance with the ITU-T 1992 (and later) recommendations. If set to 0, on detection of RPO, the signaling link is taken out of service and restoration commences. This bit should normally be set to 1.
  - Bit 20 is set to 1 to select 16-bit point codes in the MTP routing label for use in Japanese networks or 0 to allow the use of 24-bit or 14-bit point codes as set using bit 9 (see above).
  - Bit 21 should be set to 1 for use in Japanese networks, otherwise it should be set to 0.

  All other bits are reserved for future use and should be set to 0.

  Note: For correct ANSI operation, bits 8, 9, 10, 11, and 18 must be set to 1. This gives a typical <options> field value of 0x00040f00 for ANSI configurations.

7.2.2 MTP_LINKSET - Configure a Linkset

Synopsis

Configuration of a linkset to an adjacent signaling point.

Syntax

```
MTP_LINKSET <linkset_id> <adjacent_spc> <num_links> <flags> <local_spc> <ssf>
```

Example

```
MTP_LINKSET 0 321 2 0x0000 456 0x8
```

Parameters

The `MTP_LINKSET` command includes the following parameters:

- **<linkset_id>**

  The logical identity of the linkset, in the range of 0 to one less than the number of linksets supported. The linkset_id is used in other commands for reference.

- **<adjacent_spc>**

  The point code of the adjacent signaling point.

- **<num_links>**

  The number of links to be allocated to the linkset.

- **<flags>**

  This field is a 16-bit field containing run-time configuration options for the link set as follows:

  - Bit 3 is set to 1 to enable MTP restart procedures for this link set.
  - All other bits are reserved for future use and should be set to zero.

- **<local_spc>**

  The point code of the signaling point itself.

- **<ssf>**

  The value to be used in the Sub-Service Field (SSF) of all MTP3 messages and checked for by the discrimination function in all received messages. This is a 4-bit value. For ANSI operation, each of the two least significant bits should be set to 1.

  Note: For correct operation, the adjacent point code must also appear in an **MTP_ROUTE** declaration.

7.2.3 MTP_LINK - Configure a Link

Synopsis

Configuration of a signaling link.

Syntax

```
MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <board_id>
                                  <blink> <stream> <timeslot> <flags>
```

Example

```
MTP_LINK 0 0 2 2 0 1 1 16 0x0006
```

Parameters

The `MTP_LINK` command includes the following parameters:

- **<link_id>**

  The unique logical identity of the link. It must be in the range of 0 to one less than the total number of signaling links supported.

- **<linkset_id>**

  The logical identity of the linkset to which the link belongs. The linkset must already have been configured using the `MTP_LINKSET` command.

- **<link_ref>**

  The logical identity within the linkset of the signaling link. It should be in the range of 0 to one less than the number of links in the linkset.

- **<slc>**

  The signaling link code for the signaling link. This must be unique within the linkset and is typically the same as <link_ref>. The valid range is 0 to 15.

- **<board_id>**

  The board ID of the signaling processor allocated for this signaling link.

- **<blink>**

  This parameter indicates the channel of the link within the board. Each link can be used only once per board. The parameter must be between 0 and one less than the number of links supported.

- **<stream>**

  When the <timeslot> parameter is set to a non-zero value, the <stream> parameter is the logical identity of the T1/E1 Line Interface Unit (LIU) (liu_id) containing the signaling link. It should be in the range of 0 to one less than the number of LIUs. In these cases, the timeslot should be the signaling processor's signaling link in the range of 0 to 31.

- **<timeslot>**

  The timeslot used for signaling in the range of 0 to 31. The valid ranges are:

  For an E1 interface: 1 to 31.

  For a T1 interface: 1 to 24.

- **<flags>**

  A 16-bit value containing additional run-time options. The bit significance is as follows:

  - Bit 0 is set to 1 to force the use of the emergency proving period during link alignment or 0 to use the appropriate proving period according to the MTP3 recommendations.
  - Bit 1 is set to 1 to cause a signaling link test (in accordance with ITU-T Q.707 / ANSI T1.111.7) to be carried out before a link is put into service, or 0 if a test is not required.

- Bit 2 is set to 1 to cause a signaling link test (in accordance with ITU-T Q.707 / ANSI T1.111.7) to be carried out every 30 seconds. This bit is ignored unless bit 1 is set to 1.
- Bit 8 is used to select the MTP2 error correction mode. It is set to 1 to select PCR (Preventive Cyclic Retransmission) operation or 0 for the Basic Method of Error Correction.
- Bits 10 and 11 together select the appropriate operating bit rate for the link. The table below specifies the appropriate values for 48, 56, or 64 kbps.

Bit 11	**Bit 10**	**Date Rate (pattern)**
0	0	64 kbps (0xFF)
0	1	48 kbps (0x7E)
1	0	56 kbps (0xFE)
1	1	56 kbps (0xFD)

- Bit 15 is set to 1 to disable the link. This bit should be set to 0 to enable normal link operation.

The other bits are reserved for future use and should be set to 0.

7.2.4 MTP_ROUTE - Configure a Route

Synopsis

Configuration of a route for use with one or more user parts.

Syntax

```
MTP_ROUTE <dpc> <norm_ls> <user_part_mask> <flags> <second_ls>
```

Example

```
MTP_ROUTE 567 0 0x0020 0x0000 0
```

Parameters

The `MTP_ROUTE` command includes the following parameters:

- **<dpc>**

  The point code of the remote signaling point for which this command is configuring routing data. It may be either an adjacent point code or a point code accessible via an adjacent Signaling Transfer Point (STP).

- **<norm_ls>**

  The linkset_id of the normal linkset used to reach the specified destination. This parameter must be a linkset_id that has already been configured using the `MTP_LINKSET` command. The normal linkset may be any of the following:

  - The only linkset used to reach the destination.
  - The preferred linkset used to reach the destination.
  - One of a pair of links sets forming a combined linkset.

  In the latter two cases, a second linkset, <second_ls>, must also be specified.

  Within a linkset, messages are automatically load shared across links using the Signaling Link Selection (SLS) field in the message.

- **<second_ls>**

  The linkset_id of an optional second linkset used to reach the specified destination. This may be either of the following options:

  - The secondary linkset used to reach the destination only on failure of the preferred linkset.
  - One of a pair of link sets forming a combined linkset over which load sharing takes place. In this case, bit 1 must also be set in the **<flags>** parameter of the command.

  When a second linkset is specified, the user must also set bit 0 in the **<flags>** field of this command.

- **<user_part_mask>**

  This is a 16-bit field used to identify the user parts that are supported over this route. The bits are labelled 0 to 15. For each user part supported, the bit corresponding to the Service Indicator for that user part should be set. For example, to support just ISUP messages, the ISUP Service Indicator is 5. Bit 5 should be set and therefore a value of 0x0020 is appropriate.

- **<flags>**

  A 16-bit field containing run-time configuration options for the route as follows:

  - Bit 0 is set to 1 to indicate that a second linkset is specified within the command. If set to 0, the <second_ls> parameter is ignored.
  - Bit 1 is used to determine whether or not to load share messages across the two linksets. It is only used when two linksets are specified for the route. When set, the MTP3 module load shares messages for the destination equally across each of the two specified linksets. Otherwise, the MTP3 module considers the normal linkset to be the preferred linkset and only uses the second linkset in the event of failure of the normal linkset. The bit should be set to 1 to enable load sharing across the two linksets or 0 to disable load sharing and use preferred and secondary linksets.

  All other bits are reserved for future use and must be set to 0.

7.3 ISUP Configuration Commands

The ISUP configuration commands are:

ISUP_CONFIG - Configure ISUP below

ISUP_CFG_CCTGRP - Configure an ISUP Circuit Group on page 53

ISUP_TIMER - Configure ISUP Timers on page 55

7.3.1 ISUP_CONFIG - Configure ISUP

Synopsis

The global configuration parameters for the ISUP module.

Syntax

```
ISUP_CONFIG <res1> <res2> <user_id> <options> <num_grps> <num_ccts> [<partner_id>]
```

Example

```
ISUP_CONFIG 0 0 0x2d 0x0435 4 128
```

Parameters

The `ISUP_CONFIG` command includes the following parameters:

- **<res1> , <res2>**

  Reserved for backwards compatibility. These fields should be set to 0.

- **<user_id>**

  The module ID of the application running on the host that uses the ISUP module.

- **<options>**

  A 16-bit value that contains global run-time options for the operation of the ISUP module. The meaning of each bit is as defined for the options parameter in the ISUP Configure Request message as detailed in the Dialogic® SS7 Protocols ISUP Programmer's Manual.

- **<num_grps>**

  The maximum number of ISUP circuit groups that the user intends to use. This must not exceed the maximum number of circuit groups supported, otherwise module configuration fails. Typically, this parameter should be set to the maximum number of circuit groups supported.

- **<num_ccts>**

  The maximum number of ISUP circuits that the user intends to use. This must not exceed the maximum number of circuits supported, otherwise module configuration fails. Typically, this parameter is set to:

  - 32 times the number of groups for E1 operation
  - 24 times the number of circuit groups for T1 operation

  Note: The valid range for the circuit identifier (cid) is from 0 to one less than the maximum cid value.

- **<partner_id>**

  Optional parameter for use when operating in dual resilient configuration. This parameter is the module_id of the partner ISUP module (equivalent to the module_id field in the ISUP Configure Request message as documented in the Dialogic® SS7 Protocols ISUP Programmer's Manual).

7.3.2 ISUP_CFG_CCTGRP - Configure an ISUP Circuit Group

Synopsis

The configuration parameters for a group of ISUP circuits. Typically, a group is all the circuits in a single E1 or T1 interface.

Syntax

```
ISUP_CFG_CCTGRP <gid> <dpc> <base_cic> <base_cid> <cic_mask> <options> <user_inst>
                                          <user_id> <opc> <ssf> <variant> <options2>
```

Example

```
ISUP_CFG_CCTGRP 0 3 1 1 0x7fff7fff 0x00000003 0 0x2d 2 0x8 4 0x00000000
```

Parameters

The `ISUP_CFG_CCTGRP` command includes the following parameters:

- **<gid>**

  The group ID of the circuit group in the range of 0 to one less than the number of groups supported.

- **<dpc>**

  The destination point code for all circuits in the circuit group.

- **<base_cic>**

  The Circuit Identification Code (CIC) that is allocated to the first circuit in the circuit group.

- **<base_cid>**

  The logical ID for the first circuit in the circuit group. It must lie in the range of 0 to one less than the number of circuits supported.

- **<cic_mask>**

  A 32-bit mask with bits set to indicate which circuits are to be allocated to the circuit group. Bit 0 must always be set since it represents the <base_cic> / <base_cid>. Subsequent bits represent the subsequent circuits. ANSI circuit groups are not permitted to contain more than 24 circuits.

- **<options>**

  A 32-bit value containing run-time options for the ISUP circuit group. Refer to the Configure Circuit Group Request section of the Dialogic® SS7 Protocols ISUP Programmer's Manual. Bits 0 to 15 are equivalent to the options field and bits 16 to 31 represent the **ext_options** field as detailed in the ISUP Programmer's Manual.

- **<user_inst>**

  The instance number of the user application. Typically, only a single user application exists so this field should be set to 0.

- **<user_id>**

  The module ID of the user application.

- **<opc>**

  Originating Point Code. The local point code for all circuits in the group.

- **<ssf>**

  The value to be used in the Sub-Service Field (SSF) of all ISUP messages for this circuit group.

- **<variant>**

  The protocol variant for this circuit group. Refer to the Dialogic® SS7 Protocols ISUP Programmer's Manual for details.

- **<options2>**

  A 32-bit value containing additional run-time options for the ISUP circuit group. Refer to the Configure Circuit Group Request section of the Dialogic® SS7 Protocols ISUP Programmer's Manual. Bits 0 to 31 are equivalent to the ext_1_options parameter as detailed in the ISUP Programmer's Manual.

7.3.3 ISUP_TIMER - Configure ISUP Timers

Synopsis

The `ISUP_TIMER` command provides the ability to configure the ISUP protocol timers from the config.txt file.

Syntax

```
ISUP_TIMER <reserved> <timer_id> <value>
```

Example

```
ISUP_TIMER 0 t4 550
```

Parameters

The `ISUP_TIMER` command includes the following parameters:

- **<reserved>**

  Must be set to 0. Reserved for future use.

- **< timer_id >**

  The text identifier for the timer to be configured. The supported set of timer mnemonics is shown in the table below.

- **< value >**

  The timer value in seconds, except T29 and T30 that are in multiples of tenths of a second (100 ms). Any timers not explicitly set are set to their default values, as shown in the table below.

Timer Mnemonic	**Default Value (Seconds)**	**Timer Mnemonic**	**Default Value (Seconds)**	**Timer Mnemonic**	**Default Value (Seconds)**
T1	10	T15	60	T28	10
T2	180	T16	10	T29	0,5
T3	180	T17	60	T30	8
T4	300	T18	10	T33	14
T5	60	T19	60	T34	3
T6	180	T20	10	T35	20
T7	25	T21	60	T36	13
T8	13	T22	10	T38	150
T9	45	T23	60	T39	10
T10	5	T24	2	T103	20
T12	10	T25	5	T104	3
T13	60	T26	120		
T14	10	T27	240		

Chapter 8: Host Utilities and Host Protocols

This chapter describes the following host utilities and protocols that can be used with the Dialogic® DSI SS7 Software for Dialogic® Diva® Media Boards:

- s7_log
- s7_play
- gctload
- tim (tim_lnx)
- tick (tick_lnx)
- s7_mgt
- ssdd
- mtpsl
- actlinks
- mtp_lnx6
- isp_lnx6

When installing the DSI SS7 Software for Diva Media Boards, these utilities and protocols are copied into the install directory /usr/lib/opendiva/dlgss7.

8.1 s7_log

Description

The s7_log utility is a console application program that receives messages and displays them as text on the host console. Maintenance and status events are interpreted as text; other messages are typically displayed in hexadecimal format. The s7_log utility can optionally print the date and time of when a message is received by the utility.

Syntax

```
s7_log [-m<module_id>] [-o<options>] [-f<filename>] [-t[t|d]]
```

Command Line Options

The s7_log utility supports the following command line options:

- **-m<module_id>**

  Specifies the unique module identifier assigned to s7_log for the inter-process communication (IPC) environment. Any message sent to this module ID is displayed by the s7_log utility as text on the console. The module ID may be entered in decimal or hexadecimal (prefixed by "0x") format. If the module ID is not specified, s7_log uses a module ID of "0xef". The module ID that is assigned to s7_log must have a corresponding LOCAL entry in the host's system.txt file and must not be in use by any other process on the host.

- **-o<options>**

  A 16-bit value that specifies the type of message reporting that occurs. If not specified, a value of "0xaf0d" is used. Each bit that is set to "1" enables reporting of a particular message group or parameter field as described in the following table:

Bit	Function
0	Enable text interpretation of all recognized messages
1	Display ALL received messages (including those interpreted as text) as hexadecimal
2	Decode and display Management trace messages
3	Decode and display Management Trace Event "time stamp" field
4	Decode message header **src** and **dst** fields as text if recognised
5	Not used. Must be set to 0
6	Not used. Must be set to 0
7	Not used. Must be set to 0
8	Display message type field
9	Display message **id** field
10	Display message **src** field
11	Display message **dst** field
12	Display message **rsp_req** field
13	Display message **status** field
14	Display message **err_info** field
15	Display message **parameter** field

- **-f<filename>**

  Optionally specifies a file to which all screen output is written. If the specified file does not exist, it is created. If the specified file already exists, it is overwritten. The data is stored in the file in ASCII format.

- **-t[t|d]**

  Specifies the format of timestamp values derived from the host clock. The timestamp information is printed after the "S7L:" label in the log. The format options are:

  `-tt` specifies short timestamp format, that is, the time only

  `-td` specifies full timestamp format, that is, the date and time

  Note: Since the timestamps related to this option are derived from the host clock, values can be affected by host loading.

- **-n<maximum number of files>**

  Sets the maximum number of log files to be generated. You can specify that the S7_log utility generates between 2 and 99 log files. If this option is not specified, up to 5 log files will be generated by default. When the maximum number of log files is reached, the oldest log file is discarded.

  Example:

  The following command line entry would cause a maximum of ten logging files named s7log.txt ... s7log.txt.9.

  ```
  s7_log -fs7log.txt -n10
  ```

- **-s<maximum log file size in kilobytes>**

  Sets the maximum log file size. You can specify a maximum log file size of between 1 kilobyte and 100,000 kilobytes. If this option is not specified, the maximum log file size is set to 1000 kilobytes by default.

  When the maximum log file size is reached, it is renamed to <filename>.txt.1 and a new log file is created. This procedure is repeated each time the log file reaches the specified maximum size.

  Example

  The following command line entry would cause log files (prefixed with s7log.txt) to be created with a maximum size of 1000 kilobytes.

  ```
  s7_log -fs7log.txt -s1000
  ```

Example

To run s7_log as module ID "0xef" and enable all tracing options, the command line is:

```
s7_log -m0xef -o0xff1f
```

8.2 s7_play

Description

The s7_play utility is a console application that reads commands from as ASCII text file and then executes the commands. Each command can specify either:

- a message to be sent to a destination process
- a delay to apply before the next command is executed

Syntax

```
s7_play -m<module_id> -f<filename>
```

Command Line Options

The s7_play utility supports the following command line options:

- **-m<module_id>**

  Specifies the unique module ID that is assigned to the s7_play utility for the inter process communication (IPC) environment. Any message that is sent to this module ID is displayed by the s7_log utility as text on the host console. The module ID may be entered in decimal or hexadecimal (prefixed by "0"') format. If the module ID is not specified, the s7_play utility uses a module ID of "0xef". The module ID assigned to the s7_play utility must have a corresponding LOCAL entry in the host's system.txt file and must not be used by any other process on the host.

- **-f<filename>**

  Specifies the text file that contains the commands to be executed by the s7_play utility.

  Example

  To run s7_play with module ID "0x3d" and accept commands from a file called cmd.txt, the command is:

  ```
  ./s7_play -m0x3d -fcmd.txt
  ```

Text File Format

Each line in the text file must begin with one of the command specifiers in the following table:

Character	Function
M	Send a message
D	Delay
W	Send a message and wait for a response
P	Pause and wait for a specified message type to be received
*	Ignore (comment line)

The **Delay function (D)** takes a single parameter specifying the delay in either milliseconds (m) or seconds (s). Some examples:

```
D-s0001 * Delay for 1 second
D-m0001 * Delay for 1 millisecond
```

Note: The delay value may be in the range of 0000 to FFFF.

The **Send Message function (M)** allows the fields of the message to be specified in the following format:

```
M-I<inst>-t<type>-i<id>-f<src>-d<dst>-r<rsp_req>-e<err_info>-s<status>-p<param>
```

The meaning of the various options is shown in the following table:

Field Identifier	Length (in Characters)	Message Field
l	2	Instance
t	4	type
i	4	id
f	2	src
d	2	dst
r	4	rsp_req
e	8	err_info
s	2	status
p	2 to 640 (variable)	param

Each field identifier is optional and causes the corresponding message field to be set to 0 if not present. All values are entered in hexadecimal format. For example:

```
M-tc701-i0000-f1d-d23-s00-p0000ffffffff
```

The following command file sends a reset circuit group message to the first ISUP group, waits for 5 seconds, then sends a reset group message for group 1.

```
*
* Example s7_play command file
*
M-tc701-i0000-f1d-d23-s00-p0000ffffffff
*
D-s0005
*
M-tc701-i0001-f1d-d23-s00-p0000ffffffff
```

The **Send and Wait Message Response function (W)** instructs the module to issue a message and then wait for a response to that message.

Note: Care must be taken to ensure that the destination for the response (as set in the -f field) is the same as the module ID for the s7_play module (as set in the command line); otherwise the response will not reach the s7_play.

The **Wait Message Response function (P)** causes the module to pause until it receives the specified message types.

Note: Care must be taken to ensure that the destination for the response (as set in the -f field) is the same as the module ID for the s7_play module (as set in the command line); otherwise the response will not reach the s7_play.

Typical Script

A typical script is as follows:

```
M-t7680-I0000-f1d-d20-rffff-s00-i0000-p2000001d000000
P-t3680
M-t7689-i0000-fef-d20-s00-p0002 0004 0006 0008 0010
M-t7680-I0000-f1d-d20-s00-i0000-p2000001d0000000000000000
W-t7680-i0000-fef-d20-s00-p200000cf70637337332e64633100
```

8.3 gctload

Description

The gctload utility is a task that initializes the host system environment and starts up all other processes (such as ssd), deriving the process and message queue configuration from a text file. For further details of the operation of the gctload utility, refer to the Dialogic® SS7 Protocols Software Environment Programmer's Manual. The gctload task derives its configuration from a text file, typically called system.txt.

The gctload task can be run on an active system to provide tracing information that indicates the system state (-t1, -t2 flags) and it can also be used to terminate an active system (-x flag).

Syntax

```
gctload [-c<filename> -m<message pool size> -Ci<congestion module id>
-Co<congestion onset threshold> -Ca<congestion abatement threshold>
-d -v -t1 -t2 -x]
```

Command Line Options

The gctload utility supports the following command line options:

- **-c<filename>**

  Specifies the system configuration file <filename>. If no default file name is defined, the name system.txt is assumed.

- **-m<message pool size>**

  Specifies the message pool size, that is the number of messages available on the host. If this option is not defined, the default message pool size is 200.

  Note: For Linux systems, the kernel.msgmnb value might require to be increased to ensure stable operation.

- **-Ci<congestion module id>**

  Specifies the congestion-handling module ID. If this ID is set to the module ID of ssdd (0x20), then ssdd stops reading messages from the board until the congestion abates.

- **-Co<congestion onset threshold>**

  Specifies the congestion (overload) onset threshold, that is, the percentage of the total number of available messages that must be allocated before the system starts congestion procedures. The default is 50% of the messages in the message pool defined by the `-m` option. Once this threshold is reached, the congestion-handling module specified by the `-Ci` option is notified and should take steps to reduce the system loading.

- **-Ca<congestion abatement threshold>**

  Specifies the congestion abatement threshold, that is, the percentage of the total number of messages that must be available before the system stops congestion procedures. The default is 10% of the messages in the message pool defined by the `-m` option. Once the message pool size drops back below this threshold, the congestion-handling module, as specified by the `-Ci` option, is notified and can return the system to normal loading levels.

- **-t1**

  Displays system trace information (short). See System Status (gctload -t1) for more information.

- **-t2**

  Displays system trace information (long). See Show All Currently Allocated API messages (gctload -t2) for more information.

- **-v**

  Displays version information.

- **-d**

  Enables diagnostic tracing.
- **-x**

  Terminates a running system. An active instance of the gctload module, together with any forked binaries, is terminated if a subsequent call of gctload binary is made with the `-x` parameter.

Example

To run gctload with the system.txt file as the configuration file, a congestion onset value of 70, a congestion abatement value of 30, and a message pool size of 2000, the command is:

```
./gctload -csystem.txt -Co70 -Ca30 -m2000
```

8.3.1 System Status (gctload -t1)

For diagnostic purposes, it is possible to determine message queue statistics using the gctload utility with an additional command line option. When a host is running (having already started gctload), run gctload a second time with either the `-t1` or `-t2` option to display message statistics to the console. The `-t1` option causes gctload to print the current system statistics.

For example, the command:

```
./gctload -t1
```

generates output similar to the following:

```
GCTLOAD System status:
MSGs in system: 2000
MSGs allocated: 1
MSGs free: 1999
Maximum MSGs allocated: 155
Out of MSG count: 0
Internal system error: 0
Congestion module Id: 0x20
Congestion onset: 1000
Congestion abate: 200
Congestion status: 0
Congestion count: 0
GCTLIB library: V1.19
```

A rising number of allocated messages indicates that there is an issue, e.g., messages being sent to a non-existent queue or no process in the system is reading from the associated destination queue. The behavior of the system after it has run out of messages may be unstable and in these conditions, the gctload environment should be restarted. The contents of the currently allocated messages may be shown using the `-t2` option, see Show All Currently Allocated API messages (gctload -t2) below.

8.3.2 Show All Currently Allocated API messages (gctload -t2)

Note: The `gctload` command with the `-t2` option should not be used on live systems, since it locks the system until all messages have been printed out, an operation that can take a significant amount of time. The `-t2` option is intended for use during fault finding on a system that has not been configured correctly.

Issuing the `gctload` command with the `-t2` option generates a printout of all the currently allocated messages to the console. Messages are displayed in hexadecimal format as follows:

```
M t<type> i<id> f<src> d<dst> s<status> e<err_info> p<param>
```

where each field contains the value of the corresponding message field in hexadecimal format.

For example, the following command:

```
./gctload -t2
```

generates output similar to the following:

```
M-t0f83-i0000-fb0-def-s02
M-t0f83-i0000-fb0-def-s01
M-t0f0d-i0000-fdf-def-s19
M-t0201-i0000-f71-def-s03
M-t0201-i0000-f71-def-s02
M-t0201-i0000-f71-def-s03
M-t0201-i0000-f71-def-s02
```

The output above indicates that there are messages sent to a destination module ID "`0xef`" in the IPC system. Under normal operation, the message queues for destination tasks should either be empty or contain a small number of messages. If this is not the case, this may be due to one of the following reasons:

- No module has been configured to read messages for the listed destination queue.
- The destination task may have stopped reading from its message queue or may have stopped running.
- There may be a missing REDIRECT statement in the host's system.txt file to redirect messages from the listed destination to a running task.

8.4 tim (tim_lnx)

Description

The tim utility starts the tim process that receives periodic tick notification from tick processes and handles protocol timers for all other processes.

Syntax

```
tim_lnx [-v]
```

Command Line Options

The tim utility supports the following command line options:

- **-v**

  Displays version information.

Example

The tim process is typically only started by forking a process using gctload by including the following line in the system.txt file:

```
FORK_PROCESS ./tim_lnx
```

8.5 tick (tick_lnx)

Description

The tick utility starts the tick process that sends periodic tick notification to the tim process which in turn handles protocol timers.

Syntax

```
tick_lnx [-v]
```

Command Line Options

The tick utility supports the following command line options:

- **-v**

Displays version information.

Example

The tick process is typically only started by forking a process using gctload by including the following line in the system.txt file:

```
FORK_PROCESS ./tick_lnx
```

8.6 s7_mgt

Description

The s7_mgt utility performs one-time protocol configuration for all protocol modules, deriving the configuration parameters from a text file (config.txt by default). This process is optional. As an alternative, the user may select to perform protocol configuration by sending messages directly to the other modules in the system. For more information, see Protocol Configuration Using Discrete Messages on page 86.

Syntax

```
s7_mgt [-v -k<config_file> -m<module_id> -i<notify_id> -d]
```

Command Line Options

The s7_mgt utility supports the following command line options:

- **-v**

  Displays version information.
- **-k<config file>**

  Specifies the SS7 configuration file. The default is config.txt.
- **-m<module id>**

  Specifies the unique module ID that is assigned to s7_mgt for the Inter Process Communication (IPC) environment. The module ID may be entered in decimal or hexadecimal (prefixed by "0x") format. If the module ID is not specified, the utility uses a module ID of `0xcf`. The module ID assigned must have a corresponding LOCAL entry in the system.txt file and must not be in use by any other process on the host.
- **-i<notify module id>**

  The module to which an indication is sent when the configuration is complete.
- **-d**

  Enables diagnostic tracing.

Example

To run the s7_mgt utility as module ID `0xdf` with the file my_config.txt as its configuration file and notifying the module 0xef on completion, the command is:

```
./s7_mgt -m0xdf -kmy_config.txt -i0xef
```

8.7 ssdd

Description

The ssdd interfaces with the device driver for passing messages to and from the board, checks the status of the boards configured and verifies the required MTP2 licenses.

If ssdd is defined as the congestion-handling module for gctload then it will stop retrieving messages from the board until the congestion abates. Other congestion handling steps may be required depending on the system configuration and state.

Note: This process is often referred to in a generic manner as "ssd" although the name of the binary for use with Dialogic® Diva® Media Boards is in fact "ssdd".

Syntax

```
ssdd [-m<module_id>] [-d<trace_mask>] [-v] [-Lt] [-Lp<path>] [-t]
```

Command Line Options

The ssdd utility supports the following command line options:

- **-m<module id>**

  Defines the module ID of ssdd (default: 0x20).

- **-d<trace mask>**

  Enables diagnostic tracing and defines the initial trace mask of ssdd as hexadecimal number (default:0x0):

Bit	Meaning
0	Initialisation and state changes
1	Errors
2	Warnings
3	Internal status changes
4	Protocol wrapper
5	Internal verbose
6	Messages/primitves
7	-
8	CAPI - NCCI state machine
9	CAPI - PLCI state machine
10	CAPI - controller state machine
11	Memory allocation
12	Timer handling

  The debug information is written into the Dialogic® Diva® System Release standard tracing utilities, see the Dialogic® Diva® System Release LIN Software Reference Guide.

  Note: The trace mask used on active systems, should not define more than bit 0, 1, and 2 since it may harm the overall system performance significantly. The `-d` option is intended for use during fault finding on a system that has not been configured correctly.

  Note: The trace mask can be changed while ssdd is up and running via the management interface, see Management Interface of the SSDD Process on page 73.

- **-v**

  Displays version information and exits.

- **-Lt**

  License Test displays MTP2 license information and exits.

  Example of typical output

  ```
  Searching path: "." for a valid Vendor 1 license file token "MTP2DV4_LNX".
  License file is valid.
  SSDD: found license for 4 MTP2 links
  ```

- **-Lp<path>**

  Sets the path to the license file (default: ".").

- **-t**

  Starts ssdd in trial mode (no MTP2 license required). It will terminate after 10 hours.

  Example

  When the required license is not in the current directory but in/usr/lib/opendiva/license:

  ```
  ./ssdd -Lp/usr/lib/opendiva/license
  ```

8.8 mtpsl

Description

MTPSL is a utility to manually activate and deactivate single MTP2 signaling links

Note: The link activation is done automatically by Dialogic® SS7 Software for Dialogic® Diva® Interfaces

Syntax

```
mtpsl <cmd> <linkset id> <link ref>
```

Command Line Options

The mtpsl utility supports the following command line options:

- **<cmd>**

  `ACT` - activate specified link

  `DEACT` - deactivate specified link
- **<linkset id>**

  Defines the linkset ID the link to activate/deactivate belongs to.
- **<link ref>**

  Defines the link reference of the link ID the link to activate/deactivate belongs to.

  Example

  Activate link 0 of linkset 0:

  ```
  ./mtpsl ACT 0 0
  ```

8.9 actlinks

Description

actlinks is a shell script to manually activate a number of links in a number of linksets by calling mtpsl.

Note: The link activation is done automatically by Dialogic® SS7 Software for Dialogic® Diva® Interfaces

Syntax

```
actlinks <linksets> <links>
```

Command Line Options

The actlinks utility supports the following command line options:

- **<linksets>**

  Number of linksets to be activated.

- **<links>**

  Number of links in each linkset to be activated.

Example

Activate two linksets with 2 links each:

```
./actlinks 2 2
```

Note: The linkset ID and link ref start always at 0. The above example will generate the following calls of mtpsl:

```
./mtpsl ACT 0 0
./mtpsl ACT 0 1
./mtpsl ACT 1 0
./mtpsl ACT 1 1
```

8.10 mtp_lnx6

Description

The mtp_lnx6 is the MTP3 host protocol module.

Syntax

```
./mtp_lnx6 [-m<module_id>][-v] [-Lt] [-Lp<path>] [-t]
```

Command line options

- **-m<module id>**

  Defines the module ID of mtp_lnx6 (default: 0x22)
- **-v**

  Shows version information and exits.
- **-Lt**

  License test displays MTP3 license information and exits.

  Example of typical output

  Searching path: "." for a valid license file token "MTP3_LNX".

  License file is valid.
- **-Lp<path>**

  Sets the path to the license file (default: ".").
- **-t**

  Starts mtp_lnx6 in trial mode (no MTP3 license is required). The trial mode will terminate after 10 hours.

8.11 isp_lnx6

Description

The isp_lnx6 is the ISUP host protocol module.

Syntax

```
./isp_lnx6 [-m<module_id>][-v] [-Lt] [-Lp<path>] [-t]
```

Command line options

- **-m<module id>**

  Defines the module ID of isp_lnx6 (default: 0x23).
- **-v**

  Shows version information and exits.
- **-Lt**

  License test displays ISUP license information and exits.

  Example of typical output

  Searching path: "." for a valid license file token "ISUP_LNX".

  License file is valid.
- **-Lp<path>**

  Sets the path to the license file (default: ".").
- **-t**

  Starts isp_lnx6 in trial mode (no ISUP license required). The trial mode will terminate after 10 hours.

Chapter 9: Management Interface of the SSDD Process

The Dialogic® Diva® System Release LIN software provides a common management utility to access configuration, status, statistics and debugging information while running. The ssdd process has implemented a management interface which can be accessed with the standard Diva System Release method, see Dialogic® Diva® System Release LIN Reference Guide.

This chapter describes the parameters of the ssdd management interface.

- Config Directory on page 74
- State Directory on page 77
- Statistics Directory on page 79
- License Directory on page 84
- Debug Directory on page 85

The management interface of the ssdd has the logical board number 1021; this number is fixed on all systems. Accessing the management interface with the option `-c 1021` registers with the management interface of the ssdd.

Reading the root directory of the ssdd management interface:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r""
```

will generate the following information:

```
--------hit-[MIF Version ................] = 0x117
--------asz-[Build ......................] = 108-76 Sep  5 2008 12:59:13
--------asz-[StartUpTime ................] = Fri Sep  5 17:49:10 2008
--------dir-[Config .....................]
--------dir-[State ......................]
--------dir-[Statistics..................]
--------dir-[License ....................]
--------dir-[Debug ......................]
```

Values

- **MIF Version**

  Identifies the version of the management information base

- **Build**

  Displays available version information of the ssdd

- **StartUpTime**

  Displays system time at start of the ssdd

- **Config**

  Directory containing configuration information, see Config Directory on page 74

- **State**

  Directory containing the current state of the ssdd, see State Directory on page 77

- **Statistics**

  Directory containing available statistics of the ssdd, see Statistics Directory on page 79

- **License**

  Directory containing extracted license information, see License Directory on page 84

- **Debug**

  Directory containing and controlling the current debug state, see Debug Directory on page 85

9.1 Config Directory

Description

Via the config directory of the ssdd, you can retrieve configuration information received by the ssdd.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Config"
```

Output

```
--------dir-[Boards .....................]
--------dir-[LIUs .......................]
--------dir-[Links ......................]
```

Values

- **Boards** below

  Directory containig board configuration information
- **LIUs on page 75**

  Directory containing information about the line interface unit (LIU) configuration
- **Links on page 76**

  Directory containing information about the signaling link configuration

9.1.1 Boards

Description

The boards directory lists configuration information about all configured boards.

When calling:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Config\Boards"
```

the boards that have been reset by the **SSD_MSG_RST_BOARD** or **DIVA_BOARD** command (see Physical Interface Configuration Command on page 44) are enumerated.

```
--------uit-[NumberBoards ...............] = N
--------dir-[Board 1 ....................]
...
--------dir-[Board N ....................]
```

Where N is the number of boards reset.

Reading the configuration of a single board is described below.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Config\Boards\Board X"
```

where X is the Link management number.

Output (example)

```
--------int-[board_id ...................] = 0
--------asz-[serial_number ..............] = 1234
--------int-[board_type .................] = 81
--------int-[board_typeS ................] = Dialogic Diva V-4PRI/E1/T1-120 PCI v1
--------int-[number_of_LIUs .............] = 4
```

Values

- **board_id**

  board_id as defined by the **SSD_BOARD_RESET** or **DIVA_BOARD** command.
- **serial_number**

  serial_number as defined by the **SSD_BOARD_RESET** or **DIVA_BOARD** command.
- **board_type**

  Dialogic® Diva® Media Board **board_type** as defined by **SSD_BOARD_RESET** or **DIVA_BOARD** command.
- **board_typeS**

  Diva Media Board **board_type** as ASCII string.
- **number_of_LIUs**

  Number of line interface unit(s) (LIU) of the board.

9.1.2 LIUs

Description

The LIUs directory lists configuration information about the configured line interface units (LIU).

When calling:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Config\LIUs"
```

the LIUs configured for signaling are enumerated:

```
--------uit-[NumberLIUs ...............] = N
--------dir-[LIU 1 ....................]
...
--------dir-[LIU N ....................]
```

Where `N` is the number of LIUs.

Reading the configuration of a single LIU is described below.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Config\LIUs\LIU X"
```

where `X` is the LIU management number

Output (example)

```
--------int-[board_id ...................] = 0
--------asz-[serial_number ..............] = 1234
--------int-[liu_id .....................] = 0
--------hit-[board_mgmt_id ..............] = 0x8e
--------hit-[mtp2_module_id .............] = 0xff
```

Values

- **board_id**

  Board ID of the Diva Media Board.
- **serial_number**

  Serial number of the Diva Media Board.
- **liu_id**

  LIU ID of this LIU.
- **board_mgmt_id**

  Module ID of board management module.

- **mtp2_module_id**

  Module ID of MTP2 module.

9.1.3 Links

Description

The links directory lists configuration information about all MTP2 links.

When calling:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Config\Links"
```

all configured links which were received in a **MGT_L1_CONFIG** request message are enumerated:

```
--------uit-[NumberOfLinks ...........] = N
--------dir-[Link 1 ..................]
...
--------dir-[Link N ..................]
```

Where `N` is the number of links.

Reading the configuration of a single link is described below.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Config\Links\Link X"
```

where `X` is the LIU management number

```
Output (example)
--------int-[board_id ...................] = 0
--------asz-[serial_number ..............] = 1234
--------int-[liu_id .....................] = 0
--------int-[llid .......................] = 0
--------bit-[timeslotmask ...............] = 0000:0000:0000:0000:1000:0000:0000:0000
--------bit-[channelBitMask .............] = 1111:1111
--------hit-[mtp2_module_id .............] = 0x71
--------asz-[mgt_msg_L1_config ..........] =
      I0000-t7f17-i0000-fcf-d8e-r8000-c00-s00-e00000000-p(040)000000000000000000000000
```

Values

- **board_id**

  Board ID of the Diva Media Board.

- **serial_number**

  Serial number of the Diva Media Board.

- **liu_id**

  LIU ID of the Diva Media Board.

- **llid**

  Logical link ID (llid) of the link.

- **timeslotmask**

  Shows channel used for this link (example shows timeslot 16).

- **channelBitMask**

  Shows bits utilized for this link (example shows all bits - 64k).

- **mtp2_module_id**

  Module ID of MTP2 module.

- **mgt_msg_L1_config**

  Displays **MGT_L1_CONFIG** request message.

9.2 State Directory

Description

The state directory gives hints when the stack is not working properly (e.g., CAPI link down).

Via the **State** directory, state information of the ssdd can be retrieved.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"State"
```

Output

```
--------dir-[Links ......................]
```

Value

Links below

Directory containing information about the signaling link states

9.2.1 Links

Description

The links directory lists state information about all MTP2 links.

When calling:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"State\Links"
```

All configured links, the ssdd is maintaining, are enumerated:

```
--------uit-[NumberOfLinks ...............] = N
--------dir-[Link 1 ....................]
...
--------dir-[Link N ....................]
```

Where N is the number of links.

Reading the state of a single link is described below.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"State\Links\Link X"
```

where X is the link management number

Output (example)

```
--------int-[board_id ...................] = 0
--------int-[liu_id .....................] = 0
--------int-[llid .......................] = 0
--------hit-[LastPrimitiveRX ............] = 0x8f01
--------hit-[LastPrimitiveTX ............] = 0xcf00
--------int-[MaxDataB3Req ...............] = 2
--------int-[CurrentDataB3Req ...........] = 0
--------int-[maxMsgCapiReq ..............] = 0
--------int-[CurrentMsgCapiReq ..........] = 0
--------int-[CapiState ..................] = 2
--------asz-[CapiStateS .................] = CONNECTED
--------asz-[ChangeTime .................] = Mon Sep  8 09:27:03
```

Values

- **board_id**

  Board ID of the Diva Media Board.
- **liu_id**

  LIU ID of the Diva Media Board.
- **llid**

  Logical link ID (llid) of the link.
- **LastPrimitiveRX**

  Shows the message type of the last primitive received from the board.

  In this example: 0x8f01 - **API_MSG_RX_IND**
- **LastPrimitiveRX**

  Shows the message type of the last primitive sent to the board.

  In this example: 0xcf00 - **API_MSG_TX_REQ**
- **MaxDataB3Req**

  Maximum number of CAPI requests of the link queued towards the board.
- **CurrentDataB3Req**

  Current number of CAPI requests of the link queued towards the board.
- **MaxMsgCapiReq**

  Maximum number of Dialogic® DSI primitives in CAPI request of the link queued towards the board.
- **CurrentDataB3Req**

  Current number of Dialogic® DSI primitives in CAPI request of the link queued towards the board.
- **CapiState**

  Current CAPI state.
- **CapiStateS**

  Current CAPI state as ASCII string.
- **ChangeTime**

  System time when entered the above state.

9.3 Statistics Directory

Description

The statistics directory gives further hints when the stack is not working properly (load balancing, …).

Via the statistics directory of the ssdd, information about the number of messages/octets received or transmitted is retrieved. The values can be reset.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Statistics"
```

Output

```
--------dir-[All ........................]
--------dir-[Boards .....................]
--------dir-[LIUs .......................]
--------dir-[Links ......................]
```

Values

- **All** below

  The **All** directory lists the numbers of all links.
- **Boards on page 80**

  The **Boards** directory lists the numbers of all links running on that board.
- **LIUs on page 81**

  The **LIUs** directory lists the numbers of all links running on that line interface unit (LIU).
- **Links on page 82**

  The **Links** directory lists the numbers of the corresponding links.

9.3.1 All

Description

The **All** directory lists the statistics numbers of all links maintained by ssdd.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Statistics\All"
```

Output (example)

```
--------int-[NumberPrimitivesRX .........] = 16
--------int-[NumberPrimitivesTX .........] = 28
--------int-[NumberOctetsRX .............] = 388
--------int-[NumberOctetsTX .............] = 388
--------fn -[Reset ......................]
```

Values

- **NumberPrimitivesRX**

  Total number of Dialogic® DSI primitives received from all link instances maintained by ssdd.
- **NumberPrimitivesTX**

  Total number of Dialogic® DSI primitives sent to all link instances maintained by ssdd.
- **NumberOctetsRX**

  Total number of octets in Dialogic® DSI primitives received from all link instances maintained by ssdd.

- **NumberOctetsTX**

  Total number of octets in Dialogic® DSI primitives sent to all link instances maintained by ssdd.

- **Reset**

  Function call to reset all statistics numbers of all link instances maintained by ssdd:

  ```
  /usr/lib/opendiva/divas/divactrl mantool -c 1021 -e"Statistics\All\Reset"
  ```

9.3.2 Boards

Description

The **Boards** directory lists the total number of all links maintained by the ssdd running on that board.

When calling:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Statistics\Boards"
```

All boards are enumerated:

```
--------uit-[NumberBoards ...............] = N
--------dir-[Board 1 ....................]
...
--------dir-[Board N ....................]
```

Where N is the number of boards.

Reading the statistics of a single board is described below.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Statistics\Boards\Board X"
```

where X is the board management number

Output (example)

```
--------int-[board_id ...................] = 0
--------int-[NumberPrimitivesRX .........] = 16
--------int-[NumberPrimitivesTX .........] = 28
--------int-[NumberOctetsRX .............] = 388
--------int-[NumberOctetsTX .............] = 388
--------fn -[Reset ......................]
```

Values

- **board_id**

  Board ID as received in the **SSD_BOARD_RESET** request message.

- **NumberPrimitivesRX**

  Total number of Dialogic® DSI primitives received from link instances assigned to the board and maintained by ssdd.

- **NumberPrimitivesTX**

  Total number of Dialogic® DSI primitives sent to all link instances assigned to the board and maintained by ssdd.

- **NumberOctetsRX**

  Total number of octets in Dialogic® DSI primitives received from all link instances assigned to the board and maintained by ssdd.

- **NumberOctetsTX**

  Total number of octets in Dialogic® DSI primitives sent to all link instances assigned to the board and maintained by ssdd.

- **Reset**

  Function call to reset all statistics numbers of all link instances assigned to the board and maintained by ssdd.

  ```
  /usr/lib/opendiva/divas/divactrl mantool -c 1021 -e"Statistic\Boards\Board X \Reset"
  ```

Note: The numbers of all link instances maintained by ssdd will be reset to 0.

9.3.3 LIUs

Description

The LIUs directory lists the total number of all links maintained by the ssdd running on that line interface unit (LIU).

When calling:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Statistics\LIUs"
```

The LIUs configured to be used for signaling are enumerated:

```
--------uit-[NumberLIUs ...............] = N
--------dir-[LIU 1 ....................]
...
--------dir-[LIU N ....................]
```

Where N is the number of LIUs.

Reading the statistics of a single LIU is described below.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Statistics\LIUs\LIU X"
```

where X is the LIU management number.

Output (example)

```
--------int-[board_id ...................] = 0
--------int-[liu_id .....................] = 0
--------int-[NumberPrimitivesRX .........] = 6
--------int-[NumberPrimitivesTX .........] = 12
--------int-[NumberOctetsRX .............] = 172
--------int-[NumberOctetsTX .............] = 172
--------fn -[Reset ......................]
```

Values

- **board_id**

  board_id, reference to the board

- **liu_id**

  LIU id of this LIU

- **NumberPrimitivesRX**

  Total number of Dialogic® DSI primitives received from link instances assigned to that LIU and maintained by ssdd.

- **NumberPrimitivesTX**

  Total number of Dialogic® DSI primitives sent to all link instances assigned to that LIU and maintained by ssdd.

- **NumberOctetsRX**

  Total number of octets in Dialogic® DSI primitives received from all link instances assigned to that LIU and maintained by ssdd.

- **NumberOctetsTX**

  Total number of octets in Dialogic® DSI primitives sent to all link instances assigned to that LIU and maintained by ssdd.

- **Reset**

  Function call to reset all statistics numbers of all link instances assigned to that LIU and maintained by ssdd.

  ```
  /usr/lib/opendiva/divas/divactrl mantool -c 1021 -e"Statistics\LIUs\LIU X \Reset"
  ```

9.3.4 Links

Description

The links directory lists the statistics numbers of each MTP2 link maintained by ssdd.

When calling:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Statistics\Links"
```

all configured links which were received in a **MGT_L1_CONFIG** request message are enumerated:

```
--------uit-[NumberOfLinks ...............] = N
--------dir-[Link 1 ....................]
...
--------dir-[Link N ....................]
```

Where N is the number of links.

Reading the statistics of a single link is described below.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Statistics\Links\Link X"
```

where X is the link management number.

Output (example)

```
--------int-[board_id ...................] = 0
--------int-[liu_id .....................] = 0
--------int-[llid .......................] = 0
--------int-[NumberPrimitivesRX .........] = 3
--------int-[NumberPrimitivesTX .........] = 6
--------int-[NumberOctetsRX .............] = 86
--------int-[NumberOctetsTX .............] = 86
--------fn -[Reset ......................]
```

Values

- **board_id**

  Board ID referencing to the board.

- **liu_id**

  LIU ID referencing to the LIU of the board.

- **llid**

  Logical link ID (llid) of the link.

- **NumberPrimitivesRX**

  Total number of Dialogic® DSI primitives received from this link instance.

- **NumberPrimitivesTX**

  Total number of Dialogic® DSI primitives sent to this link instance.

- **NumberOctetsRX**

  Total number of octets in Dialogic® DSI primitives received from this link instance.

- **NumberOctetsTX**

  Total number of octets in Dialogic® DSI primitives sent to this link instance.

- **Reset**

  Function call to reset the statistic numbers of this link instance.

  ```
  /usr/lib/opendiva/divas/divactrl mantool -c 1021 -e"Statistic\Links\Link X \Reset"
  ```

9.4 License Directory

Description

Via the license directory of the ssdd, the current license status can be obtained.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"License"
```

Output (example)

```
--------asz-[HostId .....................] = 00123456789A
--------uit-[NumberOfLinks ..............] = 4
--------uit-[NumberOfLinksInUse .........] = 4
--------asz-[LicensePath ................] = ./.
--------bol-[TestMode ...................] = FALSE
```

Values

- **HostId**

  Displays the host ID of the system.

- **NumberOfLinks**

  If a valid license is found, **NumberOfLinks** displays the number of licensed MTP2 links.

- **NumberOfLinksInUse**

  Number of links currently configured.

- **LicensePath**

  Path to license found.

- **TrialMode**

  Indicates if running in test mode (ssdd -t).

9.5 Debug Directory

Description

Via the debug directory, the debug information of the ssdd written into the Dialogic® Diva® System Release standard tracing utilities (see the Dialogic® Diva® System Release LIN Software Reference Guide) can be controlled.

Syntax

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -r"Debug"
```

Output (example)

```
-w------hit-[debug_mask .................] = 0x47
--------fn -[DebugStop ..................]
```

Values

- **debug_mask**

  Current debug mask.

Bit	Meaning
0	Initialisation and state changes
1	Errors
2	Warnings
3	Internal status changes
4	Protocol wrapper
5	Internal verbose
6	Messages/primitves
7	-
8	CAPI - NCCI state machine
9	CAPI - PLCI state machine
10	CAPI - controller state machine
11	Memory allocation
12	Timer handling

Note: The trace mask used on live systems, should not define more than bit 0, 1, and 2 since it may harm the overall system performance significantly. The -d option is intended for use during fault finding on a system that has not been configured correctly.

The value of the debug_mask can be changed during runtime:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -w"Debug\debug_mask=0x7"
```

will change the current debug mask to 0x7 (initialisation and state changes AND errors AND warnings)

- **DebugStop/DebugStart**

  Enables or disables debugging.

  When debugging is enabled, **DebugStop** will be displayed. By calling:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -e"Debug\DebugStop"
```

debugging may be disabled.

When debugging is disabled, **DebugStart** will be displayed. By calling:

```
/usr/lib/opendiva/divas/divactrl mantool -c 1021 -e"Debug\DebugStart"
```

debugging may be enabled.

Chapter 10: Protocol Configuration Using Discrete Messages

This section provides guidelines for protocol configuration using individual messages.

As an alternative to using the s7_mgt protocol configuration utility (see Protocol Configuration Using the s7_mgt Utility on page 22), it is possible to perform protocol configuration by building and sending messages directly to the ssdd. This approach means that it is necessary to write some application code to handle configuration, but enables the application, if required, to reconfigure the board/ssdd without restarting the application.

Communication with the board is achieved by sending and receiving messages via the ssdd. The configuration sequence is described below. The application should allocate a message structure using the **getm( )** library function and send it to the board using the **GCT_send( )** library function. The application should periodically call the **GCT_receive( )** or **GCT_grab( )** library functions to receive messages from the board. The **GCT_receive( )** function blocks until a message is available, while the **GCT_grab( )** function returns immediately. Once the application has finished processing the received message, it should release the message structure back to the system by calling the **relm( )** library function. The library functions are described in the Dialogic® SS7 Protocols Software Environment Programmer's Manual.

To configure the board using individual messages, the following sequence should be used.

Note: The format of all the messages is described in Message Reference on page 27.

1. Build and send an **SSD Reset Request** (see SSD_MSG_RESET - SSD Reset Request on page 28) to the SSD module. This message contains the parameters required to initialize the SSD module.
2. Wait for the response and check if the status is 0 (success).
3. Then build and send a **Board Reset Request** (see SSD_MSG_RST_BOARD - Board Reset Request on page 29) for each Dialogic® Diva® Media Board in the system with an MTP2 link configured. This message contains type and serial number of the board. It causes the board to be detected and reset. For each board, the application should wait until a **Board Status Indication** (see SSD_MSG_STATE_IND - Board Status Indication on page 38) is received and inspect the status field to determine if the reset operation was successful. On failure, the user should check carefully the parameters and try again. On success, the user should continue with the next step.
4. To stay consistent with the configuration of the Dialogic® DSI SS7 board families SS7HD and SS7SPCI, a **MGT_MSG_CONFIG0** can be transmitted. If implementing this message, the board type parameter has to be set to 3, to enable the link configuration via MGT_MSG_L1_CONFIG - Layer 1 Configuration Request on page 31 messages to configure each link. All other parameters should be set to 0.
5. Build and send a **Layer 1 Configuration Request** (see MGT_MSG_L1_CONFIG - Layer 1 Configuration Request on page 31) to set up the physical configuration parameters for the link. This message should be sent to the on-board management module. Wait for the confirmation message and check the status.

For each link in the system:

1. Build and send an **MTP2 Link Configuration Request** (see SS7_MSG_CONFIG - MTP2 Link Configuration Request on page 34) to set up the MTP2 configuration parameters. See the Dialogic® SS7 Protocols MTP2 Programmer’s Manual for the message definition. Wait for the confirmation message and check the status.
2. Build and send an MTP3 Module Reset Message (**MTP_MSG_RESET**) to reset the MTP3 module. See the Dialogic® SS7 Protocols MTP3 Programmer’s Manual for the message definition. Wait for the confirmation message and check the status.
3. Build and send an MTP3 Module Configuration Request (**MTP_MSG_CONFIG**) to set up configuration parameters that relate to the MTP3 environment (number of link sets and links to support, module_ids for user part modules etc.). See the Dialogic® SS7 Protocols MTP3 Programmer’s Manual for the message definition. Wait for the confirmation message and check the status.

For each link in the link set, perform the following:

1. Build and send an **MTP3 Signaling Link Configuration Request** (**MTP_MSG_CNF_LINK**) to set up configuration parameters for the individual link. See the Dialogic® SS7 Protocols MTP3 Programmer's Manual for the message definition. Wait for the confirmation message and check the status.

For each link set in the system, perform the following:

1. Build and send an **MTP3 Link Set Configuration Request** (**MTP_MSG_CNF_LINKSET**) to set up configuration parameters for the individual link set (for example, local and adjacent point codes and the number of links in the link set). See the Dialogic® SS7 Protocols MTP3 Programmer's Manual for the message definition. Wait for the confirmation message and check the status.
2. For each destination that needs to be accessed (including all adjacent signaling points), build and send an **MTP Route Configuration Request** (**MTP_MSG_CNF_ROUTE**) to set up configuration parameters for the route. See the Dialogic® SS7 Protocols MTP3 Programmer's Manual for the message definition. Wait for the confirmation message and check the status.

Proceed now with the User Part configuration procedure. Once this is complete, issue an **MTP Link Activation Request** (**MTP_MSG_ACT_SL**) for each link in the system as required to bring the link into service.

Further links, link sets and routes may be dynamically added at run time using the same message sequences.

Glossary

config.txt A text file used for protocol configuration.

DPC Destination Point Code. Identifies the address (point code) of the SS7 network node to which a Message Signal Unit (MSU) should be directed.

DSI Distributed Signaling Interface

gctload A program that handles the initialization sequence and creates inter-process communication.

INAP Intelligent Network Application Part. An SS7 stack layer that defines the messages and protocols used to communicate between applications (deployed as subsystems) in SS7 nodes. INAP uses the Transaction Capabilities Part (TCAP). See TCAP below.

IS41 An ANSI signaling standard used in mobile networks.

ISUP ISDN User Part. A SS7 stack layer that defines the messages and protocols used in the establishment and tear down of voice and data calls over the public switched network, and to manage the trunk network on which they rely.

Link A physical and logical connection between two signaling points.

Linkset One or more signaling links that are connected to adjacent signaling points.

LIU Line Interface Unit.

MAP Mobile Application Part (MAP). An SS7 stack layer supporting messages sent between mobile switches and databases to support user authentication, equipment identification, and roaming.

MSU Message Signal Unit. A data unit that carries signaling information for call control, transaction processing, network management and maintenance. Typically, the MSU is carried in the Signaling Information Field (SIF) of SS7 messages.

MTP Message Transfer Part. Layers 1 to 3 of the SS7 protocol stack broadly equivalent to the Physical, Data Link and Network layers in the OSI protocol stack. See also MTP1, MTP2, and MTP3 below.

MTP1 Message Transfer Part Level 1. An SS7 stack layer that defines the physical and electrical characteristics of the signaling links of the SS7 network. Signaling links use DS0 channels and carry raw signaling data at a rate of 48, 56, or 64 kbps.

MTP2 Message Transfer Part Level 2. An SS7 stack layer that provides link-layer functionality. Ensures that two end points of a signaling link can reliably exchange signaling messages. It provides error checking, flow control, and sequence checking.

MTP3 Message Transfer Part Level 3. An SS7 stack layer that provides network-layer functionality. Ensures that messages can be delivered between signaling points across the SS7 network regardless of whether the signaling points are directly connected. It provides node addressing, routing, alternate routing and congestion control.

mtpsl An example utility that can also be used to activate and deactivate signaling links.

route An MTP3 concept that determines how signaling is distributed over linksets. A route consists of a destination point code and the linkset ID of one or two linksets over which traffic to the destination node should be routed. When two linksets are provided, the user can choose to load share traffic or treat the linksets as primary and secondary.

s7_log A utility that enables messages received from the protocol stack to be logged in a text file. Typically used for diagnostic purposes.

s7_mgt A utility that performs one time protocol configuration of all protocol modules using configuration parameters from the config.txt file.

s7_play A utility that can be used to generate messages from a text file and send them to the system. Typically used for diagnostic purposes.

SCCP Signal Connection Control Part. An SS7 stack layer that allows a software application at a specific node in an SS7 network to be addressed.

SLS Signaling Link Selection field. A field in the MTP3 routing label used to determine the selection of an outgoing link for messages being routed to another point code.

SS7 Signaling System Number 7

SS7 Protocol Stack A set of software modules that implement the various layers of the SS7 protocol stack.

ssdd Process to interface with the device driver of Dialogic® Diva® Media Boards for passing messages to and from the board(s) and for configuring the Dialogic® DSI SS7 part running on the board(s).

STP Signaling Transfer Point.

system.txt A text file used for system configuration.

TCAP Transaction Capabilities Application Part. An SS7 stack layer that enables the deployment of intelligent network and mobile services by supporting non-circuit related information exchange between signaling points using the SCCP connectionless service.

TUP Telephone User Part. An SS7 stack layer that is the predecessor to ISUP (Integrated Services User Part). TUP was employed for call control purposes within and between national networks, both wireline and wireless. ISUP adds support for data, advanced ISDN, and IN (Intelligent Networks). See also ISUP.

About Customer Services

Dialogic provides various options and arrangements for obtaining technical support for your Dialogic® product. We recommend that you visit the Dialogic Services & Support site, as it includes detailed information about a variety of topics. In the unusual case that the information on the Services & Support site does not adequately addresses your support issue, you can contact Dialogic Customer Support.

For more information see:

- Dialogic® Services & Support Web Site below
- Dialogic Customer Support below

Dialogic® Services & Support Web Site

If your supplier is unable to help you to address your issue, you can visit the Dialogic Services & Support web site. There, you can get access to:

- detailed information about the Dialogic® Pro™ Services (1 or 5 year 24/7 service contracts) at http://www.dialogic.com/support/DialogicPro/
- a help web section for Dialogic® products at http://www.dialogic.com/support/helpweb
- a download section, to install the current version of your software at http://www.dialogic.com/support/downind.asp
- a training section, with information about webinars as well as online and onsite trainings at http://www.dialogic.com/training/default.htm
- a manuals section, that includes currently available documentation, at http://www.dialogic.com/manuals/default.htm
- technical discussion forums about different developer-specific Q&A at http://www.dialogic.com/forums/category-view.asp
- the Dialogic Customer Support web site. For detailed information about how to contact Dialogic Customer Support, see Dialogic Customer Support below.

Dialogic Customer Support

If the information on the Services & Support site did not help you address your issue, contact Dialogic Customer Support. See www.dialogic.com/support/contact for details on how to contact Dialogic.