TAM-1001-013

DMS-100 Family

MPCDebug CC Data Analyzer User Guide

Technical Assistance Manual

BCS34 and up   Standard 01.01   July 1992

northern telecom

DMS-100 Family

MPCDebug CC Data Analyzer User Guide

Technical Assistance Manual

Publication number: TAM-1001-013
Product release: BCS34 and up
Document release: Standard 01.01
Date: July 1992

Printed in Canada and printed in the United States of America.

Information is subject to change without notice. Northern Telecom reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

Publication history

July 1992

BCS34 Standard 01.01 first release of this document

Contents

About this document

MPCDebug CC Data Analyzer adds the ability for the central control (CC) to collect and display data from a multi-protocol controller (MPC) card that would otherwise be visible only with a data analyzer. This capability is supported with the X.25, 1980, and subsequent developments.

When to use this document

Northern Telecom (NT) software releases are referred to as batch change supplements (BCS) and are identified by a number, for example, BCS29. This document is written for DMS-100 Family offices that have BCS34 and up.

More than one version of this document may exist. The version and issue are indicated throughout the document, for example, 01.01. The first two digits increase by one each time the document content is changed to support new BCS-related developments. For example, the first release of a document is 01.01, and the next release of the document in a subsequent BCS is 02.01. The second two digits increase by one each time a document is revised and rereleased for the same BCS.

To determine which version of this document applies to the BCS in your office, check the release information in *DMS-100 Family Guide to Northern Telecom Publications,* 297-1001-001.

How to identify the software in your office

The *Office Feature Record (D190)* lists your current BCS and the NT feature packages in it. You can view similar information on a MAP (maintenance and administration position) terminal by typing

```
>PATCHER;INFORM LIST;LEAVE
```

and pressing the Enter key. ↵

Where to find information

The chart below lists the documents that you require to understand the content of this document, or to perform the tasks it describes. These documents are also referred to in the appropriate places in the text.

More than one version of these documents may exist. To determine which version of a document applies to the BCS in your office, check the release information in *DMS-100 Family Guide to Northern Telecom Publications,* 297-1001-001.

Number	Title
297-1001-001	*Master Index of Practices*
297-1001-010	*Maintenance and Administration Position (MAP)*
297-1001-015	*MPC Product Guide*
297-1001-100	*System Description*
297-1001-103	*Peripheral Modules*
297-1001-129	*Input/Output System Reference Manual*
297-1001-510	*Log Report Manual*
297-1001-515	*Peripheral Modules Maintenance Reference Manual*
TAM-1001-000	*Technical Assistance Manual Index of Documents*

How commands, parameters, and responses are represented in command descriptions

Two command conventions exist:

- command expansion - representations of commands including all parameters, variables and syntactic characteristics
- command example - representations of commands as they are entered

Command expansion conventions

A command table is used for a command expansion. This table consists of the following two sections:

- the command expansion, which contains
  - all parameters
  - all variables
  - hierarchy (the order in which elements must be entered)
  - syntax
  - truncated and abbreviated forms when allowed
  - defaults
- the parameter and variable descriptions. This section follows the command expansion and contains an alphabetical listing of all parameters and variables with a description of each.

Command elements are represented exactly as they are entered, except when Italic font is used to indicate that an element is a variable name or a certain default.

Commands

The command is represented in bold type. When commands are not case-sensitive, they are in lowercase.

The command appears to the left of all other elements (parameters and variables).

When truncated or abbreviated forms of a command are allowed, they appear directly beneath the long form of the command.

Parameters

Parameters are represented in unbolded type. When parameters are not case-sensitive, they are in lowercase.

Variables

Variables are represented in italic. Italics indicates that the variable, as represented, is not entered, but replaced with an element, a value, range, number, or item from a list.

The numbers, values, ranges, and lists are described in detail for each variable in the parameters and variables description section below the expansion.

Hierarchy

The order in which command elements are entered is represented by their order of appearance, from left to right. When several elements appear in a vertical list, only one of them may be selected for that position.

Defaults

A default parameter is underlined.

The action the system takes when an element in a vertical list is not required is called a default action, and is usually an action indicated by one of the elements that can be selected. Occasionally, the default action is something different than one indicated. These non-selectable defaults are represented by the word, “default,” in italics, to indicate that it is never entered. The default is then described in the parameters and variables section.

Related groups of elements

When an element is directly followed by another element, the second element is required when the first element is selected.

To distinguish which elements relate to which, brackets surround those elements that, as a group, pertain to other elements. Only those elements that horizontally directly precede or follow the brackets are related to the elements within the brackets. When elements are not in brackets, only those elements that directly precede or follow them are related.

The following is an example of a command expansion.

bsy command parameters and variables

Command	**Parameters and variables**			
bsy	link	*ps_link*	*noforce*	*wait*
	pm		force	nowait
	unit	*unit_no*		

Parameters and variables	Description
force	This parameter overrides all other commands and states in effect on the specified units. If the whole PM is to be taken out-of-service, confirmation, yes or no, is required.
link	This parameter busies one of the P-side links specified by *ps_link.*
noforce	This parameter indicates default condition when “force” is not entered.
nowait	This parameter enables the MAP to be used for other command entries before bsy force is confirmed. Nowait is used only with force.
pm	This parameter busies both units of the peripheral module.
ps_link	This variable specifies which of the P-side links is to be busied. Range is 0 to 3.
unit	This parameter busies one unit of the PM specified by *unit_no.*
unit_no	This variable specifies which unit of the PM is to be busied. Range is 0 to 1.
wait	This parameter indicates default condition when “nowait” is not entered.

Command examples

Command examples use the same conventions as a command expansion, except that all command elements are bold and are entered just as represented. If the variable is shown with a value, it is entered exactly like a command or parameter. If the variable name is used, it is in bold italics to indicate that it is not entered as represented. The following two examples illustrate this difference.

- This is a command example containing a variable name.

  bsy link *ps_link*

  and pressing the Enter key. ↵

- This is a command example containing a variable value.

  ```
  bsy  link  2
  ```

  and pressing the Enter key. ↵

MPCDebug overview

MPCDebug CC Data Analyzer adds the ability for the central control (CC) to collect and display data from a multi-protocol controller (MPC) card that would normally be visible only with a data analyzer. It is intended as a first stop method of determining why links are having problems. This capability is supported with the X.25, 1980, and subsequent developments.

Use of MPCDebug

MPCDebug CC Data Analyzer provides a method to capture and display copies of datalink messages between an MPC and a remote node. It captures messages and records information that relates to each message. Recorded information includes the type of message, the MPC number, the link number, a timestamp, the direction, as well as some additional information that may aid in the debugging process. Once messages are captured, they may be displayed. MPCDebug is located on the central control (CC).

At DMS restart time, reinitialization removes the user from MPCDebug. Message capturing is reinitialized to off. Once MPCDebug is entered, messages may be captured and displayed by the user. This tool is intended for internal BNR lab use and for field support.

Only one user should use MPCDebug at a time. If there are multiple users, MPCDebug will be a share resource between the users. If multiple users are working with MPCDebug, each user needs to pay careful attention to each command to ensure that the command accomplishes its desired function.

MPCDebug message display

MPCDebug displays two types of message: data and SWER. At present, both data and SWER messages are captured together, and may not be captured separately. The heading of the display informs the user of the type of display and other data relating to the transmission. A data display shows the user hexadecimal code on the left side, and ASCII on the right. A SWER shows only the ASCII code. A “.” signals unreadable data. Figure 1-1 shows the format of the data display header, followed by an example in Figure 1-2. Figure 1-3 shows the format of the SWER display header, followed by an example in Figure 1-4.

Figure 1-1xxx
Data display heading

```
MSG:AAA(DATA)  MPC:BBB LINK:C  FLOW:DDD   SIZE:EE  mm/dd/yy hh:mm:ss

       AAA              Message number
       DATA             Signals data display
       BBB              MPC number
       C                Link number
       DDD              Direction that the message travelled
       EE               Size of the message
       mm/dd/yy         Date
       hh:mm:ss         Timestamp
```

Figure 1-2xxx
Data display example

```
MSG:896(DATA)   MPC:214   LINK:2   FLOW:OUT   SIZE:55   11/22/88   10:55:17
01 46 10 0F 0B 88 14 70 04 00 14 00 14 00 00 00  *.F.....p........**
00 00 00 41 4D  41 20 50 4F 4C 4C 20 43 54 52        **...AMA POLL CTR**
```

Figure 1-3xxx
SWER display heading

```
MSG:AAA(SWER)  MPC:BBB   mm/dd/yy  hh:mm:ss

       AAA              Message number
       SWER             Signals SWER display
       BBB              MPC number
       mm/dd/yy         Date
       hh:mm:ss         Timestamp
```

Figure 1-4xxx
SWER display example

```
MSG:897(SWER)   MPC:214    11/22/88   10:55:35
“TMR Q not empty, TMR_TSK, TMR_LO”
```

MPCD_DIR level commands

Use the MPCD_DIR level of the MAP to access the MPCDebug CC Data Analyzer.

Accessing the MPCD_DIR level

To access the MPCD_DIR level, enter the following from the CI level:

MPCDebug ***mpc_number***
and pressing the Enter key. ↵

This command also indicates the “path” from the CI level that is required to reach this level. The MPC number, *mpc_number*, specified does not affect the MPC numbers set by the Data Analyzer. If the system responds with a message such as

```
No datafill exists for MPC mpc_num.
```

then the MPC you have chosen does not have MPCDebug capabilities, or the MPC does not exist.

To determine what MPC numbers exist in the system, enter Table MPC and list all of the table’s contents. The commands follow:

Table MPC
and pressing the Enter key. ↵

List All
and pressing the Enter key. ↵

MPCD_DIR commands

All the commands available at the MPCD_DIR MAP level are described in this chapter. They are arranged in alphabetical order. The table below lists every command and indicates the page where its description is located.

MPCD_DIR commands

Command	**Directory**	**Page**
capture	MPCD_DIR	2-3
dealloc	MPCD_DIR	2-7
display	MPCD_DIR	2-11
query	MPCD_DIR	2-15
quit	MPCD_DIR	2-17
startmsgs	MPCD_DIR	2-19
stopmsgs	MPCD_DIR	2-23
zapdata	MPCD_DIR	2-27

capture

Function

Use the capture command to allocate a message table, and begin and end message capture.

capture command parameters and variables

Command	Parameters and variables		
capture	on off	*msg_size*	*number_ msgs*

Parameters and variables	Description
msg_size	This variable specifies the number of bytes for each message to be captured. The size may be 32, 128, or 244 bytes. The message size defaults to 32 bytes per message.
number_msgs	This variable specifies the number of messages to be stored. The number of messages can be any integer such that the total space allocated is less than or equal to 64K. The number of messages defaults to 1920 for a 32 byte message, 480 for a 128 byte message, and 250 for a 244 byte message.
off	This parameter stops the capture of messages and performs the command string query.
on	This parameter begins the capture of messages. If the message size and number of messages to be captured have been previously specified, all variables are ignored. If the message size and number of messages have not been previously specified, each is set according to the variables or default values.

Qualifications

Each captured message includes the MPC link number, the length of the message received from the MPC, and timestamp values.

The memory table is circular, and overwrites the oldest messages if the data expand beyond the table limits.

A capture command without any parameters displays the current status.

If an HX or logout occurs while the user is capturing messages, the capture continues. The capture and deallocation (dealloc) of the table must be done manually.

capture

Examples

The following table show examples of the capture command usage.

Examples of the capture command

Example **Task, response, and explanation**

capture on *32 500*
and pressing the Enter key. ↵
where

32 is the *msg_size.*
500 is the *number_msgs.*

Task: Allocate a table and collect messages.

Response:

```
Request Passed.
Message Size= 32   Message Table Size= 500
```

Explanation: The system checks for previous table allocation, and applies the previous table. Otherwise a table of 500 messages, each being 32 bytes, is created. The system begins collecting messages.

capture off
and pressing the Enter key. ↵

Task: Stop collecting messages. Issue a query command.

Response:

```
Request Passed.
Current Index= 131             Message Table Wrap=  NO
Message Size= 32               Message Table Size=  1920
Number of MSGS Captured= 131   Currently Capturing= NO
Started MPCS:
    1    3
```

Explanation: The system stops collecting messages, and issues the command string query.

Responses

The following table shows some possible responses to the capture command.

Responses for the capture command

MAP output	Meaning and action
`Error allocating Message Table.`	**Meaning:** A capture on was done. The message table was not previously allocated, and the attempt to allocate it failed. The system does not begin collecting messages. **Action:** Re-enter the command. Investigate SWERs. There must be an SOS memory problem which will be indicated by a SWER.
`Capture is currently on.`	**Meaning:** The capture command was entered without any parameters. Currently, capture is turned on. **Action:** None
`Capture was already on.`	**Meaning:** The capture command was previously specified to be on. The system is presently collecting messages. The user is also notified if the command string capture off is entered when the capture is already turned off. **Action:** None

`Second parameter ignored.`
`To change table size, you must first dealloc the table.`
`Capture is on.`
	Meaning: A capture on command was given with a table size variable. Capture was previously off. The message table was previously allocated. capture is now on, and the system is collecting messages.
	Action: If a new table size is desired, deallocate (dealloc) the present table and re-enter the command.

dealloc

Function

Use the dealloc command to deallocate the memory used to store messages.

dealloc command parameters and variables

Command	Parameters and variables
dealloc	nostop

Parameters and variables	Description
nostop	This parameter, when specified, prohibits the command string stopmsgs all from executing.

Qualifications

Unless otherwise specified, dealloc issues the command string stopmsgs all.

If an HX or logout occurs when the table is allocated, the table will not be deallocated. The table must be deallocated manually.

Examples

Examples of the dealloc command

Example	Task, response, and explanation
dealloc and pressing the Enter key. ↵	**Task:** Deallocate the memory used to store messages. **Response:** `Request Passed.` **Explanation:** The memory is deallocated, and the command string stopmsgs all was issued.

-continued-

dealloc

Examples of the dealloc command (continued)

Example **Task, response, and explanation**

dealloc nostop
and pressing the Enter key. ↵

Task: Deallocate the memory used to store messages, but do not issue the command string stopmsgs all.

Response:

```
Request Passed.
```

Explanation: The memory is deallocated, but the command string stopmsgs all was not issued.

-end-

Responses

Responses for the dealloc command

MAP output **Meaning and action**

```
There is no message table to deallocate.
```

Meaning: No memory was allocated to the message table. The command string stopmsgs all was not issued.

Action: None

```
WARNING: MPCDEBUG DATA Table Deallocation FAILED!!!
```

Meaning: There was an error in deallocating the data table.

Action: Investigate SWERs. There must be an SOS memory problem. A SWER should indicate the problem.

```
WARNING:  MPCDEBUG MSG Table Deallocation FAILED!!
```

Meaning: There was an error in deallocating the message table.

Action: Investigate SWERs. There must be an SOS memory problem. A SWER should indicate the problem.

-continued-

dealloc

Responses for the dealloc command (continued)

MAP output	Meaning and action
`WARNING:  MPCDEBUG MSG and DATA Table Deallocation Failed!!!`	
	Meaning: There was an error deallocating the message and data table. **Action:** Investigate SWERs. There must be an SOS memory problem. A SWER should indicate the problem.

-end-

display

Function

Use the display command to display captured messages in both hexadecimal and ASCII format.

display command parameters and variables

Command	**Parameters and variables**
display	*start_num* *stop_num*

Parameters and variables	**Description**
start_num	This variable indicates the message number to begin the display with.
stop_num	This variable indicates the last message number to be displayed.

Qualifications

If you are presently capturing messages, the display command issues the command string capture off. This will stop new messages from being captured while the user displays the messages already captured. To capture new messages, the command string capture on must be issued after displaying the messages.

If variables do not accompany the display command, all messages will be shown, starting with the oldest message.

If only one variable accompanies the display command, the messages shown begin at that number and continue through the most recently captured message.

If the *stop_num* is smaller than the *start_num*, the messages shown will begin with the *start_num*, continue through the most recently captured message, and then show the messages 0 through the *stop_num*.

display

Examples

Examples of the display command

Example	Task, response, and explanation
display ***896 897*** and pressing the Enter key. ↵ *where* 896 is the *start_num* 897 is the *stop_num*	
	Task: Display messages 896 through 897. **Response:** `The system will display messages 896 and 897.  Each message will be either a data message or a SWERR.` **Explanation:**896 is the starting message number. 897 is the ending number.

Responses

Responses for the display command

MAP output	Meaning and action
`There is no message table allocated.`	
	Meaning: The message table is not currently allocated. There are not any messages to display. **Action:** None
`The message table is empty.`	
	Meaning: There are not any messages in the message table. **Action:** None
`ERROR: Parameter not in current table range.`	
	Meaning: Either the first or the second variable is not in the current table range. **Action:** Perform a query command to find the table range, and then re-enter the display command.

-continued-

Responses for the display command (continued)

MAP output	Meaning and action
`<End Of Messages.>`	
	Meaning: There is not a valid message at the requested starting index. **Action:** None

-end-

query

Function

Use the query command to display the current index into the message table, whether valid messages exist after the current index, the size of the message table, the number of messages captured, and a list of MPCs in the start mode.

query command parameters and variables

Command	Parameters and variables
query	There are no parameters or variables.

Examples

Examples of the query command

Example **Task, response, and explanation**

query
and pressing the Enter key. ↵

Task: Display message table information.

Response:

```
Request Passed.
Current Index= 131            Message Table Wrap=  NO
Message Size= 32              Message Table Size=  1920
Number of MSGS Captured= 131   Currently Capturing= NO
Started MPCS:
    1    3
```

Explanation: This query command shows a current index of 131, no messages have been captured after numer 131, each message is 128 bytes, the table has space for 450 messages, messages are currently being captured, and MPCs 1 and 3 are in start mode.

quit

Function

Use the quit command to exit MPCDebug.

quit command parameters and variables

Command	Parameters and variables
quit	There are no parameters or variables

Qualifications

The quit command does not deallocate memory for the message table.

The quit command does not stop messages from being captured.

The quit command does not remove MPCs from start mode.

If memory is allocated for the table, messages are being captured, or MPCs are in start mode, the user will be warned and asked for additional confirmation prior to exiting MPCDebug with the quit command. Confirmations are shown in the Responses section.

If a logout or HX occurs in MPCDebug, the message table is not deallocated, and capture of messages continues.

Examples

Examples of the quit command

Example **Task, response, and explanation**

quit
and pressing the Enter key. ↵

Task: Exit MPCDebug.

Response:

```
Request Passed.
Current Index= 131            Message Table Wrap=  NO
Message Size= 0               Message Table Size=  0
Number of MSGS Captured= 131  Currently Capturing= NO
Started MPCS:
    NONE.
```

Explanation: The system shows the present status, and exits MPCDebug.

Responses

The following responses occur when the user must confirm a quit command.

quit

Responses for the quit command

MAP output	Meaning and action
WARNING: You are currently capturing messages.	
	Meaning: The capture of messages has not been stopped. **Action:** Type the command string stopmsgs prior to the quit command, or confirm the quit command to continue message capture.
WARNING: The message table is currently allocated.	
	Meaning: The memory for the message table has not been deallocated. **Action:** Type the command string dealloc prior to the quit command, or confirm the quit command to exit without memory deallocation.
WARNING: There are MPCs currently in the Start mode.	
	Meaning: One or more MPCs are still being monitored. **Action:** Type the command string stopmsgs prior to the quit command, or confirm the quit command to exit while monitoring MPCs.

startmsgs

Function

Use the startmsgs command to have the MPC board begin sending copies of SWERs and data messages.

startmsgs command parameters and variables

Command	Parameters and variables
startmsgs	*mpc_num*

Parameters and variables	Description
mpc_num	This variable indicates which MPC is to be monitored for incoming and outgoing messages. The value may range from 0 to 255.

Qualifications

SWERs and data messages coming in and going out on a link will be sent.

The startmsgs command does not start capturing messages unless the command string capture on was given previously and the user is still in the capture mode.

If an MPC is busied, the state of capture is not affected; however, messages cannot be captured from it. It is up to the user to issue another startmsgs command.

The startmsgs command may be issued to any MPC, and is not restricted to the MPC specified at the outset of the session.

MPC card level operations and restarts can undetectably change the status of an MPC. Therefore, the startmsgs command may have to be issued more than once to return the MPC to its desired state.

startmsgs

Examples

Examples of the startmsgs command

Example	Task, response, and explanation
startmsgs ***3*** and pressing the Enter key. ↵ *where* 3 is the *mpc_num*	**Task:** Begin monitoring MPC number 3. **Response:** `Request Passed for Data Msgs.` `Request Passed for SWERR Msgs.` **Explanation:**The startmsgs command was successful for both data and SWER messages.

Responses

Responses for the startmsgs command

MAP output	Meaning and action
`Unable to start data msgs for requested MPC.`	
	Meaning: The startmsgs command failed. **Action:** Re-enter the command.
`Unable to start SWERR msgs for requested MPC.`	
	Meaning: The startmsgs command failed. **Action:** Re-enter the command.
`Requested MPC is not in a valid state.`	
	Meaning: The requested MPC was either not in the comm_active state, or the startmsgs command failed. **Action:** None

-continued-

Responses for the startmsgs command (continued)

MAP output	Meaning and action
`Requested MPC is not datafilled.`	
	Meaning: The requested MPC was either not datafilled, or the startmsgs command failed. **Action:** None
`CC Data Analyzer not supported on MPC` *`mpc_num`*`.`	
	Meaning: The requested MPC is using a peripheral load that does not support the Data Analyzer feature. **Action:** None

-end-

stopmsgs

Function

Use the stopmsgs command to stop copies of SWERs and data messages from being sent.

stopmsgs command parameters and variables

Command	**Parameters and variables**
stopmsgs	all *mpc_num*

Parameters and variables	**Description**
all	This parameter indicates that monitoring of all the MPCs will be stopped. When this parameter is issued, not all error messages are shown. This avoids excess error messages if the stopmsgs all command fails.
mpc_num	This variable indicates the MPC number to stop monitoring. The MPC number ranges from 0-255.

Qualifications

The stopmsgs command stops SWER and data messages coming in or going out on a link.

The stopmsgs command may be issued to any MPC, and is not limited to the MPC specified at the start of the session.

MPC card level operations and restarts can undetectably change the status of an MPC. Therefore, the stopmsgs command may need to be issued more than once to return the MPC to its desired state.

stopmsgs

Examples

Examples of the stopmsgs command

Example	Task, response, and explanation
stopmsgs28 and pressing the Enter key. ↵ *where* 28 is the *mpc_num*	**Task:** Stop the monitoring of MPC number 28. **Response:** `Request Passed.` **Explanation:**The stopmsgs command was successful.

Responses

Responses for the stopmsgs command

MAP output	Meaning and action
`Unable to stop data msgs for MPC` *`mpc_num`*`.`	**Meaning:** An error occurred when sending the stopmsgs command to the requested MPC. **Action:** Re-enter the command.
`Unable to stop SWERR msgs for MPC` *`mpc_num`*`.`	**Meaning:** An error occurred when sending the stopmsgs command to the requested MPC. **Action:** Re-enter the command.
`MPC` *`mpc_num`* `is not datafilled.`	**Meaning:** The MPC requested is not datafilled. **Action:** None

-continued-

Responses for the stopmsgs command (continued)

MAP output	Meaning and action
`MPC` *`mpc_num`* `is not in a valid state.`	
	Meaning: The MPC requested is not in a comm_active state. **Action:** None
`CC Data Analyzer not supported on MPC` *`mpc_num`*`.`	
	Meaning: The requested MPC is using a peripheral load that does not support the Data Analyzer feature. **Action:** None

-end-

zapdata

Function

Use the zapdata command to clear all previously captured messages.

zapdata command parameters and variables

Command	Parameters and variables
zapdata	There are no parameters or variables.

Qualifications

While the zapdata command is being executed, the process is run locked so the command can run to completion without interruption.

Examples

Examples of the zapdata command

Example	Task, response, and explanation
zapdata and pressing the Enter key. ↵	
	Task: Clear all previously captured messages. **Response:** `Request Passed.` **Explanation:** The message table has been cleared.

Responses

Responses for the zapdata command

MAP output	Meaning and action
`The message table is empty.`	
	Meaning: The message table does not contain any messages and cannot be cleared. **Action:** None

-continued-

zapdata

Responses for the zapdata command (continued)

MAP output	Meaning and action
`ERROR:  There is No Message Table to Zap.`	**Meaning:** The message table is not allocated, and cannot be cleared. **Action:** None

-end-

MPCDebug logs

Logs and log format

MPCDEBUG CC Data Analyzer is associated with the the following logs:

- MPCDEBUG Tables allocated for 32 Byte elements.
- MPCDEBUG Tables allocated for 128 Byte elements.
- MPCDEBUG Tables allocated for 244 Byte elements.
- MPCDEBUG Tables Deallocated.
- MPCDEBUG MSG Table Deallocation Failed!!!
- MPCDEBUG DATA Table Deallocation Failed!!!
- MPCDEBUG MSG and DATA Table deallocation failed!!!

Each log report also shows the date, timestamp, log sequence number, reason code, and the maximum number of table entries. Log reports are generated for two reasons. One, when another user changes message and data table allocation, and two, when an error occurs while allocating or deallocating a table. Figure 3-1 shows the format of a log report. Figure 3-2 shows a sample log report.

Figure 3-1xxx
Log report format

```
MPC101  mmmdd  hh:mm:ss  ssdd INFO MPC_INFORMATION_REPORT
      REASON: rrr
      reasontext, nnnn
      MPC = MPCs involved

         MPC                Log name
         101                Log number
         mmmdd              Date
         hh:mm:ss           Timestamp
         ssdd               Log sequence number
         rrr                Reason number associated with the log
         reasontext         Text description of why the log was generated
         nnnn               Maximum number of table entries
```

Figure 3-2xxx
Log report example

```
MPC101  DEC20  15:08:12  0300 INFO MPC_INFORMATION_REPORT
      REASON: 593
      MPCDEBUG Tables allocated for 32 byte elements, 1920
      MPC = NONE
```

Reason 593

Log report example

A sample of the log with reason 593 follows:

```
MPC101 SEP05 18:14:33 4827 INFO MPC_INFORMATION_REPORT
    REASON: 593
    MPCDEBUG Tables allocated for 32 Byte elements, 1920
    MPC = NONE
```

Explanation

An MPCDebug user has allocated a table that captures 32 byte elements. The table holds 1920 elements.

Action to be taken

Action is only required if another user changed the setup. Otherwise, this log is for information only.

Reason 594

Log report example

A sample of the log with reason 594 follows:

```
MPC101 SEP05 18:14:33 4827 INFO MPC_INFORMATION_REPORT
    REASON: 594
    MPCDEBUG Tables allocated for 128 Byte elements, 480
    MPC = NONE
```

Explanation

An MPCDebug user has allocated a table that captures 128 byte elements. The table holds 480 elements.

Action to be taken

Action is only required if another user changed the setup. Otherwise, this log is for information only.

Reason 595

Log report example

A sample of the log with reason 595 follows:

```
MPC101 SEP05 18:14:33 4827 INFO MPC_INFORMATION_REPORT
    REASON: 595
    MPCDEBUG Tables allocated for 244 Byte elements, 250
    MPC = NONE
```

Explanation

An MPCDebug user has allocated a table that captures 244 byte elements. The table holds 250 elements.

Action to be taken

Action is only required if another user changed the setup. Otherwise, this log is for information only.

Reason 596

Log report example

A sample of the log with reason 596 follows:

```
MPC101 SEP05 18:14:33 4827 INFO MPC_INFORMATION_REPORT
    REASON: 596
    MPCDEBUG Tables Deallocated
    MPC = NONE
```

Explanation

A user has deallocated the MPCDebug tables.

Action to be taken

Action is only required if another user changed the setup. Otherwise, this log is for information only.

Reason 597

Log report example

A sample of the log with reason 597 follows:

```
MPC101 SEP05 18:14:33 4827 INFO MPC_INFORMATION_REPORT
    REASON: 597
    MPCDEBUG MSG Table Deallocation Failed!!!
    MPC = NONE
```

Explanation

The attempt to deallocate the message table failed. Data table deallocation was successful.

Action to be taken

Investigate SWERs for an SOS memory problem.

Reason 598

Log report example

A sample of the log with reason 598 follows:

```
MPC101 SEP05 18:14:33 4827 INFO MPC_INFORMATION_REPORT
    REASON: 598
    MPCDEBUG DATA Table Deallocation Failed!!!
    MPC = NONE
```

Explanation

The attempt to deallocate the data table failed. The message table was deallocated successfully.

Action to be taken

Investigate SWERs for an SOS memory problem.

Reason 599

Log report example

A sample of the log with reason 599 follows:

```
MPC101 SEP05 18:14:33 4827 INFO MPC_INFORMATION_REPORT
    REASON: 599
    MPCDEBUG MSG and DATA Table deallocation failed!!!
    MPC = NONE
```

Explanation

The attempts to deallocate both the message and data table failed.

Action to be taken

Investigate SWERs for an SOS memory problem.

List of terms

ASCII

American standard code for information interchange

American standard code for information interchange (ASCII)

Coded character set used for the interchange of information among information processing systems, communication systems, and associated equipment. ASCII defines one format in which data are exchanged between an input/output device and the device controllers of the DMS-100 Family of switches.

Batch change supplement (BCS)

A DMS-100 Family software release.

BCS

Batch change supplement

Bell-Northern Research (BNR)

Part of the tricorporate structure consisting of Bell Canada, Northern Telecom Ltd., and Bell Northern Research

BNR

Bell-Northern Research

Buffer

A storage device used to compensate for a difference in rate of data flow or time of occurrence of an event when transmitting data from one device to another.

Capture

A command used by MPCDebug to allocate a message table, and begin and end message capture.

CC

Central control

Central control (CC)

Comprises the data processing functions of the DMS-100 Family, with associated data store and program store.

Command

- A control signal.
- In man-machine language, the specification of an expected action or function by the system.

Dealloc

A command used by MPCDebug to deallocate the memory used to store captured messages.

Digital Multiplex System (DMS)

A central office switching system in which all external signals are converted to digital data and stored in assigned time slots. Switching is performed by reassigning the original time slots. DMS is a trademark of Northern Telecom.

Display

A command used by MPCDebug to show the captured messages in both hexadecimal and ASCII format.

DMS

Digital Multiplex System

HX

A command used to halt process execution.

HEX

Hexadecimal

Hexadecimal

A numerical system using base 16. HEX provides a convenient notation for 4-bit and larger binary numbers. The system uses ten numeric digits (0-9) and six alphabetic digits (A-F).

LEN

Line equipment number

Line equipment number (LEN)

A seven-digit function-reference used to identify line circuits.

Log reports

Used by DMS software to record the occurrence of all significant events, and then report the events to the operating company.

Logout

A command to exit the software tool or system.

Maintenance and administration position (MAP)

A group of components that provide a man-machine interface between operating company personnel and the DMS-100 Family systems. A MAP consists of a visual display unit and keyboard, a voice communications module, test facilities, and MAP furniture. MAP is a trademark of Northern Telecom.

MAP

Maintenance and administration position.

Memory

Location where instructions and data is stored.

Message

- The unit of information-transfer between nodes in the DMS-100 system. A message is incoming if it is sent from a peripheral to the central control, and outgoing if sent from the CC to a peripheral.
- (MSG) A type of control message used in the input/output system of the DMS-100 Family. The MSG byte specifies that the information to come is a data message.

Message table

A memory array used to store captured messages.

MPC

Multi-protocol controller

MPCD_DIR

The directory that MPCDebug resides in.

Multi-protocol controller (MPC)

A general-purpose data communication card that allows data communications between a DMS-100 Family switch and an external computer. The MPC card resides on the input/output controller shelf. The MPC card's protocol software is downloaded from the DMS-100 central processing unit and then supports software routines for data packet network communication.

Query

A command used by MPCDebug to display the current index into the message table, whether valid messages exist after the current index, the size of the the message table, the number of messages captured, and a list of MPCs in start mode.

Quit

A command used to exit MPCDebug.

SWER

Software error

Software error (SWER)

An interrupt in the software process resulting in a foreseen problem.

Startmsgs

A command used by MPCDebug to have the board begin sending copies of SWERs and data messages to the message table.

Stopmsgs

A command used by MPCDebug to have the board stop sending copies of SWERs and data messages to the message table.

SOS memory

Memory for the support operating system.

Timestamp

Output showing the time and date that the message was captured.

X.25

CCITT-defined, network layer protocol that is used in packet switching to establish, maintain, and clear virtual circuit connections between an ISDN terminal and a destination in the packet-switched network.

Zapdata

A command used by MPCDebug to clear the message table.

DMS-100 Family

MPCDebug CC Data Analyzer User Guide

Technical Assistance Manual

Public Networks
Northern Telecom
4001 E. Chapel Hill Nelson Hwy.
P.O. Box 13010
Research Triangle Park, NC 27709
(919) 481-5839

Information is subject to change without notice. Northern Telecom reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

Publication number: TAM-1001-013
Product release: BCS34 and up
Document release: Standard 01.01
Date: July 1992

Printed in Canada and printed in the United States of America.

northern
telecom