TAM-1001-003

DMS-100 Family

DISPCALL user guide

Technical Assistance Manual

BCS34 and up Standard 3.01 July 1992

northern telecom

DMS-100 Family

DISPCALL user guide

Technical Assistance Manual

Publication number: TAM-1001-003
Product release: BCS34 and up
Document release: Standard 3.01
Date: July 1992

Publication history

March 1991

BCS32 Standard

- restructured and released as standard

June 1992

BCS34 Standard 03.01

- added help command
- converted commands to software format

Revision bars in the table of contents identify the sections where technical information has been changed. Revision bars in the outside margin of a page indicate text that has been added or revised.

Contents

About this document

The DISPCALL utility captures all available data associated with the death of a call or a call that is being held for trouble analysis. The formatted output displays the Call Condense Block (CCB) and the Call Data Block (CDB). DISPCALL can also be used to capture and display data about calls in progress.

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
297-1001-100	*DMS-100 Family System Description*
TAM-1001-000	*Technical Assistance Manual Index of Documents*
297-1001-001	*Master Index of Practices*
297-1001-106	*Maintenance System Description*
297-1001-107	Maintenance and Administration Tools Description

What precautionary messages mean

Danger, warning, and caution messages in this document indicate potential risks. These messages and their meanings are listed in the following chart.

Message	Significance
DANGER	Possibility of personal injury
WARNING	Possibility of equipment damage
CAUTION	Possibility of service interruption or degradation

Examples of the precautionary messages follow.

DANGER
Risk of electrocution
The inverter contains high voltage lines. Do not open the front panel of the inverter unless fuses F1, F2, and F3 have been removed first. Until these fuses are removed, the high voltage lines inside the inverter are active, and you risk being electrocuted.

WARNING
Damage to backplane connector pins
Use light thumb pressure to align the card with the connectors. Next, use the levers to seat the card into the connectors. Failure to align the card first may result in bending of the backplane connector pins.

CAUTION
Loss of service
Subscriber service will be lost if you accidentally remove a card from the active unit of the peripheral module (PM). Before continuing, confirm that you are removing the card from the inactive unit of the PM.

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 italics. 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 pm unit	*ps_link* *unit_no*	*noforce* force	*wait* nowait

Parameters and variables	Description
force	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	busies one of the P-side links specified by *ps_link.*
noforce	indicates default condition when “force” is not entered.
nowait	enables the MAP to be used for other command entries before bsy force is confirmed. Nowait is used only with force.
pm	busies both units of the peripheral module.
ps_link	specifies which of the P-side links is to be busied. Range is 0 to 3.
unit	busies one unit of the pm specified by *unit_no.*
unit_no	specifies which unit of the pm is to be busied. Range is 0 or 1.
wait	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 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.

DISPCALL utility

This part describes the DISPCALL utility, DISPCALL uses, and restrictions and limitations.

DISPCALL access level

DISPCALL is a low level internal diagnostic tool that monitors data associated with the death of a call or with a call being held for analysis. This data includes the originating and terminating ports of the call and the software and hardware resources used on the call.

In addition to displaying data relating to calls that have died, DISPCALL captures and displays data about calls in progress.

DISPCALL displays call processing data areas such as Call Condense Blocks (CCBs), Call Data Blocks (CDBs), message buffers, extension blocks, and protected and unprotected data for call processing agents. A call processing agent is an originator or receiver of a call in the system.

DISPCALL data can be displayed in either formatted (English text) form or in hexadecimal form. The format routines can be accessed from the DISPCALL command level. For more information on the DISPCALL command level, refer to Chapter 3 on page 3-1.

Access the DISPCALL utility by entering the DISPCALL command and the appropriate parameters while in the Command Interpreter (CI) access level of the MAP. Refer to Chapter 3 for correct command syntax.

When DISPCALL is used

Data obtained with DISPCALL is used to analyze trouble conditions that cause call deaths and to analyze AUDIT log reports. Analysis of trouble conditions involves looking for missing or corrupted data on a repeating equipment unit. For such analysis it is recommended that hard copies of the DISPCALL outputs be retained.

The death of a call can occur due to any of the following reasons:

- call processing errors

- translation omissions
- network module speech errors or faults.

The death of a call can cause any of the following customer problems:

- cutoff

  Note: Cutoffs in the DMS-100 caused by network module troubles or by the death of a call are counted by the CPSUIC counter in the CP table of the Operational Measurements (OM).

- No Ringing Signal (NRS)
- Dial Tone Returns (DTR).

Restrictions and limitations

DISPCALL stops saving call data once the buffers allocated for call data are full. For information on allocating buffers, refer to SET on page 3-17.

Calculating node and terminal numbers

Terminal identifiers

A terminal is an external connection to the DMS-100, such as a line, a trunk, or a data link. DISPCALL recognizes which terminal to monitor by the terminal identifier entered by the DISPCALL user.

A terminal identifier is composed of a node number and a terminal number. A node number is a unique number assigned by the system to a node. A terminal number is a number assigned to a specific terminal attached to a node. Terminal 0 is reserved for maintenance messaging, and the remaining terminals (1 to n) are associated with individual lines, trunks, and so on.

Terminal 0

Each node has a terminal 0 that sends and receives messages specific to the peripheral processor.

When a terminal requires maintenance activity, such as Return to Service, a message indicating that maintenance is needed is sent from terminal 0 to the Central Control (CC).

When a peripheral module controller receives maintenance action commands from the MAP, such as load, busy, or test, messages are sent to terminal 0.

Calculating terminal identifiers

Before calculating internal node and terminal numbers, you must first determine the external node and terminal numbers.

To determine the Peripheral Module (PM) node number and terminal number composing a terminal identifier, use one of the following methods.

- Method 1. Enter the CI command QDN with the directory number of the line to be traced as follows:

  QDN <directory number>

  Refer to Figure 2-1 on page 2-2 for an example of the QDN command.

Figure 2-1xxx
QDN command example

```
CI:
>qdn 6213010
   ------------------------------------------------------
   DN:      6213010
   TYPE: SINGLE PARTY LINE
   SNPA: 613
   LINE EQUIPMENT NUMBER:    REM1 00 0 00 23
   LINE CLASS CODE:   1FR
   SIGNALLING TYPE:  DIGITONE
   LINE TREATMENT GROUP:     0
   LINE ATTRIBUTE INDEX:        0
   CARDCODE: 2X17AB  GND: N  PADGRPL STDLN  BNV: NL MNO: N
   PM NODE NUMBER       :  21
   PM TERMINAL NUMBER   :  27
   OPTIONS:
   DGT
   ------------------------------------------------------
```

In the preceding example, the node number is 21 and the terminal number is 27.

- Method 2. Enter the CI command QLEN with the LEN as follows:
  QLEN <len>

  Refer to Figure 2-2 on page 2-3 for an example of the QLEN command.

Figure 2-2xxx
QLEN command example

```
CI:
>qlen 1 0 11 1
-----------------------------------------------------
LEN:     HOST 01 0 11 01
TYPE: SINGLE PARTY LINE
SNPA: 613
DIRECTORY NUMBER:   6215111
LINE CLASS CODE:  1FR
SIGNALLING TYPE:  DIGITONE
LINE ATTRIBUTE INDEX:         32
CARDCODE  6X17 GND N PADGRP STDLN BNV NL MNO N
OPTIONS:
DGT
PM NODE NUMBER       :  63
PM TERMINAL NUMBER   :  354
-----------------------------------------------------
```

In the preceding example, the node number is 63 and the terminal number is 354.

The LMNT command is used the same way as the QLEN command.

- Method 3. Use QUERYPM in the PM access level to find the node number as follows:

  POST <node type> <device number>
  QUERYPM

  Refer to Figure 2-3 on page 2-4 for an example of the QUERYPM command.

Figure 2-3
QUERYPM command example

```
PM:
>post dtc 0
POST:
>querypm
    PM TYPE DTC PM NO.: 0 Int. No.: 1  Node_No.:24
PMs Equipped: 38  Loadname: DT723AY1E,CHKSUM: 018 1
WARM SWACT is supported: VALID FNAME: BTMIA01
DTC 0 is included in the REX schedule.
REX on DTC 0 has not been performed
Node Status: {OK, FALSE}
Unit 0   Act, Status: {OK, FALSE}
Unit 1 Inact, Status: {OK, FALSE}
Site Flr RPos  Bay_Id   Shf Description  Slot  EqPEC
HOST  00 C00   LTE  00  51  DTC : 000          6X02AA
```

In the preceding example, the node number is 24.

- Method 4. Use the NODENO command in PMIST to find the node number as follows:

  NODENO <node type> <device class> <device number>

  Refer to Figure 2-4 on page 2-4 for an example of the NODENO command.

Figure 2-4
NODENO subcommand example

```
PMIST MULTI USER:
>nodeno tm_node tm8 1
NODENO=16
PMIST MULTI USER:
>
```

Calculate the terminal number for Digital Trunk Controllers (DTCs) and Line Trunk Controllers (LTCs) as follows:

terminal number = (carrier * 32) + <channel> + 1

Note: Determine the carrier number, channel number, circuit number, and similar information by posting the peripheral.

Calculate the terminal number for Maintenance Trunk Modules (MTMs) and TM8s as follows:

terminal number = <channel> + 1

DISPCALL level commands

Use the DISPCALL level of the MAP to monitor data associated with the death of a call or with a call being held for analysis.

Accessing the DISPCALL level

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

dispcall

and pressing the Enter key. ↵

This command also indicates the "path" from the CI level that is required to reach this level.

The following table illustrates responses to a user's attempt to enter the DISPCALL command level.

Responses for the DISPCALL command

MAP output	Meaning and action
CAN'T ALLOCATE COMMAND DIRECTORY	
	Meaning: The system does not have enough resources to allocate a DISPCALL command directory.
	Action: Wait until the CC is less busy and try again. If the problem persists, initiate a service report.
CAN'T EXTEND THE SYMBOL TABLE	
	Meaning: The system does not have enough resources to allocate a DISPCALL symbol table.
	Action: Wait until the CC is less busy and try again. If the problem persists, initiate a service report.

End

DISPCALL commands

All of the commands available at the DISPCALL 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.

DISPCALL commands (continued)

Command		Page
help	DISPCALL	3-3
clear	DISPCALL	3-5
death	DISPCALL	3-6
disptid	DISPCALL	3-8
free	DISPCALL	3-10
query	DISPCALL	3-11
quit	DISPCALL	3-12
savelen	DISPCALL	3-13
savetid	DISPCALL	3-15
set	DISPCALL	3-17
show	DISPCALL	3-19

End

help

Function

Use the help command to access online information on DISPCALL commands.

help command parameters and variables

Command	Parameters and variables
help	*command*
h	dispcall

Parameters and variables	Description
command	displays brief information and syntax of the DISPCALL command specified. The following commands are valid entries: quit set show savetid savelen clear query death free disptid
dispcall	displays DISPCALL commands and a brief description.

End

Qualifications

None

Examples

The following table shows examples of the help command.

help

Examples of the help command

Example	Task, response, and explanation

help dispcall
and pressing the Enter key. ↵

Task: Display information about DISPCALL commands.

Response:
```
QUIT    - Quits from the DISPCALL environment
SET     - Set various parameters
SHOW    - Display saved calls
SAVETID - Save a call based on tid
SAVELEN - Save a call based on len
CLEAR   - Clear all buffers
QUERY   - Query the status of various things
DEATH   - Set call death parameters
FREE    - Free allocated buffers
DISPTID - Convert TID to CPID
```

Explanation: An explanation of each command in DISPCALL is given.

help set
and pressing the Enter key. ↵

Task: Display information about the set command.

Response:
```
Set various parameters
Parms: <TYPE>   {EXT <NUM> {0 TO 20}
                CCB <NUM> {0 TO 30}
                CDB <NUM> {0 TO 31}
                MBUFF <NUM> {0 TO 200}
                PROT <NUM> {0 TO 20}
                UNPROT <NUM> {0 TO 17}}
```

Explanation: Information on the syntax of the set command is displayed.

End

Responses

None

clear

Function

Use the clear command to clear all the buffers of their contents.

clear command parameters and variables	
Command	**Parameters and variables**
clear	
End	

Qualifications

None

Examples

The following table shows examples of the clear command.

Examples of the clear command	
Example	**Task, response, and explanation**
clear and pressing the Enter key. ↵	**Task:** Clear all data buffers. **Response:** None. No display appears on the screen. **Explanation:** Any buffers that contain information are cleared.
End	

death

Function

Use the death command to enable or disable data collection during the death of a call.

death command parameters and variables

Command	Parameters and variables
death	pup [on / off] search [on / off]

Parameters and variables	Description
pup	when set to ON, saves the Protected or Unprotected (PUP) data for the agents in the call.
search	when set to ON, searches for agents linked to the call but not appearing in any port in the call.
on	saves a call death.
off	does not save the call.

End

Qualifications

The saved data include the following items:

- CCB
- Extension blocks
- Protected and unprotected data for the agents
- CDB

Use the query command to see the current death settings.

Responses

The following table lists possible responses to the death command.

death

Responses for the death command

MAP output	Meaning and action
UNDEFINED COMMAND	**Meaning:** Something other than ON or OFF was entered as a parameter. **Action:** Verify command string and reenter.
End	

disptid

Function

Use the disptid command to convert the terminal identifier (TID) number of an agent to a call process identifier (CPID) designation (text). To determine the TID of an agent, refer to "Calculating Terminal Identifiers" on page 2-1.

disptid command parameters and variables

Command	Parameters and variables	
disptid	nodeno	termno

Parameters and variables	Description
nodeno	is the node number. Range is 0 to 4095.
termno	is the terminal number. Range is 0 to 4095.

End

Qualifications

A valid node number and a valid terminal number must be entered.

Examples

Following is a possible example of the disptid command.

Examples of the disptid command

Example	Task, response, and explanation
disptid 25 54 and pressing the Enter key. ↵ *where* 25 is the node number 54 is the terminal number.	**Task:** Display the LEN and DN associated with the given TID. **Response:** **Explanation:**

End

Responses

The following table shows possible responses to the disptid command.

Responses for the disptid command	
MAP output	**Meaning and action**
BAD TID	
	Meaning: The terminal identifier entered was not valid. **Action:** Verify terminal identifier and reenter entire command string.
End	

free

Function

Use the free command to deallocate and free all buffers.

free command parameters and variables

Command	Parameters and variables
free	

End

Qualifications

To see how many buffers are currently allocated, use the query command.

Responses

The following table shows possible responses to the free command.

Responses for the free command

MAP output	Meaning and action
UNABLE TO DEALLOCATE STORE	**Meaning:** The data for deallocating store in DISPCALL is corrupt. **Action:** Quit from DISPCALL and reenter. If the message still occurs, press the Break key and type STOP. If the message still occurs, quit from DISPCALL, logout, log back in, and reenter DISPCALL. If problem persists, initiate a service report.

End

query

Function

Use the query command to display

- the number of allocated buffers
- the number of allocated buffers currently in use
- the current settings of the death parameters.

query command parameters and variables

Command	Parameters and variables
query	
End	

Qualifications

Buffers in use are incremented from 0 to 9.

Examples

The following table shows an example of the query command.

Examples of the query command

Example	Task, response, and explanation

query and pressing the Enter key. ↵	**Task:** Display information on buffers and death parameters.
	
	Response:
	`BUFFER  ALLOCATED  IN USE`
	`CCB  10  1`
	`CDB  10  0`
	`EXT  10  0`
	`PROT  10  2`
	`UNPROT  10  2`
	`MBUFF  10  0`
	
	Explanation: This example shows that 10 of each type of buffer have been allocated and that one CCB, two PROT, and two UNPROT buffers are actually in use. It also indicates that the DEATH, SEARCH, and PUP options are not activated.

End	

quit

Function

Use the quit command to exit DISPCALL.

quit command parameters and variables	
Command	**Parameters and variables**
quit	
End	

Qualifications

All buffers are maintained even after the quit command is executed.

savelen

Function

Use the savelen command to take a snapshot of the call on which a given line is currently active. The savelen command saves the CCB, and protected and unprotected data associated with the call.

savelen command parameters and variables

Command	Parameters and variables	
savelen	*site*	len

Parameters and variables	Description
len	is the line equipment number.
site	is the location of the line; therefore, the LEN.

End

Qualifications

You must specify a desired site. Savelen does not default to the host, and is available only in offices equipped with lines.

Examples

The following table shows examples of the savelen command.

savelen

Examples of the savelen command

Example	Task, response, and explanation
savelen host 00 0 05 07 and pressing the Enter key. ↵ *where* host is the site 00 0 05 07 is the LEN.	**Task:** Specify a call by LEN 00 0 05 07. **Response:** No response appears on the screen. The current call information is stored in buffers. **Explanation:** The saved data includes the following: CCB extension blocks protected and unprotected data for the agents (if the Death Pup On command is processed).
End	

savetid

Function

Use the savetid command to take a snapshot of the call. The call is specified by supplying the TID of an agent or terminal in the path of the call. The savetid command saves the CCB, protected and unprotected data associated with the call.

savetid command parameters and variables

Command	Parameters and variables	
savetid	nodeno	termno

Parameters and variables	Description
nodeno	is the node number. The range is 0 to 4095.
termno	is the terminal number. The range is 0 to 4096.

End

Qualifications

A valid node number and a valid terminal number must be entered.

Examples

The following table shows examples of the savetid command.

savetid

Examples of the savetid command

Example	Task, response, and explanation
savetid 03 02 and pressing the Enter key. ↵ *where* 03 is the node number 02 is the terminal number	**Task:** This command displays information saved from node 03, terminal 02. **Response:** No response. The current call information is stored in the current buffer space and can be displayed by using the show command. **Explanation:** The saved data includes the following information: CCB Extension blocks Protected and unprotected data for the agents (if the PUP is enabled by using the death command).

End

Responses

The following table shows possible responses to the savetid command.

Responses for the savetid command

MAP output	Meaning and action
BAD TID	**Meaning:** The terminal identifier entered is not valid. **Action:** Verify terminal identifier and reenter entire command string.

End

set

Function

Use the set command to allocate the buffers needed to save the call data.

set command parameters and variables

Command	Parameters and variables	
set	*type*	*num*

Parameters and variables	Description
num	is the number of buffers to allocate and has the following ranges per bufer type: 0 to 30 for *ccb* 0 to 31 for *cdb* 0 to 600 for *mbuff* 0 to 34 for *ext* 0 to 20 for *prot* 0 to 17 for *unprot*
type	is the type of buffer. Acceptable types include the following: *ccb* Call condense blocks *cdb* Call data blocks *mbuff* Message buffers (currently unused) *ext* Extension blocks *prot* Protected terminal table *unprot* Unprotected terminal table

End

Qualifications

None

Examples

The following table shows examples of the set command.

set

Examples of the set command

Example	Task, response, and explanation
set ccb 10 and pressing the Enter key. ↵ *where* *ccb* is the type of buffer *10* is the number of ccb buffers to allocate.	**Task:** Allocate 10 CCB buffers. **Response:** **Explanation:**

End

Responses

The following table shows possible responses to the set command.

Responses for the set command

MAP output	Meaning and action
UNABLE TO ALLOCATE STORE	**Meaning:** The system is unable to allocate the buffer designated in the command string. **Action:** Wait and try again when the CC is less busy.

End

show

Function

Use the show command to display the collected data.

show command parameters and variables

Command	Parameters and variables		
show	*what*	*which*	*format*

Parameters and variables	Description
format	indicates the format to use when dumping the DATA. Acceptable formats include: h hex f formatted
what	is the type of buffer to be displayed. Acceptable types include the following: *call* all buffers containing data belonging to a particular call. *ccb* Call condense blocks *cdb* Call data blocks *prot* Protected terminal table *ext* Extension blocks *unprot* Unprotected terminal table *P1P* port 1 permanent data for the given CCB *P2P* port 2 permanent data for the given CCB
which	indicates the buffer number to be displayed. Buffers are filled from 0 upward. The range is 0 to 255. The query command indicates the number of buffers used.

End

Qualifications

None

Examples

The following table shows examples of the show command.

show

Examples of the show command

Example	Task, response, and explanation
show cdb 0 f and pressing the Enter key. ↵ *where* *cdb* is the *what* variable and specifies the type of buffer to be displayed. *0* is the *which* variable and indicates the buffer number to be displayed. *f* is the format variable and indicates the format to use when dumping the DATA.	**Task:** Display the data associated with the CDB 0 buffer in a formatted form. **Response:** (See Figure 3-1 on page 3-21 for the response to the show cdb 0 f command.) **Explanation:**

End

Responses

The following table shows possible responses to the show command.

Responses for the show command

MAP output	Meaning and action
NOT IN USE	**Meaning:** The buffer identified in the command string is not being used. **Action:** Determine which buffer you wish to display (one that is in use) and reenter the command.
OUT OF RANGE	**Meaning:** The number of the buffer identified in the command string is not within the valid number range (0 to 255). **Action:** Determine which buffer you wish to display, and reenter the command with a valid buffer number.

End

Figure 3-1
Example of show cdb 0 f command

```
SHOW CDB 0 F
>
** CDB Dump (0) for CALLID 426908 saved at 1987/02/04
CPMB:
   LINK/CPTLBPTR = 2C0D90
   MAILBOX:
      MBQLINK = FFFFFF MBPROCQD = N CLONED = N MBINDEX = 0
      MBLETTERCOUN = 0002 MBPROCPTR = 37E9E2  MBLETTERQ = FFFFFF
STATUS = DISPLAYING
MESSAGE:
   MT = 0016 - DIGITS_MSG
   BODY:
      4127    0999    FF0F    7FFF    8987    7000    CCCC   CCCC  CCCC   CCCC
      CCCC   CCCC   CCCC  CCCC   CCCC  CCCC   CCCC   CCCC  CCCC   CCCC
      CCCC   CCCC   CCCC  CCCC   CCCC
   TID =NODE_NO = 011  TRMNL_NO_MSN = 0 TRMNL_NO_LSB = 41
TEMPTIMESTAMP = 036 BEB5 NETWORK_CONNECTION_MADE = N NCCLS
= 0
CDBRTEB:
   GROUP_CPID         CKT              TTT     1
   TERMINAL_CPID     CKT              TTT     1
   CONNECTION_TYPE=D REROUTE_ELEMENT  =N CST_L                  =N
   CST_H                =N  RECREATE_CDBRTEB=N ERWT_ACCEPTED = N
   WAIT_FOR_COT_TO =N GRPSC                  =Y
FAST  PROC = 0
FF
```

Collecting and displaying call data

Several methods exist for collecting and displaying call processing data information. These methods and the procedures associated with them are described in this part.

Recommendations for operation

Following are recommendations for displaying accurate and complete DISPCALL output.

1 Before using DISPCALL to analyze a call death, perform a Translation Verification (TRAVER) on the called number to verify that the translation information is complete.

2 Use the SET command to allocate ten buffers for each buffer type. For example, to allocate ten CCB buffers, type `SET CCB 10`. Repeat this command for CDB, EXT, PROT, UNPROT, and MBUFF buffers. Refer to SET on page 3-17 for more information on the SET command.

3 Set the DEATH option ON, in order to run DISPCALL continuously. While DEATH is ON, every call that dies causes the appropriate data to be captured and displayed, thus allowing rapid data analysis. Refer to DEATH on page 3-6 for more information on the DEATH command.

4 Perform QUERY periodically to determine the number of buffers in use. Refer to QUERY on page 3-11 for more information on the QUERY command.

5 The call to be analyzed must be in progress before the SAVETID or SAVELEN commands can be issued.

6 Perform a SHOW of all in-use buffers to obtain a hard copy of the data. Retain the hard copies for trouble analysis. Specifically, the SHOW CALL command with the FORMAT option displays notification of a call death, including information such as calling and called numbers, route data, equal access data, and whether or not a feature was used. Refer to SHOW on page 3-19 for more information on the SHOW command.

7 Periodically CLEAR the buffers to make room for storing more call data. Refer to CLEAR on page 3-5 for more information on the CLEAR command.

8 If a Network is running in simplex, loss of integrity can cause a call to drop. In this instance analyzing a call death at the integrity analysis (NETINTEG) sublevel of the Network (NET) level of the MAP can aid in troubleshooting. The NETINTEG sublevel provides tools for troubleshooting integrity problems.

9 Buffers are shared in DISPCALL; therefore, if more than a single map is using DISPCALL simultaneously, the buffer space is shared, making the task of keeping track of your call more difficult.

10 Buffers are not cleared when you quit DISPCALL. It is possible to retrieve a call's information if you exited a session accidentally or DISPCALL was aborted.

Collecting data when a call dies

To collect and display call data when a call dies, perform the following steps:

1 To access DISPCALL type:

DISPCALL

The system returns the prompt 'DISPCALL >>'.

2 Ensure that buffers are available to hold call information by typing `QUERY DISPCALL`.

The display should be as follows:

BUFFER	ALLOCATED	IN USE
CCB	10	0
CDB	10	0
EXT	10	0
PROT	10	0
UNPROT	10	0
MBUFF	10	0

If 0 appears in the ALLOCATED column, type `SET type num`, where the type of buffer is CCB, CDB, MBUFF, EXT, PROT, or UNPROT, and the number of buffers is 10. For example, to initialize ten CCB buffers, type `SET CCB 10`. For more information on the SET command, refer to SET on page 3-17.

3 To enable data collection, type `DEATH ON`.

Processing this command causes DISPCALL to run continuously (until DEATH OFF is entered). Whenever a call dies, the relevant call processing information is captured.

4 While the DEATH option is ON, do the following:

a. Periodically determine how many buffers are in use by entering `QUERY DISPCALL`.

b. Display all the information associated with the call by entering `SHOW CALL buffer format`.

For example, to display the data contained in the first buffer in a formatted form, type `SHOW CALL 0 F`.

To obtain a hard copy of the data of each in-use buffers, enter `SHOW type buffer format` for each buffer, where the type of buffer is one of CCB, CDB, MBUFF, EXT, PROT, or UNPROT. For example, to display the data associated with the first CCB buffer in a formatted form, type `SHOW CCB 0 F`. For information on the SHOW command, refer to SHOW on page 3-19.

Retain the hard copies for trouble analysis.

c. Periodically clear the buffers in order to make room for storing more failures, by entering `CLEAR`.

5 To disable data collection, type `DEATH OFF`.

6 To exit DISPCALL type `QUIT`.

Collecting data for a call specified by terminal identifier

To collect and display the data associated with an active call specified by a given Terminal Identifier (TID), perform the following steps:

1 To access DISPCALL type `DISPCALL`.

The system returns the prompt 'DISPCALL >>'.

2 Ensure that buffers are available to hold call information by typing `QUERY DISPCALL`.

The display should be as follows:

BUFFER	ALLOCATED	IN USE
CCB	10	0
CDB	10	0
EXT	10	0
PROT	10	0
UNPROT	10	0
MBUFF	10	0

If 0 appears in the ALLOCATED column, type `SET type num`, where the type of buffer is CCB, CDB, MBUFF, EXT, PROT, or UNPROT, and the number of buffers is 10. For example, to initialize ten CCB buffers,

type `SET CCB 10`. For more information on the SET command, refer to SET on page 3-17.

3 To save the CCB, and protected and unprotected data for the specified node and terminal number, type `SAVETID nodeno termno`.

Note: The call must be up before the SAVETID command is processed.

For example, to specify a call on node 03 and terminal 02, type `SAVETID 03 02`. For more information on the SAVETID command, refer to SAVETID on page 3-15.

4 Display all the information associated with the call by entering `SHOW CALL buffer format`.

For example, to display the data contained in the first buffer in a formatted form, type `SHOW CALL 0 F`.

To display the collected information for each buffer type, enter `SHOW type buffer format`.

For example, to display the data associated with the first CCB buffer in a formatted form, type `SHOW CCB 0 F`. For information on the SHOW command, refer to SHOW on page 3-19.

5 To exit DISPCALL type `QUIT`.

Collecting data for a call specified by line equipment number

To collect and display the data associated with an active call specified by the Line Equipment Number (LEN), perform the following steps:

1 To access DISPCALL type `DISPCALL`.

The system returns the prompt 'DISPCALL >>'.

2 Ensure that buffers are available to hold call information by typing `QUERY DISPCALL`.

The display should be as follows:

BUFFER	ALLOCATED	IN USE
CCB	10	0
CDB	10	0
EXT	10	0
PROT	10	0
UNPROT	10	0
MBUFF	10	0

If 0 appears in the ALLOCATED column, type `SET type num`, where the type of buffer is CCB, CDB, MBUFF, EXT, PROT, or UNPROT, and the number of buffers is 10. For example, to initialize ten CCB buffers,

type `SET CCB 10`. For more information on the SET command, refer to SET on page 3-17.

3 To save the CCB, and protected and unprotected data for a specified LEN, type `SAVELEN len`.

Note: The call must be up before the SAVELEN command is processed.

Note: A site must be specified in the SAVELEN parameter list. In the above example `len` includes site in this context.

For example, to specify a call by LEN 00 0 05 07, type `SAVELEN HOST 0000507`. For more information on the SAVELEN command, refer to SAVELEN on page 3-13.

4 Display all the information associated with the call by entering `SHOW CALL buffer format`.

For example, to display the data contained in the first buffer in a formatted form, type `SHOW CALL 0 F`.

To display the collected information for each buffer type, enter `SHOW type buffer format`.

For example, to display the data associated with the first CCB buffer in a formatted form, type `SHOW CCB 0 F`. For information on the SHOW command, refer to SHOW on page 3-19.

5 To exit DISPCALL type `QUIT`.

Interpreting DISPCALL output

This part provides instructions for troubleshooting a call death using DISPCALL, log reports, and SWERRs.

Log reports and SWERRs

DISPCALL can be used in association with LOGUTIL to troubleshoot the source of a call death. In this part, the sample call death is a valid DN dialing a maintenance circuit. For this example, Log Reports AUD398, AUD395, AUD433, and AUD103 are generated. These log reports and the associated SWERR are shown in Figure 5-1 on page 5-2.

Figure 5-1xxx
Log reports for call death

```
AUD398 OCT14 16:41:35 7622 INFO CDB DUMP
   CALLID:      820001
   4B00 1E00 FFFF FFFF FD80 0002 A5F1 3700 FFFF FFFF
   FDFD FDFD FD88 0016 A127 0AAA 550F 7055 9887 BE04
   8540 8E02 9C71 8190 9090 9090 9080 9090 8080 9090
   0289 0202 0202 0202 0202 0202 0202 0202 0202 0011
   0041 23BA 0024 0000 FDE0 0030 1B00 0030 1B00 FD00
   FDFD 0000 FDFD FDFD FDFD FDFD FDFD FDFD FDFD FDFD
AUD395 OCT14 16:41:35 7723 INFO CCB DUMP
   CALLID:      820001
   442C 1F00 0321 C203 FFFF FFFF FFFF FFFF FFFF FFFF
   FFFF 0000 FFFF FFFF 01D9 0000 00C1 1C00 0008 0410
   3C41 2400 056F 5780 022F 0A29 0011 0041 01C1 7100
   01C0 7100 A166 7100 FE00 FFFF FFFF FFFF 086B 0000
   0000 0000 0000 7F81 0400 040A 0000 0000 B337 6C00
   FFFF FFFF FFFF FFFF FFFF FFFF FF07 FFFF 0895 0000
   3803 0C07 0000 0000 0000 2800 A127 FAAA FFFF FFFF
   47FF FD00 8001 FFFF FFFF FFFF 19FF 2000 0000 0000
   0000 0000 0000 0000 0000 0000 0000 0400 0001 0017
   01D0 0000 0000 0000 0000 0000 00C8 1A28 4005 1000
   00B0 00B0 E000 0080 0040 03E7 0000 0000 0000 0000
   0000 0000 0000 0000 0002 22C2 0024 0000 0000 C0000
AUD433 OCT14 16:41:35 7824 INFO LINKED TID
   LEN HOST 02 0 02 00   DN 7210061
   CALLID:      820001  NODE NO: 17 TERMINAL NO: 65

AUD103 OCT14 16:41:35 4815 INFO CP OBITUARY
   DEATHTYPE:  SUICIDE  PROC: A140  2008  REASON:  0
   CALLID:       820001

SWERR   OCT14 16:41:35 8125 MISC
   REASON=0029, PROCID=#A140 #C003: CALLP, TEXT= 820001
   10F100=CPTABUI.EO19:CALLERRO+#000E
   10F100=CPTABUI.EO19:NIL_PROC+#0019
   853F62=LSETPRCI.ZA01:LINE_SETUP_P#1568
   84EFF3=LNSTART.ZA01:LINE_STA+#0561
   15D8B8=CALLP.FP08:CALL_PRO+#0036
   01BE85=MODULES.ZA:INITIALIZEP+#0009
   00D965=PROCS.DT12:LIVEANDD+#0007
```

The call identifier in field CALLID of the log reports is 820001, which indicates that these log reports are associated with the same call. The identifier given in field TEXT of the SWERR indicates that this SWERR is also associated with the same call death.

The SWERR in Figure 5-1 on page 5-2 provides a traceback of the procedures called prior to the call death. In this example, the procedures called prior to the call death indicate a normal call progression until

procedure LINE_SETUP_PROCESSOR calls NIL_PROCESSOR. This procedure call indicates that the line setup processor cannot determine how to route the call to the terminating agent. This information indicates that the terminating agent is invalid. Therefore, procedure NIL_PROCESSOR calls CALLERROR, which causes the call to die.

Log Report AUD103 is generated when a process has been terminated. The process identifier shown in field PROC of AUD103 is the same process identifier given in field PROCID of the SWERR. This association verifies that the procedure call to CALLERROR is the source of the call death.

Log Report AUD433 identifies the node and terminal number of the calling DN. This information can be used with the SAVETID command, if troubleshooting requires that other calls be made from the originating DN, or with other tools, such as PMIST.

Log Report AUD395 and AUD398 can be used in conjunction with DISPCALL.

In Figure 5-1 on page 5-2, the call identifier (CALLID) referred to in the log reports is 820001. This CALLID also identifies call information displayed with DISPCALL.

If the user processes the SHOW CALL command in DISPCALL and specifies H as the formatting parameter (refer to SHOW on page 3-19), the user displays the call death in a format nearly identical to Log Reports AUD395 and AUD398. Refer to Figure 5-2 on page 5-4 for the hexadecimal output for CALLID 820001.

Since the hexadecimal dump of the CCB and the CDB is not easy to interpret, the DISPCALL user can dump the information on a call in symbolic form. Refer to Figure 5-3 on page 5-5 for the formatted output for CALLID 820001.

Notice that the Call Data Block Routing Block (CDBRTEB) in Figure 5-3 on page 5-5 contains the fields GROUP_CPID and TERMINAL_CPID, which verify that the terminating agent was a Transmission Test Trunk (TTT), which is an invalid terminating agent.

Figure 5-2xxx
DISPCALL output with hexadecimal format

```
DISPCALL>>
>>>query
   BUFFER  ALLOCATED  IN USE
    CCB        10        1
    CDB        10        1
    EXT        10        0
    PROT       10        1
    UNPROT     10        1
    MBUFF      10        0
 DEATH ON  SEARCH ON  PUP ON
>>> show call 0 h
**
** Call Data Dump (0) for CALLID 820001 saved at 1987/10/14
                                                16:41:35.766 WED.
**
** CCB Dump (0) for CALLID 820001 saved at 1987/10/14
                                                16:41:35.766 WED.
  442C  1F00  0321  C203  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF
  FFFF  0000  FFFF  FFFF  01D9  0000  00C1  1C00  0008  0410
  3C41  2400  056F  5780  022F  0A29  0011  0041  01C1  7100
  01C0  7100  A166  7100  FE00  FFFF  FFFF  FFFF  086B  0000
  0000  0000  0000  7F81  0400  040A  0000  0000  B337  6C00
  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FF07  FFFF  0895  0000
  3803  0C07  0000  0000  0000  2800  A127  FAAA  FFFF  FFFF
  47FF  FD00  8001  FFFF  FFFF  FFFF  19FF  2000  0000  0000
  0000  0000  0000  0000  0000  0000  0000  0400  0001  0017
  01D0  0000  0000  0000  0000  0000  00C8  1A28  4005  1000
  00B0  00B0  E000  0080  0040  03E7  0000  0000  0000  0000
  0000  0000  0000  0000  0002  22C2  0024  0000  0000  C0000
** CDB Dump (0) for CALLID 820001 saved at 1987/10/14
                                                16:41:35.766 WED.
  4B00  1E00  FFFF  FFFF  FD80  0002  A5F1  3700  FFFF  FFFF
  FDFD  FDFD  FD88  0016  A127  0AAA  550F  7055  9887  BE04
  8540  8E02  9C71  8190  9090  9090  9080  9090  8080  9090
  0289  0202  0202  0202  0202  0202  0202  0202  0202  0011
  0041  23BA  0024  0000  FDE0  0030  1B00  0030  1B00  FD00
  FDFD  0000  FDFD  FDFD  FDFD  FDFD  FDFD  FDFD  FDFD  FDFD
```

Figure 5-3xxx
Formatted DISPCALL output (part 1 of 3)

```
>>>dispcall
DISPCALL>>
>>>query dispcall
BUFFER     ALLOCATE  IN USE
 CCB          10        0
 CDB          10        0
 EXT          10        0
 PROT         10        0
 UNPROT       10        0
 MBUFF        10        0
DEATH ON   SEARCH ON    PUP ON
% make call
>>>query dispcall
BUFFER    ALLOCATED  IN USE
 CCB          10        1
 CDB          10        1
 EXT          10        0
 PROT         10        0
 UNPROT       10        1
 MBUFF        10        1
DEATH ON   SEARCH ON    PUP ON
>>>show call 0 F
**
** Call Data Dump (0) for CALLID 820001 saved at 1987/10/14
                                          16:41:35.766 WED.**
** CCB Dump (0) for CALLID 820001 saved at 1987/10/14
                                          16:41:35.766 WED.**
CPTLB :
 LINK/CPMBPTR=1F442C MYINDEX        =21 06  PROCQD    =N
 STATE       =UNAVAI AUDIT          =  0000 LINKCOUNT =0002
 LETTERCOUNT  =   0003 LETTERC      =  C203 WAKEID    =FFFF
 LETTERQ     =FFFFFF CCBTIMEQ.SUCC =FFFFFF CCBTIMEQ.PREV=
                                                    FFFFFF
 FASTECCBINDEX =  0000 UP_OVER_WARM  = N   ECCBINDEX  =00 00
EXTPTR  = FFFFFF  SEQNO   = 0019  CS   = TRANSI  XBITS =0000
CMI     =  0000  CTRLPORT  =   0000  CCBFC  =  0000
RECEIVER    = CKT      RCVRDGT    3
CCBFA :
 0008  0410
PORT1PERM :
  AGENT  = LEN HOST 02 0 02 00  DN 7210061
 PATHEND :
  CHNL = 000F PORT = 002B NM_PAIR = 0000 NM_CHNL  = 056F
                                              LOGICAL = 0000
  GAIN = 0000 LOSS = 0003 INTEG_VAL = 000AF PREFERRED_PL =
                                                EVEN_PLANE
 PMCHNL =   22F  PORT = 01   CFWBIT   = N  AGENT_SUSPECT = N
 THREAD = 0029  FMTCODE = 05  UTR_AVAILABLE = N
 TID : NODE_NO = 011 TRMNL_NO_MSN = 0 TRMNL_NO_LSB = 41 TSI=0
 FMTAREA:
 01C1   7100  01C0  7100  A166  7100  FE00  FFFF  FFFF  FFFF
 MBUFFPTR = 00086B
PORT2PERM :
  AGENT  = <NIL>
 PATHEND :
```

Figure 5-4xxx
Formattted DISPCALL output (part 2 of 3)

```
  CHNL = 0000 PORT = 0000 NM_PAIR = 0000 NM_CHNL  = 0000
                                                  LOGICAL = 0001
  GAIN = 0000 LOSS = 0000 INTEG_VAL = 00FF PREFERRED_PL =
                                                    EVEN_PLANE
 PMCHNL =  400  PORT  = 02   CFWBIT   = N  AGENT_SUSPECT = N
 THREAD = 000A  FMTCODE = 02  UTR_AVAILABLE = N
 TID : NODE_NO = 000 TRMNL_NO_MSN = 0 TRMNL_NO_LSB = 00 TSI=0
 FMTAREA:
  B337  6C00  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FF07  FFFF
  MBUFFPTR = 000895
CHB:
 XLAB:
  RC          = XLA_RO TXSEL     = 0000 TX_TRMT     =
  IBN_TRMT_SET= N      MIN_DIGITS= 0007 MAX_DIGITS  = 0007
  PREFIX_FENCE=  0000  DATA_VER  = N    OC           = LCL
  TRAN_SYS    = NA     TYPECALL  = NP   NPA_ADDED    = N
  TX_POS      = NONE   TXROUTE/TERM_AGENT  = DNT_RTER 000000
  CALLED_DR   =        7210000
  RTE         = Y      POS       = N    RETRANSLAT  = N
  CALLING_DR/AUTH_CODE_DR =
  SCREENING_IN= N      CAN_LCS   = N    CAN_SCRN    = N
  XLASTAGE    = BALANC ADP       = Y    BLK_OVLP    = Y
  LONGHAUL    = N      HTRP      = 0    SNPA        = 919
  XLT_FROM    =      0 HTRC      = N    VALID_/PRIVL= N
 ANI_INFO :
    ONI = N    HOT = N   TDN = N     COIN = N
 RSP = N ANI_FAIL = N SPARE2 = N   SPARE3 = N
 AMADATA :
  0000  0000  0000  0000  0000  0000  0000  0000  0000
 SOURCEPARMS :
  VALID_SOURCEPARMS  = N   NCOS    = 0000   DESTIN
                                                  = NIL_DE
  CUSTGRP            =     SUBGRP  = 0000   CALL_CHARACTER
                                                  =  0000
  DGCOLL_TABLE       = 0000SOURCE  = NIL_SO OWAT_ZONE
                                                  =  0000
  DGCOLL_TABLE       = 0000SOURCE  = NIL_SO OWAT_ZONE
                                                  =  0000
  SOURCE_TRC        = 0000 SMDR       = N SMDRB       = N
  ACR               = N    INTRAGROUP  = N ENABLE_CRL   = N
  CRL_REQUIRED      = N    ATTDNDOV    = N
  CALLED_DR_SHIFTED = N    TDN         = N DOD_DESTIN   = N
  GROUP_INTERCOM    = N    ATTX        = N LINE_HAS_LNR = N
RTEB:
  ROUTE          = OFC_RT               = RTR_ME RTE_TYPE
                                                  = NATRK
  ROUTE_CHAIN    = IN_CHA PREV_SATELLITE   = NCHOICE= DIRECT
  OHQT_APPLIED   = N      QUEUEING_ACTIVE  = NOHQ    = N
  CANCSTDL       = N      INHIBIT_QUEUEING = NHUNT   = Y

TRMT      = UNDT  POS     = 0000    TFR      = NIL_TF
ANI_SPILL = N     ANI_SPILL_9 = N   RECORDING_REQD = N
TS_OMREG      = N  SEIZE_FAILURE   = N   AC_FTR_IN_EFFECT = N
```

Figure 5-5xxx
Formatted DISPCALL output (part 3 of 3)

```
RESELECT      = N   DIST_RING        = N
FTR_IN_EFFECT = N
LCO_CALL      = N   CHARGE_TREATMENT = CHGA
FORCE_CC_TIMI = N   AC_ATTACHED      = N
EAEO_DATA:
 EA_CARRIER  = NILC     EA_LOCAL = N    EA_OCS_NEEDED = N
 CALL_EVENT  = INITIAL_EVENT:

OVLPD :
 REALCMI = 0  DRCOUNT = 0  OLSTATE = OLFIRST
PSTATE : PROCESSOR = CROSS_PRCR STATE = 0000
MBI       = 0000 CHARGE   = N     DATA_CALL = N TIMESTAMP  =
                                                    0024 22C2
ORIGDISP  = 0000 TERMDISP = 0000  OCC_CALL  = N OCC_INCOM  = N
NO_INTRAS= N    EA_CALL  = N     PIC_CALL  = N LATA_CALL  =
                                                   INTRALATA
STATE_CAL=INTRAS TRD_TIMIN = NO_TRD TPS_CP = N  NSC_CALL = N
** CDB Dump (0) for CALLID 820001 saved at 1987/10/14
                                          16:41:35.766 WED. **
CPMB:
 LINK/CPTLBPTR = 1E4B00
 MAILBOX:
  MBQLINK     = FFFFFF  MBPROCQD = N  CLONED = N  MBINDEX =
                                                       0000
  MBLETTERCOUN =    0002  MBPROCPTR = 37A5F1 MBLETTERQ =
FFFFFF
 STATE = CPMBIDLE  MILTIWAIT = N
 STATUS = DISPLAYING
 MESSAGE :
  MT = 0016 - DIGITS_MSG
  BODY:
   A127  0AAA  550F  7055  9887  BE04  8E02  9C71  8190
   9090  9090  9080  8080  9090  0280  0202  0202  0202
   0202  0202  0202  0202  0202
  TID = NODE_NO = 011  TRMNL_NO_MSN = 0  TRMNL_NO_LSB = 41
 TEMPTIMESTAMP=012 23BA NETWORK_CONNECTION_MADE= N NCCLS = 0
CDBRTEB:
   GROUP_CPID    = CKT         TTT 1
   TERMINAL_CPID = CKT         TTT1
   CONNECTION_TYPE = D REROUTE ELEMENT = N  CST_L     = N
   CST_H           = N RECREATE_CDBRTEB= N ERWT_ACCEPTED = N
   WAIT_FOR_COT_TO = N  GRPSC            = Y
FAST  PROC = 0
** PROT Dump (0) for CALLID 820001 saved at 1987/10/14
                                          16:37:42.228 WED. **
  0887 0291 2C01 0000
** UNPROT Dump (0) for CALLID 820001 saved at 1987/10/14
                                          16:37:42.228 WED. **
 0029 0718
```

CCB dump

The Call Condense Block (CCB) is a data structure that contains information about a call in progress.

A CCB may be displayed using the SAVETID and SAVELEN commands or when a call death occurs. This section provides an example of the formatted CCB dump shown in Figure 5-3 on page 5-5.

Only those fields associated with CALLID 820001 are described. Many other fields that are application-specific exist.

Refer to 297-1001-510 (AUD395) for descriptions of other CCB fields.

For more information on the fields associated with a CCB, refer to PROTEL module CPDATAUI.

Call processing terminal linkage block (CPTLB)

The Call Processing Terminal Linkage Block (CPTLB) is the part of the CCB used by the Input/Output (I/O) system. Figure 5-6 on page 5-8 contains an example of the CPTLB.

Figure 5-6xxx
Example of CPTLB area of CCB dump

```
**
** Call Data Dump (0) for CALLID 820001 saved at
                                1987/10/14 16:37:42.228 WED.**
** CCB Dump (0) for CALLID 820001 saved at 1987/10/14
                                           16:37:42.228 WED.**
CPTLB :
 LINK/CPMBPTR = FFFFFF MYINDEX       =23 04  PROCQD    =N
 STATE        = LINKED AUDIT         =  0000 LINKCOUNT =0002
 LETTERCOUNT  =   0003 LETTERC       =  C201 WAKEID    =FFFF
 LETTERQ      = FFFFFF CCBTIMEQ.SUCC =FFFFFF CCBTIMEQ.PREV=
                                                    FFFFFF
 FASTECCBINDEX =  0000  UP_OVER_WARM  = N   ECCBINDEX    =
                                                   00 00
EXTPTR  = FFFFFF   SEQNO  = 0019  CS  = TALKIN  XBITS  = 0000
FORCEUNAVAIL = N   IBN    = N     SA    = N  FASTSEQNO = 01D9
CMI     =  0003  CTRLPORT  =   0000  CCBFC  =  0000FF
RECEIVER    = <NIL>
CCBFA :
 0028 0310
```

The fields of the CPTLB are as follows:

LINK/CPMBPTR is an overlay of:

1 LINK: a queue link while the CCB is enqueued on any of the CCB queues: the available queue, the progress queue, or the origination queue.

2 CPMBPTR: a pointer to the associated Call Data Block (CDB) while the CCB is linked to a call process.

MYINDEX the index of the CCB in the segmented store table.

PROCQD a boolean bit used while the CCB is linked to a call process. If true, it indicates that the call process is suspended on its own SOS mailbox waiting for a message.

STATE the state or configuration of the CCB, as follows:

- 0: ONFREEQ - available to be selected.
- 1: LINKED_NOCPMB - linked to a terminal but not attached to a call processing mailbox. When an incoming message comes through the CPTLB, a call processing mailbox is taken off the queue.
- 2: LINKED_CPMB - linked to a terminal and to a call processing mailbox.
- 3: UNAVAILABLE - not available, for example during an audit.
- 4: ONREADYQ1 - on CCB ready queue.
- 5: ONREADYQ2 - on READYORIGS1 ready queue. CCBs are removed from this queue when no CCBs are on the CCB ready queue.
- 6: ONREADYQ3 - on READYORIGS2 ready queue. This queue is used to handle originations under overload conditions.

AUDIT used by the call processing audit process. Values are 0 to 31.

LINKCOUNT number of terminals linked to the CCB (0 - 63).

LETTERCOUNT number of call processing letters enqueued on the CCB (0 - 3).

LETTERC number of call processing letters enqueued on the CCB (0 - 3). This field is used for fast access.

WAKEID used by call processing wakeup facility.

LETTERQ queue header for the one-way pointer queue of call processing letters enqueued on the CCB.

CCBTIMEQ.SUCC points to the link of the next item in the two-way pointer queue.

CCBTIMEQ.PREV points to the link of the preceding item in the two-way pointer queue.

FASTECCBINDEX index of an Extended Call Condense Block (if one is involved in the call). This field is used for fast access.

UP_OVER_WARM if true, indicates that a call can survive a warm restart.

ECCBINDEX index of an Extended Call Condense Block (if one is involved in the call).

EXTPTR pointer to extension block (if used).

SEQNO sequence number used in conjunction with the CCB index to generate a unique call ID for each call. Each time the CCB is used for a new call, the sequence number changes.

CS CCB call state. Call states are as follows:

- 0: idle — a call is idle.
- 1: originating — a receiver is being requested.
- 2: receiving — digits are being received and the call is not condensed.
- 3: dialing — call is condensed and waiting for digits.
- 4: outpulsing — digits are being outpulsed.
- 5: ovlpulsing — overlapped outpulsing or reception is occurring.
- 6: talking — network connection has been established and the call is condensed.
- 7: twowaycon — a two-way network connection has been established. The call is not condensed.
- 8: onewaycon — a one-way network connection has been established, such as an announcement.
- 9: multiport — multi-party calling.
- 10: servicetone — service tone, such as a reorder tone.
- 11: termtone — ringing or busy tone is being generated, or talking has begun on a fast call.
- 12: held — a special state while the call is condensed in a feature call.
- 13: testing — a special state that is not timed by an audit.
- 14: transient — a transient state with a call process.
- 15: custom — for custom calling features.
- 16: waiting — a call is condensed and timing is being performed.
- 17: origtone — the originator is ringing or receiving a tone. This is not a one port call.
- 18: pfheld — used for business sets.
- 19: setup — all messages are driven into the setup processor from the starter.

- 20: xproc — all messages are driven from the starter to the cross processor.
- 21: preserved — calls preserved over WARM SWACT, LM/RLM WARM takeback/takeover.
- 22: one_end_disc — a call is waiting on the last agent in the call to be disconnected, which usually occurs after talking and before idle.

The allowable condensed call states are as follows:

Note: If a call is condensed with a state other than those listed as allowable condensed call state, the call will be killed by an audit.

- dialing
- talking
- multiport
- termtone
- held
- testing
- ovlpulsing
- custom.

XBITS special call processing bits assigned to a terminal identifier (TID) with the state of SELECTCPTLB. When a CPTLB is selected for the TID, the bits are copied into the selected CPTLB.

FORCEUNAVAIL a boolean indicating whether the terminal linkage block state has been forced to unavailable.

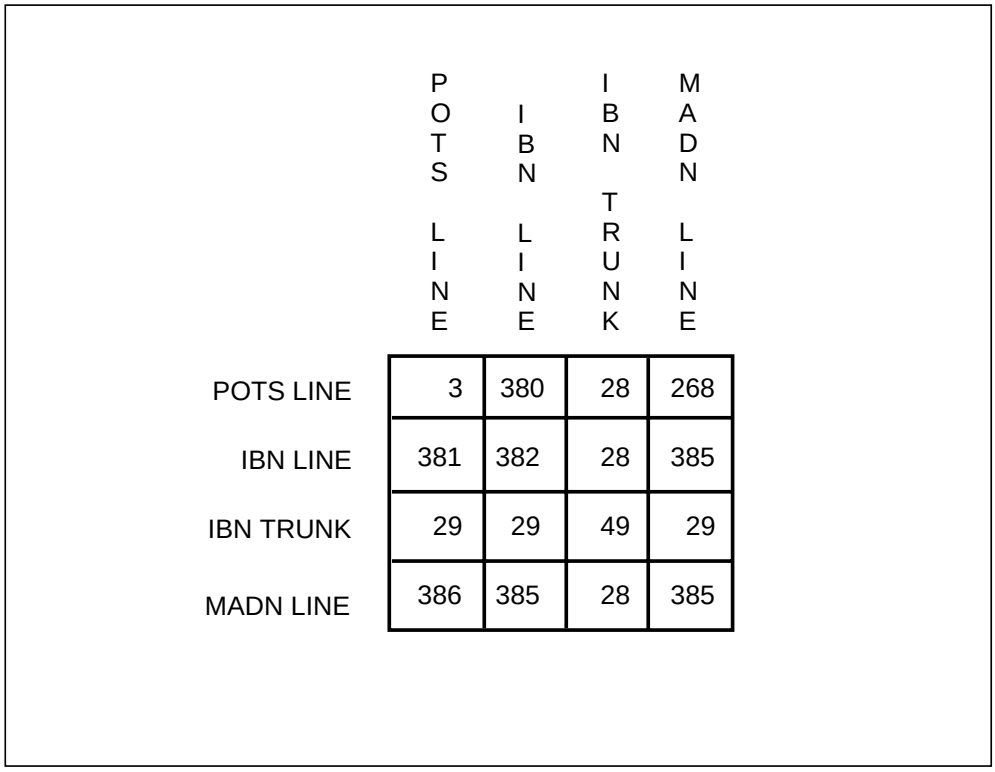
IBN indicates an IBN line; boolean.

SA if true, this bit indicates service analysis.

FASTSEQNO is an integer used only by CPIO system.

CMI Cross Matrix Index is used to categorize a connection, such as a line-to-line or a trunk-to-trunk. Refer to Figure 5-7 on page 5-12 for CMI diagram.

Figure 5-7xx
Cross matrix index diagram



CTRLPORT
call processing port number (0-31).

CCBFC CCB Format Code (0-3).

RECEIVER
call processing identifier (CPID) of the receiver.

CCBFA CCB Format Area often used to define formats for local and toll calls.

PORT1PERM and PORT2PERM

The PORT1PERM area contains information on the originating agent.

The PORT2PERM area contains information on the terminating agent.

Figure 5-8 is an example of the PORT1PERM and PORT2PERM areas.

Figure 5-8xxx
Example of PORT1PERM and PORT2PERM areas of CCB pump

```
PORT1PERM :
  AGENT  = LEN HOST 02 0 02 00  DN 7210061
 PATHEND :
  CHNL=000F PORT=001B NM_PAIR = 0001 NM_CHNL = 0B6F LOGICAL
                                                      = 0000
  GAIN=0000 LOSS=0003 INTEG_VAL = 0007E PREFERRED_PL =
                                                   ODD_PLANE
 PMCHNL =  20F PORT     = 01 CFWBIT   = N  AGENT_SUSPECT = N
 THREAD =0029  FMTCODE  = 05 UTR_AVAILABLE = N
 TID : NODE_NO = 011 TRMNL_NO_MSN = 0 TRMNL_NO_LSB = 41TSI=9
 FMTAREA:
  01C1  7100  01C0  7100  A166  7100  FE00  FFFF  FFFF  FFFF
  MBUFFPTR = 00086B
PORT2PERM :
  AGENT  = LEN HOST 00 1 02 00  DN 7210021
 PATHEND :
  CHNL=001E PORT=0034 NM_PAIR = 0000 NM_CHNL = 069E LOGICAL
                                                      = 0000
  GAIN=0001 LOSS=0003 INTEG_VAL = 0007E PREFERRED_PL =
                                                  EVEN_PLANE
 PMCHNL =   4DE  PORT   = 02  CFWBIT  = N  AGENT_SUSPECT = N
 THREAD = 0001  FMTCODE = 05  UTR_AVAILABLE = N
 TID : NODE_NO = 018 TRMNL_NO_MSN = 0 TRMNL_NO_LSB = 41TSI=0
 FMTAREA:
  4A6B  7100  4A6A  7100  BAA8  7100  FE05  FFFF  FFFF  FFFF
  MBUFFPTR = 000895
```

The fields of the PORT1PERM and PORT2PERM area are as follows:

AGENT LEN and DN associated with the dialed number; <NIL> if no DN is assigned.

PATHEND path end data area contains terminal information that is connection dependent. The fields are as follows:

- CHNL network module channel number.
- PORT network module port number.
- NM_PAIR network module pair (0-31).
- NM_CHNL network module channel.

- LOGICAL — logical connection; boolean. This field is set to true if the call is an intraswitch call. If true, there if no physical pathend and all other fields are meaningless. If this field is false, a physical connection has been made and the field is set to false.
- GAIN — pad gain value (0 - 7), adjusted based on the other pathend.
- LOSS — pad loss value (0 - 7), adjusted based on the other pathend.
- INTEG_VAL — integrity value used at the other end of the connection (0-254).
- PREFERRED_PL — preferred plane; values are ODD_PLANE (1) or EVEN_PLANE (0).

PMCHNL
peripheral module C-side channel number.

PORT call processing port number (0-31).

CFWBIT call forwarding flag.

AGENT_SUSPECT a boolean indicating whether an agent failure has occurred. If this field is pegged twice, the system will busy the faulty agent and run diagnostics.

THREAD Agent's thread index. The thread table maps from agent thread values and call processing functions to indexes into auxiliary tables that lead to specific call processing procedures. Thread values are as follows:

- 1 - POTS DP (POTS Dial Pulse)
- 2 - POTS DGT (POTS Digitone)
- 9 - POTS XTD DP (POTS extended dial pulse)
- 41 - POTS XTD DGT (POTS extended Digitone)

  Note: An extended (XTD) line is a line that has features (excluding Digitone).

- 110 - IBN DP
- 111 - IBN DGT
- 112 - IBN XTD DP
- 113 - IBN XTD DGT
- 114 - IBN CONSOLE
- 115 to 117 - IBN INCOMING TRUNK
- 129 to 131 - IBN OUTGOING TRUNK

- 132 to 134 - IBN TWO-W AY TRUNK
- 120 - 4-WIRE SUBSCRIBER
- 199 - KEY SET
- 203 - MADN DP
- 204 - MADN DGT.

FMTCODE agent format code that indicates which of the 32 refinements of the TEMP_FORMAT_AREA is being used.

UTR_AVAILABLE indicates whether the agent is a DTMF line or a multifrequency trunk on an XPM equipped with universal tone receivers (UTRs); boolean.

TID terminal ID of the agent.

NODE_NO node number.

TRMNL_NO_MSN
most significant nibble of the terminal number.

TRMNL_NO_LSB
least significant byte of the terminal number.

TSI telport supervision indicator, one of:

- 0 - no tsi
- 1 - receive tsi
- 2 - send tsi
- 3 - single tsi
- 4 - control tsi
- 5 - slave tsi
- 6 - calling tsi
- 7 - called tsi
- 8 - integ tsi
- 9 - l l tsi
- 10 - l 0 tsi
- 11 - l t tsi.

FMTAREA Agent format area contains agent pointers, progress marks, and miscellaneous bits of information related to the agents in the call.

MBUFFPTR pointer to call processing message buffer.

Call handler block (CHB)

The Call Handler Block contains the translation and routing blocks associated with a call.

Figure 5-9 on page 5-16 contains an example of the CHB.

Figure 5-9xxx
Example of CHB area of CCB dump

```
CHB:
 XLAB:
  RC           =XLA_RO  TXSEL      = 0000 TX_TRMT    =
  IBN_TRMT_SET=N        MIN_DIGITS = 0007 MAX_DIGITS = 0007
  PREFIX_FENCE= 0000    DATA_VER   = N    OC          = LCL
  TRAN_SYS     =NA      TYPECALL   = NP   NPA_ADDED   = N
  TX_POS       =NONE    TXROUTE/TERM_AGENT = LEN 00 1 02 00
                                                 DN 7210021
  CALLED_DR    =7210021
  RTE          =Y       POS        = N    RETRANSLAT = N
  CALLING_DR/AUTH_CODE_DR =
  SCREENING_IN=N        CAN_LCS    = N    CAN_SCRN    = N
  XLASTAGE     =BALANC  ADP        = Y    BLK_OVLP    = Y
  LONGHAUL     =N       HTRP       =     0SNPA        = 919
  XLT_FROM     =    0   HTRC       = N    PRIVL       = N
 ANI_INFO :
    ONI = N     HOT = NTDN = N    COIN = N
 RSP = N ANI_FAIL = NSPARE2  =    NSPARE3 = N
 AMADATA :
  0000  0000  0000  0000  0000  0000  0000  0000  0000
 SOURCEPARMS :
  VALID_SOURCEPARMS=N    NCOS   =0000   DESTIN       =NIL_DE
  CUSTGRP          =     SUBGRP =0000   CALL_CHARACTER= 0000
  DGCOLL_TABLE     =0000 SOURCE =NIL_SOOWAT_ZONE     =  0000
  SOURCE_TRC        =0000 SMDR   =N      SMDRB        =N
  ACR               =N    INTRAGROUP = N ENABLE_CRL   =N
  CRL_REQUIRED      =N    ATTDNDOV   = N
  CALLED_DR_SHIFTED=N     TDN        = N DOD_DESTIN   =N
  GROUP_INTERCOM    =N    ATTX       = N LINE_HAS_LNR =N
```

Descriptions of the fields of the CHB follow.

Translation Block (XLAB)

The Translation Block (XLAB) contains translation information. Information in this area can indicate if a call failed in translation. The fields are as follows:

RC translation return code. The return codes are as follows:

- 0: XLA_MORE_DIGITS more digits are required to complete translations at the current translation step.
- 1: XLA_CONTINUE the current translation step is complete. Translations continues with the next step.
- 2: XLA_COMPLETE translation is complete.
- 3: XLA_ROUTE_SET route identifier determined.
- 4: XLA_POS_SET route identifier determined, but the position must be decoded.
- 5: XLA_TRMT_SET treatment will be decoded.
- 6: XLA_RESTART begin digit collection again.
- 7: XLA_PROTOCOL analyze protocol.
- 8: XLA_WAIT call condenses and waits for another event, such as wait for equal access ANI or called number.
- 9: XLA_REENTER reenter translations with a new translation result.
- 10: XLA_DBQUERY_RQD for Service Switching Point (SSP) features, proceed to setup. Retranslation required after DB query.
- 11: XLA_STOP_DGCOL call processing stops digit collection and sets Automatic Data Processing (ADP). Reenter translations.

TXSEL translation selector, one of:

- 0: NIL_XLA_SELECTOR
- 1: SCCF_UPD
- 2: PRETRANSLATION_SELECTOR
- 3: HNPA_XLA_SELECTOR
- 4: INWO_XLA_SELECTOR
- 5: INWT_XLA_SELECTOR
- 6: INWS_XLA_SELECTOR
- 7: INWC_XLA_SELECTOR
- 8: DDO_XLA_SELECTOR
- 9: AMDIG_XLA_SELECTOR
- 10: CCTRNSL_XLA_SELECTOR
- 11: INPRTRNS_XLA_SELECTOR
- 12: NSC_XLA_SELECTOR.

TX_TRMT extended treatment.

IBN_TRMT_SET set to yes when the IBN treatment is also set. This prevents a Generalized No Circuit Treatment (GNCT) after an IBN treatment; boolean.

MIN_DIGITS minimum digits required.

MAX_DIGITS maximum digits required.

PREFIX_FENCE number of digits to be consumed by translations.

DATA_VER boolean set to true to initiate a trace of translations, route, and DMI.

OC originating source; values are:

- LCL
- NLCL.

TRAN_SYS translation system, such as:

- National
- International
- International Partitioned
- Nil Translation System.

TYPECALL call type, such as

- direct dial
- operator assisted
- no prefix local
- nil type of call.

NPA_ADDED boolean indicating whether a numbering plan area was added.

TX_POS translation position.

TXROUTE/TEM_AGENT translation route identifier or terminating agent CPID.

CALLED_DR called directory number.

RTE route; boolean.

POS position; boolean.

RETRANSLATE retranslation required. This is a special boolean used for the DMS-250.

CALLING_DR/AUTH_CODE_DR calling directory number or authorization code for the DMS-250.

SCREENING_IN boolean indicating whether screening was incomplete.

CAN_LCS boolean to cancel local call screening.

CAN_SCRN boolean to cancel screening.

XLASTAGE translation stage, such as:

- initial stage
- decipher stage
- six digit translation stage
- balance digits stage
- vacant code stage
- initial with prefix
- prefix added
- custom calling feature program
- ESN stage
- CCIS INWATS translation stage.

ADP boolean indicating whether all digits are present.

BLK_OVLP boolean to block overlap outpulse.

LONGHAUL a boolean indicating the presence of a longhaul operation. It is used primarily with DMS 300.

HTRP CC digit register number.

SNPA serving numbering plan area.

XLT_FROM short digit count (0 - 15).

HTRC hard to reach code.

PRIVL boolean indicating whether a terminating agent is valid.

ANI_INFO special Automatic Numbering Identification information for Equal Access.

- ONI Operator Number Identification; boolean.
- HOT hotel; boolean.

- TDN toll denied; boolean.
- COIN coin phone; boolean.
- RSP Restricted Sent Paid, ANI digit 7; boolean.
- ANI_FAIL boolean indicating whether the ANI received is in the correct format. This field is used by Service Switching Point (SSP) number services code call; boolean.
- SPARE2 spare space for call data.
- SPARE3 spare space for call data.

AMADATA used for IBN extension registers. Part of this structure is an overlay with call data.

SOURCEPARMS
The sourceparms data area contains information associated with the type of caller and the call characteristics.

VALID_SOURCEPARMS indicates whether the sourceparm area is valid for a call; boolean. The value is Y if the call is an IBN call.

NCOS Network Class of Service (0 - 255).

DESTIN destination type, one of the following:

- NIL_DEST destination not determined
- IAGRP_LINE an IBN line INTRA or INTER group
- IAGRP_TRUNK an IBN trunk INTER or INTRA group
- ROUTE non-INTRAGRP route
- FEATURE_ACTIVATION activate feature, such as SCCF (Speed Call and Call Forward) Programming
- SC_CODE speed call code usage
- ATTEND routed to attendant
- NETWORK destination is one of EPSCS, CCSA, ETN,, OWT, DOD
- DIAL_ATT ‘DIAL 0’ calls - for OM’S
- TONE_OR_ANN_TRMT destination is a treatment
- NSC_DEST network speed call

- DIAL_CALL_WAITING_DEST dial call waiting
- GROUP_INTERCOM group intercom
- LNR_CODE last number redial code
- FEATURE_ACTIVATION_VIA_ROUTING a feature is activated when the FTR terminator is called
- VMX_DEST voice message exchange.

CUSTGRP customer group (0 - 4095).

SUBGRP subgroup number (0 - 7).

CALL_CHARACTER characteristics associated with the call; defined by user datafill.

DGCOLL_TABLE index into DGCOL_PHYS_TABLE (0 - 127).

SOURCE source type of IBN agent; values are as follows:

- NIL_SOURCE
- IBNLINE
- IBNTRUNK
- IBN_PROP_SET
- ATTCONS
- EXTERNAL (line or trunk)
- DISA (Direct Inward System Access).

OWAT_ZONE OUTWATS zone (0 - 13).

SOURCE_TRC source terminating restriction code (0 - 7).

SMDR a boolean specifying station message detail recording.

SMDRB a boolean specifying station message detail recording (billable calls only).

ACR a boolean specifying whether an account code is required.

INTRAGROUP a boolean set to true if the call is an intragroup call.

ENABLE_CRL a boolean specifying enable code restrictions.

CRL_REQUIRED a boolean for a requirement to check Code Restriction Levels.

ATTDNDOV a boolean indicating attendant override for Do Not Disturb (DND).

CALLED_DR_SHIFTED a boolean indicating open-ended numbering.

DOD_DESTIN a boolean for direct outward dial destination, which is used when there is a need to know that the destination was Direct Outward Dial (DOD).

GROUP_INTERCOM a boolean set to true if Group Intercom Call (GIC) is in effect.

ATTX a boolean for an attendant extended call. The boolean is set to true when NCOS is set.

LINE_HAS_LNR a boolean indicating whether last number redial in effect.

Routing Block (RTEB)

The Routing block contains routing information. Figure 5-10 contains an example of the routing block.

Figure 5-10xxx
Example of RTEB area of CCB dump

```
RTEB:
  ROUTE        = DNT_RT RC              = 0 RTE_TYPE=DNT
  ROUTE_CHAIN  = IN_CHA PREV_SATELLITE  = N CHOICE   =DIRECT
  OHQT_APPLIED = N      QUEUEING_ACTIVE = N OHQ      =N
  CANCSTDL     = N      INHIBIT_QUEUEING= N HUNT     =Y
TRMT        = UNDT  POS      = 0000  TFR    = NIL_TF
ANI_SPILL = N      ANI_SPILL_9 = N  RECORDING_REQD = N
TS_OMREG     = Y   SEIZE_FAILURE   = N   AC_FTR_IN_EFFECT= N
RESELECT     = N   DIST_RING       = N
FTR_IN_EFFECT= N
LCO_CALL     = N   CHARGE_TREATMENT = CHGA
FORCE_CC_TIMI= N   AC_ATTACHED     = N
EAEO_DATA:
 EA_CARRIER = NILC     EA_LOCAL = N    EA_OCS_NEEDED = N
 CALL_EVENT = INITIAL_EVENT:
OVLPD :
 REALCMI = 0  DRCOUNT = 0  OLSTATE = OLFIRST
PSTATE : PROCESSOR = CROSS_PRCR STATE = 0000
MBI      = 0000 CHARGE   = N     DATA_CALL = N TIMESTAMP  =
                                                 0023 0000
ORIGDISP = 0000 TERMDISP = 0000  OCC_CALL  = N OCC_INCOM  = N
NO_INTRAS = N   EA_CALL  = N    PIC_CALL  = N LATA_CALL  =
                                                INTRALATA
STATE_CAL=INTRAS TRD_TIMIN = SHORT  TPS_CP = N  NSC_CALL = N
```

The fields are as follows:

ROUTE route identifier.

RC router return code. Return codes are as follows:

- 0: RTR_INIT — set when a route identifier is set into the routing block before a route call. When RTR_INIT is returned from the routing operations, the terminator requires that all digits must be present before termination is permitted.

- 1: CPID_RTR_SET — terminal agent identifier has been found by routing operations and is usually set into the call processing block.
- 2: RTE_EXHAUST — router exhausted the route list specified by the route identifier. Treatment is determined by the client procedure.
- 3: TRMT_RTR_SET — treatment set in CHB is determined by the client procedure.
- 4: RTR_ABORT — no terminator or allocator procedure is available. A reorder is returned.
- 5: RTR_MEMBER_ADVANCE — an agent in a group (either a trunk or a line hunt group) was busy or not available. The next member of the group is selected for termination.
- 6: RTR_FEATURE — enter feature processing.
- 7: RTR_REENTER — reenter the router with the same route element.

RTE_TYPE router type, one of:

- NATRK
- DNT
- INAT_TRK
- IBN_RTE_TYPE.

ROUTE_CHAIN reroute history; values are:

- IN_CHAIN
- OUT_OF_CHAIN_THIS_OFFICE
- OUT_OF_CHAIN_LAST_OFFICE
- OUT_OF_CHAIN_PREVIOUS_OFFICE.

PREV_SATELLITE boolean indicating whether the previous leg of the call was on a satellite link.

CHOICE route choice, as follows:

- direct route
- alternate route.

OHQT_APPLIED boolean indicating OHQT/OHQA applied.

QUEUEING_ACTIVE boolean indicating whether queueing was in process.

OHQ boolean indicating whether to allow queueing.

CANCSTDL boolean indicating whether to cancel standard outpulsing logic.

INHIBIT_QUEUEING boolean indicating whether no queuing is allowed.

HUNT a boolean, when set to false, suppresses terminators from advancing through the member list.

TRMT treatment.

POS Position.

TFR Termination failed reason, one of the following:

- 0: NIL_TFR - initialized value
- 1: NOT_AVAILABLE_TFR - no idle outgoing circuits available for terminating the call
- 2: GROUP_CONGESTION_TFR - the connecting switcher is unable to complete the call; no idle circuits available
- 3: SWITCH_CONGESTION_TFR - the connecting switcher is unable to connect the call through its office
- 4: SUB_CONGESTION_TFR - the subscribers equipment is unavailable for termination
- 5: NO_CONGESTION_TFR - no failure.

ANI_SPILL ANI spill required; boolean.

ANI_SPILL_9 a boolean, when set to Y the identification digit 9 is to be outpulsed on intercept calls.

RECORDING_REQD boolean indicating whether billing details are required.

TS_OMREG traffic separation operational measurement register.

SEIZE_FAILURE boolean indicating whether an attempt to seize an outgoing agent has failed.

AC_FTR_IN_EFFECT boolean indicating whether and Attendant Console Feature in effect.

RESELECT boolean indicating whether an outgoing agent is reselected; boolean.

DIST_REG boolean indicating distinctive ringing.

FTR_IN_EFFECT boolean indicating whether the Feature Processing Environment (FPE) needs to be accessed to implement special features.

LCO_CALL boolean indicating local coin overtime charging.

CHARGE_TREATMENT charge treatment type for TDC (time dependency control); values are as follows:

- CHGA
- CHGB
- CHGC
- CHGD.

FORCE_CC_TIMI boolean indicating whether to force CC timing (use CC timing for AMA).

AC_ATTACHED boolean indicating whether an Attendant Console is attached.

EAEO_DATA Equal Access End Office data area. This area holds the Equal Access carrier name for the duration of the call coming from the EAEO. This field is valid only if an Equal Access call is in progress.

- EA_CARRIER Equal Access carrier name.
- EA_LOCAL Equal Access local; boolean.
- EA_OCS_NEEDED equal access overlap carrier selection required; boolean.
- CALL_EVENT call event status; values are as follows:
  - INITIAL_EVENT
  - FIRST_WK_REC
  - SEC_START_DIAL_REC
  - ACK_WK_REC
  - OP_ANI_REQ_REC
  - SEC_START_DIAL_TO
  - SEC_START_DIAL_OFF
  - ACK_WK_TO
  - OP_ANI_REQ_TO.

PFXLT_FROM translate-from digit pointer. It is used in the digit register to give a starting point.

PFIXCNT prefix count (0 - 3). This field is no longer used.

DSTSW number of the destination switch.

ECHOSUP CPID of the echo suppressor. Entry is NIL if no echo suppressor is associated with the call.

OVLPD Overlap data contains information on the trunks associated with a call.

REALCMI cross matrix index.

DRCOUNT digit register count.

OLSTATE Overlap state:

- OLFIRST
- OLSUBSEQ
- OLTRANS
- OLFINAL
- OLEXIT
- OLHOLD.

PSTATE Processor entry point. Composed of the type of processor to be entered and the entry point within the processor. Processor types are as follows:

- 0: NIL_PRCR - not using PSTATE
- 1: SETUP_PRCR - enter (originating) setup processor
- 2: CROSS_PRCR - enter cross processor
- 3: RECALL_PRCR - enter recall processor
- 4: XDISCONNECT_PRCR - enter cross-disconnect processor (International only)
- 5: DISCONNECT_PRCR - enter disconnect processor
- 6: XERROR_PRCR - enter cross-error processor (International only)
- 7: ERROR_PRCR - enter error processor
- 8: FEATURE_PRCR - enter Feature Processing Environment
- 9: CUSTOM_PRCR - enter custom processor
- 10: MESSAGE_PRCR - enter message processor
- 11: REENTER_PRCR - return PROCESSOR_REENTER to starter
- 12: CONDENSE_PRCR - return PROCESSOR_CONDENSE to starter
- 13: COMPLETED_PRCR - return PROCESSOR_COMPLETED to starter

- 14: TERM_SETUP_PRCR - enter terminating setup processor. (International only).

Entry point is from 0 to 15.

MBI Message Billing Index, used for MUMR.

CHARGE boolean; default is false.

DATA_CALL boolean for Datapath use only.

Note: Datapath is a trademark of Northern Telecom.

TIMESTAMP CP timestamp.

ORIGDISP traffic separation information.

TERMDISP traffic separation information.

OCC_CALL boolean indicating an Other Common Carrier call.

OCC_INCOM boolean indicating whether an Other Common Carrier is incoming.

NO_INTRAS set to deny intraswitching; boolean.

EA_CALL boolean indicating equal access call.

PIC_CALL boolean indicating a Primary InterLATA carrier call. Default is false.

LATA_CALL values are INTRALATA or INTERLATA; default is INTRALATA.

STATE_CAL values are INTRASTATE or INTERSTATE; default is INTRASTATE.

TRD_TIMIN time release disconnect; values are:

- NO TRD
- SHORT TRD
- LONG TRD.

TPS_CP boolean used by call processor to decide whether to call agent starters or TPS.

NSC_CALL Number Service Call.

Call data block

The Call Data Block (CDB) is a data structure that is used during active call processing. Each call process has a permanently associated CDB. The CDB contains the temporary data that is associated with a call when a call is being processed. The CDB contains the mailbox and messages that were sent to the call.

A CDB may be displayed when a call death occurs. This section provides an example of the formatted CDB dump shown in Figure 5-11.

Only those fields associated with CALLID 820001 are described. Many other fields that are application-specific exist.

Refer to 297-1001-510 (AUD398) for descriptions of other CDB fields.

For more information on the fields associated with a CDB, refer to PROTEL module CPDATAUI.

Figure 5-11 on page 5-29 contains an example of a CDB dump.

Figure 5-11xxx
Example of CDB dump

```
** CDB Dump (0) for CALLID 820001 saved at 1987/10/14
                                              16:41:35.766 WED.
CPMB:
 LINK/CPTLBPTR = 1E4B00
 MAILBOX:
  MBQLINK   = FFFFFF MBPROCQD = N CLONED = N  MBINDEX = 0000
  MBLETTERCOUN =  0002 MBPROCPTR = 37A5F1 MBLETTERQ = FFFFFF
 STATE = CPMBIDLE  MILTIWAIT = N
 STATUS = DISPLAYING
 MESSAGE :
  MT = 0016 - DIGITS_MSG
  BODY:
   A127  0AAA  550F  7055  9887  BE04  8E02  9C71  8190
   9090  9090  9080  8080  9090  0280  0202  0202  0202
   0202  0202  0202  0202  0202
  TID = NODE_NO = 011  TRMNL_NO_MSN = 0  TRMNL_NO_LSB = 41
 TEMPTIMESTAMP=012 23BA NETWORK_CONNECTION_MADE=N NCCLS=0
CDBRTEB:
   GROUP_CPID    = CKT         TTT 1
   TERMINAL_CPID = CKT         TTT 1
   CONNECTION_TYPE= D REROUTE ELEMENT = N  CST_L     = N
   CST_H          = N RECREATE_CDBRTEB= N ERWT_ACCEPTED = N
   WAIT_FOR_COT_TO= N GRPSC           = Y
FAST  PROC = 0
```

The sections of the CDB are as follows:

Call processing mailbox (CPMB) data area

The Call Processing Mailbox data area contains information concerning the call processing mailboxes and associated queues as well as the pointer to the CPTLB for the call.

The fields are as follows:

LINK/CPTLBPTR an overlay of

1 LINK: a queue link that is used under the following conditions:
   - While the CDB is enqueued on the queue of unused CDBs.
   - While the call process associated with the CDB is suspended on the queue of available call processing mailboxes.
2 CPTLBPTR: a pointer to the associated Call Processing Terminal Linkage Block (CPTLB).

MAILBOX a SOS mailbox used to receive messages for a call process when the process suspends itself by invoking one of the call processing waiting procedures. While the call process is suspended on this mailbox, the CCB PROCQD field is true.

MBQLINK queue that links the mailboxes together when a process is waiting for more than one mailbox. The value of this field is NIL if only one mailbox or a cloned mailbox is being waited on.

MBPROCQD indicates if a process is waiting on the mailbox.

CLONED indicates if a mailbox has been cloned.

MBINDEX indicates the index of the mailbox on MBQLINK.

MBLETTERCOUN indicates the number of letters that can be placed on the letter queue before the queue is full. The queue is full if the value of this field is NIL.

MBPROCPTR pointer to the process waiting on the mailbox.

MBLETTERQ queue header for the one-way queue containing the letters in the mailbox.

STATE one of the following:

- CPMBQUEUED - the call process associated with the CDB is suspended on the available queue. This state also indicates that the CDB can be linked to a CPTLB when needed.
- CPMBLINKED - the CDB is linked to a CPTLB.
- CPMBUNAVAIL - the CDB is linked to other processes, such as audits or during recovery.

- CPMBIDLE - initial state when the CDB is allocated; also used as a transient state during recovery from error conditions.

MULTIWAIT boolean indicating whether a process can wait for up to 16 mailboxes simultaneously.

STATUS status of the call processing system. Status indicates what action should be taken. STATUS is normally set to PROCESSING; when errors occur, this status is used as a progress mark to prevent infinite loops during recovery. Possible values are:

- PROCESSING
- DISPLAYING
- CLEANING
- RECOVERING.

MESSAGE information on the incoming message that is currently being processed:

MT message type, such as digits message.

BODY message content; the format is dependent on the message type.

TID terminal identifier of the originator. The subfields for TID are:

NODE_NO node number

TRMNL_NO_MSN
terminal number most significant nibble

TRMNL_NO_LSB
terminal number least significant nibble

TEMPTIMESTAMP call processing timestamp.

NETWORK_CONNECTION_MADE boolean indicating whether a connection has been made to the network.

NCCLS No Circuit Class type; one of the following:

- 0 - NCRT
- 1 - NCTC
- 2 - NCLT
- 3 - NOSC
- 4 - NCBN
- 5 - NCID
- 6 - NCOT

- 7 - NCIT
- 8 - NCIM
- 9 - NCON
- 10 - NCOF.

Call data block routing block (CDBRTEB)

Call Data Block Routing Block (CDBRTEB) contains data concerning how the call routed to the terminator. Information contained in this area can indicate whether a call failed during routing. The fields are as follows:

GROUP_CPID Call Processing Identifier of the terminating group.

TERMINAL_CPID Call Processing Identifier of the terminating terminal.

CONNECTION_TYPE type of connection; one of:

- direct
- alternate
- tandem.

REROUTE ELEMENT a boolean set to true if the previous route element was a reroute element.

CST_L boolean indicating whether the class of tone is low.

CST_H boolean indicating whether the class of tone is high.

RECREATE_CDBRTEB boolean indicating whether the Call Data Block Routing Block is valid. If the CDBRTEB is NIL, the field is set to true.

ERWT_ACCEPTED boolean indicating whether the caller has accepted an expensive route.

WAIT_FOR_COT_TO_TERMINATE boolean used by CCIS6 trunks to delay termination to lines until a voice path is established.

GRPSC boolean indicating the group speed calling option.

FAST PROC causes calls to enter a streamlined processing environment from the cross processor. This field is initialized to NOT FAST. Possible values are as follows:

- 0 - not fast
- 1 - fast possible
- 2 - intertoll to intertoll

- 3 - INCAMA to intertoll
- 4 - local to local.

PROT dump

The PROT dump area is not currently used.

UNPROT dump

The UNPROT dump area is not currently used.

List of terms

Agency

An optional plug-in software Module or subsystem at the top of the hierarchy of modules in the system. It encloses data, sequences run-time commands sent to terminals, and processes calls through their different stages. See also Call Processing Agency.

Agent

See Call Processing Agent.

ANI

Automatic Number Identification

Automatic Number Identification (ANI)

A system in which a calling number is identified automatically and transmitted to the AMA office equipment for billing.

Call Condense Block (CCB)

A data block that is associated with a call from initiation through completion. The CCB contains enough information to describe a basic call, and can be extended for calls that require more data.

Call Condensing

Consists of a) condensing the data stored in the Call Data Block (CDB) so that only the information necessary to handle further functions associated with the call is retained; and b) idling the process previously associated with the call and freeing the CDB.

Call Data Block (CDB)

A data block that is associated with a call only while it is being processed. The CDB has the capability of being extended if required.

Call Handler Block (CHB)

A data block that contains routing and translation information.

Call Processing

The software system that handles the processes involved in setting up connections through the DMS-100 Family network between calling and called parties.

Call Processing Agency

A subsystem of software containing the code and data relating to a specific set of call processing features, and representing a number of Call Processing Agents with similar functions.

Call Processing Agent

(1) An originator or receiver of a call in the system. (2) A member of a call processing agency uniquely identified by a CPID.

Call Processing Utility

A subsystem of software containing the code and data for executing those call processing functions that are similar for all calls in the system.

Call Processing Identifier (CPID)

Uniquely identifies a call processing agent. The CPID consists of two parts a) a call processing selector, which is a number that identifies a particular call processing agency b) an agent identifier that identifies a specific agent within the agency.

CCB

Call Condense Block

CDB

Call Data Block

CHB

Call Header Block

CI

Command Interpreter

Command Interpreter (CI)

A support operating system component that functions as the main interface between machine and user. Its principal roles are:

1 to read lines entered by a terminal user

2 to break each line into recognizable units

3 to analyze the units

4 to recognize command input-numbers on the input lines

5 to invoke these commands.

CPID

Call Processing Identifier

Directory Number (DN)

The full complement of digits required to designate a subscriber's station within one NPA - usually a three-digit Central Office code followed by a four-digit station number.

DN

Directory Number

EXT

Extension Block

Extension Block (EXT)

In DMS-100 Call Processing, used to provide additional data space for a Call Condense Block (CCB) or a Call Data Block (CDB).

FPE

Feature Processing Environment

I/O

Incoming and/or Outgoing

Line Equipment Number (LEN)

Composed of the site, frame number, unit number, drawer number, and circuit number. For example, the LEN HOST 00 0 05 08 has the site HOST, frame number 00, unit number 0, drawer number 05, and circuit number 08.

LEN

Line Equipment Number

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

MBUFF

Message Buffer

Node

Any unit that can accept or originate messages.

Node Number

A system assigned number unique to a node.

Peripheral Module Intercept System Test (PMIST)

A debugging tool that traces messages between the Peripheral Modules.

PMIST

Peripheral Module Intercept System Test

PROT

Protected Terminal Table

PUP

Protected or Unprotected Data

SWERR

Software Error

Terminal Identifier (TID)

The node number and the terminal number.

Terminal Number

A number given to a specific terminal attached to a node.

TID

Terminal Identifier

UNPROT

Unprotected Terminal Table

User

A person, organization, or other group that uses the services of a DMS switch.

DMS-100 Family

DISPCALL user guide

Technical Assistance Manual

Publication number: TAM-1001-003
Product release: BCS34 and up
Document release: Standard 3.01
Date: July 1992

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