



# **Dialogic® NaturalAccess™ MTP2 Layer Developer's Reference Manual**

## Copyright and legal notices

---

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

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

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

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

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

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

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

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

## Revision history

| Revision                     | Release date    | Notes             |
|------------------------------|-----------------|-------------------|
| B.1.0                        | July, 1998      | GJG               |
| 9000-6477-11                 | September, 1998 | GJG               |
| 9000-6477-12                 | March, 1999     | GJG               |
| 9000-6477-14                 | November, 2000  | GJG; SS7 3.6      |
| 9000-6477-15                 | June, 2001      | GJG, SS7 3.61     |
| 9000-6477-16                 | August, 2001    | GJG; SS7 3.8 Beta |
| 9000-6477-17                 | February, 2002  | MVH; SS7 3.8      |
| 9000-6477-18                 | November, 2003  | MCM; SS7 4.0      |
| 9000-6477-19                 | July 2006       | LBZ, SS7 4.3      |
| 64-0464-01                   | July 2009       | LBG, SS7 5.1      |
| Last modified: June 23, 2009 |                 |                   |

Refer to [www.dialogic.com](http://www.dialogic.com) for product updates and for information about support policies, warranty information, and service offerings.



# Table Of Contents

|  |           |
|--|-----------|
| <b>Chapter 1: Introduction .....</b>                         | <b>7</b>  |
| <b>Chapter 2: NMS MTP 2 overview .....</b>                   | <b>9</b>  |
| NMS MTP 2 components .....                                   | 9         |
| MTP 2 service users .....                                    | 11        |
| MTP 2 interface .....  | 12        |
| Binding .....  | 12        |
| Establishing a connection .....                              | 12        |
| Transferring data .....                                      | 13        |
| Status and control .....                                     | 14        |
| Clearing a connection .....                                  | 15        |
| CPI library interface .....                                  | 15        |
| TX driver message format .....                               | 16        |
| Byte ordering considerations .....                           | 17        |
| Configuring MTP 2 .....                                      | 18        |
| Configuring TDM ports .....                                  | 18        |
| Configuring MTP 2 links .....                                | 18        |
| Downloading to the board .....                               | 18        |
| Running txalarm .....  | 22        |
| Running a demonstration program .....                        | 23        |
| <b>Chapter 3: MTP 2 function reference .....</b>             | <b>25</b> |
| MTP 2 function summary .....                                 | 25        |
| Using the MTP 2 function reference .....                     | 25        |
| Mtp2GetLinkCfg .....   | 26        |
| Mtp2InitLinkCfg .....  | 27        |
| Mtp2LinkStats .....  | 33        |
| Mtp2LinkStatus .....   | 35        |
| Mtp2MgmtInit .....   | 37        |
| Mtp2MgmtTerm .....   | 38        |
| Mtp2SetLinkCfg .....   | 39        |
| <b>Chapter 4: Demonstration programs and utilities .....</b> | <b>41</b> |
| Summary of the demonstration programs and utilities .....    | 41        |
| MTP 2 configuration: mtp2cfg .....                           | 42        |
| MTP 2 manager: mtp2mgr .....                                 | 47        |
| Using the MTP 2 standalone task: mtp2app and mtp2itu .....   | 52        |
| <b>Chapter 5: MTP 2 message reference .....</b>              | <b>55</b> |
| MTP 2 message summary .....                                  | 55        |
| Using the MTP 2 message reference .....                      | 56        |
| Bind request .....   | 56        |
| Connect confirmation .....                                   | 57        |
| Connect request .....  | 57        |
| Data confirmation .....                                      | 58        |
| Data indication .....  | 58        |
| Data request .....   | 59        |
| Disconnect confirmation .....                                | 59        |
| Disconnect indication .....                                  | 60        |
| Disconnect request .....                                     | 60        |

|                               |    |
|-------------------------------|----|
| Disconnect response .....     | 61 |
| Flow control indication ..... | 61 |
| Flow control request .....    | 62 |
| Flow control response .....   | 62 |
| Status confirmation .....     | 63 |
| Status indication .....       | 64 |
| Status request .....          | 65 |
| Status response .....         | 66 |

---

# 1

## Introduction

---

The *NMS MTP 2 Developer's Reference Manual* explains how to implement the SS7 MTP 2 layer using NMS MTP 2. This manual explains how to create applications using NMS MTP 2 and presents a detailed specification of its signaling procedures and functions.





---

# 2

## NMS MTP 2 overview

---

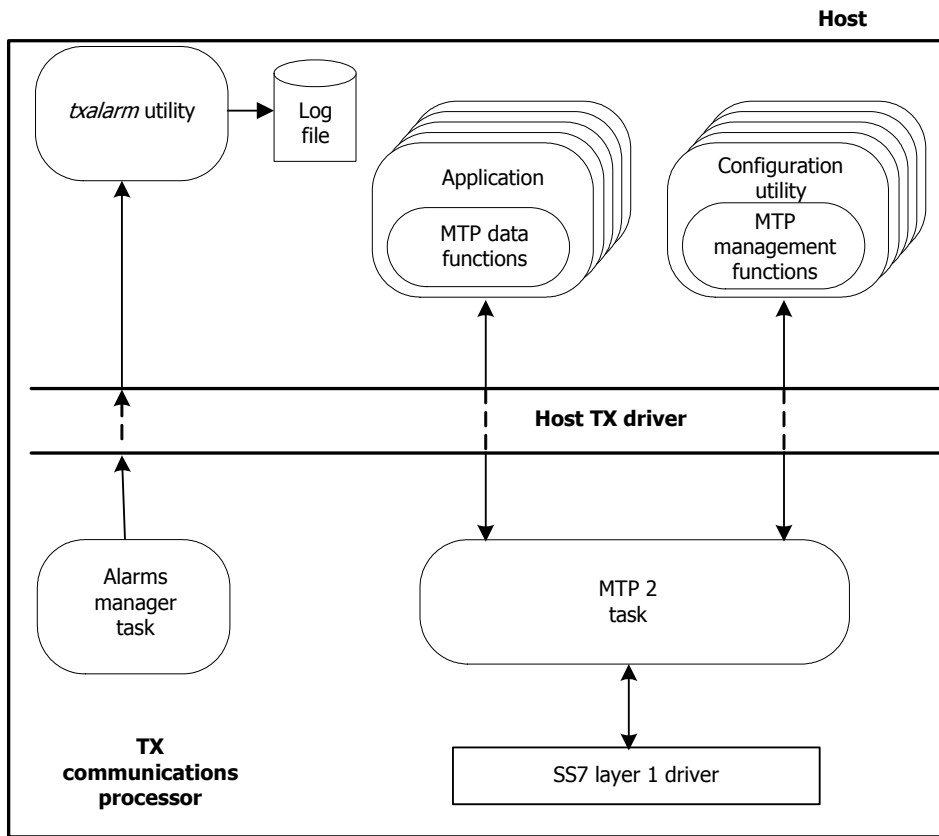
### NMS MTP 2 components

---

A typical NMS MTP 2 implementation consists of the following components:

- MTP 2 task that provides the signaling link functions defined by the ITU-T (Q.703), ANSI (T1.111), and TTC standards:
  - Signal unit delimitation and alignment
  - Error detection and correction
  - Link alignment
  - Link error monitoring
  - Flow control
- High speed links (HSL) meet the ANSI T1.111-1996 and Q.703/Annex A standards. Each HSL occupies a full (unchannelized) T1/E1 and transfers data at the rate of 2.0 (1.544) Mbps. For information about configuring high speed links, refer to *Mtp2InitLinkCfg* on page 27.
- The TX alarms manager task that collects unsolicited alarms (status changes) generated by the SS7 tasks and forwards them to the host for application-specific alarm processing.
- The *txalarm* utility for displaying and optionally saving alarms from the MTP 2 task to disk.
- A TX driver for the host operating system that provides low level access to the TX boards from the host.
- The MTP 2 interface consisting of a set of messages passed between the application and the MTP 2 task on the TX communications processor. These messages enable the application to initialize, transfer data, and exchange status information with MTP 2, and implement the link and traffic management procedures defined in the relevant standards for MTP 3.
- The *mtp2cfg* utility for downloading text file configurations to the MTP 2 task. *mtp2cfg* is provided in source and executable form.
- The *mtp2mgr* utility for managing the MTP 2 task. *mtp2mgr* is provided in source and executable form.
- The *mtp2app* and *mtp2itu* demonstration programs, also provided in source and executable form. These programs accept user commands and send messages to the MTP 2 task.
- MTP 2 functions that manage the MTP 2 task. *mtp2cfg* and *mtp2mgr* use these functions.

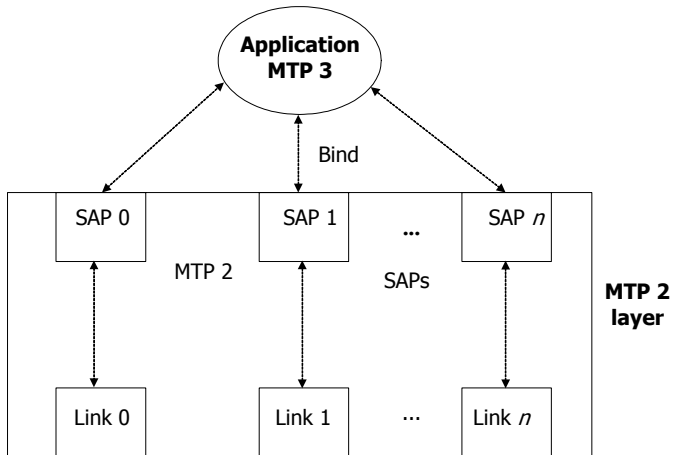
The following illustration shows a typical MTP 2 implementation:



## MTP 2 service users

A TX board can support as many as 32 SS7 links numbered 0 through 31, depending on the board model and board configuration. Additional TX boards can be added for more links.

MTP 2 uses SAPs (service access points) to support one or more applications. Each SAP corresponds to a single SS7 link controlled by the MTP 2 task. An application binds to a particular SAP at initialization time, specifying the link number (0 through 31). An application implementing MTP 3 repeats the bind process for all SS7 links on a TX board. Only one application is allowed to bind to each MTP 2 SAP (link). The following illustration shows MTP2 SAPs:



## MTP 2 interface

The MTP 2 interface consists of messages that govern communications between an application on the host and the MTP 2 task on the TX board.

MTP 2 messages perform the following tasks:

- Binding
- Establishing a connection
- Transferring data
- Status and control
- Clearing a connection

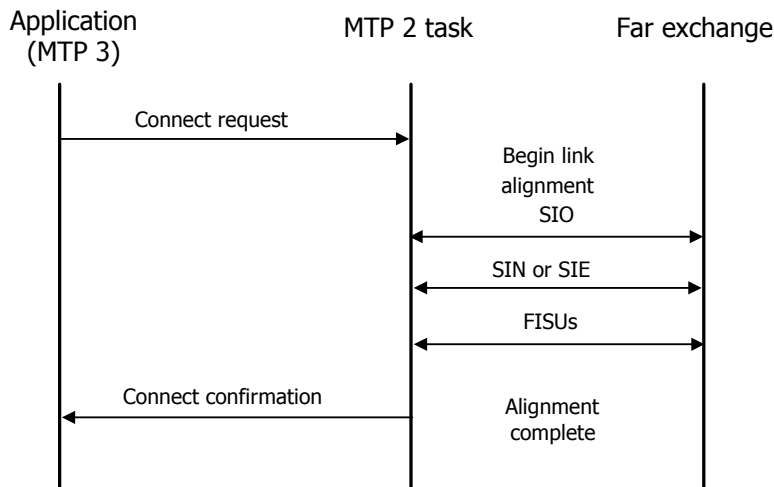
### Binding

The binding phase establishes the host application (MTP 3) as the user of the MTP 2 interface. The application sends a single bind request message to MTP 2, for which there is a bind confirmation response. The application must send separate bind request messages to MTP 2 for each link on each TX board that it wants to use.

### Establishing a connection

The application establishes a connection by sending a connect request message and attempting to bring up the link and initiate link alignment procedures with the far exchange. MTP 2 sends a connect confirmation message back to the application when link alignment is successfully established. The application requests connections separately for each SS7 link.

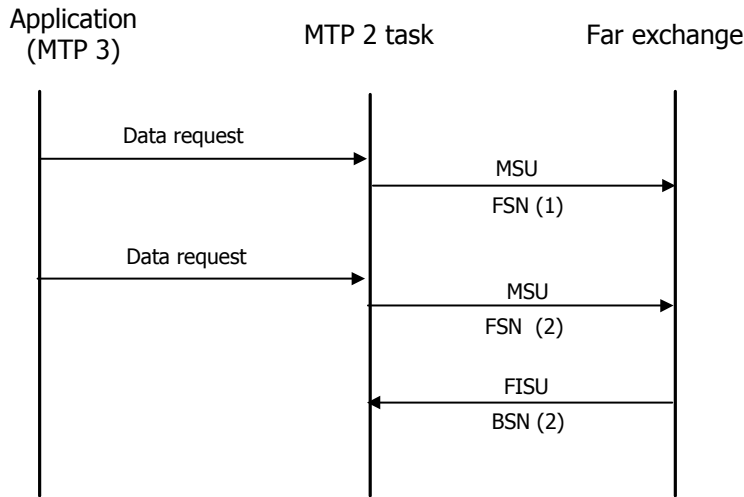
No data packets are transferred until the application receives the connect confirmation message. The following illustration shows the message flow for establishing a connection:



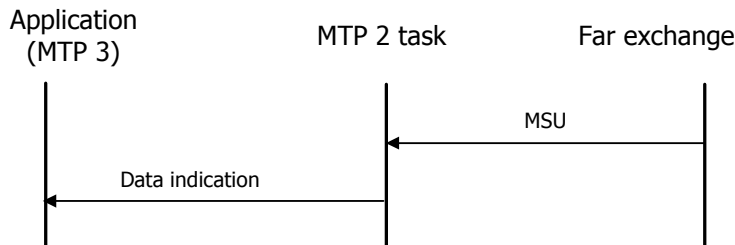
### Transferring data

After the MTP 2 task returns a connect confirmation message to the application, the application can begin transferring data.

The application sends a data request message to MTP 2 to request transmission of an SS7 packet on a particular link. When a message or messages are acknowledged by the far exchange there is no corresponding notification sent to the application. The following illustration shows the message flow for an outgoing data transfer:



MTP 2 sends a data indication message to notify the application of an incoming data packet. The application does not respond to MTP 2. The following illustrates an incoming data packet notification:



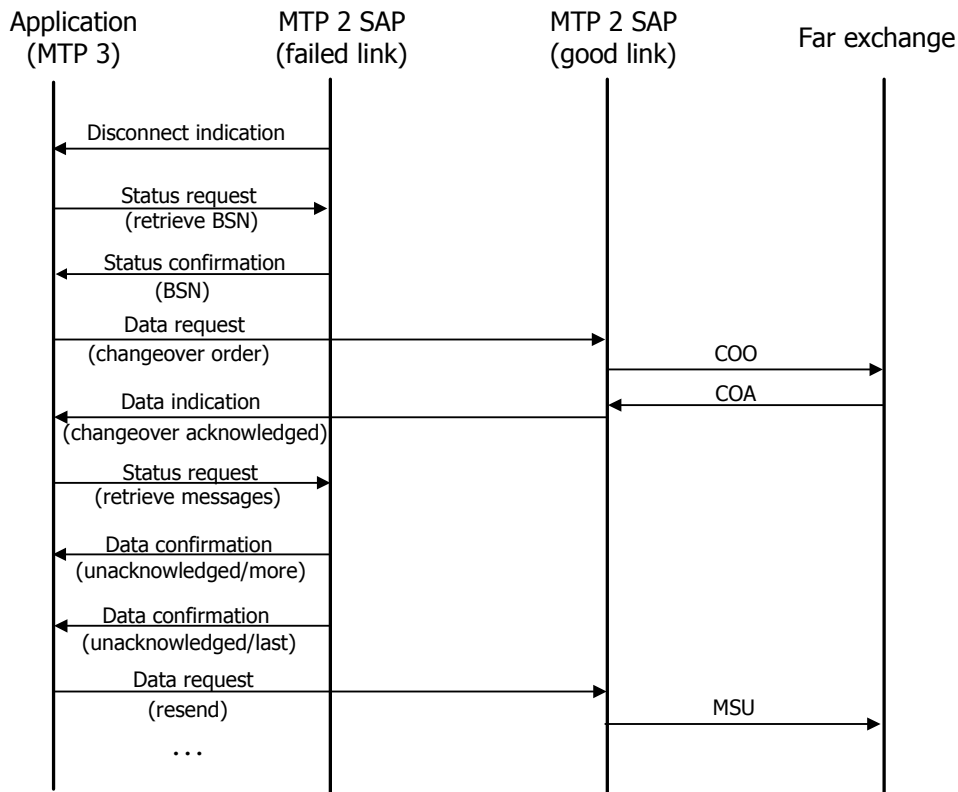
During data transfer, the application can send a flow control request to apply flow control to a link (or to stop flow control). MTP 2 does not respond to the application. MTP 2 can also send an unsolicited flow control indication message to the application to indicate that congestion started or ended on a link.

### Status and control

The MTP 2 status request provides the following functions that help MTP 3 implement link and traffic management procedures:

- Retrieves the current BSN (last acknowledged sequence number) for a link. This function is useful in implementing changeover procedures.
- Retrieves all MSUs (message signal units) transmitted on a link but not yet acknowledged. This function is useful in implementing changeover procedures.
- Requests MTP 2 to drop all queued messages.
- Notifies MTP 2 of an emergency on or off condition on a link.
- Notifies MTP 2 of a local processor up or down condition (passed on to the far exchange).

MTP 2 returns a status confirmation message containing the current BSN for a link to the application in response to a retrieve BSN status request. MTP 2 can also return a status confirmation message in response to a retrieve messages status request, but only if there are no unacknowledged messages to be retrieved. If there are unacknowledged messages to be retrieved, a data confirmation message indicating a status of unacknowledged [more|last] is sent to the application for each message. The last message indicates that it is the last. In this case, no status confirmation message is returned to the application. The following illustration shows how to use retrieve status requests to implement link changeover:



## Clearing a connection

---

The application can send a disconnect request message to MTP 2 to disable the link. MTP 2 does not respond to a disconnect request message. MTP 2 can send an unsolicited disconnect indication message to the application to notify it that a link disconnected and the reason for the disconnect.

MTP 2 sends unsolicited status indication messages to the application to notify it of changes in the link status (up or down).

Refer to the *MTP 2 message summary* on page 55 for additional information

## CPI library interface

---

TX device drivers implement the message-passing interface between the host and the TX processor (CP). The physical interface between the host and the CP (shared dual-port RAM or DPR) is subdivided into a set of virtual channels to support communication between processes on the host and on the TX board. The channel concept enables multiple processes on the host and the CP to exchange messages without interfering with each other. The TX board currently supports 256 channels numbered 0 through 255. To minimize conflicts, NMS recommends the following channel usage:

| Channel         | Usage                             |
|-----------------|-----------------------------------|
| 0 through 31    | Reserved                          |
| 32 through 127  | Available for use by applications |
| 128 through 255 | Reserved                          |

A process on either the host or the CP wanting to receive messages registers with the driver for a specific channel number. For a host-resident process, the channel registration can be implicit with the device open request, or it can be a separate operation. Channel numbers are assigned statically at compile time. There is no built-in facility to dynamically learn the channel assigned to a particular function. Only one process can receive messages for a particular channel. Each message contains a header supplied by the sender of the message that identifies the destination board and channel, the source board and channel, and the length of the message.

Each application must have a unique entity and instance ID to route messages between the processes in the system. Entity IDs are single byte values in the range of 0x00 through 0xFF. Allocate entity IDs as follows:

| Range                                  | Usage   |
|--|---|
| 0x00 through 0x1F<br>0x80 through 0xFF | Reserved for system utilities, configuration utilities, and management utilities. |
| 0x20 through 0x7F                      | Reserved for applications.  |

Instance IDs identify the processor on which the entity executes. The host is always processor 0 (zero). All host-resident MTP 2 applications must be coded to zero. All tasks on TX board 1 receive an instance ID of 1, all tasks on TX board 2 receive an instance ID of 2, and so on.

Each process on the CP and on the host in an SS7 implementation registers for the channel that corresponds to that layer's (or application's) entity ID. MTP 2 registers for channel 0x16. To send a message to the MTP 2 task on board *n*, the destination channel is set to 0x16, the destination board is set to *n*, and the source channel and board are set to the sender's entity ID and board number.

Call the TX driver through the CPI library. Call **cpi\_init** before issuing any other CPI calls. Issue a call to **cpia\_open** for each board with which the application wants to communicate. Then call **cpi\_send** (synchronous interface) or **cpia\_send** (asynchronous interface) and **cpi\_get\_data** (synchronous interface) or **cpia\_get\_data** (asynchronous interface) to send and retrieve data packets to and from the board. See the *CPI Library Developer's Reference Manual* for more details.

### TX driver message format

The following table describes the format of messages passed between the MTP 3 task on the host and the MTP 2 task on the CP through the TX driver:

| Octet   | Message format   |
|---------|--|
| 0       | Source channel (source entity ID)  |
| 1       | Source board (source instance ID). The source entity and instance IDs define the sender of the message.  |
| 2       | Destination channel (destination entity ID)  |
| 3       | Destination board (destination instance ID). The destination entity and instance IDs define the intended recipient of the message.   |
| 4 and 5 | Message length (2 octets). Contains the byte length of the entire message, including headers, filler, and parameters.  |
| 6 and 7 | Message start offset (2 octets). Contains the number of octets of filler included following the message end offset before the message type parameter. Include at least four bytes of filler on messages sent to MTP 2 to guarantee MTP 2 has room to prepend its header to the actual data portion of a message without having to copy the data to a new buffer. On messages received from MTP 2, the layer 3 application must use this parameter to determine where in the buffer the actual message content (the message type field) begins. |
| 8 and 9 | Message end offset (2 octets)  |
| 10..i   | Filler   |
| i + 1   | Message type   |
| i + 2   | ...  |
| ...     | MTP 2 message parameters   |
| n       | ...  |

The MTP 2 demonstration programs *mtp2app* and *mtp2itu* provide include files containing structures for message formats. For further details on the MTP 2 messages, see *MTP 2 message summary* on page 55.



## Byte ordering considerations

The following byte ordering conventions are used for 16-bit and 32-bit parameters in the messages to and from MTP 2. The host application is responsible for converting byte order between network and host, if necessary, in both the send and receive directions. The message start and end offsets are part of the interface between the host application and the MTP 2 task, and must be passed in the network byte order as shown in this illustration.

**Note:** This is not the same byte order used for short and long integers on Intel 80x86 machines. The message length field is part of the DPR header interface between the host application and the host device driver, and must be passed in host native byte order, not in the byte order shown in this topic.

### 16-bit fields

Address

|      |                       |
|------|-----------------------|
| Low  | High order byte (MSB) |
| High | Low order byte (LSB)  |

### 32-bit fields

Address

|      |                       |
|------|-----------------------|
| Low  | High order word (MSB) |
|      | High order word (LSB) |
| High | Low order word (MSB)  |
|      | Low order word (LSB)  |

The following nomenclature is used for data types:

| Data type | Description              |
|-----------|--------------------------|
| U8        | Unsigned 8-bit quantity  |
| S16       | Signed 16-bit quantity   |
| U32       | Unsigned 32-bit quantity |

## Configuring MTP 2

---

After installing the TX board and the software, configure MTP 2 as follows:

1. Configure TDM ports.
2. Configure MTP 2 links.
3. Download to the board.
4. Run *txalarm*.
5. Run a demonstration program.

### Configuring TDM ports

---

If the TX board running the MTP 2 task sends and receives on a T1 or E1 timeslot, configure the TDM ports that the MTP 2 task uses. NMS SS7 provides sample TDM configuration files for ANSI standalone and redundant configurations and ITU standalone and redundant configurations. On a Windows operating system, the files are *tdmcp1* and *tdmcp2*, located in `\nms\tx\config\`. On a UNIX system, the files are *TDMcp1* and *TDMcp2*, and are located in `/opt/nmstx/etc/`. You can modify these files for your specifications.

For more information on configuring TDM ports, see the *NMS SS7 Configuration Manual*.

### Configuring MTP 2 links

---

MTP 2 allows applications to configure the number of links and the parameters associated with those links. NMS SS7 provides sample link configuration files for ANSI standalone and redundant configurations and ITU standalone and redundant configurations. These files support back-to-back configurations. They contain a definition for one TDM port and a commented-out V.35 port. If the installation uses V.35 ports, comment in the example serial port definition in the file and comment out the TDM port.

On a Windows operating system, the files are *mtp3cp1* and *mtp3cp2*, located in `\nms\tx\config\`. On a UNIX system, the files are *MTP3cp1* and *MTP3cp2*, and are located in `/opt/nmstx/etc/`.

For more information on configuring MTP 2 links, see the *NMS SS7 Configuration Manual*.

### Downloading to the board

---

After you configure the TDM ports and links, download the configuration to the TX board using *mtp2load*. *mtp2load* accepts one parameter, the board number to load. *mtp2load* downloads the following software to the TX board:

- The proper kernel
- Support tasks for alarms and debugging
- A console program
- The T1/E1 managers
- The MTP 2 task
- The MTP 2 configuration

*mtp2load* is located in `\nms\tx\bin` on the Windows operating system and in `/opt/nmstx/bin` on the UNIX operating systems.

### Windows version of *mtp2load*

```
@echo off
REM *****
REM          TX Series COMMUNICATIONS PROCESSOR BOOT FILE (MTP2)
REM
REM  Execute this file to perform the following:
REM    - Reset the TX board
REM    - Synchronize the on-board flash image with the installed software
REM    - Download TDM configuration
REM    - Download all TX-based tasks
REM    - Configure MTP
REM *****

REM *****
REM  Define all script parameters
REM
set TXUTIL=\nms\tx\bin
set TXCP=\nms\tx\cp
set TXCONFIG=\nms\tx\config\ansi

REM *****
REM  Process arguments - Get the board number
REM
set BRD=1
if not "%1"==" " set BRD=%1

REM *****
REM  Clear driver statistics
REM
%TXUTIL%\txstats -b %BRD% -z -q

REM *****
REM  Get the model number (TX board type)
REM
%TXUTIL%\cpmodel -b %BRD%
if errorlevel 4000 goto boot4000
if errorlevel 3220 goto boot3220
echo ERROR! TX board number %BRD% not available.
goto end

REM *****
REM  Perform board type-specific boot for TX 3220 or TX 3220C
REM
:boot3220
set TASKTYPE=lot
REM  Reset TX board (and verify TX flash image in sync with installed software)
%TXUTIL%\txflash -s %TXCP%\cpk3220.bin -b %BRD%
if errorlevel 1 goto failedreset

REM load the diagnostic operator console task
%TXUTIL%\cplot -c %BRD% -f %TXCP%\diag3220.lot          -n diag          -p 2 -a

REM load TDM configuration
%TXUTIL%\cplot -c %BRD% -f %TXCONFIG%\TDMcp%BRD%.bin -g tdm

REM load ARP and INF (alarm forwarding task)
%TXUTIL%\cplot -c %BRD% -f %TXCP%\arp.lot              -n arp              -p 17 -a
%TXUTIL%\cplot -c %BRD% -f %TXCP%\inf.lot              -n inf              -p 16 -a

REM load the MVIP and T1/E1 manager tasks to enable use
REM of the MVIP and T1/E1 host APIs; NOTE: if you do not
REM use either of these APIs, remove the following 2 lines.
%TXUTIL%\cplot -c %BRD% -f %TXCP%\mvip.lot            -n mvip            -p 4 -a
%TXUTIL%\cplot -c %BRD% -f %TXCP%\tlelmgr.lot        -n tlelmgr        -p 15 -a

REM To enable packet tracing in the ISUP or TUP layer, make the following
```

```

REM command active to download the ETP trace collector on the board.
REM %TXUTIL%\cplot -c %BRD% -f %TXCP%\etp.lot          -n etp          -p 14 -a

goto loadcommon

REM *****
REM Perform board type-specific boot for TX 4000
REM
:boot4000
set TASKTYPE=elf
REM Reset TX board (and verify TX flash image in sync with installed software)
%TXUTIL%\txflash -s %TXCP%\cpk4000.fls -b %BRD%
if errorlevel 1 goto failedreset

REM load TDM configuration
%TXUTIL%\txconfig -b %BRD% -f %TXCONFIG%\txcfg%BRD%.txt

goto loadcommon

REM *****
REM Load all TX-based tasks that are common to all board types
REM
:loadcommon

REM To use txdbg,
REM you should uncomment the following line to load the debug task
REM %TXUTIL%\cplot -c %BRD% -f %TXCP%\debug.%TASKTYPE%  -n debug      -p 11 -a

REM Load MTP12 task
REM
%TXUTIL%\cplot -c %BRD% -f %TXCP%\mtp12.%TASKTYPE%      -n mtp          -p 20 -a -s 12000

REM *****
REM Configure SS7 MTP2
REM
%TXUTIL%\mtp2cfg -b %BRD% -f %TXCONFIG%\MTP2cp%BRD%.cfg
goto end

REM *****
REM Report reset error
REM
:failedreset
echo ERROR! Unable to reset TX board number %BRD%.
goto end

REM *****
REM Exit load script
:end
set TXMODE=
set TXUTIL=
set TXCP=
set TXCONFIG=
set BRD=
set TASKTYPE=

```

### UNIX version of *mtp2load*

```

#!/bin/ksh
*****
#
#           TX Series COMMUNICATIONS PROCESSOR BOOT FILE (MTP2)
#
# Execute this file to perform the following:
#   - Reset the TX board
#   - Synchronize the on-board flash image with the installed software
#   - Download TDM configuration
#   - Download all TX-based tasks
#   - Configure MTP
*****
*****

```

```

# Define all script parameters
#
if [ -z "$TXUTIL" ]
then
    TXUTIL=/opt/nmstx/bin
fi
if [ -z "$TXBASE" ]
then
    TXBASE=/opt/nmstx/cp
fi
if [ -z "$TXCONFIG" ]
then
    TXCONFIG=/opt/nmstx/etc/ansi
fi

#*****
# Process arguments - Get the board number
#
case $# in
    0)
        BRD=1
        ;;
    1)
        BRD=$1
        ;;
    *)
        echo "Usage: mtp2load <board#>"
        exit 1
esac

#*****
# Clear driver statistics
#
$TXUTIL/txstats -b $BRD -z -q

#*****
# Get the model number (TX board type)
#
BOARDTYPE=`$TXUTIL/cpmodel -b$BRD | tail -1 | cut -d' ' -f5`

case $BOARDTYPE in
    TX4000)
        FLASH="cpk4000.fls"
        ;;
    TX3220)
        FLASH="cpk3220.bin"
        ;;
    *)
        echo "ERROR! Board number $BRD not available"
        exit 1
        ;;
esac

#*****
# Reset TX board (and verify TX flash image in sync with installed software)
#
$TXUTIL/txflash -s $TXBASE/$FLASH -b$BRD

#*****
# Perform TX board type-specific load
#
if [ $BOARDTYPE = "TX3220" ]
then

# Perform board type-specific boot for TX 3220 or TX 3220C
TASKTYPE=lot

# load the diagnostic operator console task
$TXUTIL/cplot -c $BRD -f $TXBASE/diag3220.lot -n diag -p 2 -a

# load TDM configuration

```

```

    $TXUTIL/cplot -c $BRD -f $TXCONFIG/TDMcp${BRD}.bin -g tdm

#    load ARP and INF (alarm forwarding task)
$TXUTIL/cplot -c $BRD -f $TXBASE/arp.lot      -n arp      -p 17 -a
$TXUTIL/cplot -c $BRD -f $TXBASE/inf.lot      -n inf      -p 16 -a

#    load the MVIP and Tl/E1 manager tasks to enable use
#    of the MVIP and Tl/E1 host APIs. Note: if you do not
#    use either of these APIs, remove the following two lines
$TXUTIL/cplot -c $BRD -f $TXBASE/mvip.lot    -n mvip     -p 4  -a
$TXUTIL/cplot -c $BRD -f $TXBASE/tlelmgr.lot -n tlelmgr  -p 15 -a

#    To enable packet tracing in the ISUP or TUP layer, make the following
#    command active to download the ETP trace collector on the board.
$TXUTIL/cplot -c $BRD -f $TXBASE/etp.lot     -n etp      -p 14 -a

else

#    Perform board type-specific boot for TX 4000
TASKTYPE=elf

#    load TDM configuration
$TXUTIL/txconfig -b $BRD -f $TXCONFIG/txcfg$BRD.txt
fi

#*****
#    Load all TX-based tasks that are common to all board types
#

#    To use txdg,
#    you should uncomment the following line to load the debug task
#$TXUTIL/cplot -c $BRD -f $TXBASE/debug.$TASKTYPE      -n debug     -p 11 -a

#    Load MTP task
#
$TXUTIL/cplot -c $BRD -f $TXBASE/mtp12.$TASKTYPE      -n mtp       -p 20 -a -s 12000

#*****
#    Configure SS7 MTP2
#
$TXUTIL/mtp2cfg -b $BRD -f $TXCONFIG/MTP2cp${BRD}.cfg

#*****
#    Exit load script
exit 0

```

## Running txalarm

The *txalarm* utility captures messages from the boards, displays them on the screen, and optionally saves them to a disk file. *txalarm* is the primary tool for monitoring what is happening on the link as you download the board and bring links up.

Run *txalarm* from a separate window according to the following syntax:

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

where **-f *filename*** specifies the file to which messages are copied.

The following example shows *txalarm* messages when the board is downloaded without sending a new MTP 2 configuration. This shows only the first four links for brevity. There can be alarms for up to 32 links.

```

<07/25/06 14:52:25> mtp      1  16387 Registering MTP Layer 2
<07/25/06 14:52:25> mtp      1  16387 Configuring MTP Layer 1
<07/25/06 14:52:25> mtp      1  16387 MTP1 Initializing.
<07/25/06 14:52:25> mtp      1  16387 MTP1 General Configuration
<07/25/06 14:52:25> mtp      1  16387 MTP1 Configuring link 0: TDM, External
<07/25/06 14:52:25> mtp      1  16387 MTP1 Configuring link 1: TDM, External
<07/25/06 14:52:25> mtp      1  16387 MTP1 Configuring link 2: TDM, External
<07/25/06 14:52:25> mtp      1  16387 MTP1 Configuring link 3: TDM, External
<07/25/06 14:52:25> mtp      1  16387 Configuring MTP Layer 2
<07/25/06 14:52:25> mtp      1  16387 MTP2: General Configuration
<07/25/06 14:52:25> mtp      1  16387 MTP2: Link 0 Configuration
<07/25/06 14:52:25> mtp      1  16387 MTP2: Link 1 Configuration
<07/25/06 14:52:25> mtp      1  16387 MTP2: Link 2 Configuration
<07/25/06 14:52:25> mtp      1  16387 MTP2: Link 3 Configuration

```

## Running a demonstration program

After the board is loaded, start the demonstration program. *mtp2app* is an ANSI version and *mtp2itu* is an ITU-T (formerly CCITT) version of the demonstration program. See *Using the MTP 2 standalone task: mtp2app and mtp2itu* on page 52 for more details.

If you are using the TX board with test equipment, run an application then press **Enter** to display a command prompt. Enter connect 0 to start the connect procedure on link 0. The *txalarm* messages look like the following sample if the link aligns properly:

```

<07/25/06 15:30:08> mtp      1  16387 Flushing Buffers (OutPktCnt=0) (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Sent SIOS (Lnk 0) (Start in PowerOff)
<07/25/06 15:30:08> mtp      1  16387 Flushing Buffers (OutPktCnt=0) (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Starting Alignment (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Sent SIO (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Rcvd SIO (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Sent SIN (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Rcvd SIO (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Rcvd SIN (Lnk 0)
<07/25/06 15:30:11> mtp      1  16387 Timer 4 Expired (LINK 0 ALIGNED at layer 2)

```

If you are using two TX boards back-to-back, after downloading each board separately, start an application on each board (for example, *mtp2app -b 2* to run on board 2), and enter connect 0 on each board. The *txalarm* messages look like the following sample if the link aligns properly:

```

<07/25/06 15:30:08> mtp      1  16387 Flushing Buffers (OutPktCnt=0) (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Sent SIOS (Lnk 0) (Start in PowerOff)
<07/25/06 15:30:08> mtp      1  16387 Flushing Buffers (OutPktCnt=0) (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Starting Alignment (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Sent SIO (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Rcvd SIO (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Sent SIN (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Rcvd SIO (Lnk 0)
<07/25/06 15:30:08> mtp      1  16387 Rcvd SIN (Lnk 0)
<07/25/06 15:30:08> mtp      2  16387 Flushing Buffers (OutPktCnt=0) (Lnk 0)
<07/25/06 15:30:08> mtp      2  16387 Sent SIOS (Lnk 0) (Start in PowerOff)
<07/25/06 15:30:08> mtp      2  16387 Flushing Buffers (OutPktCnt=0) (Lnk 0)
<07/25/06 15:30:08> mtp      2  16387 Starting Alignment (Lnk 0)
<07/25/06 15:30:08> mtp      2  16387 Sent SIO (Lnk 0)
<07/25/06 15:30:09> mtp      2  16387 Rcvd SIOS (Lnk 0)
<07/25/06 15:30:09> mtp      2  16387 Rcvd SIO (Lnk 0)
<07/25/06 15:30:09> mtp      2  16387 Sent SIN (Lnk 0)
<07/25/06 15:30:09> mtp      2  16387 Rcvd SIO (Lnk 0)
<07/25/06 15:30:09> mtp      2  16387 Rcvd SIN (Lnk 0)
<07/25/06 15:30:11> mtp      2  16387 Timer 4 Expired (LINK 0 ALIGNED at layer 2)
<07/25/06 15:30:11> mtp      1  16387 Timer 4 Expired (LINK 0 ALIGNED at layer 2)

```

The middle column containing 1s and 2s identifies the board generating the alarm.





# 3

## MTP 2 function reference

### MTP 2 function summary

MTP 2 consists of the following synchronous functions:

| Function               | Description   |
|------------------------|---|
| <b>Mtp2GetLinkCfg</b>  | Obtains the current configuration values of the specified link number.  |
| <b>Mtp2InitLinkCfg</b> | Initializes the provided data link configuration structure with default values and the specified link number and link type. |
| <b>Mtp2LinkStats</b>   | Obtains and potentially resets statistical information about the specified link number.                                     |
| <b>Mtp2LinkStatus</b>  | Obtains status information about the specified link number.   |
| <b>Mtp2MgmtInit</b>    | Initializes internal structures and opens communication with the MTP 2 process on the TX board.                             |
| <b>Mtp2MgmtTerm</b>    | Terminates the dual port RAM channel binding specified in <b>Mtp2MgmtInit</b> for this application.                         |
| <b>Mtp2SetLinkCfg</b>  | Configures the MTP 2 task with the data link configuration values contained in the provided MTP2LinkCfg structure.          |

### Using the MTP 2 function reference

This section provides an alphabetical reference to the MTP 2 functions. A typical function includes:

|                      |   |
|----------------------|---|
| <b>Prototype</b>     | The prototype is shown followed by a list of the function arguments. NMS Communications data types include: <ul style="list-style-type: none"><li>• U8 (8-bit unsigned)</li><li>• S16 (16-bit signed)</li><li>• U32 (32-bit unsigned)</li><li>• Bool (8-bit unsigned)</li></ul> If a function argument is a data structure, the complete data structure is defined. |
| <b>Return values</b> | The return value for a function is either MTP2_SUCCESS or an error code.  |

## Mtp2GetLinkCfg

Obtains the current configuration values of the specified link number.

### Prototype

MTP2\_STATUS **Mtp2GetLinkCfg** (U8 *board*, MTP2LinkCfg \**pLinkCfg*, S16 *linkNo*)

| Argument        | Description  |
|-----------------|--|
| <i>board</i>    | TX board number on which the desired MTP 2 task resides.   |
| <i>pLinkCfg</i> | Pointer to an MTP2LinkCfg structure where the data link configuration values are filled.               |
| <i>linkNo</i>   | Link number for which to obtain configuration information (0 through maximum number of links minus 1). |

### Return values

| Return value | Description   |
|--------------|---|
| MTP2_SUCCESS |   |
| MTP2_BOARD   | Invalid board number.                                 |
| MTP2_HANDLE  | <b>Mtp2MgmtInit</b> never called for specified board. |
| MTP2_TIMEOUT | No response from board.                               |

### Details

This function obtains the current data link configuration parameters of the specified link on the specified board. An application must provide a pointer to a buffer large enough for the MTP2LinkCfg structure. This function can be called any time after **Mtp2MgmtInit**.

Refer to *Mtp2InitLinkCfg* on page 27 for more information on the MTP2LinkCfg structure.

### See also

**Mtp2InitLinkCfg, Mtp2MgmtInit**

## Mtp2InitLinkCfg

Initializes the provided data link configuration structure with default values and the specified link number and link type.

### Prototype

MTP2\_STATUS **Mtp2InitLinkCfg** (U8 *board*, MTP2LinkCfg \**pLinkCfg*, S16 *linkNo*, U8 *linkType*)

| Argument        | Description  |
|-----------------|--|
| <b>board</b>    | TX board number on which the desired MTP 2 task resides.   |
| <b>pLinkCfg</b> | <p>Pointer to the MTP2LinkCfg structure to be initialized:</p> <pre>typedef struct _MTP2LinkCfg /* MTP level 2 data link SAP configuration */ {     MemoryId mem;           /* memory region and pool id */     MemoryId memMac;       /* memory region and pool id for MAC */     S16 swtch;             /* protocol type */     U8 priorDl;           /* priority for data link layer */     U8 routeDl;          /* route for data link layer */     U8 selectorDl;       /* upper interface selector */     U8 entMac;           /* entity for MAC */     U8 instMac;          /* instance for MAC */     U8 priorMac;         /* priority for MAC layer */     U8 routeMac;         /* route for MAC layer */     U8 selectorMac;      /* lower interface selector */     U8 maxOutsFrms;     /* maximum outstanding frames */     Bool errType;        /* cyclical or normal error checking */     TimerCfg t1;         /* timer 1 configuration structure */     TimerCfg t2;         /* timer 2 configuration structure */     TimerCfg t3;         /* timer 3 configuration structure */     TimerCfg t5;         /* timer 5 configuration structure */     TimerCfg t6;         /* timer 6 configuration structure */     TimerCfg t7;         /* timer 7 configuration structure */     TimerCfg t10;        /* timer 10 configuration structure */     TimerCfg t11;        /* timer 11 configuration structure */     TimerCfg t12;        /* timer 12 configuration structure */     TimerCfg t13;        /* timer 13 configuration structure */     TimerCfg congTmr;    /* congestion timer configuration structure */     U16 provEmrgcy;      /* emergency proving period */     U16 provNormal;     /* normal proving period */     U8 lssuLen;         /* one or two byte LSSU length */     U8 hsBits;          /* High speed link bits defined above */     U16 maxFrmLen;      /* max frame length for MSU */     U8 cfmType;         /* when to send connect confirm */     Bool congDisc;      /* congestion discard TRUE or FALSE */     U16 sdD;            /* Error count decrement rate &lt;JMP&gt; */     U8 sdT;             /* SUERM error rate threshold */     U8 sdTie;          /* AERM emergency error rate threshold */     U8 sdTin;          /* AERM normal error rate threshold */     U8 sdN1;           /* maximum number of MSUs for retransmission */     U16 sdN2;          /* maximum number of MSU octets for retrans */     U8 sdCp;           /* maximum number of alignment attempts */     U8 align2;         /* alignment byte */     S16 spIdSE;        /* service provider id */     U16 dstProcId;     /* processor id */     U16 isoThresh;     /* Number of packets to queue before sending                        /* SIPOs     U16 txqThresh1;    /* Threshold level one on MTP2's Tx Queue     U16 txqThresh1Abate; /* Abatement threshold for level two on                        /* MTP2's Tx Queue     U16 txqThresh2;    /* Threshold level two on MTP2's Tx Queue     U16 txqThresh2Abate; /* Abatement threshold for level two on                        /* MTP2's Tx Queue </pre> |

| Argument        | Description  |
|-----------------|--|
|                 | <pre> U16    sapThresh;    /* Threshold level on MTP2's upward bound */                 /* SAP queue */ U16    sapThreshAbate; /* Abatement threshold level for MTP2's */                 /* SAP Queue */ /* MTP level 1 parameters &lt;JMP&gt; S16    baud;        /* baud rate (4.8, 9.6, 19.2, 28.8, 38.4, */                 /* 56, 64) */ U8     portNum;     /* port number */ U8     portType;    /* port type */ S16    physIntType; /* Physical interface type (DCE/DTE) */ U8     dataEncoding; /* Data encoding (NRZ/NRZI) */ U8     flgShare;    /* Share flags ? */ U8     flgIdle;     /* Use idles (True) or flags (False) */ U8     numFlg;      /* Number of additional flag chars */                 /* between frames */ U32    idleFreq;    /* Transmit frequency of FISU's when done by */                 /* software (in ms). Done at config */ U32    rtFreq;      /* Retransmit frequency of other SU's when */                 /* done by software (in ms). Done at transmit*/  } MTP2LinkCfg; </pre> <p>See the Details section for field descriptions.</p> |
| <b>linkNo</b>   | Link number to assign to this data link. Valid range is 0 through the maximum number of links minus 1, depending on hardware configuration and whether high speed links are defined.   |
| <b>linkType</b> | Type of link. Valid values are:<br>MTP2LNK_ANSI<br>MTP2LNK_ANSI88<br>MTP2LNK_CCITT   |

**Return values**

| Return value | Description        |
|--------------|--------------------|
| MTP2_SUCCESS |                    |
| MTP2_LNKTYPE | Invalid link type. |

**Details**

This function enables an application to initialize an MTP2LinkCfg structure before calling **Mtp2SetLinkCfg** to set the data link configuration parameters.

**Mtp2InitLinkCfg** sets the link number and link type as specified and initializes all other fields in the Mtp2SetLinkCfg structure to their defaults.

A pointer to an MTP2LinkCfg structure is passed to **Mtp2InitLinkCfg** where default values are set. After initialization, an application can override zero to all of these values and then pass the pointer to **Mtp2SetLinkCfg**, which sets the configuration. The default values provided by **Mtp2InitLinkCfg** are adequate for most applications.

The following table lists the MTP2LinkCfg fields that can be modified. Unless otherwise specified, the timer names correspond to the CCITT specification.

**Note:** All timer values are specified in tenths of a second.

| Field      | Description   |
|------------|---|
| swtch      | Protocol type. Default is the <b>linkType</b> parameter. Valid values:<br>MTP2LNK_ANSI<br>MTP2LNK_ANSI88<br>MTP2LNK_CCITT                                 |
| maxOutsFrm | Maximum outstanding frames to layer MTP 1. NMS recommends leaving the default at 2.   |
| errType    | Normal or cyclical (for PCR) error checking. Valid values:<br>ERR_NRM (default)<br>ERR_CYC  |
| t1         | Timer aligned and ready. Valid range is 1 through 65535. Default is 130 (ANSI) and 400 (ITU).   |
| t2         | Timer not aligned. Valid range is 1 through 65535. Default is 115 (ANSI) and 100 (ITU).   |
| t3         | Timer aligned. Valid range is 1 through 65535. Default is 115 (ANSI) and 15 (ITU).  |
| t5         | Timer sending SIBs (busy indications). Valid range is 1 through 65535. Default is 1.  |
| t6         | Timer remote congestion. Valid range is 1 through 65535. Default is 60.   |
| t7         | Timer excessive delay of acknowledgement. Valid range is 1 through 65535. Default is 20.  |
| t10        | Amount of time MTP 2 can be isolated from a remote MTP 3 before SIPO (sending processor outage). Valid range is 1 through 65535. Default is 30.           |
| t11        | Time to wait for a flow control acknowledgement from MTP 3 before sending another flow control indication. Valid range is 1 through 65535. Default is 20. |
| t12        | Time to wait for a status confirmation from MTP 3 before sending another status indication. Valid range is 1 through 65535. Default is 20.                |
| t13        | Time to wait for a disconnect confirmation from MTP 3 before sending another disconnect indication. Valid range is 1 through 65535. Default is 20.        |
| provNormal | Normal proving period. Valid range is 1 through 65535. Default is 23 (ANSI) and 82 (ITU).   |
| provEmrgcy | Emergency proving period. Valid range is 1 through 65535. Default is 6 (ANSI) and 5 (ITU).  |
| IssuLen    | One or two byte LSSU length. Valid values:<br>1<br>2 (default)  |

| Field      | Description   |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
|------------|---|--|-------|-------------|---------|------|-------------------------------|------------|------|--|---------|-------------|---------|--|------------|---|
| hsBits     | <p>High speed link bits valid values:</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>HS_LINK</td> <td>0x01</td> <td>High speed link is in effect.</td> </tr> <tr> <td>HS_EXT_SEQ</td> <td>0x02</td> <td>Extended sequence numbers are in effect.</td> </tr> </tbody> </table> <p>The following table describes these high speed link bits:</p> <table border="1"> <thead> <tr> <th>HSL bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>HS_LINK</td> <td>Setting this bit to TRUE, notifies MTP that high speed links are in effect and automatically sets HS_EXT_SEQ to TRUE. Set HS_EXT_SEQ to FALSE for high speed links with normal sequence numbers.</td> </tr> <tr> <td>HS_EXT_SEQ</td> <td>Setting this bit to TRUE, notifies MTP that extended sequence numbers are in effect and changes the size of: <ul style="list-style-type: none"> <li>FSN, BSN, and LI fields in MTP 2 packets</li> <li>The last FSN field of COO and COA messages at layer 3</li> </ul> Sequence numbers increase from 7 to 12 bits and the length indicator increases from 6 to 8 bits. Setting HS_EXT_SEQ to TRUE automatically sets HS_LINK to TRUE. Normal speed links with extended sequence numbers are not supported.</td> </tr> </tbody> </table> <p>A combination of high and normal speed links is not supported.</p> | Bit                                      | Value | Description | HS_LINK | 0x01 | High speed link is in effect. | HS_EXT_SEQ | 0x02 | Extended sequence numbers are in effect. | HSL bit | Description | HS_LINK | Setting this bit to TRUE, notifies MTP that high speed links are in effect and automatically sets HS_EXT_SEQ to TRUE. Set HS_EXT_SEQ to FALSE for high speed links with normal sequence numbers. | HS_EXT_SEQ | Setting this bit to TRUE, notifies MTP that extended sequence numbers are in effect and changes the size of: <ul style="list-style-type: none"> <li>FSN, BSN, and LI fields in MTP 2 packets</li> <li>The last FSN field of COO and COA messages at layer 3</li> </ul> Sequence numbers increase from 7 to 12 bits and the length indicator increases from 6 to 8 bits. Setting HS_EXT_SEQ to TRUE automatically sets HS_LINK to TRUE. Normal speed links with extended sequence numbers are not supported. |
| Bit        | Value   | Description                              |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| HS_LINK    | 0x01  | High speed link is in effect.            |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| HS_EXT_SEQ | 0x02  | Extended sequence numbers are in effect. |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| HSL bit    | Description   |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| HS_LINK    | Setting this bit to TRUE, notifies MTP that high speed links are in effect and automatically sets HS_EXT_SEQ to TRUE. Set HS_EXT_SEQ to FALSE for high speed links with normal sequence numbers.  |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| HS_EXT_SEQ | Setting this bit to TRUE, notifies MTP that extended sequence numbers are in effect and changes the size of: <ul style="list-style-type: none"> <li>FSN, BSN, and LI fields in MTP 2 packets</li> <li>The last FSN field of COO and COA messages at layer 3</li> </ul> Sequence numbers increase from 7 to 12 bits and the length indicator increases from 6 to 8 bits. Setting HS_EXT_SEQ to TRUE automatically sets HS_LINK to TRUE. Normal speed links with extended sequence numbers are not supported.   |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| maxFrmLen  | Maximum frame length for MSU. Valid range is 64 through 1024. NMS recommends leaving the default at 272.  |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| congDisc   | Congestion discard. Valid values:<br>TRUE<br>FALSE (default)  |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| sdT        | SUERM error rate threshold. Valid range is 1 through 255. Default is 64.  |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| sdD        | SUERM decrement rate. Valid range is 1 through 65535. Default is 256.   |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| sdTie      | AERM emergency error rate threshold. Valid range is 1 (default) through 255.  |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| sdTin      | AERM normal error rate threshold. Valid range is 1 through 255. Default is 4.   |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| sdN1       | Maximum number of MSUs for retransmission (when using PCR). Valid range is 1 through 255. Default is 30.  |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| sdN2       | Maximum number of MSUs octets for retransmission (when using PCR). Valid range is 1 through 65535. Default is 220.  |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |
| sdCp       | Maximum number of proving failures. Valid range is 1 through 255. Default is 5.   |  |       |             |         |      |                               |            |      |  |         |             |         |  |            |   |

| Field           | Description   |
|-----------------|---|
| baud            | Baud rate. Valid values:<br>BD_4800<br>BD_9600<br>BD_19200<br>BD_28800<br>BD_38400<br>BD_48000<br>BD_56000 (default)<br>BD_64000  |
| physIntType     | Physical interface type (DCE/DTE) with or without modem signal detection. Modem signal usage is the only way to assure detection of connection loss and re-establishment. Valid values:<br>INT_DTE (default)<br>INT_DCE<br>INT_DTE_MODEM<br>INT_DCE_MODEM<br><b>Note:</b> Modem signal detection requires an additional pod jumper on the DCE side connecting DSR and CD. |
| dataEncoding    | Data encoding (NRZ/NRZI). Valid values:<br>DAT_NRZ (default)<br>DAT_NRZ1  |
| flgShare        | Share flags between frames. Valid values:<br>FLAG_SHARE (default)<br>NO_FLAG_SHARE  |
| FlgIdle         | Use idles (TRUE) or flags (FALSE). Default is FALSE.  |
| numFlg          | Number of additional flag characters between frames. Valid range is 0 (default) through 15.   |
| isoThresh       | Number of messages queued to MTP 3 while isolated that causes MTP 2 to begin SIPOs (processor outages). Valid range is 1 through 65535. Default is 1000.  |
| txqThresh1      | Transmission queue length at which the outbound flow control level is raised to one. Valid range is 1 through 65535. Default is 50.   |
| txqThresh1Abate | Transmission queue length at which the outbound flow control level is lowered to zero. Valid range is 1 through 65535. Default is 20.   |
| txqThresh2      | Transmission queue length at which the outbound flow control level is raised to two. The subsequent indication causes MTP 3 to cease all transmission to MTP 2 until the flow control level returns to one or zero. Valid range is 1 through 65535. Default is 200.   |
| txqThresh2Abate | Transmission queue length at which the outbound flow control level is lowered to one. Valid range is 1 through 65535. Default is 100.   |
| sapThresh       | Number of messages queued to MTP 3 while inbound flow control is in effect that causes MTP 2 to send SIBs (busy indications). Valid range is 1 through 65535. Default is 500.   |
| sapThreshAbate  | Number of messages queued to MTP 3 while inbound flow control is in effect that causes MTP 2 to stop sending SIBs (busy indications). Valid range is 1 through 65535. Default is 100.   |

| Field    | Description  |
|----------|--|
| idleFreq | Frequency at which FISUs are sent by the software (in ms). 0 (default) indicates that hardware constantly retransmits duplicate FISUs.<br><br>Non-zero frequencies can be used by switches that process all FISUs (including duplicate FISUs) in the software. Valid range is 1 through 65535.                           |
| rtFreq   | Frequency at which other retransmitted SUs (LSSUs) are sent by the software (in ms). 0 (default) indicates that hardware constantly retransmits duplicate LSSUs.<br><br>Non zero frequencies can be used by switches that process all LSSUs (including duplicate LSSUs) in the software. Valid range is 1 through 65535. |

**See also****Mtp2SetLinkCfg**



## Mtp2LinkStats

Obtains and potentially resets statistical information about the specified link number.

### Prototype

MTP2\_STATUS **Mtp2LinkStats** (U8 *board*, S16 *linkNo*, MTP2LinkStats \**pStats*,  
 BOOL *bReset*)

| Argument      | Description  |
|---------------|--|
| <b>board</b>  | TX board number on which the desired MTP 2 task resides.   |
| <b>linkNo</b> | Link number for which to obtain statistical information. Must have been previously defined through <b>Mtp2SetLinkCfg</b> .   |
| <b>pStats</b> | <p>Pointer to a buffer provided by the application where the requested statistical information is returned:</p> <pre>typedef struct _Mtp2LinkStats /* MTP Level 2 link statistics */ {   U32  inService;          /* start of in-service          */   U32  lclBusy;           /* duration of local-busy       */   U32  slFailAll;         /* all reasons for a signaling link                           * failure                      */   U32  slFailAb;          /* Abnormal FIBR/BSNR link failure */   U32  slFailAck;         /* delay of acknowledgment link                           * failure                      */   U32  slFailErr;         /* excessive error rate link failure */   U32  slFailCong;        /* excessive congestion link failure */   U32  slFailAlign;       /* alignment link failure       */   U32  slNSUErr;          /* number of signal units in error */   U32  slNegAck;          /* number of negative acknowledgments */   U32  nSifSioTx;         /* number of SIF/SIO packets                           * transmitted                   */   U32  nRetrans;          /* number of octets transmitted */   U32  mTx;               /* information frames transmitted */   U32  lTx;               /* receive ready frames transmitted */   U32  fTx;               /* receive not ready frames                           * transmitted                   */   U32  nSifSioRx;         /* number of SIF/SIO packets received */   U32  mRx;               /* information frames received */   U32  lRx;               /* receive ready frames received */   U32  fRx;               /* receive not ready frames received */   U16  txCnt;             /* Number of messages on the transmit                           * queue                         */   U16  hiTxCnt;           /* High water number of messages on                           * the transmit queue           */   U16  rtbCnt;            /* Number of messages on the                           * retransmit queue            */   U16  hiRtbCnt;          /* High water number of messages on                           * the retransmit queue        */   U16  ltxCnt;            /* Number of messages on the lower                           * transmit queue              */   U16  hiLtxCnt;          /* High water number of messages on                           * the lower transmit queue     */   U16  sapCnt;            /* Number of messages on the inbound                           * sap queue                   */   U16  hiSapCnt;          /* High water number of messages on                           * the inbound sap queue       */ } MTP2LinkStats;</pre> |
| <b>bReset</b> | If true (non-zero), statistics are reset (set to zero) after returning the current values. If false (zero), no reset is performed.   |

## Return values

| Return value | Description   |
|--------------|---|
| MTP2_SUCCESS |   |
| MTP2_BOARD   | Invalid board number.                               |
| MTP2_HANDLE  | <b>Mtp2MgmtInit</b> not called for specified board. |

## Details

Statistics include counts for:

- Various message types and their acknowledgments, both transmitted and received.
- Various failure counts.
- Queue levels and high water marks.

## See also

**Mtp2MgmtInit**, **Mtp2SetLinkCfg**

## Mtp2LinkStatus

Obtains status information about the specified link number.

### Prototype

MTP2\_STATUS **Mtp2LinkStatus** (U8 *board*, S16 *linkNo*, MTP2LinkStatus *\*pStatus*)

| Argument       | Description   |
|----------------|---|
| <i>board</i>   | TX board number on which the desired MTP 2 task resides.  |
| <i>linkNo</i>  | Link number for which to obtain status information. Must have been previously defined through <b>Mtp2SetLinkCfg</b> .   |
| <i>pStatus</i> | <p>Pointer to a buffer provided by the application where the requested status information is returned:</p> <pre>typedef struct _Mtp2LinkStatus /* MTP Level 2 - signaling link status */ {     DateTime DT;                /* Date / time stamp                */     U8       hlSt;              /* sap state                        */     U8       psOutsFrm;        /* outstanding frames to MAC        */     S16     cntMaDrop;         /* frames dropped by MAC            */     U8       llSt;             /* link level matrix state          */     U8       lclBsy;          /* local busy                        */     Bool     remBsy;          /* remote busy                       */     U8       fsn;              /* forward sequence number          */     U8       bsn;              /* backward sequence number         */     U8       flcSt;           /* flow control state                */ } MTP2LinkStatus;</pre> <p>See the Details section for field descriptions.</p> |

### Return values

| Return value | Description   |
|--------------|---|
| MTP2_SUCCESS |   |
| MTP2_BOARD   | Invalid board number.                               |
| MTP2_HANDLE  | <b>Mtp2MgmtInit</b> not called for specified board. |

## Details

This function obtains status information about the specified link number, including the link state, flow control state, and busy states.

The following table describes the MTP2LinkStatus fields:

| Field     | Description   |
|-----------|---|
| hlSt      | SAP state. Valid values:<br>ST_UNBOUND<br>ST_CONFIGURED<br>ST_BND_DISABLED<br>ST_BND_ENABLED  |
| psOutsFrm | Outstanding frames to MAC. Valid range is 0 through 0xFF.   |
| cntMaDrop | Frames dropped by MAC. Valid range is 0 through 0xFF.   |
| llSt      | Link level matrix state. Valid values:<br>ST_UNUSED<br>ST_OUT_OF_SERVICE<br>ST_INITIAL_ALIGNMENT<br>ST_ALIGNED_READY<br>ST_ALIGNED_NOT_READY<br>ST_PROC_OUT<br>ST_IN_SERVICE<br>ST_IDLE<br>ST_NOT_ALIGNED<br>ST_IS_ALIGNED<br>ST_PROVING<br>ST_LOCAL_PROC_OUT<br>ST_REMOTE_PROC_OUT<br>ST_BOTH_PROC_OUT<br>ST_MONITORING<br>ST_CONGESTION<br>ST_POWER_OFF |
| lclBsy    | Whether the local MTP 2 is sending SIBs. Valid values:<br>TRUE<br>FALSE   |
| remBsy    | Whether the remote MTP 2 is sending SIBs. Valid values:<br>TRUE<br>FALSE  |
| fsn       | Last forward sequence number sent. Valid range is 0 through 127.  |
| bsn       | Backward sequence number to send next. Valid range is 0 through 127.  |
| flcSt     | Outbound flow control level. Valid range is 0 through 2. 2 = congested.   |

## See also

**Mtp2MgmtInit, Mtp2SetLinkCfg**

## Mtp2MgmtInit

---

Initializes internal structures and opens communication with the MTP 2 task on the TX board.

### Prototype

MTP2\_STATUS **Mtp2MgmtInit** (U8 *board*, U8 *srcEnt*, U8 *srcInst*)

| Argument       | Description         |
|----------------|---------------------|
| <i>board</i>   | TX board number.    |
| <i>srcEnt</i>  | Source entity ID.   |
| <i>srcInst</i> | Source instance ID. |

### Return values

| Return value | Description  |
|--------------|--|
| MTP2_SUCCESS |  |
| MTP2_BOARD   | Invalid board number.  |
| MTP2_DRIVER  | Low-level driver returned an error trying to open the specified dual port RAM channel. <i>srcEnt</i> should be from 0x20 through 0x31 and cannot be in use by another application. |

### Details

Call this function before any other management function. Source entity must be unique for each application accessing the MTP 2 (or other) task.

An application that opens both the management and data functions must use different entity IDs for the **Mtp2MgmtInit** and data bind request calls.

## Mtp2MgmtTerm

---

Terminates the communication with the MTP 2 task for this application.

### Prototype

MTP2\_STATUS **Mtp2MgmtTerm** (U8 *board*)

| Argument     | Description  |
|--------------|--|
| <i>board</i> | TX board number with which to terminate communication. |

### Return values

| Return value | Description   |
|--------------|---|
| MTP2_SUCCESS |   |
| MTP2_BOARD   | Invalid board number.                               |
| MTP2_HANDLE  | <b>Mtp2MgmtInit</b> not called for specified board. |

### Details

Call this function to free up resources when an application terminates or finishes communication with the MTP 2 task.

### See also

**Mtp2MgmtInit**

## Mtp2SetLinkCfg

Configures the MTP 2 task with the data link configuration values contained in the provided MTP2LinkCfg structure.

### Prototype

MTP2\_STATUS **Mtp2SetLinkCfg** (U8 *board*, MTP2LinkCfg \**pLinkCfg*)

| Argument        | Description  |
|-----------------|--|
| <i>board</i>    | TX board number on which the desired MTP 2 task resides.   |
| <i>pLinkCfg</i> | Pointer to the MTP2LinkCfg structure containing the data link configuration values. Refer to <i>Mtp2InitLinkCfg</i> on page 27 for more information. |

### Return values

| Return value | Description   |
|--------------|---|
| MTP2_SUCCESS |   |
| MTP2_BOARD   | Invalid board number.                               |
| MTP2_HANDLE  | <b>Mtp2MgmtInit</b> not called for specified board. |
| MTP2_TIMEOUT | No response from the board.                         |

### Details

This function sets the data link configuration parameters of the MTP 2 task on the specified board. Before calling this function, an application must set the field values in the MTP2LinkCfg structure in one of the following ways:

- Call **Mtp2InitLinkCfg**.
- Set each field from within the application.
- Use a combination of the two, such as calling **Mtp2InitLinkCfg** and then overriding specific fields before passing the MTP2LinkCfg structure to this function.

This function is typically called once for each configured link.

### See also

**Mtp2InitLinkCfg**, **Mtp2MgmtInit**





---

# 4

## Demonstration programs and utilities

---

### Summary of the demonstration programs and utilities

---

NMS MTP 2 provides the following demonstration programs and utilities:

| Program        | Description  |
|----------------|--|
| <i>mtp2cfg</i> | Scans the MTP 2 configuration text file and downloads the configuration to the MTP task on the TX board. |
| <i>mtp2mgr</i> | Monitors and manages the status of the MTP 2 layer.  |
| <i>mtp2app</i> | Demonstrates the use of the MTP 2 standalone task. For ANSI links.                                       |
| <i>mtp2itu</i> | Demonstrates the use of the MTP 2 standalone task. For ITU-T links.                                      |

## MTP 2 configuration: mtp2cfg

### Name

*mtp2cfg*

### Purpose

Scans the MTP 2 configuration text file and downloads the configuration to the MTP task on the TX board. *mtp2cfg* is run as part of *mtp2load* when the board is initially downloaded. You can run *mtp2cfg* after initial configuration to dynamically update some configuration parameters.

### Usage

*mtp2cfg options*

### Prerequisites

Requires a single computer with a TX board installed and the following software:

- Windows or UNIX
- Natural Access

### Procedure

To run *mtp2cfg*, enter the following command from the command line prompt:

```
mtp2cfg options
```

where **options** include:

| Option             | Description   |
|--------------------|---|
| -b <b>board</b>    | Board number to which the MTP 2 configuration is downloaded. Default = 1. |
| -f <b>filename</b> | Name and location of the MTP 2 configuration file to be downloaded.       |

*mtp2cfg* scans the information in the MTP 2 configuration file specified with the -f option and downloads the information to the task on the TX board.

### Notes

The following table shows the default assignment of ports to SS7 links for TX board configurations. There can be up to 32 links. The following nomenclature is used:

- **S1, S2, ..., Sn** refer to ports 1, 2, ..., **n** on the serial port adaptor pod (V.35 or V.24/RS232).
- **T1, T2, ..., T16** refer to the **Port1, Port2, ..., Port16** entries defined in the TDM configuration file for T1/E1/H.100/H.110 channel timeslots assigned as SS7 links.

|                               |    |    |    |    |    |    |    |    |    |     |     |     |
|-------------------------------|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| <b>Link (SAP)</b>             | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | ... | 14  | 15  |
| <b>Serial port adaptor</b>    | S1 | S2 | S3 | S4 | T1 | T2 | T3 | T4 | T5 | ... | T11 | T12 |
| <b>No serial port adaptor</b> | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | ... | T15 | T16 |

## MTP 2 link configuration parameters

The following table lists the MTP 2 link configuration parameters. All parameters are optional except LINK and the terminating END statement. If the default port assignments and link attributes are correct for your application, you do not need to run *mtp2cfg* at load time. If the default attributes are not correct for your application, create a text configuration file and run *mtp2cfg* to override these defaults.

All layer 2 times are specified in tenths of seconds (60 = 6 seconds).

| Parameter | Description   |
|-----------|---|
| LINK      | Starts link definition and assigns port to link. Valid values:<br>S1 through S4<br>T1 through T32<br>Default is none.                                     |
| LINK_TYPE | Protocol variant. Valid values:<br>ANSI (default)<br>ANSI88<br>CCITT  |
| ERR_TYPE  | Error correction method. Valid values:<br>NORMAL = normal (default)<br>PCR = preventive cyclic retransmission   |
| L2_T1     | Timer aligned and ready. Valid range is 1 through 65535. Default is 130 (ANSI) and 400 (ITU-T).   |
| L2_T2     | Timer not aligned. Valid range is 1 through 65535. Default is 115 (ANSI) and 100 (ITU-T).   |
| L2_T3     | Timer aligned. Valid range is 1 through 65535. Default is 115 (ANSI) and 15 (ITU-T).  |
| L2_T4_N   | Normal proving period. Valid range is 1 through 65535. Default is 23 (ANSI) and 82 (ITU-T).   |
| L2_T4_E   | Emergency proving period. Valid range is 1 through 65535. Default is 6 (ANSI) and 5 (ITU-T).  |
| L2_T5     | Timer sending SIB (busy indication). Valid range is 1 (default) through 65535.  |
| L2_T6     | Timer remote congestion. Valid range is 1 through 65535. Default is 60.   |
| L2_T7     | Timer excessive delay of acknowledgement. Valid range is 1 through 65535. Default is 20.  |
| L2_T10    | Amount of time MTP 2 can be isolated from a remote MTP 3 before sending SIPO (processor outage). Valid range is 1 through 65535. Default is 30.           |
| L2_T11    | Time to wait for a flow control acknowledgement from MTP 3 before sending another flow control indication. Valid range is 1 through 65535. Default is 20. |
| L2_T12    | Time to wait for a status confirmation from MTP 3 before sending another status indication. Valid range is 1 through 65535. Default is 20.                |
| L2_T13    | Time to wait for a disconnect confirmation from MTP 3 before sending another disconnect indication. Valid range is 1 through 65535. Default is 20.        |

| Parameter        | Description  |
|------------------|--|
| LSSU_LEN         | LSSU length. Valid values:<br>1<br>2 (default)   |
| MAX_FRAME        | Maximum frame length for MSU. Valid range is 64 through 1024. Default is 272.  |
| SUERM_THRESH     | Signal unit error rate monitor threshold (bad frames). Valid range is 1 through 255. Default is 64.  |
| SUERM_D_RATE     | Signal unit error rate monitor decrement rate (frames). Valid range is 1 through 65535. Default is 256.  |
| AERM_THRESH_E    | Alignment error rate monitor error rate threshold (emergency alignment). Valid range is 1 (default) through 255.                                       |
| AERM_THRESH_N    | Alignment error rate monitor error rate threshold (normal alignment). Valid range is 1 through 255. Default is 4.                                      |
| MAX_RTБ_MSGS     | Maximum number of MSUs for retransmission (only when using PCR error correction). Valid range is 1 through 255. Default is 127.                        |
| MAX_RTБ_OCTETS   | Maximum number of MSU octets for retransmission (only when using PCR error correction). Valid range is 1 through 65535. Default is 34544.              |
| MAX_PROV_ABORT   | Maximum number of proving failures. Valid range is 1 through 255. Default is 5.  |
| BAUD             | Baud rate for serial ports only (in bits per second). Valid values:<br>4800<br>9600<br>19200<br>28800<br>38400<br>48000<br>56000 (default)<br>64000    |
| INT_TYPE         | Interface type for serial ports only. Valid values:<br>DTE (default)<br>DCE  |
| DATA_ENC         | Data encoding. Valid values:<br>NRZ (default)<br>NRZI  |
| SHARE_FLAGS      | Allow single flag to be shared between frames. Valid values:<br>TRUE (default)<br>FALSE  |
| USE_FLAGS        | Use flags (TRUE) or idles (FALSE) between frames. Default is TRUE.   |
| MIN_FLAGS        | Minimum number of additional flags between frames (in addition to shared flag). Valid range is 0 (default) through 15.                                 |
| ISO_THRESH       | Number of messages queued to MTP 3 while isolated that cause MTP 2 to begin SIPOs (processor outage). Valid range is 1 through 65535. Default is 1000. |
| L2_TXQ_THRESH1   | Transmission queue length at which the outbound flow control level is raised to one. Valid range is 1 through 65535. Default is 50.                    |
| L2_TXQ_THRESH1_A | Transmission queue length at which the outbound flow control level is lowered to zero. Valid range is 1 through 65535. Default is 20.                  |

| Parameter        | Description  |
|------------------|--|
| L2_TXQ_THRESH2   | Transmission queue length at which the outbound flow control level is raised to two. The subsequent indication causes MTP 3 to cease all transmission to MTP 2 until the flow control level returns to one or zero.<br>Valid range is 1 through 65535. Default is 200.   |
| L2_TXQ_THRESH2_A | Transmission queue length at which the outbound flow control level is lowered to one. Valid range is 1 through 65535. Default is 100.  |
| L2_SAP_THRESH    | Number of messages queued to MTP 3 while inbound flow control is in effect that cause MTP 2 to send SIBs (busy indications).<br>Valid range is 1 through 65535. Default is 500.  |
| L2_SAP_THRESH_A  | Number of messages queued to MTP 3 while inbound flow control is in effect that cause MTP 2 to stop sending SIBs (busy indications).<br>Valid range is 1 through 65535. Default is 100.  |
| IDLE_FREQ        | Frequency at which FISUs are sent by the software (in ms). 0 (default) indicates that hardware constantly retransmits duplicate FISUs.<br>Non-zero frequencies can be used by switches that process all FISUs (including duplicate FISUs) in the software. Valid range is 1 through 65535.                           |
| RT_FREQ          | Frequency at which other retransmitted SUs (LSSUs) are sent by the software (in ms). 0 (default) indicates that hardware constantly retransmits duplicate LSSUs.<br>Non-zero frequencies can be used by switches that process all LSSUs (including duplicate LSSUs) in the software. Valid range is 1 through 65535. |

## Sample configuration file

The following sample is an ANSI configuration file for a two-link configuration. The sample configuration file shows all relevant parameters, even those for which the default values are not changed. A more typical configuration file would list only those parameters with values that are being changed from the defaults.

```
#
#--- MTP 2 link parameters
#=====
# Link 0 - serial (v.35)
#=====

LINK           S1           # V.35 Serial Port 1
LINK_TYPE      ANSI
ERR_TYPE       NORMAL
L2_T1          130
L2_T2          115
L2_T3          115
L2_T4_N        23
L2_T4_E        6
L2_T5          1
L2_T6          60
L2_T7          20
LSSU_LEN       2
MAX_FRAME      272
SUERM_D_RATE   256
SUERM_THRESH   64
AERM_THRESH_N  4
AERM_THRESH_E  1
MAX_PROV_ABORT 5

#--- MTP 1 link parameters
BAUD           64000
INT_TYPE       DCE
DATA_ENC       NRZ
USE_FLAGS      TRUE
MIN_FLAGS      0
SHARE_FLAGS    TRUE
END

#=====
# Link 1 - TDM (from T1/E1 daughter card)
#=====

LINK           T1           # Port1 from TDM cfg file
LINK_TYPE      ANSI
ERR_TYPE       NORMAL
L2_T1          130
L2_T2          115
L2_T3          115
L2_T4_N        23
L2_T4_E        6
L2_T5          1
L2_T6          60
L2_T7          20
LSSU_LEN       2
MAX_FRAME      272
SUERM_D_RATE   256
SUERM_THRESH   64
AERM_THRESH_N  4
AERM_THRESH_E  1
MAX_PROV_ABORT 5

#--- MTP1 link parameters
USE_FLAGS      TRUE
MIN_FLAGS      0
SHARE_FLAGS    TRUE
END
```

## MTP 2 manager: `mtp2mgr`

---

### Name

`mtp2mgr`

### Purpose

After downloading the MTP 2 configuration to the TX board with `mtp2cfg`, run `mtp2mgr` to monitor the status of the MTP 2 layer. `mtp2mgr` provides a command line interface from which statistics can be viewed.

### Usage

`mtp2mgr -b board`

### Prerequisites

Requires a single computer with a TX board installed and the following software:

- Windows or UNIX
- Natural Access

### Procedure

To run `mtp2mgr`, enter the following command at the command line prompt:

```
mtp2mgr -b board
```

where **board** is the TX board number to open.

`mtp2mgr` supports the following commands:

| Command            | Description  |
|--------------------|--|
| STATS <b>link</b>  | Retrieves the statistics for the specified <b>link</b> .               |
| STATUS <b>link</b> | Retrieves status information for the specified <b>link</b> .           |
| MTP2FLOW ON   OFF  | Deprecated.  |
| BOARD <b>board</b> | Switches to a new target board.  |
| Q                  | Quits the application.   |
| ? <b>command</b>   | Lists available commands or parameters of a specified <b>command</b> . |

The following sample shows output from *mtp2mgr* with the STATS command:

```

mtp2mgr[1]>stats link 0
Board 1, Link 0 MTP2 Stats:

inService = 1037      lclBusy   = 0          slFailAll  = 1
slFailAb   = 0        slFailAck = 0          slFailErr  = 1
slFailCong = 0        slFailAlign = 0       slNSUErr  = 0
slNegAck   = 0        nRetrans  = 0

Msg:      Rx      Tx      Msg:      Rx      Tx
----      --      --      -
SIF/SIO   3        2        MSU       417     413
LSSU*    12        12       FISU*    37984   37497
* LSSU & FISU counts do not include those filtered by the firmware

          Current  High
          -
Tx Queue:  0        270
Rtb Queue:  0         4
lTx Queue:  0       125
Sap Queue:  0         0
    
```



The following table describes the *mtp2mgr* statistics:

| <b>Statistic</b> | <b>Description</b>  |
|------------------|---|
| inService        | Amount of time (in 1/10 of a second) that the link has been in service (aligned).   |
| lclBusy          | Amount of time (in 1/10 of a second) that the link has been in congestion.  |
| sIFailAb         | Number of bad sequence numbers in received frames.  |
| sIFailAck        | Number of times an acknowledgment was not received within the T7 timeout.   |
| sIFailErr        | Number of link failures due to the SUERM threshold being reached.   |
| sIFailCong       | Number of link failures due to the T5 congestion timer expiring.  |
| sIFailAlign      | Number of alignment failures due to T2 expiration.  |
| sIFailAll        | Total number of link failures due to the link failure reasons.  |
| sINSUErr         | Not used.   |
| sINegAck         | Number of negative acknowledgments received.  |
| nRetrans         | Number of retransmissions performed.  |
| SIF/SIO<br>Tx    | Number of SIO frames sent during alignment (normal error checking).   |
| SIF/SIO<br>Rx    | Number of SIO frames received during alignment (both normal and PCR error checking).  |
| MSU<br>Tx/Rx     | Number of message signal units transmitted and received.  |
| LSSU<br>Tx/Rx    | Number of LSSUs transmitted and received. Counts do not include frames filtered by the firmware.  |
| FISU<br>Tx/Rx    | Number of FISUs transmitted and received. Counts do not include frames filtered by the firmware.  |
| Tx Queue         | Current count and high water mark of the number of messages queued on the transmit queue waiting to be sent out to the network.   |
| RtbQueue         | Current count and high water mark of the number of messages queued on the retransmission queue. These are messages that have been transmitted but not yet acknowledged by the remote side.                            |
| ITx Queue        | Current count and high water mark of the number of messages queued on the lower transmit queue. These are messages that were taken off the Tx Queue to be sent but could not be because layer 1 was busy at the time. |
| Sap<br>Queue     | Current count and high water mark of the number of messages queued on the inbound sap queue waiting to be sent to MTP 3.  |

The following sample shows output from *mtp2mgr* with the STATUS command:

```
mtp2mgr[1]>status link 0
Board 1, Link 0 MTP2 Status:

High lvl state = ENABLED   Frames out   = 1   Frames dropped = 0
Low  lvl state = IN_SERVICE

---- Current Link (0) Layer 2 configuration ----

Link Name = T1
maxFrmLen = 272      congDisc = FALSE
errType   = NORMAL  lssuLen  = 2      SUERMD rate = 256
SUERM Thrsh= 64     AERM Thrsh N= 4   AERM Thrsh E= 1

Max RTB Msg= 127     Max RTB Oct = 34544   Max Prov Abt= 5
IsoThresh  = 1000   TxqThresh1  = 50     TxqThresh1A = 20
TxqThresh2 = 200   TxqThresh2A = 100   SapThresh   = 500
SapThreshA = 100

----- Layer 2 Timers (in tenths/sec) -----
L2 t1      = 400     L2 t2      = 100     L2 t3      = 15
L2 t4 norm = 82     L2 t4 emrg = 5      L2 t5      = 1
L2 t6      = 60     L2 t7      = 20     L2 t10     = 30
L2 t11     = 20     L2 t12     = 20     L2 t13     = 20
```

The following table describes the *mtp2mgr* status information:

| Status         | Description  |
|----------------|--|
| High lvl state | Valid values are:<br>UNBOUND = Initial state<br>CONFIGURED = Link has been configured. Still unbound<br>DISABLED = Upper layer (MTP 3) has bound but not connected<br>ENABLED = Upper layer (MTP 3) has bound and connected successfully to a peer MTP 2 |
| Low lvl state  | See Low-level state values.  |
| Frames out     | Number of frames outstanding to layer 1 (transmission not yet complete by layer 1).  |
| Frames dropped | Number of transmitted frames dropped due to level 1 transmit errors.   |

**Low-level state values**

| Value             | Description                        |
|-------------------|------------------------------------|
| UNUSED            | Default to unused.                 |
| OUT_OF_SERVICE    | Link out of service.               |
| INITIAL_ALIGNMENT | Link in initial alignment.         |
| ALIGNED_READY     | Link aligned and ready.            |
| ALIGNED_NOT_READY | Link aligning and not ready.       |
| PROC_OUT          | Processor outage detected.         |
| IN_SERVICE        | Link in service - data state.      |
| IDLE              | Link idling - data state.          |
| NOT_ALIGNED       | Link not yet aligned.              |
| IS_ALIGNED        | Link is aligned.                   |
| PROVING           | Link is in initial proving phase.  |
| LOCAL_PROC_OUT    | Local processor outage - level 3.  |
| REMOTE_PROC_OUT   | Remote processor outage - level 3. |
| BOTH_PROC_OUT     | Both processors outage - level 3.  |
| MONITORING        | Link monitoring.                   |
| CONGESTION        | Link congested - level 2.          |
| POWER_OFF         | Power off state.                   |

Refer to *Mtp2InitLinkCfg* on page 27 for a description of all configuration parameters.

## Using the MTP 2 standalone task: mtp2app and mtp2itu

### Name

*mtp2app* (for ANSI links)

*mtp2itu* (for ITU-T links)

### Purpose

Demonstrate the use of the MTP 2 standalone task. Both programs exercise the MTP 2 task on the TX board.

### Usage

*mtp2app* **options**

*mtp2itu* **options**

where **options** are:

| Option          | Description  | Default ITU | Default ANSI |
|-----------------|--|-------------|--------------|
| -s <b>sio</b>   | Service information octet for sent SLTMs (signaling link test messages). | 0xb1        | 0xb2         |
| -b <b>board</b> | Board number   | 1           | 1            |
| -o <b>opc</b>   | Local point code   | 1           | 1.1.1        |
| -d <b>dpc</b>   | Adjacent point code  | 2           | 1.1.2        |

### Prerequisites

Requires a single computer with a TX board installed and the following software:

- Windows or UNIX
- Natural Access

### Procedure

To run the demonstration programs:

| Step | Action   |
|------|--|
| 1    | <p>Enter one of the following commands from the command line prompt:</p> <pre>mtp2app -b <b>board</b></pre> <pre>mtp2itu -b <b>board</b></pre> <p>where <b>board</b> is the TX board number to open.</p> <p>At startup, the demonstration programs attempt to bind to links 0 through 3. If some of these links fail, the programs still properly execute on the available links. The programs start in monitor mode, printing out to the screen whenever an MTP 2 event is received. In monitor mode, if an SLTM (signaling link test message) is received, an SLTA (signaling link test acknowledgement) is automatically transmitted. If links 0 and 1 are connected and a disconnect indication is received for one of those links, a changeover is automatically performed to the other link.</p> |

| Step                | Action  |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
|---------------------|---|---------|-------------|---------------------|---------------------------------------|-------------------|--|---------------------|---------------------------------------|---------------------|-------------------------------|------------------|------------------------------------|--------------------|--|------------------|-------------|---|--|---|-------------------------------|
| 2                   | <p>To enter command mode, press <b>Enter</b> to display the following prompt:</p> <pre>Command?&gt;</pre> <p><i>mtp2app</i> and <i>mtp2itu</i> support the following commands:</p> <table border="1" data-bbox="370 369 1214 989"> <thead> <tr> <th>Command</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>connect <i>link</i></td> <td>Connects normal mode: levels 1 and 2.</td> </tr> <tr> <td>econn <i>link</i></td> <td>Connects emergency mode: levels 1 and 2.</td> </tr> <tr> <td>powconn <i>link</i></td> <td>Places <i>link</i> in power-on state.</td> </tr> <tr> <td>disconn <i>link</i></td> <td>Disconnects the <i>link</i>.</td> </tr> <tr> <td>send <i>link</i></td> <td>Sends an SLTM on the <i>link</i>.</td> </tr> <tr> <td>status <i>link</i></td> <td>Sets status conditions on the specified <i>link</i>. Options include:<br/>Start = Start emergency state<br/>End = End emergency state<br/>Up = Local processor up<br/>Down = Local processor down</td> </tr> <tr> <td>flow <i>link</i></td> <td>Deprecated.</td> </tr> <tr> <td>q</td> <td>Quits the MTP 2 demonstration program.</td> </tr> <tr> <td>?</td> <td>Lists the supported commands.</td> </tr> </tbody> </table> <p>The following sample shows output from the MTP 2 demonstration program:</p> <pre>Command?&gt;connect 0 Link 0:Connecting Link 0:Disconnect Indication received - reason 2 Command?&gt;connect 0 Link 0:Connecting Link 0:Connect Confirmation received Link 0:Received SLTM, sending SLTA in response Link 0:Data Confirmation received acking 1 messages Command?&gt;send 0 Link 0:Sending a SLTM Link 0:Data Confirmation received acking 1 messages Link 0:Received a correct SLTA Command?&gt;q</pre> <p>The demonstration programs attempt to bind with four links. If you defined less than four links, the programs still execute properly on the existing links. However, the following alarm is visible in <i>txalarm</i> output:</p> <pre>&lt;10/05/2003 15:51:58&gt; mtp 1 13191 Internal Error [4]</pre> | Command | Description | connect <i>link</i> | Connects normal mode: levels 1 and 2. | econn <i>link</i> | Connects emergency mode: levels 1 and 2. | powconn <i>link</i> | Places <i>link</i> in power-on state. | disconn <i>link</i> | Disconnects the <i>link</i> . | send <i>link</i> | Sends an SLTM on the <i>link</i> . | status <i>link</i> | Sets status conditions on the specified <i>link</i> . Options include:<br>Start = Start emergency state<br>End = End emergency state<br>Up = Local processor up<br>Down = Local processor down | flow <i>link</i> | Deprecated. | q | Quits the MTP 2 demonstration program. | ? | Lists the supported commands. |
| Command             | Description   |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| connect <i>link</i> | Connects normal mode: levels 1 and 2.   |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| econn <i>link</i>   | Connects emergency mode: levels 1 and 2.  |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| powconn <i>link</i> | Places <i>link</i> in power-on state.   |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| disconn <i>link</i> | Disconnects the <i>link</i> .   |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| send <i>link</i>    | Sends an SLTM on the <i>link</i> .  |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| status <i>link</i>  | Sets status conditions on the specified <i>link</i> . Options include:<br>Start = Start emergency state<br>End = End emergency state<br>Up = Local processor up<br>Down = Local processor down  |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| flow <i>link</i>    | Deprecated.   |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| q                   | Quits the MTP 2 demonstration program.  |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| ?                   | Lists the supported commands.   |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |
| 3                   | Press <b>Enter</b> again to exit command mode and return to monitor mode.   |         |             |                     |                                       |                   |  |                     |                                       |                     |                               |                  |                                    |                    |  |                  |             |   |  |   |                               |



# 5

## MTP 2 message reference

### MTP 2 message summary

The following table lists the MTP 2 task messages. Messages also contain a message type and a link ID.

| Message                 | Direction       | Parameters   | Description  |
|-------------------------|-----------------|--|--|
| Bind request            | MTP3 -><br>MTP2 | Service Provider ID<br>Service User ID<br>Source Entity ID<br>Source Instance ID | Binds user application as sole user of MTP 2 link.   |
| Connect request         | MTP3 -><br>MTP2 | Service Provider ID<br>Status  | Requests MTP 2 to bring up a physical link. Use the status parameter to specify status to bring link up in (for example, normal, emergency, or power on only). |
| Connect confirmation    | MTP2 -><br>MTP3 | Service User ID  | Notifies application that link connection successfully completed (link is aligned).  |
| Data request            | MTP3 -><br>MTP2 | Service Provider ID<br>Data  | Requests MTP 2 to transmit an MSU (message signal unit) on a particular link.  |
| Data confirmation       | MTP2 -><br>MTP3 | Service User ID<br>Status<br>Data  | Sends transmitted messages that have not been acknowledged by the far exchange when the application is performing a changeover.                                |
| Data indication         | MTP2 -><br>MTP3 | Service User ID<br>Data  | Notifies the application of an incoming data packet.   |
| Disconnect request      | MTP3 -><br>MTP2 | Service Provider ID<br>Reason  | Requests MTP 2 to disconnect a link.   |
| Disconnect confirmation | MTP2 -><br>MTP3 | Service User ID  | Notifies the application that the disconnect request was received.   |
| Disconnect indication   | MTP2 -><br>MTP3 | Service User ID<br>Reason  | Notifies the application that a link disconnected.   |
| Disconnect response     | MTP3 -><br>MTP2 | Service User ID  | Notifies MTP 2 that the application received the disconnect indication.  |
| Flow control request    | MTP3 -><br>MTP2 | Service Provider ID<br>Action  | Deprecated.  |
| Flow control indication | MTP2 -><br>MTP3 | Service User ID<br>Action<br>FlcLevel  | Deprecated.  |
| Flow control response   | MTP3 -><br>MTP2 | Service User ID<br>Action<br>FlcLevel  | Deprecated.  |
| Status request          | MTP3 -><br>MTP2 | Service Provider ID<br>Action<br>Status  | Requests a special status or control function.   |

| Message             | Direction    | Parameters                          | Description  |
|---------------------|--------------|-------------------------------------|--|
| Status confirmation | MTP2 -> MTP3 | Service User ID<br>Action<br>Status | Notifies the application that the status request was received.   |
| Status indication   | MTP2 -> MTP3 | Service User ID<br>Status<br>Value  | Notifies the application that a link has gone up or come down or that level 1 buffer depletion occurred. |
| Status response     | MTP3 -> MTP2 | Service User ID<br>Status           | Confirms that MTP 3 received a status indication.  |

## Using the MTP 2 message reference

This section provides an alphabetical reference to the MTP 2 messages. A typical message includes message direction, message description, and message parameters. NMS MTP 3 uses the following conventions for data types:

- U8 (8-bit unsigned)
- S16 (16-bit signed)
- U32 (32-bit unsigned)
- Bool (8-bit unsigned)

## Bind request

### Direction

MTP 3 -> MTP 2

### Description

Establishes the sender as the user of an SS7 data link. Bind request is the first message sent from the layer 3 application to MTP 2 and is sent once for each link on a TX board that the application uses.

The following table lists the message parameters:

| Type | Field               | Description  |
|------|---------------------|--|
| U8   | Message Type        | 0x04   |
| S16  | Service Provider ID | MTP 2 reference number for the link on the TX board. Valid values are 0 through 15.  |
| S16  | Service User ID     | Application reference number for this link (typically a link number is global to all links known by layer 3). MTP 2 passes this number in all messages to the application associated with this link. |
| U8   | Source Entity ID    | Application entity ID.   |
| U8   | Source Instance ID  | Application instance ID.   |



## Connect confirmation

---

### Direction

MTP 2 -> MTP 3

### Description

Notifies the MTP 3 application that the specified link is aligned and is ready to transfer data. After sending a connect request, the application must wait for this confirmation before data can be sent over the link.

The following table lists the message parameters:

| Type | Field           | Description  |
|------|-----------------|--|
| U8   | Message Type    | 0x0D   |
| S16  | Service User ID | Application reference number for the link from <b>Bind request</b> . |

## Connect request

---

### Direction

MTP 3 -> MTP 2

### Description

Requests MTP 2 to bring up the associated link and start alignment procedures. The application must send a connect request and wait for a connect confirmation for each link from MTP 2 before data can be sent over the link.

The following table lists the message parameters:

| Type | Field               | Description   |
|------|---------------------|---|
| U8   | Message Type        | 0x0C  |
| S16  | Service Provider ID | MTP 2 reference number for the link on this TX board. Valid values are 0 through 15.  |
| U8   | Status              | The requested connection mode. Possible values are:<br>0 = Connect level 1 only<br>1 = Connect level 2 with emergency status<br>2 = Connect level 2 with normal status<br>3 = Connect levels 1 and 2 with emergency status<br>4 = Connect levels 1 and 2 with normal status<br>5 = Power on start only<br>6 = Send a start after a power on |

## Data confirmation

---

### Direction

MTP 2 -> MTP 3

### Description

Data confirmation is also sent in response to an application status request to retrieve unacknowledged messages. The status field indicates that this message is unacknowledged. It also indicates whether more unacknowledged messages are coming or whether this is the last. One data confirmation is returned for each unacknowledged message.

The following table lists the message parameters:

| Type     | Field           | Description   |
|----------|-----------------|---|
| U8       | Message Type    | 0x15  |
| S16      | Service User ID | Application reference number for the link from <b>Bind request</b> .  |
| U8       | Status          | Status of the confirmation. Valid values are:<br>0 = Unacknowledged message, more to follow.<br>1 = Unacknowledged message, last one. |
| Variable | Data            | The contents of the unacknowledged message.   |

## Data indication

---

### Direction

MTP 2 -> MTP 3

### Description

Notifies the layer 3 application of an MSU (message signal unit) received on the specified data link.

The following table lists the message parameters:

| Type     | Field           | Description  |
|----------|-----------------|--|
| U8       | Message Type    | 0x16   |
| S16      | Service User ID | Application reference number for the link from <b>Bind request</b> .   |
| Variable | Data            | Contents of the received MSU. The first byte of data is typically the SIO (service information octet), followed by the routing label, and protocol dependent data. Layer 2 strips the flags, sequence number, length indicator, and frame check sequence. The maximum length of the data field is 274 octets including the SIO. There is no field to specify the length of the data field. The application derives the length of the data from the message start and end offsets in the message. |

## Data request

---

### Direction

MTP 3 -> MTP 2

### Description

Asks MTP 2 to transmit an MSU (message signal unit) on the specified data link.

The following table lists the message parameters:

| Type     | Field               | Description  |
|----------|---------------------|--|
| U8       | Message Type        | 0x14   |
| S16      | Service Provider ID | MTP 2 reference number for the link on the TX board. Valid values are 0 through 15.  |
| Variable | Data                | The contents of the MSU to transmit. The first byte of data is typically the SIO (service information octet), followed by the routing label, and protocol dependent data. Layer 2 inserts the flags, sequence number, length indicator, and frame check sequence. The maximum length of the data field is 274 octets including the SIO. There is no field to specify the length of the data field. MTP 2 derives the length of the data from the message start and end offsets in the message. |

## Disconnect confirmation

---

### Direction

MTP 2 -> MTP 3

### Description

Notifies MTP 3 that the disconnect request was received.

The following table lists the message parameters:

| Type | Field           | Description  |
|------|-----------------|--|
| U8   | Message Type    | 0x11   |
| S16  | Service User ID | Application reference number for the link from <b>Bind request</b> . |

## Disconnect indication

---

### Direction

MTP 2 -> MTP 3

### Description

Notifies the user application that a link was disconnected.

The following table lists the message parameters:

| Type | Field           | Description   |
|------|-----------------|---|
| U8   | Message Type    | 0x12  |
| S16  | Service User ID | Application reference number for the link from <b>Bind request</b> .                |
| S16  | Reason          | An explanation of the disconnect. Possible values are:<br>0 = Layer manager request |

## Disconnect request

---

### Direction

MTP 3 -> MTP 2

### Description

Requests MTP 2 to bring down the associated link, which results in the loss of data.

The following table lists the message parameters:

| Type | Field               | Description  |
|------|---------------------|--|
| U8   | Message Type        | 0x10   |
| S16  | Service Provider ID | MTP 2 reference number for the link on the TX board. Valid values are 0 through 15.  |
| S16  | Reason              | An explanation of the disconnect. Possible values are:<br>0 = Management request (drop layer 2 only)<br>10 = Disconnect level 1 also<br>(may drop modem signals, reset transmitter and receiver) |

## Disconnect response

---

### Direction

MTP 3 -> MTP 2

### Description

Notifies MTP 2 that the disconnect indication was received. Timer T13 is stopped.

The following table lists the message parameters:

| Type | Field           | Description  |
|------|-----------------|--|
| U8   | Message Type    | 0x13   |
| S16  | Service User ID | Application reference number for the link from <b>Bind request</b> . |

## Flow control indication

---

### Direction

MTP 2 -> MTP 3

### Description

This message has been deprecated. Do not use.

The following table lists the message parameters:

| Type | Field           | Description  |
|------|-----------------|--|
| U8   | Message Type    | 0x2A   |
| S16  | Service User ID | Application reference number for the link from <b>Bind request</b> .   |
| S16  | Action          | Always DL_FLC_SETLVL.  |
| U8   | FlcLevel        | Expected values are 0, 1, or 2, where 2 indicates severe congestion and 0 indicates no congestion. NMS Communications advises applications to send normally at 0, reduce traffic at 1, and cease traffic at 2. |

## Flow control request

---

### Direction

MTP 3 -> MTP 2

### Description

This message has been deprecated. Do not use.

The following table lists the message parameters:

| Type | Field               | Description   |
|------|---------------------|---|
| U8   | Message Type        | 0x28  |
| S16  | Service Provider ID | MTP 2 reference number for the link on the TX board. Valid values are 0 through 15.         |
| S16  | Action              | The action to take. Possible values are:<br>0 = Start flow control<br>1 = Stop flow control |

## Flow control response

---

### Direction

MTP 3 -> MTP 2

### Description

This message has been deprecated. Do not use.

The following table lists the message parameters:

| Type | Field           | Description  |
|------|-----------------|--|
| U8   | Message Type    | 0x2B   |
| S16  | Service User ID | Application reference number for the link from <b>Bind request</b> . |
| S16  | Action          | Always DL_FLC_SETLVL.  |
| U8   | FlcLevel        | 0, 1, or 2, matching the flcLevel in flow control indication.        |

## Status confirmation

---

### Direction

MTP 2 -> MTP 3

### Description

Sent to the application in response to a retrieve BSN status request. If there are no unacknowledged messages to be retrieved, MTP 2 also returns a status confirmation message in response to a retrieve messages status request. If there are unacknowledged messages to be retrieved, one or more data confirmation messages are returned to the application.

All other status requests are unconfirmed.

The following table lists the message parameters:

| Type | Field           | Description  |
|------|-----------------|--|
| U8   | Message Type    | 0x79   |
| S16  | Service User ID | Application reference number for the link from <b>Bind request</b> .   |
| S16  | Action          | Action being reported in this confirmation. Possible values are:<br>1 = Response to retrieve backwards sequence number status request;<br>status field contains BSN.<br>3 = Response to retrieve messages status request;<br>no unacknowledged messages to retrieve. |
| U32  | Status          | Current value of the backwards sequence number for the link when action = 1.   |

## Status indication

---

### Direction

MTP 2 -> MTP 3

### Description

Notifies the application that a processor up or down event occurred, or that level 1 buffer depletion occurred. Level 1 buffer depletion indicates that incoming traffic exceeded the layer 3 ability to read it. The application may or may not be able to do anything to slow the incoming traffic, such as requesting the other side's layer 3 to slow down. A status response is not required for the buffer depletion indication.

The following table lists the message parameters:

| Type | Field           | Description  |
|------|-----------------|--|
| U8   | Message Type    | 0x7A   |
| S16  | Service User ID | Application reference number for the link from <b>Bind request</b> .   |
| U8   | Status          | The condition being reported. Possible values are:<br>0 = Remote processor down<br>1 = Remote processor up<br>2 = Level 1 buffer depletion |
| U8   | Value           | Buffer depletion level. Unused for processor status notifications.   |



## Status request

---

### Direction

MTP 3 -> MTP 2

### Description

Requests a special function supported by the MTP 2 layer.

The following table lists the message parameters:

| Type | Field               | Description   |
|------|---------------------|---|
| U8   | Message Type        | 0x78  |
| S16  | Service Provider ID | MTP 2 reference number for the link on the TX board. Valid values are 0 through 15.   |
| S16  | Action              | Action to take. Possible values are:<br>0 = No action; indicate change in status to MTP 2 as specified in Status field<br>1 = Retrieve backward sequence number<br>2 = Retrieve unacknowledged messages<br>3 = Drop all queued transmit messages  |
| U32  | Status              | Status change being indicated to MTP 2; relevant if action is 0 or 2.<br>When action = 0, possible values are:<br>0 = Local processor is down<br>1 = Local processor is up<br>2 = Emergency condition<br>3 = Emergency ceases<br><br>When action = 2, the only possible value is the starting sequence number, usually retrieved from an MTP3 COO or COA, which retrieves unacknowledged messages from the board. |

### Details

The MTP 2 layer supports the following special functions:

- Retrieves the current BSN (last acknowledged sequence number) for a link. This information is used for implementing changeover procedures.
- Retrieves all MSUs transmitted on a link but not yet acknowledged, starting from a particular sequence number. This information is used for implementing changeover procedures.
- Requests MTP 2 to drop all queued messages.
- Notifies MTP 2 of an emergency on or off condition on a link.
- Notifies MTP 2 of a local processor up or down condition. This information is passed on to the far exchange.

MTP 2 returns a status confirmation message containing the current BSN (last acknowledged sequence number) for a link to the application in response to a retrieve BSN status request. MTP 2 also returns a status confirmation message in response to a retrieve messages status request, but only if there are no unacknowledged messages to be retrieved. If there are unacknowledged messages to be retrieved, a data confirmation message indicating a status of unacknowledged [more | last] is sent to the application for each message. The last message indicates that it is the last. No status confirmation message is returned to the application.

## Status response

---

### Direction

MTP 3 -> MTP 2

### Description

Confirms that the application received a status indication message. Timer T12 is stopped if this contains the same action and FlcLevel as the flow control indication. Layer 3 is not required to send a status response message for a buffer depletion status indication.

The following table describes the message parameters:

| Type | Field           | Description  |
|------|-----------------|--|
| U8   | Message Type    | 0x7B   |
| S16  | Service User ID | Application reference number for the link from <b>Bind request</b> .                                       |
| U8   | Status          | The condition being reported. Possible values are:<br>0 = Remote processor down<br>1 = Remote processor up |

# Index

## A

alarms 22

## B

Bind request 56

binding 12, 56

byte order 17

## C

channel usage 15

clearing a connection 15, 59, 60, 60, 61

configuration 18

    downloading to the board 18

    link parameters 43

    MTP 2 links 18

    mtp2cfg utility 42

    TDM ports 18

Connect confirmation 57

Connect request 57

connecting 12, 57, 57

CPI library 15

cpi\_get\_data 15

cpi\_init 15

cpi\_open 15

cpi\_send 15

cpia\_get\_data 15

cpia\_send 15

## D

Data confirmation 58

Data indication 58

Data request 59

demonstration programs 41, 52

Disconnect confirmation 59

Disconnect indication 60

Disconnect request 60

Disconnect response 61

disconnecting 15, 59, 60, 60, 61

downloading to the board 18

## E

entity IDs 15

## F

flow control 61, 62, 62

Flow control indication 61

Flow control request 62

Flow control response 62

freeing resources 38

function summary 25

## I

initialization 27, 37

instance IDs 15

## L

link configuration 26, 27, 39, 43, 46

link statistics 33

link status 14, 35, 63, 64, 65, 66

## M

message format 16

messages 12

    Bind request 56

    Connect confirmation 57

    Connect request 57

    Data confirmation 58

    Data indication 58

    Data request 59

    Disconnect confirmation 59

    Disconnect indication 60

    Disconnect request 60

    Disconnect response 61

    Flow control indication 61

    Flow control request 62

- Flow control response 62
- Status confirmation 63
- Status indication 64
- Status request 65
- Status response 66
- summary 55
- MTP 2 components 9
- MTP 2 link configuration 18
- MTP 2 status 47
- MTP 2 utilities 41, 42, 47
- MTP2\_SUCCESS 25
- mtp2cfg utility 42
- Mtp2GetLinkCfg 26
- Mtp2InitLinkCfg 27
- mtp2itu demonstration program 52
- MTP2LinkCfg structure 27
- Mtp2LinkStats 33
- MTP2LinkStats structure 33
- Mtp2LinkStatus 35
- MTP2LinkStatus structure 35
- mtp2load utility 18
- Mtp2MgmtInit 37
- Mtp2MgmtTerm 38
- mtp2mgr utility 47
- Mtp2SetLinkCfg 39
- mtp3cp 1 configuration file 18
- mtp3cp2 configuration file 18
- S**
- SAPs 11
- service access points 11
- statistics 33, 47
- status 35, 47, 63, 64, 65, 66
- Status confirmation 63
- Status indication 64
- Status request 65
- Status response 66
- T**
- TDM configuration 18
- tdmcp1 configuration file 18
- tdmcp2 configuration file 18
- termination 38
- transferring data 13, 58, 58, 59
- TX driver 15
- txalarm utility 22