TAM-1001-012

DMS-100 Family

CallTrak User Guide

Technical Assistance Manual

BCS34 and up Standard 01.01 July 1992

northern telecom

DMS-100 Family

CallTrak User Guide

Technical Assistance Manual

Publication number: TAM-1001-012
Product release: BCS34 and up
Document release: Standard 01.01
Date: July 1992

Publication history

July 1992

BCS34 Standard 01.01 first release of this document

Contents

List of figures

List of tables

About this document

The CallTrak utililty provides the ability to trace calls from one or more terminals, either line or trunk. CallTrak has multiple session capability, allowing a user to manipulate sessions as desired.

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	*DMS-100 Family Guide to Northern Telecom Publications*
297-1001-010	*Maintenance and Administration Position (MAP)*
297-1001-100	*System Description*
297-1001-103	*Peripheral Modules*
297-1001-129	*Input/Output System Reference Manual*
297-1001-509	*Command Reference Manual*
297-1001-510	*Log Report Manual*
297-1001-513	*Input/Output Devices Man-Machine Interface Description*
297-1001-515	*Peripheral Modules Maintenance Reference Manual*
TAM-1001-000	*Technical Assistance Manual Index of Documents*
TAM-1001-001	*TAS Non-res Tool Listing*
TAM-1001-003	*Display Call (DISPCALL) User Guide*
TAM-1001-007	*Peripheral Module Intercept System Test (PMIST) User Guide*

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 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. ↵

CallTrak utility

This chapter describes the CallTrak utility, its uses, and its restrictions and limitations. This chapter also gives general usage ideas.

CallTrak description

CallTrak provides the ability to trace individual calls by selecting the originating terminal of the call. When CallTrak detects a message from a selected terminal, the utility marks the call as being traced. The terminal may be either a line or trunk terminal.

CallTrak contains individual tools to collect and display data. Currently CallTrak supports the tools PGMTRACE, MSGTRACE, and TIMECALL. PGMTRACE is based on the existing tool, CALLCT, and provides procedure call tracing for the call process CALLP. MSGTRACE is based on the existing tool PMIST, and provides incoming and outgoing message monitoring for all messages to and from a traced call. TIMECALL provides a listing of the call events, along with the real-time cost of those events. TIMECALL also provides the total, real-time cost for the call.

When to use CallTrak

CallTrak is a call-processing specific tool. CallTrak should be used to collect data for call-processing applications. CallTrak is not designed to replace the CALLCT and PMIST tools.

CallTrak real-time impact

CAUTION
Real-time impact
CallTrak has a substantial real-time impact on call. Users should be extremely cautious when they enter the CallTrak environment. CallTrak should never be used during peak hours.

Users should be extremely cautious when they enter the CallTrak environment. CallTrak will have a substantial effect on calls. Due to the real-time impact of CallTrak it should never be used on a switch during peak hours. PGMTRACE has the largest real-time impact, capable of slowing

calls by a factor of more than 200 times the call. MSGTRACE will slow calls by a factor of 2. TIMECALL slows calls by 25-50%.

CallTrak Modules and Subsystems

CallTrak is packaged in several subsystems. If subsystems or modules are not present certain CallTrak commands or tools will not function. TCALLSUB is the main CallTrak subsystem. All of the other subsystems provide optional functions and commands.

Table 1-1 lists all of the subsystems and modules associated with CallTrak. If a command or function fails to operate correctly, the user should reference Table 1-1 to find the module or subsystem required to perform the command, and then check the load to see if the subsystem is included.

Table 1-1xxx
CallTrak Modules and Subsytems

Subsystem	Module, function, and comments
TCALLSUB	TCALLSUB is the main CallTrak subsystem. It requires that the ToolMaster subsystem TOOLMBSE is present. TCALL-SUB contains the following: TCBASECI: CI base utilities. CALLTRAK: CallTrak CI commands. TCTTIMECL: The TIMECALL tools base. This module does not contain the display routines.
TDCMSUB	TDCMSUB is an optional subsystem. It contains the following: TCDISCM: TIMECALL display software (CM).
TDLTSSUB	TDLTSSUB is an optional subsystem. It contains the following: TCDISLTS: TIMECALL display software (LTS).
TCMSGSUB	TCMSGSUB is an optional subsystem. It requires that module DIOMSG in the PMISTSUB subsystem be present. TCMSGSUB contains the following: MSGTRACE: The MSGTRACE tool.
TCPGMSUB	TCPGMSUB is an optional subsystem. It requires that CALLCTX in the CALLCSUB subsystem be present. TCPGMSUB conatins the following: PGMTRACE: The PGMTRACE tool.
TCLLTRK	TCLLTRK is an optional subsystem for lines. It contains the fol-lowing: TCTRUNK: The select trk command.
TCALLLNS	TCALLLNS is an optional subsystem for lines. It contains the following: TCLINES: The select len and select dn commands.

TCLLKLNS	TCLLKLNS is an optional subsystem for key set lines; p-phones and ISDN sets. It contains the following: TCKLINES: The select ltid command and keyset portions of the select len and select dn commands. TCACTIDS The select ac command.
CTTOPSSB	CTTOPSSB is an optional subsystem for TOPS. It contains the following: TCTOPS The select tops command.

CallTrak interaction

Normally, CallTrak will not be affected by external actions; however, CallTrak does not ensure that users will never be interrupted by external calls. If a user happens to initiate a call ending at a terminal being called by an external user, the two calls will not interfere with each other. The accuracy of the tool is not affected; however, the user may not trace the type of call planned.

For example, consider what happens if a user is trying to trace a call to a terminal, and the terminal is called from a different site just prior to the user calling the terminal. The first call received by the terminal will complete as planned. The other call will terminate in a busy signal instead of a completed call. This does not affect the call trace. The second call is traced as a “line-to-tone” call instead of the “line-to-line” call that was planned. The output reflects the call type and the user must check that it is correct.

CallTrak usage notes

The following comments are helpful when using CallTrak.

Call description

CallTrak considers a call as a connection between two agents or terminals. In cases where a message connects through more than two terminals or agents, CallTrak considers the message to be more than one call. As an example, if a message were to connect over a line, a trunk, and a line CallTrak would consider it two calls. And, for all of the data to be collected the originator of each of the calls must be selected.

Averaging calls when using CallTrak

When using CallTrak to collect times for a call it is useful to repeat the call several times to allow TIMECALL to obtain an average time. This helps to average out timing differences for individual calls, which can fluctuate considerably.

Online help

All commands in CallTrak, including the CallTrak command, will print a help message if preceded by any parameter. The help message will show the syntax of the command, and list all of the subcommands.

Sending output to a file

The nowait command option is a convenient way to send display output to a disk file or printer. The user does not need to wait while the display is taking place. Refer to the display command for more information on the this option.

In addition to the nowait option, the user may use the send command to save the output from a command to a file. The send command is used as follows:

send *file_name* or *device_name* ↵

Interactions between timing tools

Firmware timing may only be used by one measuring tool at a time. Currently, tools using firmware timing are ACTIVITY, ANALYSIS, HOGCT, and CALLCT.

When CallTrak is started with either PGMTRACE or TIMECALL turned on, CallTrak checks whether firmware timing is in use by any other tool. If so, CallTrak issues a message and cannot be started.

Number of sessions

The maximum number of sessions is 15.

Terminal selection

This chapter describes how terminals may be selected. This chapter also describes terminal identifiers and virtual identifiers, how to calculate node and terminal numbers, and how to find virtual identifiers.

Selecting terminals

Selecting the correct terminal is vital. If the user does not select the correct terminal data will not be collected. For multiple leg calls, for example line->trunk->line, it is necessary to select the originator of each of the legs. CallTrak allows selection of up to 32 terminals.

Terminals may be selected in a number of ways. Users may select terminals according to the directory number (DN), line equipment number (LEN), terminal identifier (TID), or virtual identifier (VID). See the select command for more detail on how terminals may be chosen. CallTrak does not allow a user to select a terminal that is already tracing a call.

Originator terminals

The originator terminal is the terminal that initiates a call. To trace a call, users must select the originating terminal with the select command. In trunk calls it may be necessary to manually busy some of the possible origination terminals or select all of the trunk terminals to ensure that the call originates from a known terminal. If the call originates from a terminal that has not been selected, the call will not be traced and data will not be collected.

Terminal identifiers and virtual identifiers

Users may select terminals according to the terminal identifier (TID) or the virtual identifier (VID). CallTrak performs tracing on a per-call basis. It regards terminals from the call processing points of view. Generally, terminals are identified by their TID, as in the case of plain old telephone service (POTS) lines and trunks. However, where multiple call appearances occur on a single loop, terminals are identified by their VID. Electronic business sets (EBS) and integrated services digital network (ISDN) sets, for example, require the use of a VID, as each key on the set has its own VID.

Calculating node and terminal numbers

A terminal identifier is composed of a node number and a terminal number. A node is any unit that can accept or originate messages. 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.

To determine the node number and terminal number comprising a terminal identifier the directory number may be queried, the line equipment number may be queried, the terminal may be posted, or the carrier and timeslot may be cross-referenced.

Querying the directory number

The node and terminal number may be found by querying the directory number. To do so, use the CI command qdn outside the CallTrak subsystem with the directory number of the line to be traced as follows:

qdn ***directory number*** ↵

Refer to Figure 2-1 for an example of using the qdn command. In the figure the node number is 21 and the terminal number is 27.

Figure 2-1xxx
Sample qdn command output

```
 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
------------------------------------------------------------
```

Querying the line equipment number

The node and terminal number may be found by querying the line equipment number. To do so, use the CI command qlen and the LEN outside the CallTrak subsystem as follows:

qlen ***line equipment number*** ↵

Refer to Figure 2-2 for an example of the QLEN command. In the figure, the node number is 63 and the terminal number is 354.

Figure 2-2xxx
Sample qlen command output

```
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
------------------------------------------------------------
```

Posting the terminal

The node and terminal number may be found posting by the terminal and entering the querypm command. To post the terminal enter the post command at the appropriate MAP level along with the node type and device number. Once the terminal is posted, enter the querypm command as follows:

post *node_type device_number* ↵

querypm ↵

Refer to Figure 2-3 for an example of the querypm command. In the figure, the node number is 16.

Figure 2-3xxx
Sample posting and querypm command output

```
PM:
>post TM8 1
POST:
>querypm
  PM TYPE TM8 PM NO.: 1 NODE NO.: 16
PM_STATUS: InSv NOTE_STATUS: OK,FALSE,CHKSUM: 018 1
PP LOAD: VALID PP EXECS: VALID FNAME: BTMIA01
PMS EQUIPPED: 30 PM INT.   :1
 Site Flr RPos  Bay_Id   Shf Description  Slot  EqPEC
 HOST  00 B01   TME 0001 04 LTC  :  000        6X02AA
```

Cross-referencing the carrier and timeslot

The terminal number of a DCM may be found by posting the DCM at the TTP level of the MAP. Once posted, the carrier number and timeslot associated with the DCM is visible.

The terminal number may be read from the chart shown in Figure 2-4. In the CCT column locate the carrier number and timeslot associated with the posted DCM. The terminal number associated with the carrier and timeslot number is given in the TN column.

For example, if DCM 1 is posted at the TTP level of the MAP, and the following is displayed:

DCM 1 0 09

the carrier number is 0 and the timeslot is 09. According to the chart in Figure 2-4, 0-09 is associated with terminal number 11.

To determine the node number of the DCM, post the DCM at the PM level of the MAP and issue the command querypm. The node number is contained in the output of the querypm command.

Figure 2-4xxx
DCM carrier and timeslot to terminal number cross-reference

CCT	TN	CCT	TN	CCT	TN	CCT	TN	CCT	TN
0-01	01	1-01	31	2-01	61	3-01	91	4-01	02
0-02	32	1-02	62	2-02	92	3-02	03	4-02	33
0-03	63	1-03	93	2-03	04	3-03	34	4-03	64
0-04	94	1-04	05	2-04	35	3-04	65	4-04	95
0-05	06	1-05	36	2-05	66	3-05	96	4-05	07
0-06	37	1-06	67	2-06	97	3-06	08	4-06	38
0-07	68	1-07	98	2-07	09	3-07	39	4-07	69
0-08	99	1-08	10	2-08	40	3-08	70	4-08	100
0-09	11	1-09	41	2-09	71	3-09	101	4-09	12
0-10	42	1-10	72	2-10	102	3-10	13	4-10	43
0-11	73	1-11	103	2-11	14	3-11	44	4-11	74
0-12	104	1-12	15	2-12	45	3-12	75	4-12	105
0-13	16	1-13	46	2-13	76	3-13	106	4-13	17
0-14	47	1-14	77	2-14	107	3-14	18	4-14	48
0-15	78	1-15	108	2-15	19	3-15	49	4-15	79
0-16	109	1-16	20	2-16	50	3-16	80	4-16	110
0-17	21	1-17	51	2-17	81	3-17	111	4-17	24
0-18	52	1-18	82	2-18	112	3-18	23	4-18	53
0-19	83	1-19	113	2-19	24	3-19	54	4-19	84
0-20	114	1-20	25	2-20	55	3-20	85	4-20	115
0-21	26	1-21	56	2-21	86	3-21	116	4-21	27
0-22	57	1-22	87	2-22	117	3-22	28	4-22	58
0-23	88	1-23	118	2-23	29	3-23	59	4-23	89
0-24	119	1-24	30	2-24	60	3-24	90	4-24	120

Calculating virtual identifiers

To determine the VID in loads containing the tidtovid command, users can enter the tidtovid command and receive the VID or VIDs for a given TID. See the tidtovid command for more details.

If the tidtovid command is not available, the TIMECALL or MSGTRACE output may be examined to determine the VID for a key. This process is more complicated than using the tidtovid command.

Begin by selecting all of the terminals for the desired VID loop, and enabling either TIMECALL or MSGTRACE Then, with tracing started, simply hit the desired key to produce part of a call; which will register the key VID. The output from either TIMECALL or MSGTRACE will provide the VID for the key.

Interpreting CallTrak Output

This section details CallTrak output for TIMECALL, MSGTRACE, and PGMTRACE.

TIMECALL

TIMECALL outputs call timing information.

TIMECALL header information

TIMECALL output contains a header, which indicates some of the environmental conditions at the time the data was collected. The beginning of the header is indicated by the line -- Total call timings --, followed by two lines of useful information.

The information contained in the header includes a system log message, the date and time of data collection, the platform the call took place on, the synchronization state of the switch, and the processor type and speed. Figure 3-1 shows several examples of TIMECALL headers.

Figure 3-1xxx
TIMECALL header samples

```
-- T otal call timings --
91/10/01 15:20 mbcs33bw encomc co +rtm+ 91/10/02 ds
1991/10/05 09:51:53.380 SAT. CM on 9x13 68020 InSync

-- T otal call timings --
91/10/03 09:11 bcs33bw i300 Regression Running: DO NOT DISTURB!!
1990/05/03 10:08:25.671 THU. CC on NT40 40 MHz InSync

-- T otal call timings --
91/09/20 14:27 mbcs33bu encomc 91/9/19 am
1991/09/20 09:29:45.103 FRI. CM on 9x13 68030-40MHz InSync

-- T otal call timings --
91/09/26 13:08 bcs33bv comc co 91/09/26 ds
1991/09/27 10:56:59.451 FRI. CC on NT40 36MHz InSync

-- T otal call timings --
91/10/04 12:18 rbcs33bw bncomc co 91/10/4
1991/10/10 09:24:22.925 THU. CM on 9x13 88100 InSync
```

The system log message uses the first line. It is not controlled by CallTrak, as it may be set by any user. Therefore, CallTrak does not influence the

information presented within the system log message field. Typically, the system log message includes the BCS load and switchname.

The second line of the header contains the date and time, the platform, the processor type and speed, and the synchronization state. The date and time begin the final line of the header. The date and time shown represent the DMS clock. CallTrak makes no guarantee that this value was correctly set. The platform follows the date and time. It will be either CC or CM. TIMECALL does not run on other platforms.

The TIMECALL header reports the processor type and speed to the user. Possible values include: NT40 36Mhz, NT40 40Mhz, 68020 20Mhz, 68030 33Mhz, 68030 40Mhz, and 88100.

The synchronization state indicates whether both sides of the CM or CC are synchronized or not.

Major fields in TIMECALL output

After the TIMECALL header, call specific data appears. The data may be broken down into major fields. The major fields include the agent list, the call identification number, the call information, the total times, and the number of calls. The following is a description of each of the major fields. Figure 3-2 shows an output sample.

Figure 3-2xxx
Sample TIMECALL output

```
Agent#      CP Selector     Node# Term#     Agent Description
   0          24 LINES         17   322     LEN HOST 00 1 10 01   DN 5591234
   1          24 LINES         17   148     LEN HOST 00 1 04 19   DN 5552111

Callid(s):    33451, 33651, 33921, 34251, 34571

Type      Min   Max   Avg   CMI  ConnectionType          Ag  Msgtype/Schedtype/Packets%Shared
          Time  Time  Time
INMSG     438   442   440     0                          0  #00C ORIGINATION              0
TRANS    1723  1725  1724     0
INMSG     420   430   426     0                          0  #016 DIGITS                   0
TRANS    6177  6194  6190     0
INMSG     420   430   425     3  Line to standard Line   0  #075 EXIT                     0
TRANS    1500  1501  1501     3  Line to standard Line   1  #075 EXIT                     0
INMSG     429   432   431     0
TRANS    1378  1379  1379     0


Total    12490 12538 12521

Number of calls: 5
```

Agent list

The agent list shows the origination terminals involved in this call. The agent list includes the Agent#, CP Selector, Node#, Term#, and an Agent Description. The Agent# (agent number) is used in later fields to indicate the agent involved. It is assigned by the processor. Agent# 0 is the

originating agent of the call. The CP Selector describes the class of the agent. Typical classes include lines, trunksub, and receiver. The integer preceding the class is a code for the CP Selector.

The Node# (node number) and Term# (terminal number) are the unique node and terminal numbers assigned to the agent. Together the node and terminal numbers make up the Terminal identifier (TID) of the agent.

The Agent Description provides a useful description of the agent. For example, if a TID refers to a line, the DN and LEN of that line are displayed.

Callid

The callid provides the CALLP process a way to identify each call. A separate, unique callid exists for each call.

Call information

Call information is gathered for each call. If more than one call has the same parameters, these calls are described in the same block of information. In Figure 3-2 multiple calls originated from agent 0 and terminated at agent 1. All of the parameters, except for the times and call identification, were identical. Therefore, all of the calls are represented by the same call information. The call information includes the Type, entry times, Avg Time, CMI and Connection Type, Ag, %shared, and entry type description. Each of these subfields correspond to a single call process.

The Type refers to the type of process described by the tuple. Two common types are INMSG and TRANS. INMSG indicates an incoming message. INMSGs are the times read at IO interrupt level to accept the message and hand it off to CALLP. TRANS indicates processing of the preceding INMSG. TRANS is the time spent by the CALLP process to interpret the message and complete the actions required to respond to that event. Other entry types; SCHED, INPKT, and BOUNC are described later.

The entry times list the processing times for each entry type. The Min Time (minimum time) represents the shortest amount of time taken to complete the process. The Max time, maximum time, represents the longest amount of time the process took to complete. The Avg time (average time) is the average time for all of the calls completing the process. The number of calls may be found in the Number of calls field.

The CMI and Connection Type describe the type of connection established and the type of call. The CMI is a numerical code for the connection type. CMI 0 means that a connection was not established. CMI 3 refers to a line to standard line connection. The Connection Type is the text explanation of the CMI. This description only exists during the time a connection is established. In the example, the connection exists from after the digits are

processed until the first agent exits. The Connection Type is defined in module cpsysd.

Ag (agent) refers to the Agent# defined in the agent list. The Ag only appears for INMSGs and shows the originator of the INMSG.

The entry type description provides a text description of the entry type. In Figure 3-2 the four entry type descriptions are shown. The origination message is caused by the originator going off-hook. The digits message occurs when the collected digits are sent to the CC. The exit messages, one from each agent, occur as each agent goes on-hook.

%Shared (percent shared) occurs when two or more interrupt tasks are processed, but only one timing value is known. This happens very rarely, as generally, when an interrupt occurs the interrupt is timed normally. Occasionally, however, one or more unrelated interrupts will be processed along with the related interrupt. The number given in the %Shared column indicates how many interrupts share a single timing value. For example, if an interrupt were to occur when processing and INMSG interrupt, the unrelated interrupt and the related interrupt are processed with a single timing value. The time for the just INMSG interrupt is no longer known. Instead the total time of both interrupts are known. Since the only time of interest is the INMSG time, this time is estimated by dividing the total time by two. This is indicated in the percent shared column as 50%.

Figure 3-6, forward several pages, shows an example of a %Shared entry. In the example an INMSG has a percent shared value of 25. This means that four interrupts occurred, while only one timing value was known. Of the four interrupts only one was of interest. Shared timing values are not as accurate as values measured individually for each call entry. Percent shared applies only to INMSG, SCHED, or BOUNC, entries.

Total times

Total times appear in the tuple following the call information. Total times are provided for the Min Time, Max Time, and Avg Time columns. The Min Time total represents the sum of the minimum times. The Max Time total represents the sum of the maximum times. The Avg Time total represents the sum of the average times. The total times, in effect, represent the minimum amount of time the call could have taken, the maximum amount of time the call could have taken, and the average amount of time that the call took.

In some instances the total time may not appear to be the sum of the timing entries; however, the total time is accurate. The timing values are collected and summed using the CPU cycle times, which are less than 1 microsecond. The total values are converted to the nearest microsecond when printed.

Number of calls

The number of calls shows the number of calls collected that were identical in every respect except time. The times are averaged together as described above. It is often useful to make calls of the same type and average them, thus smoothing out timing variations in individual calls.

Timing the total call

TIMECALL uses firmware timing to measure the execution times of the processes which handle a particular call. Since call processing is message driven, timing a call amounts to timing the messages and the handling of the messages associated with a call. Usually, this involves timing the incoming messages which indicate a new call, and measuring the transaction times that process the message.

For certain types of TOPS and EBS calls, a scheduler time must also be timed. The scheduler, SCHED, is used to schedule special input handlers or to cause voluntary suspension during transaction processing.

Multiple leg calls

TIMECALL output for multiple leg, line to trunk to line for example, appears in reverse chronological order. Figure 3-3 shows an example of a multiple leg call. In the example, DN 6211234 originates a call to trunk ITDP1. The trunk ITDP1 then originates a call to DN 6211235. For the user to receive all of the information both of the originating agents, DN 6211234 and trunk ITDP1, must have been selected.

Figure 3-3xxx
Sample multiple leg call

```
Agent#      CP Selector     Node# Term#      Agent Description
   0          12 TRUNKSUB   20    100       CKT              ITDP1    4
   1          43  LINES     17    303       LEN HOST 00 1 08 18    DN 6211235
Callid(s):    47451, 46651, 41921, 54251, 76571, 77301

Type      Min   Max   Avg   CMI  ConnectionType          Ag  Msgtype/Schedtype/Packets%Shared
          Time  Time  Time
INMSG     428   488   466     0                          0  #089 ORIG_DIGITS              0
TRANS    5801  5863  5819     0
INMSG     427   458   446   223  Trk to Line no ans msg   0  #011 CLEAR_FORWARD            0
TRANS     775   788   784   223  Trk to Line no ans msg
INMSG     441   501   483     0                          1  #075 EXIT                     0
TRANS    1340  1350  1346     0
Total    9214  9447  9347
Number of calls: 6

-------------------------------------------------------

Agent#      CP Selector     Node# Term#      Agent Description
   0          43  LINES     17    322       LEN HOST 00 1 10 01    DN 6211234
Callid(s):    33451, 33651, 33921, 34251, 34571, 36951

Type      Min   Max   Avg   CMI  ConnectionType          Ag  Msgtype/Schedtype/Packets%Shared
          Time  Time  Time
INMSG     422   449   440     0                          0  #00C ORIGINATION              0
TRANS    4001  4099  4057     0
                ... digits messages and TRANS removed ...
INMSG     419   431   424    28  IBN Line to Trunk        0  #019 ANSWER                   0
TRANS     850   867   854    28  IBN Line to Trunk
INMSG     426   443   434    28  IBN Line to Trunk        0  #075 EXIT                     0
TRANS    2177  2322  2255    28  IBN Line to Trunk
Total   20981 21305 21181
Number of calls: 6
```

Calls generating AMA billing records

Some calls generate a billing record handled by the AMA process. TIMECALL measures the time required to format the billing record and displays an AMA billing entry.

In BCS34 the display was updated to show the AMA billing within the call that caused it. This was done to overcome the problem of tracking timing from billing through multiple calls. Figure 3-4 shows the way AMA billing appears in BCS34.

Figure 3-4xxx
Sample TIMECALL AMA billing entry

```
Agent#      CP Selector     Node# Term#     Agent Description
   0          1 LINES          93   151     LEN HOST 02 0 04 22   DN 7224300
   1          2 TRUNKSUB       78   501     CKT             ISUPITOG   1


Type      Min   Max   Avg   CMI  ConnectionType        Ag  Msgtype/Schedtype/Packets%Shared
          Time  Time  Time
INMSG     300   300   300     0                        0  #00C ORIGINATION_MSG          0
TRANS    1241  1241  1241     0
INMSG     278   278   278     0                        0  #016 DIGITS_MSG               0
TRANS    1068  1068  1068     0
INMSG     294   294   294     0                        0  #016 DIGITS_MSG               0
TRANS    8028  8028  8028     0
INMSG     250   250   250   374  Line to ISUP          1  #130 CCS7_UP_MSG              0
TRANS     756   756   756   374  Line to ISUP
INMSG     260   260   260   374  Line to ISUP          1  #130 CCS7_UP_MSG              0
TRANS    1034  1034  1034   374  Line to ISUP
INMSG     274   274   274   374  Line to ISUP          0  #019 ANSWER_MSG               0
TRANS     774   774   774   374  Line to ISUP
INMSG     275   275   275   374  Line to ISUP          0  #075 EXIT_MSG                 0
TRANS    2005  2005  2005   374  Line to ISUP
AMAPR    3729  3729  3729     0                           #001 LAMA_PRU_FC
AMAPR    2925  2925  2925     0                           #010 SMDR_RECORDING_UNIT
Total   23483 23483 23483
Number of calls: 1

------------------------------------------------------------

Total number of calls:  1, number of call types: 1
```

INPKT entries

INPKT indicates that an incoming long message was received and that it was broken up into packets. An inpacket, INPKT, entry exists for the first packet that was received. Another INPKT entry will show the time taken for all intermediate packets that were received, if there were any. When the last packet is received, meaning the incoming message is completely received, an INMSG entry will be shown. Refer to Figure 3-5 a for an example of the INPKT entry type.

Figure 3-5xxx
Sample INPKT entry

```
Agent#      CP Selector     Node# Term#      Agent Description
   0          2 TRUNKSUB    59   173       CKT        T_ISUPIBNTI   0
   1          1 LINES       79   394       LEN HOST 10 1 07 01   DN 5214111
Callid(s):   50411, 50611, 49011

Type      Min    Max    Avg    CMI  ConnectionType        Ag  Msgtype/Schedtype/Packets%Shared
          Time   Time   Time
INPKT     275    328    294      0                        0  First Packet                    0
INPKT     501    535    520      0                        0  2 Intermediate Packets
INMSG     403    429    419      0                        0  #1B0 CCS7_LONG_IAM_MSG          0
TRANS1398714008 14000      0
INMSG     247    310    273    375  ISUP to Line          0  #130 CCS7_UP_SG                 0
TRANS 1073   1081   1077     375  ISUP to Line
INMSG     245    271    254    375  ISUP to Line          0  #011 CLEAR_FORWARD_MSG          0
TRANS 1288   1290   1288     375  ISUP to Line
INMSG     453    457    455      0                        1  #075 EXIT_MSG                   0
TRANS 1099   1131   1111      0
Total     19063 19301 19168
Number of calls: 5
```

SCHED entries

SCHED (scheduling) entries represent time used to schedule events into the call process, or time that the call process has been scheduled out of. SCHED entries come in two forms. They may be IOINTRPT entries, or CALLP entries. Further, each of the forms may be defined as either IN or OUT.

SCHED IOINTRPT entries

SCHED IOINTRPT denotes scheduler time used to schedule in another process, IOINTRPT. The events are counted if they are caused by the call, and are not part of basic system operation. SCHED IOINTRPT appears on the NT40 platform.

Figure 3-6 shows an example of SCHED IOINTRPT. In this example, a complicated message is handled by a separate process called IOINTRPT. Therefore, each INMSG entry is both preceded and followed by a SCHED entry. The SCHED IOINTRPT IN entries record the time required to schedule the IOINTRPT process in, and the SCHED IOINTRPT OUT entries record the amount of time required to schedule the process back out.

Figure 3-6xxx
Sample IOINTRP SCHED entries and %Shared entries

```
Agent#      CP Selector     Node# Term#     Agent Description
   0          24 LINES         17   322     LEN HOST 00 1 10 01   DN 6211234
   1          24 LINES         17   148     LEN HOST 00 1 04 19   DN 6211235
Callid(s):   33451, 33651, 33921, 34251

Type      Min   Max   Avg   CMI  ConnectionType          Ag  Msgtype/Schedtype/Packets%Shared
          Time  Time  Time
SCHED     428   448   438     0                              IOINTRPT IN                    0
INMSG     713   717   714     0                           0  #9C ORIG_KEY                   0
SCHED     158   162   158     0                              IOINTRPT OUT                   0
TRANS    2920  2928  2924     0
SCHED     450   456   453     0                              IOINTRPT IN                    0
INMSG     580   588   582     0                           0  #16 DIGITS                    25
SCHED     170   180   173     0                              IOINTRPT OUT                   0
TRANS    1249  1250  1249     0
   ... 4 SCHED-INMSG-SCHED-TRANS for DIGITS removed ...
SCHED     450   453   452   382  IBN Line to IBN Line          IOINTRPT IN                    0
INMSG     584   594   585   382  IBN Line to IBN Line       0  #19 ANSWER                    50
SCHED     160   161   161   382  IBN Line to IBN Line          IOINTRPT OUT                   0
TRANS     448   455   453   382  IBN Line to IBN Line
SCHED     430   442   439   382  IBN Line to IBN Line          IOINTRPT IN                    0
INMSG     585   585   585   382  IBN Line to IBN Line       1  #75 EXIT                       0
SCHED     158   165   160   382  IBN Line to IBN Line          IOINTRPT OUT                   0
TRANS    1960  1980  1973   382  IBN Line to IBN Line
SCHED     460   464   462     0                              IOINTRPT IN                    0
INMSG     582   592   583     0                           0  #75 EXIT                       0
SCHED     158   160   159     0                              IOINTRPT OUT                   0
TRANS    1770  1780  1777     0
Total   26901 27001 26951
Number of calls: 4
```

SCHED CALLP entries

CALLP (call process) entries indicate that the active call process has voluntarily scheduled itself out for a period of time. The SCHED CALLP entries record the time taken for these scheduling operations. Since the CALLP process does not normally suspend during transaction processing these calls are not incurring additional overhead.

Figure 3-7 shows a SCHED CALLP example. A SCHED CALLP entry requires a SCHED CALLP OUT to schedule time away from the call process, and a SCHED CALLP IN to return to the call process. In the example, there are multiple cases of the CALLP OUT and CALLP IN sequence. The first has an INMSG occurring between the CALLPs. The second does not have an INMSG occurring between the CALLPs.

If an incoming message is indicated between the CALLP OUT and CALLP IN, the paused process performed a WAIT_CP which was cancelled by the arrival of the incoming message.

When an INMSG does not occur between the CALLP OUT and CALLP IN, the paused process performed either WAIT_CP or a CP_DELAY for a specific period of time, and the timeout occurred successfully. Both

WAIT_CP and CP_DELAY are capable of waiting for a specified time period, but CP_DELAY is not terminated by an INMSG.

Figure 3-7xxx
Sample SCHED CALLP entries

```
Agent#      CP Selector     Node# Term#     Agent Description
   0         2 TRUNKSUB      121   386      CKT       TOPCOMAMF       0
   1        32 RECEIVER      111    16      CKT             RCVRMF   63
   2         2 TRUNKSUB      121   398      CKT          TOPOGNY     0
   3        18 DMODEM        105     1                    DMODEM     0
Callid(s):  274810


Type      Min    Max    Avg    CMI  ConnectionType          Ag  Msgtype/Schedtype/Packets%Shared
          Time   Time   Time
INMSG     458    458    458    0                            0  #00C ORIGINATION              0
TRANS    2436   2436   2436    0
INMSG     426    426    426    0                            1  #013 LAST_DIGITS              0
SCHED     400    400    400    0                               CALLP OUT                     0
INMSG     442    442    442    0                            1  #013 LAST_DIGITS              0
SCHED     366    366    366    0                               CALLP IN                      0
SCHED     600    600    600    48  TOPS trk to TOPS pos        CALLP OUT                     0
TRANS   23102  23102  23102    0
SCHED     357    357    357    48  TOPS trk to TOPS pos        CALLP IN                      0
INMSG       0      0      0    48  TOPS trk to TOPS pos     0  #087 CPWAKEUP                 0
TRANS    8830   8830   8830    48  TOPS trk to TOPS pos
INMSG     462    462    462    48  TOPS trk to TOPS pos     2  #01C DIGITS_SENT              0
TRANS    1755   1755   1755    48  TOPS trk to TOPS pos
INMSG     433    433    433    48  TOPS trk to TOPS pos     3  #05F DM_REPORT                0
TRANS    2290   2290   2290    48  TOPS trk to TOPS pos
INMSG     433    433    433    48  TOPS trk to TOPS pos     3  #05F DM_REPORT                0
SCHED     586    586    586    48  TOPS trk to TOPS pos        CALLP OUT
INMSG     435    435    435    48  TOPS trk to TOPS pos     3  #05F DM_REPORT                0
SCHED     341    341    341    48  TOPS trk to TOPS pos        CALLP IN
TRANS   12715  12715  12715    48  TOPS trk to TOPS pos
INMSG     465    465    465    48  TOPS trk to TOPS pos     2  #018 RESEIZE                  0
TRANS    1608   1608   1608    48  TOPS trk to TOPS pos
INSMG     447    447    447    48  TOPS trk to TOPS pos     0  #011 CLEAR_FORWARD            0
TRANS    3553   3553   3553    48  TOPS trk to TOPS pos
Total   62950  62950  62950
Number of calls: 1
```

BOUNC entries

For certain types of calls, messages are sent from one peripheral module to another via CM bounce. The bounce messages consume time at interrupt level, similar to an INMSG, to identify the message as a bounce message, and to send it out to the destination PM. TIMECALL is able to trace these bounce messages, and they appear with an entry type of BOUNC in the TIMECALL output. There are two distinct types of BOUNC messages traced by TIMECALL, MTS (message transport system) and JTS (JNET transport system) bounce messages.

JTS bounce messages

JTS messages are used mainly for DTC-DTC bounce between the ends of an ISUP tandem call. JTS bounce messages are automatically selected and traced when they are present. The user does not need to actively trace JTS bounce messages. They appear in the TIMECALL output as part of the call they belong to. Since the message type of the bounce message cannot be

determined, the message type field contains the length of the message, including the header, in bytes. Figure 3-8 shows how a JTS bounce message might appear.

Figure 3-8xxx
Sample JTS bounce entries

```
Agent#      CP Selector     Node# Term#     Agent Description
   0         2 TRUNKSUB      78   181      CKT          ISUPITIC      1
   1         2 TRUNKSUB      78   500      CKT       ISUPITOGTC       1
Callid(s):   194919

Type      Min    Max    Avg    CMI  ConnectionType          Ag  Msgtype/Schedtype/Packets%Shared
          Time   Time   Time
INMSG     301    301    301      0                          0  #12C CCS7_IAM_MSG              0
TRANS    5247   5247   5247      0
BOUNC     284    284    284    264  ISUP Trk to ISUP        1  #00E JTS Bounce Msg            0
BOUNC     268    268    268    264  ISUP Trk to ISUP        0  #010 JTS Bounce Msg            0
BOUNC     272    272    272    264  ISUP Trk to ISUP        0  #00E JTS Bounce Msg            0
INMSG     259    259    259    264  ISUP Trk to ISUP        0  #130 CCS7_UP_MSG               0
TRANS     519    519    519    264  ISUP Trk to ISUP
INMSG     327    327    327    264  ISUP Trk to ISUP        0  #011 CLEAR_FORWARD_MSG         0
BOUNC     124    124    124    264  ISUP Trk to ISUP        1  #012 JTS Bounce Msg            0
TRANS    1206   1206   1206    264  ISUP Trk to ISUP
Total    8804   8804   8804
Number of calls: 1
```

MTS bounce messages

MTS bounce messages are more difficult to trace and interpret. MTS bounce messages are typically used for bouncing between a DTC to an LIU for ISUP signalling. This only occurs when the ISUP signalling is sent via an LIU peripheral, and not via a MSB7.

Because MTS does not use TID based addressing CallTrak cannot selectively trace MTS bounce messages, and charge the message cost against the appropriate call. MTS bounce messages are grouped into an artificial call and appear separately from the other TIMECALL output. Figure 3-9 shows an example of TIMECALL output containing MTS bounce messages.

Figure 3-9xxx
Sample MTS bounce entries

```
Agent#      CP Selector     Node# Term#     Agent Description
   0         0 NIL            78  3842     <NIL>
   1         2 TRUNKSUB       78   136     CKT   TOOGISUPDTC15        54
   2        24 LTC_AGN_NME    78     0     DTC  15
Callid(s):   69

Type      Min    Max    Avg   CMI  ConnectionType          Ag  Msgtype/Schedtype/Packets%Shared
          Time   Time   Time
BOUNC     294    294    294     0                          0  #03D MTS Bounce To ISN          0
BOUNC     267    267    267     0                          0  #01D MTS Bounce To ISN          0
BOUNC     352    352    352     0                          1  #03E MTS Bounce To XPM          0
BOUNC     307    307    307     0                          1  #01E MTS Bounce To XPM          0
BOUNC     316    316    316     0                          0  #03D MTS Bounce To ISN          0
BOUNC     268    268    268     0                          0  #01D MTS Bounce To ISN          0
BOUNC     203    203    203     0                          1  #03E MTS Bounce To XPM          0
BOUNC     306    306    306     0                          1  #01E MTS Bounce To XPM          0
BOUNC     279    279    279     0                          2  #025 MTS Bounce To ISN          0
BOUNC     318    318    318     0                          1  #026 MTS Bounce To XPM          0
BOUNC     297    297    297     0                          2  #025 MTS Bounce To ISN          0
BOUNC     338    338    338     0                          1  #026 MTS Bounce To XPM          0
BOUNC     294    294    294     0                          2  #025 MTS Bounce To ISN          0
BOUNC     334    334    334     0                          1  #026 MTS Bounce To XPM          0
Total    4173   4173   4173
Number of calls: 1

 ---------------------------------------------------------

 Agent#      CP Selector     Node# Term#     Agent Description
    0         1 LINES          478    59     LEN HOST 03 1 01 26          DN 2331080
    1         2 TRUNKSUB       109   433     CKT            S3S5T2C7      4
 Callid(s):   69


 Type      Min    Max    Avg   CMI  ConnectionType          Ag  Msgtype/Schedtype/Packets%Shared
           Time   Time   Time
 INMSG     173     17    173     0                          0  #00C ORIGINATION_MSG            0
 TRANS     442    442    442     0
 INMSG     158    158    158     0                          0  #016 DIGITS_MSG                 0
 TRANS    3264   3264   3264     0
 INMSG     133    133    133   374  Line to ISUP            1  #130 CCS7_UP_MSG                0
 TRANS     466    466    466   374  Line to ISUP
 INMSG     162    162    162   374  Line to ISUP            0  #019 ANSWER_MSG                 0
 TRANS     195    195    195   374  Line to ISUP
 INMSG     156    156    156   374  Line to ISUP            1  #075 EXIT_MSG                   0
 TRANS     741    741    741   374  Line to ISUP
 Total    7260   7260   7260
 Number of calls: 1
```

The MTS bounce message output is made up of one call type containing all the MTS bounce messages. Messages are not sorted by call, and messages are not associated with particular calls. In this example, the output was generated by a line to ISUP which looped around to an ISUP to line call. Both of these calls contain an ISUP component, and hence contain bounce messages. The user should split the total bounce impact between the two ISUP calls to determine the total cost for each ISUP call.

MTS bounce messages indicate direction of the bounce. The message type field for an MTS bounce message contains the length of the message in bytes, not the message type.

PGMTRACE

PGMTRACE gives a list of the procedures called during the trace. The list is shown in the order that the procedures were called. The procedure address, its name, and the module where it is found is also given.

PGMTRACE header information

PGMTRACE output begins with a header for each call. The header contains the process id, the process, the callid, as well as some of the environmental conditions at the time the data was collected.

Figure 3-10 shows a sample PGMTRACE header. If begins with the process id, followed the name of the process. In this case the process id is 652B 7008, and the name is CALLP.

Following the process name is the callid. The callid is used to give each call an unique identifier. In Figure 3-10 the callid is 513-0. The environmental information follows the callid. It includes the date and time of data collection, the platform the call took place on, and a system log message.

Figure 3-10xxx
Sample PGMTRACE header

```
Trace for #652B  #7008:  CALLP CALLID:  513-0  91/09/06  18:18:16 CM on 9x13
          91/09/03  09:50  MBCS34AT Sanitized on 910903
```

The date and time of data collection follow the callid. The date and time shown represent the DMS clock. CallTrak makes no guarantee that this value was correctly set. The platform follows the date and time. It will be either CC or CM, and will include the card identifier.

The system log message uses the second line. It is not controlled by CallTrak, as it may be set by any user. Therefore, CallTrak does not influence the information presented within the system log message field. Typically, the system log message includes the BCS load and switchname.

Major fields in PGMTRACE output

After the PGMTRACE header, call specific data appears. The data may be broken down into major fields. The major fields include stack information, and procedure information. Each of the major fields may be activated through options in the command line.

Stack information

If the user specifies the stack command string, two stack fields are included in the output. The first, This stack, denotes the stack size of the current procedure. The second, total stack, denotes the accumulated stacks size for all the procedures presently executing. The total stack is relative to the first procedure that is called in the trace. Figure 3-11 shows an output sample for stack information.

Figure 3-11xxx
Sample PGMTRACE stack information output

```
Trace for #652B  #7008:  CALLP CALLID:  513-0  91/09/06  18:18:16 CM on 9x13
          91/09/03  09:50  MBCS34AT Sanitized on 910903


 This    Total
stack    stack          Procedure
    8        8          0005110C MISCMACH STOP_ERA*
    8        8          00050428 SYSDEFS UNLOCK*..
   32       32          0C3CA84 LNSTART LINE_STA
   84      116          .   0C2E75D0 LNUTILUI SET_LINE
   12      128          .   .   0C2E6A74 LNUTILUI LINE_CPI
   16      132          .   .   OC6C891C CUSTDATA REAL_UPD
    8      140          .   .   .   0C6C81D0 CUSTDATA GET_PCUST_BASIC_PTR_
   24      140          .   .   0C6CB8EC UBNUTIL REAL_LNU
   32      148          .   .   0B0E7B18 IOUI FAST_INCOMING_KEYVID_TO_L
   12      128          .   .   0B0DCF48 IOUI TID_TO_C
    8      124          .   .   0BDF2778 SMEUI REAL_GET_SME_INDEX
  152      184          .   0C3E5FE8 LSETPRCI LINE_SETUP_P
  120      304          .   .   0CE7DEB0 KSETALCI KSET_ORI
   20      324          .   .   .   0B0E7C2C IOUI FAST_INCOMING_KEYVID_TO_S
   12      316          .   .   .   0B0DCF48 IOUI TID_TO_C
   28      332          .   .   .   0B0E7CE0 IOUI FAST_INCOMING_S
    8      340          .   .   .   .   0B0E4CEC IOUI CHECK_TID
   12      316          .   .   .   0B0DCF48 IOUI TID_TO_C
   20      324          .   .   .   0BDB26E8 LINEZD GET_SUPPL_DATA_PTR
    8      312          .   .   .   0C6CACCC IBNUTIL GET_IBN_L
   24      328          .   .   .   0CFAA900 LNSELUI OLS_NO_S
    8      312          .   .   .   00055C80 CPIOUI CP_PUT_B
   84      388          .   .   .   0C6E2B84 KSETDUI LOAD_KEY
    8      396          .   .   .   .   00055C80 CPIOUI CP_PUT_B
   32      420          .   .   .   .   0D0D2CF4 FTRGRPUTL GET_FEATURE_GROUP_I
    8      396          .   .   .   .   00055C80 CPIOUI CP_PUT_B
   32      336          .   .   .   0C3E4A08 XLALCZI XL_ORIG_
```

Procedure information

The PGMTRACE output includes a listing of all the procedures called, the address of each procedure and the module containing the procedure. The listing is giving in the order that each procedure was called. Each procedure is indented to show the relationship to the previous procedure(s); a procedure is indented from the procedure that called it. Figure 3-11 show how the procedure information appears.

In addition to the procedure address, name, and location, PGMTRACE can also display the procedure edition or the return address for the procedure. The edition option appends the issue and version number to the procedure name. The retaddr option indicates where the current procedure was called

from. It gives the procedure that called it as well as the address within that procedure where the call occurred. Figure 3-12 shows an output sample with the edition option, and Figure 3-13 shows a sample of the output with the retaddr option.

Figure 3-12xxx
Sample PGMTRACE output with the edition option

```
Trace for #652B  #7008:  CALLP CALLID:  513-0  91/09/06  18:18:16 CM on 9x13
          91/09/03  09:50  MBCS34AT Sanitized on 910903
  Time
    in
  this
  proc    Procedure
 -223     0005110C MISCMACH.AZ01 STOP_ERA*
    17    00050428 SYSDEFS.EV17 UNLOCK*..
  3175    0C3CA84 LNSTART.FF02 LINE_STA
   289    .   0C2E75D0 LNUTILUI.FJ01 SET_LINE
    14    .   .   0C2E6A74 LNUTILUI.FJ01 LINE_CPI
    27    .   .   OC6C891C CUSTDATA.CE01 REAL_UPD
```

Figure 3-13xxx
Sample PGMTRACE with the retaddr option

```
Trace for #652B  #7008:  CALLP CALLID:  513-0  91/09/06  18:18:16 CM on 9x13
          91/09/03  09:50  MBCS34AT Sanitized on 910903
  Time
    in
  this
  proc    Procedure
 -223     0005110C MISCMACH STOP_ERA* from 0DA58BE8 CMCALLP CM_RESET+#0A9C
    17    00050428 SYSDEFS UNLOCK* from 0DA5904A CMCALLP CM_RESET+#0EFE..
  3175    0C3CA84 LNSTART LINE_STA from 0DA58080 CMCALLP CM_CALL+#008C
   289    .   0C2E75D0 LNUTILUI SET_LINE from 00078180 LNSTART_STA+#0180
    14    .   .   0C2E6A74 LNUTILUI LINE_CPI from 000622E8 LNUTILUI SET_LINE+#0120
    27    .   .   OC6C891C CUSTDATA REAL_UPD from 000623FA LNUTIL SET_LINE+#0232
```

PGMTRACE tracing of AMA billing records

Calls generating a billing record are handled by the amaproc process. CallTrak cannot determine which call the amaproc trace belonged with. Amaproc is output separately, indicating the type of formatting, instead of the callid of the call. CallTrak traces some types of AMA recording units. Figure 3-14 shows an example of how the amaproc appears.

Figure 3-14xxx
PGMTRACE with AMA formatting

```
Trace for #2529, #3002:  AMAPROC AMATYPE:  LAMA_PRU_FC  81/11/09  11:24:12 CM ON 9X13
          91/10/01 15:20 MBCS33BW 91/10/02

      Procedure
      0BD30478 BCAMAFMI BC_AMA_C
      .   0B010784 SYSDEFS CHARDESC
      .   0BD510DC BOCFMT GET_CCI_
      .   0BD35A44 BCAMAFMI RECORD_S
      .   0BD5A094 BOCFMT RU_FRAME
      .   .    0C271438 LNUTILUI OBTAIN_N
```

MSGTRACE

MSGTRACE monitors incoming and outgoing messages to and from terminals involved in a traced call.

MSGTRACE header information

MSGTRACE uses four different headers, two incoming message headers and two outgoing message headers. The incoming and outgoing message headers contain both a long version and a short version. Only the long versions will be shown.

MSGTRACE incoming call header information

Incoming call headers provide information on the sending agent, the receiving agent, and the message. Messages from a PM to the CC are called incoming messages or reports. Figure 3-15 shows a sample header. Each of the fields are briefly described below.

Figure 3-15xxx
Sample MSGTRACE incoming call header

```
INCOMING  13:36:27.7  NODE TYPE= LTC_NODE  CCS7_LONG_IAM_MSG
NN= 0048  TN= 0133  MSGTAG= 3A  ROUTE= 0080  ERROR= 00  LENGTH=  16
AGENT= CKT                    ISUPITIC   3
B0 01 2C 01 01 04 00 80 81 00 55 23 00
CONTROL FIELDS= 0000 0002 FFFF FFFF FFFF FFFF FFFF FFFFF 0000 0000
FENCE1= 0000  FENCE2= 0002  CALLID= 3276731  IOMSG_PTR= FFFFFF
GRDIAN_PTR= FFFFFF  AUDIT= 0000  STATUS= 2
LENGTH= 0046  REAL_MT= CCS7_IAM_MSG
```

The first line of the incoming call header begins with the type of header, and the time that the message was received. The node type of the agent sending the message follows the time. The first line ends with a verbal description of the message.

The second line provides the node number and the terminal number of the sending agent. Both are given in hexadecimal. The second line also provides the first byte of the message, MSGTAG, the route the message took

through the switch, the error rate, and the length of the message. The length is given in hexadecimal bytes.

The third line shows the call processing agent. The fourth line shows the hexadecimal version of the actual message sent or received. The remaining lines give detailed message information. The information given varies from call to call.

For virtual node and terminal numbers, the node type appears as VLM_NODE.

MSGTRACE outgoing call header information

Outgoing message headers are similar to those of incoming messages. The first line of the outgoing message header begins with the type of call and the time the message took place, as well as showing the node type of the sending agent. The second line shows the node number and terminal number of the sending agent. The second line also shows the length of the message in hexadecimal bytes. The third line shows the call processing agent, while the fourth line shows receiving information.

Figure 3-16xxx
Sample MSGTRACE outgoing call header

```
OUTGOING  13:36:28.2  NODE TYPE= LTC_NODE
NN= 0048  TN= 0133  LENGTH= 0044
AGENT= CKT           ISUPITOG  3
DEST_MTA= 3A 80 00 00 97 80 00 00     NODE= 003A    OFFSET= 8097
```

MSGTRACE output

After the MSGTRACE header, call specific data appears. Output may be displayed sequentially or sorted into calls. Messages appear as either short messages or long messages.

Short messages

Figure 3-17 shows a sample of an outgoing message. After the header, the body of the message appears.

The first two bytes of the message body are the hexadecimal representation of the message type. The message type defines what type of message was sent. Examples of the message type include SVOFSET, SETSD, and DISCHN.

The remaining bytes contain information about what has occurred at the PM. The information is specific to the PM and the message that is being sent or received.

Figure 3-17xxx
Sample MSGTRACE outgoing message

```
OUTGOING  13:36:27.7  NODE TYPE= LCM_NODE     EXIT_MSG
NN= 0048  TN= 0016  MSTAG= 3B  ROUTE= 4000  ERROR= 00  LENGTH=  10
AGENT= LEN HOST 00 1 04 18          DN 7224111
75 00 99 50 02 00 00 00

00  3A     SVOFSET
01   82    STKI3 03 00 02
05   52    SETSD
06  2E     DISCHN
07   64    STAIDSCAN
```

Long Messages

Figure 3-18 shows a sample of an outgoing long message. Long messages are traced as complete messages, and not by packet as done by PMIST. After the header, the body of the message appears in hexidecimal.

Figure 3-18xxx
Sample MSGTRACE outgoing long message

```
OUTGOING  13:36:28.2  NODE TYPE= LTC_NODE
NN= 0048  TN= 0133  LENGTH= 0044
AGENT= CKT            ISUPITOG  3
DEST_MTA= 3A 80 00 00 97 80 00 00     NODE= 003A    OFFSET= 8097
2C 01 01 04 4A 00 00 03 06 0C 03 80 90 A2 06 81 10 17 13 32 05 20 04 FF 33 33
11 20 04 FF 33 33 11 20 04 FF 33 33 11 20 04 FF 33 33 11 20 04 FF 33 33 11 20
04 FF 33 33 11 20 04 FF 33 33 11 20 04 FF 33 33 11 00
```

Notice that the outgoing message has been disassembled to show the commands actually sent to the terminal.

For more details on interpreting MSGTRACE output, see the PMIST users guide. For more information on I/O messages, refer to *Peripheral Modules*, 297-1001-103.

CallTrak level commands

Use the CallTrak level of the MAP to use TIMECALL, PGMTRACE, and MSGTRACE.

Accessing the CallTrak level

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

calltrak ↵

a password must also be entered.

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

Responses for the CallTrak command

MAP output	Meaning and action
`CallTrak:`	**Meaning:** CallTrak has been entered. **Action:** None
`Maximum user count exceeded.`	**Meaning:** Another user is already in the CallTrak environment. CallTrak has not been entered. **Action:** Wait until a user exits CallTrak or timecall and retry the command.
`Already in CallTrak. Type Q CallTrak for Help`	**Meaning:** The user previously entered the CallTrak environment. **Action:** None

-continued-

Responses for the CallTrak command (continued)

MAP output	Meaning and action
`**** WARNING **** Calltrak has a significant REALTIME impact on the switch It affects every call. Use with care. **** WARNING ****`	**Meaning:** CallTrak is running. **Action:** None

-end-

CallTrak commands

All of the commands available at the CallTrak 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.

CallTrak commands

Command		Page
alltools	CallTrak	4-3
display	CallTrak	4-7
msgtrace	CallTrak	4-13
pgmtrace	CallTrak	4-19
quit	CallTrak	4-29
remove	CallTrak	4-31
select	CallTrak	4-37
session	CallTrak	4-45
start	CallTrak	4-49
status	CallTrak	4-53
stop	CallTrak	4-55
tidtovid	CallTrak	4-57
timecall	CallTrak	4-61
vidtotid	CallTrak	4-67

alltools

Function

Use the alltools command to enable or disable all available tools.

alltools command parameters and variables

Command	Parameters and variables
alltools	off on

Parameters and variables	Description
off	This parameter disables all available tools.
on	This parameter enables all available tools.

Examples

Examples of the alltools command

Example	Task, response, and explanation
alltools on ↵	

Task: Enable all available tools.

Response:
```
TIMECALL:  On
PGMTRACE:  On  -  Display MERGEable
MSGTRACE:  On  -  Display MERGEable
```

Explanation: TIMECALL, PGMTRACE, and MSGTRACE are enabled.

Responses

Responses for the alltools command

MAP output	Meaning and action
`Cannot allocate enough memory`	

Meaning: Not enough memory could be allocated for the user data buffers.

Action: Allocate a smaller user buffer and retry the command.

-continued-

alltools

Responses for the alltools command (continued)

MAP output	Meaning and action
`MSGTRACE: On`	**Meaning:** The MSGTRACE tool has been enabled. **Action:** None
`MSGTRACE: Off`	**Meaning:** The MSGTRACE tool has been disabled. **Action:** None
`PGMTRACE: On`	**Meaning:** The PGMTRACE tool has been enabled. **Action:** None
`PGMTRACE: Off`	**Meaning:** The PGMTRACE tool has been disabled. **Action:** None
`TIMECALL: On`	**Meaning:** The TIMECALL tool has been enabled. **Action:** None
`TIMECALL: Off`	**Meaning:** The TIMECALL tool has been disabled. **Action:** None
`Tracing started - use STOP first`	**Meaning:** Tools may not be enabled or disabled while call tracing is on. **Action:** Stop tracing calls by issuing a stop command.

-continued-

alltools

Responses for the alltools command (continued)

MAP output	Meaning and action

`**WARNING - Firmware timing is in use`
`**  You will be able to SELECT and REMOVE terminals`
`**  but you cannot START tracing until the other`
`**  tool is turned off.`
	Meaning: The timing base is already in use.
	
	Action: Wait until the other tool stops and retry the command.

-end-	

display

Function

Use the display command to show the output.

display command parameters and variables

Command	Parameters and variables						
display	alltools	[nowait	*device*	[file	*filename*]	delete	(1)
	merge	[timesort] [nowait	*device*	[file	*filename*]		(2)
		callsort]					(3)
	msgtrace	[timesort] [nowait	*device*	[file	*filename*]		(4)
		callsort]					(5)
	pgmtrace	[timesort] [full]	[stack]	[nowait	*device*		(6)
		callsort	brief]	retaddr]			(7)
		processsort]		edition]			(8)
	timecall	[partavg] [nowait	*device*	[file	*filename*]		(9)
		fullavg]					(10)
		noavg]					(11)
		summary]					(12)

display (continued)	(1) noreply]				
	(2) delete noreply]				
	(4) delete noreply]				
	(6) [file	*filename*]	delete	noreply]	
	(9) delete noreply]				(end)

Parameters and variables	Description
alltools	This parameter displays all of the tracing data collected by the enabled tools.
brief	This parameter, if specified, limits the output to showing only the Time in this Proc field and the procedure name.
callsort	This parameter sorts the display according to the call.
delete	This parameter, if included, removes the session containing the data once the display is completed.
device	This variable specifies the device to display data on.
edition	This parameter, if included, appends the edition codes after the module name in the output listing.
file	This parameter specifies that a file will be worked with.

-continued-

display

display command parameters and variables (continued)

Parameters and variables	Description
filename	This variable specifies the name of the file to be worked with.
full	This parameter, if specified, includes Time in this Proc, Actual time, Total Time, and Time since last Event fields in the display.
fullavg	This option averages the collected calls together if they are the same type of call. Only the agent types are checked when averaging. For example, call of the same type, but between different directory numbers would be averaged together.
merge	This parameter specifies that the data should be displayed in merged form.
msgtrace	This parameter displays all of the call tracing data collected by MSGTRACE. See the section on MSGTRACE in the “Interpreting CallTrak Output” chapter for information on what will be displayed.
noavg	This parameter stops the averaging of calls.
noreply	This parameter, if included, prohibits the system from informing the the user when the display is complete.
nowait	This parameter creates a copy of the session and displays the data from the copy. The nowait option may be used with any other option to display a session in the background.
partavg	This option averages the collected calls together only if the agents in the list are the same. For example, calls of the same type, but between different directory numbers would not be averaged together.
pgmtrace	This parameter displays all of the call tracing data collected by PGMTRACE. See the section on PGMTRACE in the “Interpreting CallTrak Output” chapter for information on what will be displayed.
processsort	This parameter sorts the display according to the process id. The output will be identical to the timesort output, except where non-consecutive transactions are handled by the same call process.
retaddr	This parameter, if included, displays the procedure which called the present procedure. Absolute addresses and offsets are shown.
stack	This parameter, if included, displays the Stack Usage and Total Stack fields.

-continued-

display command parameters and variables (continued)

Parameters and variables	Description
summary	This parameter averages calls in the same way that fullavg does; however, the calls are not displayed. Only a count of the number of calls and the call types are shown.
timecall	This parameter displays all of the call tracing data collected by TIMECALL. See the section on TIMECALL in the "Interpreting CallTrak Output" chapter for information on what will be displayed.
timesort	This parameter sorts the display according to the time.

-end-

Qualifications

The nowait option may be used with any display command, and applies to all versions of the display command.

The display merge command displays both the MSGTRACE entries and the PGMTRACE entries. In merged form, the output will show the PGMTRACE entries in either timesorted or callsorted form, and as messages are received or transmitted the MSGTRACE output will be placed within the PGMTRACE output. The MSGTRACE output will be placed within the PGMTRACE output at the time the entries occurred.

Examples

Examples of the display command

Example	Task, response, and explanation
display ↵	
	Task: Display all of the tracing data collected by all of the enabled tools.
	Response: `See the appropriate sections of the "Interpreting CallTrak Output" chapter for sample responses.`
	Explanation: See the appropriate sections of the "Interpreting CallTrak Output" chapter for an explanation of the response.

-continued-

display

Examples of the display command (continued)

Example	Task, response, and explanation
display timecall summary nowait lp021 noreply ↵	
	Task: Display the data collected by TIMECALL.
	Response: `See the chapter clarifying TIMECALL output for a sample response.`
	Explanation: See the chapter clarifying TIMECALL output for an explanation of the output.

-end-

Responses

Responses for the display command

MAP output	Meaning and action
`Nothing to Display`	
	Meaning: There is no data to display.
	Action: Verify that tracing has been started, and the proper terminal(s) have been selected.
`xxxxxx: Nothing to Display`	
	Meaning: For the tool, xxxxx, there is no data to display.
	Action: Verify that tracing has been started, and the proper terminal(s) have been selected.
`Tracing has been stopped`	
	Meaning: This response indicates that tracing was on, but has been stopped by the display command. Data output will follow.
	Action: None

-continued-

display

Responses for the display command (continued)

MAP output	Meaning and action
`Unable to generate list of calls` `***  DISPLAY ABORTED  ***`	**Meaning:** The display procedure was unable to get a list of calls to sort. The display process was aborted. This response occurs only when using the callsort option. **Action:** Verify that the required tools are enabled, or use the timesort option to display the data.

`Unable to generate COMPLETE call list`
`***  WARNING   ***:  CALLS MAY BE MISSING`
`Use TIMESORT option to display all data`
	Meaning: The display procedure was unable to get a complete list of calls to sort. This could be due to insufficient internal storage, or simply that the tool buffers were full. The calls that the display procedure was able to find are shown. This response only occurs with the callsort option.
	
	Action: Use the timesort option to display the data. The timesort option is guaranteed to display all possible data.

`Your print job is being processed`	**Meaning:** The data is being processed, and will be sent to the appropriate device. The message will occur after using the nowait option. **Action:** None
`Session n has been printed/filed`	**Meaning:** The data gathered by session number n was sent to the appropriate device. **Action:** None

-end-

msgtrace

Function

Use the msgtrace command to monitor incoming and outgoing messages, both to and from the terminals involved in a traced call.

msgtrace command parameters and variables

Command	Parameters and variables	
msgtrace	query	(1)
	bufsize [short *size* long *size*]	(2)
	display [timesort | callsort] [nowait *device* [file *filename*]]	(3)
		(4)
	displayopts [query | set [timesort | callsort] | clear]	(5)
		(6)
		(7)
		(8)
	off	(9)
	on	(10)
msgtrace (continued)	(3) delete noreply	(end)

Parameters and variables	Description
bufsize	This parameter, if other parameters are not included, queries the buffer size. If other parameters are included, the size of the buffer will be changed.
callsort	This parameter sorts the display according to the call.
clear	This parameter clears the default display option.
delete	This parameter, if included, removes the session containing the data once the display is completed.
device	This variable specifies the device to display data on.
display	This parameter displays all of the tracing data collected by MSGTRACE. See the section on MSGTRACE in the “Interpreting CallTrak Output” chapter for information on what will be displayed.
displayopts	This parameter controls the default display options.
file	This parameter specifies that a file will be worked with.

-continued-

msgtrace

msgtrace command parameters and variables (continued)

Parameters and variables	Description
filename	This variable specifies the name of the file to be worked with.
long	This parameter specifies that the buffer for long messages is to be set.
noreply	This parameter, if included, prohibits the system from informing the the user when the display is complete.
nowait	This parameter creates a copy of the session and displays the data from the copy. The nowait option may be used with any other option to display a session in the background.
off	This parameter disables MSGTRACE, or disables MSGTRACE functions and commands.
on	This parameter enables MSGTRACE, or enables MSGTRACE functions and commands.
query	This parameter queries the status.
set	This parameter is used in conjunction with an option to set the default display option.
short	This parameter specifies that the buffer for short messages is to be set.
size	This variable specifies the size of the buffer. If set to zero, the buffer is deallocated.
timesort	This parameter sorts the display according to the time.

-end-

Qualifications

MSGTRACE output may be sorted and displayed either sequentially or by call. Output sorted into calls contains a header which includes the callid at the beginning of each call.

Long messages are traced as complete messages. Long messages are not broken into packets. Virtual node and terminal numbers are shown in the listing if a message originates or terminates at a virtual terminal.

Examples

Examples of the msgtrace command

Example	Task, response, and explanation

msgtrace ↵	**Task:** Query the status of MSGTRACE.
	
	Response:
	`MSGTRACE: On - Display MERGEable`
	`Buffersize: Short: 32 Long: 10`
	`#Entries: Short: 5 Long: 0`
	`Display Opts: CALLSORT`
	
	Explanation: The status of MSGTRACE is displayed. The status includes whether or not the tool is active, the buffersize, the entries collected, and the display options.

msgtrace bufsize short 50 long 20 ↵
where
 	
50 and 20 are the size
	Task: Allocate a short message buffer of 50 entries, and a long message buffer of 20 entries.
	
	Response:
	`5 Data entries have been lost`
	`16 Long Data entries have been lost`
	
	Explanation: The previous short message buffer held 55 short data entries, of which 5 could not be transferred to the new buffer. The previous long message buffer held 36 data entries, of which 16 could not be transferred to the new buffer.

msgtrace displayopts clear ↵	**Task:** Clear the display option. **Response:** `Display Opts: TIMESORT` **Explanation:** The option was reset to the TIMESORT default.
-end-	

msgtrace

Responses

Responses for the msgtrace command

MAP output	Meaning and action
`MSGTRACE: On - Display MERGEable`	**Meaning:** MSGTRACE has been enabled, and can be used with the display merge command. **Action:** None
`MSGTRACE: Off - Display MERGEable`	**Meaning:** MSGTRACE has been disabled, but can be used with the display merge command. **Action:** None

`MSGTRACE: On - Display MERGEable`
`Buffersize: Short: 32 Long: 10`
`#Entries: Short: 5 Long: 0`
`Display Opts: CALLSORT`
	Meaning: The MSGTRACE status. It includes the buffersize for both long and short messages, the number of entries in each buffer, and the display options.
	
	Action: None

`nn Data entries have been lost`	**Meaning:** When the data buffer was reallocated to a smaller size, nn entries were not transferred. **Action:** None
`nn Long Data entries have been lost`	**Meaning:** When the long data buffer was reallocated to a smaller size, nn long data entries were not transferred. **Action:** None

-continued-

Responses for the msgtrace command (continued)

MAP output	Meaning and action
`Tracing started - use STOP first`	**Meaning:** MSGTRACE cannot be enabled or disabled while call tracing is initiated. **Action:** Use the stop command to stop the tracing of calls, then retry the command.
`Cannot allocate enough memory`	**Meaning:** The system could not allocate enough memory (DS temp) for the data buffer. **Action:** Try allocating a smaller buffer and retry the command.
`** ERROR: only one option can be selected`	**Meaning:** Only one MSGTRACE display option may be selected at a time. **Action:** Retry the command, selecting only one option.
`Nothing to Display`	**Meaning:** MSGTRACE has not collected any data. **Action:** Verify that tracing has been initiated, and the proper terminals were selected. If so, retry the command.
`Tracing has been stopped`	**Meaning:** Call tracing was previously on, but has been stopped. Data output will follow. **Action:** None

-end-

pgmtrace

Function

Use the pgmtrace command to trace call processes through portions of the program code.

pgmtrace command parameters and variables

Command	Parameters and variables						
pgmtrace	query						(1)
	bufsize	*buffer_size*					(2)
	clear						(3)
	display	timesort / callsort / processsort	full / brief	stack / retaddr / edition	nowait	*device*	(4) (5) (6)
	event	query					(7)
		proc					(8)
		return					(9)
		jump					(10)
		clear					(11)
	include / exclude	proc	*module*	*proc*			(12)
		addr	*address*				(13)
		module	*module*				(14)
		package	*area*				(15)
	off						(16)
	on						(17)
	tracepoint	query / clear / force	start / stop / startjump / stopjump / tracepoint	proc / addr	*module* / *address*	*proc*	(18) (19) (20) (21) (22)
pgmtrace (continued)	(4) file *filename* delete noreply (18) *offset*						(end)

Parameters and variables	Description
addr	This parameter specifies that a physical address will be worked with.
address	This variable specifies a physical address to be worked with.
area	This variable specifies the name of the package or area to be worked with.
brief	This parameter, if specified, limits the output to showing only the procedure name.

-continued-

pgmtrace

pgmtrace command parameters and variables (continued)

Parameters and variables	Description
bufsize	This parameter, if other parameters are not included, queries the buffer size. If other parameters are included, the size of the buffer will be changed.
buffer_size	This variable, if included, sets the buffer to the specified size. If not entered, the current buffer size will be displayed. If set to zero, the buffer is deallocated.
callsort	This parameter sorts the display according to the call.
clear	This parameter may be used to clear individual options, or may be used to clear all of the options. To clear a single option, the option must precede the clear parameter. If the option does not precede the clear parameter, all of the options will be cleared. The clear parameter affects include, exclude, event, and tracepoint.
delete	This parameter, if included, removes the session containing the data once the display is completed.
device	This variable specifies the device to display data on.
display	This parameter displays all of the tracing data collected by PGMTRACE. See the section on PGMTRACE in the "Interpreting CallTrak Output" chapter for information on what will be displayed.
edition	This parameter, if included, appends the edition codes after the module name in the output listing.
event	This parameter selects the type of event(s) that PGMTRACE traces and displays. PGMTRACE supports procedure calls, returns from a procedure, and other nonsequential changes in flow.
exclude	This parameter selects software procedures, modules, and/or packages to be excluded from the PGMTRACE output. For more information on the exclude parameter see the Qualifications section.
file	This parameter specifies that a file will be worked with.
filename	This variable specifies the name of the file to be worked with.
force	This parameter, if included, allows a tracepoint to be set on an instruction boundary.
full	This parameter, if specified, includes Time in this Proc, Actual time, Total Time, and Time since last Event fields in the display.

-continued-

pgmtrace command parameters and variables (continued)

Parameters and variables	Description
include	This parameter selects software procedures, modules, and/or packages to be included in the PGMTRACE output. For more information on the include parameter see the Qualifications sections.
jump	This parameter traces and displays events whenever a non-sequential change in the program flow is made.
module	This parameter specifies that a module will be specified and worked with.
module	This variable specifies the name of the module to be worked with.
noreply	This parameter, if included, prohibits the system from informing the the user when the display is complete.
nowait	This parameter creates a copy of the session and displays the data from the copy. The nowait option may be used with any other option to display a session in the background.
off	This parameter disables MSGTRACE, disables MSGTRACE functions and commands, or turns the timing tool off.
offset	This variable, when included, specifies the hexadecimal offset into the procedure.
on	This parameter enables MSGTRACE, or enables MSGTRACE functions and commands.
package	This parameter specifies that a package, or area, will be specified and worked with.
proc	This parameter specifies that a procedure will be specified and worked with. In the case of an event trace, movements to or from a procedure are traced and displayed.
proc	This variable specifies the name of the procedure to be worked with.
processsort	This parameter sorts the display according to the process id. The output will be identical to the timesort output, except where non-consecutive transactions are handled by the same call process.
query	This parameter queries the status.
retaddr	This parameter, if included, displays the procedure which called the present procedure. Absolute addresses and offsets are shown.

-continued-

pgmtrace

pgmtrace command parameters and variables (continued)

Parameters and variables	Description
return	This parameter traces and displays events that return from a procedure.
stack	This parameter, if included, displays the Stack Usage and Total Stack fields.
start	This parameter causes tracing to be initiated when the tracepoint is hit.
startjump	This parameter initiates jump event tracing. It has no effect on procedure tracing.
stop	This parameter causes tracing to be stopped when the tracepoint is hit.
stopjump	This parameter stops jump event tracing. It has no effect on procedure tracing.
timesort	This parameter sorts the display according to the time.
tracepoint	This parameter defines software tracepoints in the program store. For more information on tracepoints, see the Qualifications section.

-end-

Qualifications

PGMTRACE include and exclude parameters

The include and exclude parameters work together to limit unwanted sections of code from being traced. Generally, the exclude parameter is used to block large units of code, packages or modules, from being traced, while the include parameter is used to trace subsets of the larger units. This combination allows a user to view the output from a particular module or procedure of interest without receiving unwanted output. To exclude all but a desired section of code, use the include parameter at the outset.

The include and exclude parameters are limited to a total of ten procedures and physical addresses, ten modules, and five packages.

Selective tracing with PGMTRACE tracepoints

The tracepoint parameter is used to define software tracepoints in program store. Once in place, tracepoints may be used to selectively trace portions of code. Tracepoints are placed in program store during the time that the user has tracing activated. Once in place, call tracing does not begin at in the CALLP RESETANDQUEUE process. Instead call tracing begins where the tracepoint start is defined.

pgmtrace

To create a tracepoint, the user must select the action to be performed, either start or startjump, and the location to set the tracepoint at. The user may then choose to set a tracepoint to stop call tracing someplace after the initial tracepoint. Once set, tracepoints may be cleared, queried, or additional tracepoints may be set.

Each user is limited to a maximum of six tracepoints. Tracepoints may be placed in code that runs in outside of CALLP process, or levels other than the user. These tracepoints will be hit; however, action will only be taken if the process is a CALLP process running at user level.

Examples

Examples of the pgmtrace command

Example	Task, response, and explanation
pgmtrace ↵	

	Task: Query the status of PGMTRACE.
	
	Response:
	`PGMTRACE: On - Display MERGEable`
	`Buffersize: 1024`
	`#Entries: 238`
	`Trace EVENTS: JUMP, PROC`
	`Display EVENTS: JUMP, PROC`
	`EXCLUDE Package SOSBILGE`
	`INCLUDE Module CPIOUI`
	`No TRACEPOINTS defined`
	`Display Opts:CALLSORT, BRIEF`
	`Timing is OFF with Stabilizer OFF`
	
	Explanation: The status of PGMTRACE is displayed. For a breakdown of each of the fields see the Responses section.

pgmtrace bufsize 4096 ↵ *where* 4096 is the buffer_size	
	Task: Set the number of entries in the buffer to a maximum of 4096. **Response:** `Buffersize: 4096` **Explanation:** The buffer has been set to the desired size.

-continued-

pgmtrace

Examples of the pgmtrace command (continued)

Example **Task, response, and explanation**

pgmtrace exclude addr #657483 ↵
where

657483 is the address

Task: Exclude the procedure at physical address 657483 from call tracing.

Response: >

Explanation: The desired procedure has been excluded from call tracing.

pgmtrace event jump ↵

Task: Trace all non-sequential movements through the program code.

Response:
```
Trace EVENTs:  Jump, Proc
Display EVENTs:  Jump, Proc
```

Explanation: An entry will be made by PGMTRACE each time the program uses non-sequential flow.

pgmtrace tracepoint start proc mymod myproc 20 ↵
where

mymod is the module name
myproc is the procedure name.
20 is the offset

Task: Set a start tracepoint at offset 20 in procedure myproc of module mymod.

Response: >

Explanation: A tracepoint has been set.

-end-

Responses

Responses for the pgmtrace command

MAP output	Meaning and action
`PGMTRACE: On - Display MERGEable`	**Meaning:** PGMTRACE has been enabled, and the command string display merge may be used. **Action:** None
`PGMTRACE: Off - Display MERGEable`	**Meaning:** PGMTRACE has been disabled. **Action:** None

`PGMTRACE: On - Display MERGEable`
`Buffersize: 1024`
`#Entries: 238`
`Trace EVENTS: JUMP, PROC`
`Display EVENTS: JUMP, PROC`
`EXCLUDE Package SOSBILGE`
`INCLUDE Module CPIOUI`
`No TRACEPOINTS defined`
`Display Opts: CALLSORT, BRIEF`
`Timing is OFF with Stabilizer OFF`
	Meaning: The status of PGMTRACE has been displayed. For a more precise breakdown of each line within the status see the entry for the line in question.
	Action: None

`Buffersize: 2048`	**Meaning:** The current buffersize is shown. **Action:** None
`#Entries: 167`	**Meaning:** The number of entries collected by PGMTRACE is shown. **Action:** None

-continued-

pgmtrace

Responses for the pgmtrace command (continued)

MAP output	Meaning and action
`Trace EVENTS: JUMP, PROC` `Display EVENTS: JUMP, PROC`	**Meaning:** The events that are currently selected are shown. **Action:** None
`EXCLUDE Package SOSBILGE` `INCLUDE Module CPIOUI`	**Meaning:** The sections of code currently excluded and included are shown. **Action:** None
`No TRACEPOINTS defined`	**Meaning:** No tracepoints are currently defined. **Action:** None
`START at 6F53BD MODNAME PROCNAME+#0003` `STOP at 357BE0 MODNAME PROCNAME+#03FC`	**Meaning:** A list of the tracpoints is shown. The list includes the type of tracepoint, and its location. **Action:** None
`Display Opts: CALLSORT, BRIEF`	**Meaning:** The current display options are shown. **Action:** None
`nn Data entries have been lost`	**Meaning:** When the data buffer was reallocated to a smaller six, nn entries were not transferred. **Action:** None

-continued-

Responses for the pgmtrace command (continued)

MAP output	Meaning and action
`Tracing started - use STOP first`	**Meaning:** PGMTRACE cannot be enabled or disabled while a tool is tracing calls. **Action:** Stop the trace, and retry the command.
`Cannot allocate enough memory`	**Meaning:** The system could not allocate enough memory (DS temp) for the data buffer. **Action:** Try allocating a smaller buffer or wait until another user has finished, and retry the command.

`**WARNING - Firmware timing is in use`
`**  You will be able to SELECT and REMOVE terminals`
`**  but you cannot START tracing until the other`
`**  tool is turned off.`
	Meaning: The timing base is already in use by another tool. PGMTRACE cannot use the timing base at present.
	Action: Wait until the other tool stops, and retry the command.

`Not cleared:  START at 123456 is actually a STARTJUMP`	**Meaning:** The user requested the removal of a start tracepoint at address 123456. The tracepoint at the address was a startjump. The tracepoint was not removed. **Action:** Retry the command using the correct type of tracepoint.
`No more room for TRACEPOINTS, max = 6`	**Meaning:** Each user is limited to six tracepoints. **Action:** Remove a tracepoint and retry the command.
`**  ERROR:  FULL and BRIEF are incompatible`	**Meaning:** The two options specified are not compatible and may not be chosen together. **Action:** Retry the command using only one of the parameters; either full or brief.

-continued-

pgmtrace

Responses for the pgmtrace command (continued)

MAP output	Meaning and action
`** ERROR: TIMESORT and CALLSORT are incompatible`	**Meaning:** The two options specified are not compatible and may not be chosen together. **Action:** Retry the command using only one of the parameters; either timesort, callsort, or processsort.
`** ERROR: TIMESORT and PROCESSSORT are incompatible`	**Meaning:** The two options specified are not compatible and may not be chosen together. **Action:** Retry the command using only one of the parameters; either timesort, callsort, or processsort.
`All TRACEPOINTs cleared`	**Meaning:** All of the tracepoints have been removed. **Action:** None
`All START tracepoints cleared`	**Meaning:** All of the tracepoints of type start have been removed. **Action:** None
`START at 123456 changed to STARTJUMP`	**Meaning:** The type of tracepoint located at address 123456 has been changed from start to startjump. **Action:** None
`None cleared: TRACEPOINT at 123456 not defined.`	**Meaning:** The user attempted to remove a tracepoint at an address, 123456, where a tracepoint was not defined. **Action:** Execute a tracepoint query to find the proper address to remove, then retry the command with the proper address.
-end-	

quit

Function

Use the quit command to leave the CallTrak environment and return to the CI level.

quit command parameters and variables

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

Qualifications

If the user exits from CallTrak, all of the sessions are lost and all tracing is stopped.

Examples

Examples of the quit command

Example	Task, response, and explanation
quit ↵	**Task:** Exit the CallTrak environment and return to the CI level. **Response:** `CallTrak Ends.` **Explanation:** CallTrak has been exited successfully.

quit

Responses

Responses for the quit command

MAP output	Meaning and action
`CallTrak Ends`	**Meaning:** The CallTrak environment has been successfully exited. **Action:** None

`Tracing is started.`
`Do you really want to QUIT?`
`Enter ″YES″ or ″NO″`
	Meaning: The tool is currently tracing calls. If the user exits from CallTrak the tracing will be stopped and all data will be lost.
	
	Action: To exit from CallTrak enter yes. To abort the command enter no.

remove

Function

Use the remove command to deselect originating terminal(s) or agent(s).

remove command parameters and variables

Command	Parameters and variables						
remove	ac	*clli*					(1)
	all						(2)
	all bounce						(3)
	dn	*dn*					(4)
	len	*site*	*frame*	*unit*	*drawer*	*circuit*	(5)
	ltid	*ltgrp*	*ltnum*	*lkey*			(6)
	tid	*node*	*terminal*				(7)
	tops vcct						(8)
	trk	*clli*	*ckt1*	*ckt2*			(9)
remove (continued)	(5) *key*						(end)

Parameters and variables	Description
ac	This parameter deselects call of the attendant console terminals associated with the common language name specified. If the clli is not valid, terminals are not selected.
all	This parameter deselects all terminals on the switch.
all bounce	This parameter deselects all MTS bounce messages on the switch.
circuit	This variable specifies the line equipment circuit number.
ckt1	This variable specifies a member in the trunk group, or the first member in a range of trunk group members. If not specified the entire trunk group is selected. If ckt2 is not specified, this integer variable deselects a single member in the trunk group. If ckt2 is specified, this integer variable is the first member in the range.
ckt2	This variable specifies the final member in the trunk group range.
clli	This variable specifies the common language location identifier.
dn	This parameter deselects the originating terminal according to the directory number given.
dn	This variable specifies the directory number of the originating terminal.

-continued-

remove

remove command parameters and variables (continued)

Parameters and variables	Description
drawer	This variable specifies the line equipment drawer number
frame	This variable specifies the line equipment frame number.
key	This variable, when specified, sets the logical key number on a set. This is useful when the LEN maps to a set using VID mapping. If the set does not access VID mapping, the key is ignored. If the LEN refers to a loop with multiple sets, specifying a key is invalid and results in no terminal deselection.
len	This parameter deselects the originating terminal according to the line equipment number. This parameter may only be used on a switch containing lines.
ltgrp	This variable specifies the logical terminal group name.
lkey	This variable, when specified, sets the logical key number on a set. If not specified, all call appearances are deselected.
ltid	This parameter deselects the originating terminal according to the logical terminal identifier. The LTID identifies an unique ISDN subscriber set. This parameter may only be used on a switch containing lines.
ltnum	This variable specifies the logical terminal number.
node	This variable specifies the node number.
site	This variable specifies the line equipment number site. If not specified, the site defaults to 'HOST'.
terminal	This variable specifies the terminal number.
tid	This parameter deselects the originating terminal according to the terminal identifier.
tops vcct	This parameter deselects automatic TOPS VCCT (virtual circuit).
trk	This parameter deselects the originating terminal according to the trunk identifier.
unit	This variable specifies the line equipment unit number.

-end-

Qualifications

The remove command may be repeated with any parameters, except all, to remove more than one selected terminal.

When a remove len command is issued to a loop containing EBS or ISDN sets, the command finds all possible call appearances on the loop and removes each.

Examples

Examples of the remove command

Example **Task, response, and explanation**

remove dn 4818917 ↵
where

4818917 is the directory number.

Task: Remove a terminal according to the directory number.

Response: >

Explanation: The terminal was deselected.

remove ltid isdn 1 ↵
where

isdn is the logical terminal group name.
1 is the logical terminal number.

Task: Remove a terminal according to the logical terminal identifier.

Response: `Terminal not selected.`

Explanation: The terminal was not previously selected.

remove

Responses

Responses for the remove command

MAP output	Meaning and action
`>`	**Meaning:** This is the command prompt. The terminal has been removed. **Action:** None
`TOPS VCCT Selection Disabled.`	**Meaning:** Automatic selection of TOPS VCCT terminals for tracing TOPS OC calls will no longer be performed. **Action:** None
`n terminals removed.`	**Meaning:** The remove command found *n* terminals corresponding to the command parameters. The terminals were removed. **Action:** None
`Attempted to remove n terminals` `m terminals removed.`	**Meaning:** The remove command found *n* terminals corresponding to the command parameters; however, only *m* terminals were removed. **Action:** See the failure reason(s) that preceded this response.
`Terminal not selected.`	**Meaning:** The terminal was not previously selected. **Action:** Check that the command was entered correctly, or enter a status command to see which terminals are selected.
`Too many terminals. Exceeded table size by n.` `No terminals selected.`	**Meaning:** The select len or select ltid command found more terminals than the locally allocated table could contain. No terminals were selected. **Action:** Contact the CallTrak prime to increase the size of the table.

-continued-

Responses for the remove command (continued)

MAP output	Meaning and action
`n trunk members found.`	**Meaning:** The remove trk command found *n* members. **Action:** None
`Invalid member range: {n1, n2}`	**Meaning:** The range specified in the remove trk command was incorrect. When specifying a range, n2 must be greater than n1. **Action:** Check the parameters and retry the command.
`Invalid range - member n does not exist.`	**Meaning:** One of the parameters in the member range does not exist. **Action:** Check the parameters and retry the command, or select the entire trunk group by omitting the optional range parameters.
`n MADN members found`	**Meaning:** The DN specified is a multiple appearance directory number (MADN) with *n* members. **Action:** None
`Invalid DN - expecting 7 digits.`	**Meaning:** The DN specified in the command was longer than 7 digits. Currently variable length DNs of up to 7 digits are supported. **Action:** Check the parameters and retry the command.
`Invalid DN`	**Meaning:** The DN specified is not known by the system, or contains values outside of the digit range (0 to 9). **Action:** Check the parameters and retry the command.

-continued-

remove

Responses for the remove command (continued)

MAP output	Meaning and action
`Invalid CLLI`	**Meaning:** The CLLI specified is not recognized by the system. **Action:** Check the parameters and retry the command.
`****  WARNING: Unmapped logical terminal.`	**Meaning:** The logical terminal specified is a valid terminal; however, the terminal is not connected to a line. **Action:** None
`Tracing already started - use STOP first.`	**Meaning:** The user cannot select or remove terminals while tracing is on. **Action:** Use the stop command to turn tracing off, then retry the command.
-end-	

select

Function

Use the select command to set originating terminal(s) or agent(s).

select command parameters and variables

Command	Parameters and variables						
select	ac	*clli*					(1)
	dn	*dn*					(2)
	len	*site*	*frame*	*unit*	*drawer*	*circuit*	(3)
	ltid	*ltgrp*	*ltnum*	*lkey*			(4)
	tid	*node*	*terminal*				(5)
	tops vcct						(6)
	trk	*clli*	*ckt1*	*ckt2*			(7)
select (continued)	(3) *key*						(end)

Parameters and variables	Description
ac	This parameter selects call attendant console terminals associated with the CLLI specified. If the clli is not valid, terminals are not selected.
circuit	This variable specifies the line equipment circuit number.
ckt1	This variable specifies a member in the trunk group, or the first member in a range of trunk group members. If not specified the entire trunk group is selected. If ckt2 is not specified, this integer variable deselects a single member in the trunk group. If ckt2 is specified, this integer variable is the first member in the range.
ckt2	This variable specifies the final member in the trunk group range.
clli	This variable specifies the CLLI.
dn	This parameter selects the originating terminal according to the DN given.
dn	This variable specifies the DN of the originating terminal.
drawer	This variable specifies the line equipment drawer number.
frame	This variable specifies the line equipment frame number.

-continued-

select

select command parameters and variables (continued)

Parameters and variables	Description
key	This variable, when specified, sets the logical key number on a set. This is useful when the LEN maps to a set using VID mapping. If the set does not access VID mapping, the key is ignored.
len	This parameter selects the originating terminal according to the line equipment number. This parameter may only be used on a switch containing lines.
ltgrp	This variable specifies the logical terminal group name.
lkey	This variable, when specified, sets the logical key number on a set. If not specified, all call appearances are deselected.
ltid	This parameter selects the originating terminal according to the logical terminal identifier. The LTID identifies an unique ISDN subscriber set. This parameter may only be used on a switch containing lines.
ltnum	This variable specifies the logical terminal number.
node	This variable specifies the node number.
site	This variable specifies the LEN site. If not specified, the site defaults to 'HOST'.
terminal	This variable specifies the terminal number.
tid	This parameter selects the originating terminal according to the terminal identifier.
tops vcct	This parameter selects automatic TOPS VCCT (virtual circuit). This command is required when tracing tops OC calls. The TOPS VCCT command may be used to automatically select the host vcct after the remote call originator has been selected. The remote call originator may be selected with any other command.
trk	This parameter selects the originating terminal according to the trunk identifier.
unit	This variable specifies the line equipment unit number.

-end-

Qualifications

The select command may be used to enable more than one originating terminal.

Use the select dn command when the directory number is the only directory number or the primary directory number of a MADN group. For secondary

directory number members, it is necessary to select the key VID using the select tid command.

The select len command will select all of the VIDs on a given loop, up to the maximum number of terminals that may be selected. When the LEN refers to a loop containing EBS or ISDN sets, the select len command finds all possible call appearances on the loop and selects them for the user.

Examples

Examples of the select command

Example	Task, response, and explanation
select dn 4818917 ↵ *where* 4818917 is the DN.	**Task:** Select a call according to the directory number. **Response:** >**Explanation:** The terminal was selected.
select len rem1 1 0 15 32 1 ↵ *where* rem1 is the site name. 1 is the frame number. 0 is the unit number. 15 is the drawer number. 32 is the circuit number. 1 is the logical key number.	**Task:** Select a terminal according to the line equipment number. **Response:** `Terminal already selected.` **Explanation:** The terminal was previously selected.

-continued-

select

Examples of the select command (continued)

Example	Task, response, and explanation
select trkigmf 1 4 ↵ *where* igmf is the trunk CLLI. 1 is the start of the member range. 4 is the end of the member range.	**Task:** Select a range of trunks beginning with 1 and running through 4. **Response:** `4 trunk members found.` **Explanation:** The command selected members 1 through 4 on trunk igmf.

-end-

Responses

Responses for the select command

MAP output	Meaning and action
`>`	**Meaning:** This is the command prompt. The terminal has been selected. **Action:** None
`nn VCCT nodes found` `TOPS VCCT Selection Enabled.`	**Meaning:** Automatic selection of TOPS VCCT terminals for tracing TOPS OC calls has been enabled. The number of VCCT nodes found and allocated are represented by *nn*. **Action:** None
`n terminals selected.`	**Meaning:** The select command found *n* terminals corresponding to the command parameters. The terminals were selected. **Action:** None.

-continued-

Responses for the select command (continued)

MAP output	Meaning and action
`Attempted to select n terminals` `m terminals selected.`	**Meaning:** The select command found *n* terminals corresponding to the command parameters. However, only *m* terminals were selected. **Action:** See the failure reason(s) that preceded this response.
`Terminal already selected.`	**Meaning:** The terminal has already been enabled. The terminal could not be selected. **Action:** Check that the command was entered correctly.
`There is only room to select m terminals.` `No terminals selected.`	**Meaning:** The command attempted to select more terminals than the selection list had space for. The list has space for *m* terminals. This command did not select any terminals. **Action:** Remove some of the selected terminals or select fewer terminals.
`Selection list full.` `No terminals selected.`	**Meaning:** The terminal could not be selected. The maximum number of terminals has already been reached. **Action:** Remove some of the selected terminals.
`No such terminal.`	**Meaning:** The parameter does not refer to a valid terminal. **Action:** Check the parameters and retry the command.
`Cannot allocate enough memory.`	**Meaning:** The terminal was not selected. The memory to select terminal on the node could not be allocated. **Action:** Call a local NT representative to have more memory installed.

-continued-

select

Responses for the select command (continued)

MAP output	Meaning and action
`Too many terminals.  Exceeded table size by n.` `No terminals selected.`	**Meaning:** The select len or select ltid command found more terminals than the locally allocated table could contain. No terminals were selected. **Action:** Contact the CallTrak prime to increase the size of the table.

`***  WARNING  ***  This TID maps to VIDs`
`Select the originating key VID using`
`the SELECT DN or SELECT LEN commands -OR-`
`The TIDTOVID command converts a TID to VIDs`
`select the correct VID using SELECT TID.`
	Meaning: The terminal was selected; however, the selected TID maps to one or more set and/or key VIDs. The user must select the key VID for CallTrak to collect data.
	
	Action: Use the tidtovid command to determine and select the key VIDs on this TID, or use the select len or select dn command to select one or more VIDs.

`n trunk members found.`	**Meaning:** The select trk command found *n* members. **Action:** None
`Invalid member range: {n1, n2}`	**Meaning:** The range specified in the select trk command was incorrect. When specifying a range, n2 must be greater than n1. **Action:** Check the parameters and retry the command.
`Invalid range - member n does not exist.`	**Meaning:** One of the parameters in the member range does not exist. **Action:** Check the parameters and retry the command, or select the entire trunk group by omitting the optional range parameters.

-continued-

Responses for the select command (continued)

MAP output	Meaning and action
`n MADN members found`	**Meaning:** The DN specified is a multiple appearance directory number (MADN) with *n* members. **Action:** None
`Invalid DN - expecting 7 digits.`	**Meaning:** The DN specified in the command was longer than 7 digits. Currently variable length DNs of up to 7 digits are supported. **Action:** Check the parameters and retry the command.
`Invalid DN`	**Meaning:** The DN specified is not known by the system, or contains values outside of the digit range (0 to 9). **Action:** Check the parameters and retry the command.
`Invalid CLLI`	**Meaning:** The CLLI specified is not recognized by the system. **Action:** Check the parameters and retry the command.
`****  WARNING: Unmapped logical terminal.`	**Meaning:** The logical terminal specified is a valid terminal; however, the terminal is not connected to a line and will not collect any data. **Action:** Remove the terminal, and check the command parameters. Either map the terminal to a line and retry the command, or select another terminal.
`Tracing already started - use STOP first.`	**Meaning:** The user cannot select or remove terminals while tracing is on. **Action:** Use the stop command to turn tracing off, then retry the command.

-end-

session

Function

Use the session command to control CallTrak's virtual session capability. With the session command, users may monitor, create, delete, and move between sessions.

session command parameters and variables

Command	Parameters and variables		
session	current	*sessnum*	
	delete	[no	*sessnum*]
		[all	]
	new		
	status	all	

Parameters and variables	Description
all	This parameter, when specified, applies the command entered to all of the sessions.
current	This parameter causes the chosen session to become the current session. Any commands made by the user will affect the chosen session.
delete	This parameter removes the chosen session.
new	This parameter creates a new session. The new session will have the same terminal selections, tools enabled, buffer sizes, and options as the current session; however, the new session will have empty data buffers. The new session will become the current session.
no	This parameter, when used in conjunction with a session number, specifies a particular session.
sessnum	This variable specifies a particular session.
status	This parameter displays a list of sessions. The list will contain the the session, the session's owners, the session's status, as well as indicating the current session.

Qualifications

Each CallTrak user has at least one session, the current session, connected to the MAP terminal. Most of the CallTrak commands affect only the current session. Users may create additional sessions and allow the additional sessions to operate without affecting the current session.

When the command session delete all is used, any session running immediate tasks will be deleted when the task is finished.

session

Examples

Examples of the session command

Example	Task, response, and explanation
session status ↵	

Task: Display the user's sessions.

Response: See the first sample in the Responses section.

Explanation: The system displays the user's sessions. The session number, status, user name, and last update time are displayed. The current session is also noted.

Responses

Responses for the session command

MAP output	Meaning and action

```
CallTrak:
session status
Session       Status                      User        Time
-------       ------                      ----        ----
   5          Data is being processed     OPERATOR    10:40:46.018
   6          Data HAS been processed     OPERATOR    10:50:31.363
   7          No data collected           OPERATOR    10:48:54.147
   8 <CURR>   No data collected           OPERATOR    10:50:10.428
```

Meaning: The system shows all sessions currently running. The session number, status, user, time of the last update, and current session are displayed.

Action: None

```
Tracing is STARTed.  You must STOP tracing if
you want to change the CURRENT session.
```

Meaning: The user cannot change the current session while tracing calls.

Action: Use the stop command and then retry the command.

-continued-

Responses for the session command (continued)

MAP output	Meaning and action
`Your new session number is x`	**Meaning:** A new session has been successfully created. The new session number has been designated as x. **Action:** None.
`Invalid symbol: <Dealloc option> {ALL, NO <Session No> {1 TO 32000}}`	**Meaning:** The user entered an invalid parameter. The system responds with a listing of all of the parameter options. **Action:** Retry the command using appropriate options.
`Session 'n' cannot be set to current because it is in PROCESSING state.`	**Meaning:** Session n is being processed by another command, and cannot be made current until the command finishes. **Action:** Enter the session status command. The status of the desired session will change to "Processed" when it becomes available. Retry the desired command.
`You cannot delete the current session.`	**Meaning:** The user attempted to delete the session which is currently selected. **Action:** To delete the current session the user must first change to another session, and then delete the desired session.
`Session n was deleted successfully`	**Meaning:** Session number n was successfully deleted. **Action:** None
`Session(s) were deleted successfully`	**Meaning:** All of the sessions, except the current one, were deleted successfully. Any session completing tasks will be deleted as soon as the tasks are complete. **Action:** None

-continued-

session

Responses for the session command (continued)

MAP output	Meaning and action
`Invalid Session`	**Meaning:** The command referenced a session that does not exist. **Action:** Retry the command using an existing session. The session status command will five a list of all of the sessions.
`EITHER incorrect optional parameter(s) OR too many parameters.`	**Meaning:** The parameters entered by the user were incorrect, or the user entered an additional parameter. **Action:** Execute a query command to check the parameters, and retry the command.
`Wrong type: <Session No> {1 TO 32000}` `Enter: <Session No>`	**Meaning:** Although the parameters entered were correct, one of the parameters was out of range or of the wrong type. **Action:** Enter the appropriate parameter.
-end-	

start

Function

Use the start command to begin tracing activity originating from the selected terminals.

start command parameters and variables

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

Qualifications

The start command clears all tool buffers before beginning the tracing process.

Examples

Examples of the start command

Example	Task, response, and explanation
start ↵	
	Task: Begin tracing calls originating from the selected terminals.
	Response: `Started`
	Explanation: The tracing has been initiated.

start

Responses

Responses for the start command

MAP output	Meaning and action
`Started`	**Meaning:** The tracing has been initiated. **Action:** None
`Tracing already started`	**Meaning:** Tracing was previously initiated. **Action:** None
`Firmware timing on - STOP other tool first.`	**Meaning:** The timing base for TIMECALL and PGMTRACE tools is already in use by another timing tool. **Action:** Wait until the other tool has stopped or stop the other tool, and then retry the command.
`NO TOOLS On`	**Meaning:** Tracing was not initiated because no tools were enabled. **Action:** Enable one or more tools, and retry the command.
`No Terminals Selected`	**Meaning:** Tracing was not initiated because no terminals have been selected. **Action:** Select one or more terminals, and retry the command.
`xxxxxx:  Failed to Start Tool`	**Meaning:** Tracing was not initiated because one of the tools could not be started. The tools that failed to start replaces the xxxxx. Reason(s) for failure to start follow the line shown. **Action:** See the reason for failure.

-continued-

Responses for the start command (continued)

MAP output	Meaning and action
`PGMTRACE Error: Breakpoint already set at that location. Address= aaaaaa`	**Meaning:** The tracepoint at address aaaaaa could not be inserted into program store. A tracepoint is already set at that location in store. **Action:** Move the tracepoint to another location, then retry the command.
`PGMTRACE Error: Internal Tables full. Address= aaaaaa`	**Meaning:** The tracepoint at address aaaaaa could not be handled by the DEBUG system. The maximum number of DEBUG tracepoints, currently 50, has been reached. **Action:** Remove tracepoints or wait until other users have finished tracing calls, and then retry the command.
`**** WARNING **** CPU availability for call processing is less than 25% CALLTRAK will degrade call processing. Do you really want to continue? Enter YES or NO`	**Meaning:** The switch is under a high call processing load. The warning is a safety mechanism to prevent CallTrak from being started under the high load. **Action:** To initiate call tracing type yes. To abort the start command type no.
`CALLTRAK is NOT STARTED because CPU availability for call processing is too low.`	**Meaning:** The user responded to the low CPU availability warning by aborting the start request. CallTrak was not started. **Action:** None
`PGMTRACE Error: Illegal instruction at that location. Address= aaaaaa`	**Meaning:** The tracepoint at address aaaaaa does not refer to a valid machine instruction and could not be inserted into program store. **Action:** Move the tracepoint to a valid location, and then retry the command. To move the tracepoint, check the procedure offset listing, or disassemble the procedure to find a valid location.

-continued-

start

Responses for the start command (continued)

MAP output	Meaning and action
`WARNING:  TIMECALL and PGMTRACE are both ON` `TIMECALL timings are more accurate with PGMTRACE OFF.`	**Meaning:** PGMTRACE procedure timing interacts with TIMECALL timing, and affects the TIMECALL times. To obtain accurate TIMECALL times, the user should run TIMECALL without PGMTRACE. **Action:** Turn PGMTRACE off and start the tracing. Trace calls a second time with TIMECALL off and PGMTRACE on to get the PGMTRACE trace.
-end-	

status

Function

Use the status command to display a list of the selected terminals, and the available tools.

status command parameters and variables

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

Examples

Examples of the status command

Example	Task, response, and explanation
status ↵	**Task:** Display the status of the user. **Response:** `A sample response is shown in the Responses section.` **Explanation:** The selected terminals and available tools will be displayed.

status

Responses

Responses for the status command

MAP output	Meaning and action

```
USER:  OPERATOR                    TRACING:  Disabled
Terminals Selected:
      NODE        TERMINAL          AGENT
        22            147     LEN HOST 00 1 04 18    DN 7224111
        12            103     LEN HOST 00 0 03 5    DN 7224546
         5             45     DTC 0 1 12              OTMF1   5
TOPS VCCT Selection Enabled.
ALL BOUNCE messages selected
TIMECALL:         ON
   Buffersize:    256
   #Entries:      0
   Display Opts:  PARTAVG
PGMTRACE:         OFF   -  Display MERGEable
   Buffersize:    0
   #Entries:      0
   Trace EVENTs:   PROC
   Display EVENTs:  PROC
   No INCLUDES/EXCLUDES Selected
   No TRACEPOINTS defined
   Display Opts:  TIMESORT, BRIEF
   Timing is OFF with Stabilizer OFF
MSGTRACE:         OFF   -  Display MERGEable
   Buffersize:    Short:  0   Long:  0
   #Entries:      Short:  0   Long:  0
   Display Opts:  TIMESORT
```

Meaning: The status of the user has been displayed.

Action: None

stop

Function

Use the stop command to stop call tracing activity.

stop command parameters and variables

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

Qualifications

The stop command is optional. The display command will also stop the tracing process.

Examples

Examples of the stop command

Example	Task, response, and explanation
stop ↵	
	Task: Stop call tracing activity. **Response:** `Stopped` **Explanation:** Tracing was successfully stopped.

Responses

Responses for the stop command

MAP output	Meaning and action
`Stopped`	
	Meaning: Call tracing was successfully stopped. **Action:** None
`Tracing not on`	
	Meaning: Call tracing was not previously on. **Action:** None

tidtovid

Function

Use the tidtovid command to map the specified terminal identifier (TID) to one or more associated virtual terminal identifiers (VIDs).

tidtovid command parameters and variables

Command	Parameters and variables		
tidtovid	*node*	*term*	*extbyte*

Parameters and variables	Description
extbyte	This variable, if specified, limits the tidtovid command to the VID corresponding to the TID and extension byte.
node	This variable specifies the node number.
term	This variable specifies the terminal number.

Qualifications

The tidtovid command works well for finding the VID of a key on an EBS or ISDN set.

Examples

Examples of the tidtovid command

Example **Task, response, and explanation**

tidtovid ***26 49*** ↵
where

26 is the the node number.
49 is terminal number.

Task: Map a TID to VID.

Response:
```
See the first output sample in the Responses
section.
```

Explanation: Display the TID to VID mapping for an EBS set. The user did not specify the extension byte. The tidtovid returns all key VIDs corresponding to the VID.

-continued-

tidtovid

Examples of the tidtovid command (continued)

Example **Task, response, and explanation**

tidtovid ***26 49 1*** ↵
where

26 is the the node number.
49 is terminal number.
1 is the extension byte.

Task: Map a TID to VID.

Response: `See the second output sample in the Responses section.`

Explanation: Display the TID to VID mapping for an EBS set. The user specified the extension byte. The tidtovid returns the VID corresponding to the specified TID and byte.

tidtovid ***82 3 1*** ↵
where

82 is the the node number.
2 is terminal number.
1 is the extension byte.

Task: Map a TID to VID.

Response: `See the third output sample in the Responses section.`

Explanation: Display the TID to VID mapping for an ISDN set. The user specified the extension byte. The tidtovid returns the VID corresponding to the specified TID and byte.

-end-

Responses

Responses for the tidtovid command

MAP output	Meaning and action

```
  TID   Extbyte   <==>  VID      Description
26  49     0      <==> 98   1  LEN HOST 00 0 01 16     DN 7227000 KEY: 1
26  49     1      <==> 98   2  LEN HOST 00 0 01 16     DN 7227001 KEY: 2
26  49     2      <==> 98   3  LEN HOST 00 0 01 16     DN 7227002 KEY: 3
26  49     3      <==> 98   4  LEN HOST 00 0 01 16     DN 7227003 KEY: 4
```

Meaning: The output for a TID to VID mapping of an EBS set. The command returned all key VIDs corresponding to the TID.

Action: None

```
  TID   Extbyte   <==>  VID      Description
26  49     1      <==> 98   2  LEN HOST 00 0 01 16     DN 7227001 KEY: 2
```

Meaning: The output for a TID to VID mapping of an EBS set. The command returned the VID for a specified extension byte.

Action: None

```
  TID   Extbyte   <==>  VID      Description
82   3     1      <==> 93  28  LEN HOST 01 1 00 02         NO DIRN
  LTID  Extbyte   <==> KeyVID    Description
93  28     0      <==> 56 200  LEN HOST 01 1 00 02     DN 7225034 KEY: 1
93  28     1      <==> 56 201  LEN HOST 01 1 00 02     DN 7225032 KEY: 2
93  28     2      <==> 56 202  LEN HOST 01 1 00 02     DN 7225035 KEY: 3
93  28    31      <==> 56 279  LEN HOST 01 1 00 02     DN 7225034 KEY: 32
93  28    33      <==> 56 280  LEN HOST 01 1 00 02     DN 7225034 KEY: 34
```

Meaning: The output for a TID to VID mapping of an ISDN set. The user supplied the TID of the loop and the extension byte of one of the ISDN sets on the loop. For the output, two levels of mapping are performed. First the VID corresponding to the set specified by the extension byte is shown. Second, all of the keyvids on that set are shown.

Action: None

-continued-

tidtovid

Responses for the tidtovid command (continued)

MAP output	Meaning and action
`Invalid TID`	
	Meaning: The node and terminal number specified do not comprise a valid terminal identifier.
	Action: Check the TID and retry the command.
`TID is a VID`	
	Meaning: The node and terminal number specified is actually a VID. For ISDN sets mapping is still done, if possible.
	Action: None
`ERROR: Could not convert TID to VID`	
	Meaning: The mapping could not be done. Either the TID does not use VID mapping, or the TID does not use incoming mapping.
	Action: Check the TID and retry the command.

-end-

timecall

Function

Use the timecall command to collect and output call timing information.

timecall command parameters and variables

Command	Parameters and variables	
timecall	query	(1)
	bufsize *buffer_size*	(2)
	display [partavg | fullavg | noavg | summary] [nowait *device* [file *filename*]]	(3)
		(4)
		(5)
		(6)
	displayopts [query | set [partavg | fullavg | noavg | summary] | clear]	(7)
		(8)
		(9)
		(10)
		(11)
		(12)
	off	(13)
	on	(14)
timecall (continued)	(3) delete noreply	(end)

Parameters and variables	Description
bufsize	This parameter, if other parameters are not included, queries the buffer size. If other parameters are included, the size of the buffer will be changed. If set to zero, the buffer is deallocated.
buffer_size	This variable, if included, sets the buffer to the specified size. If not entered, the current buffer size will be displayed. If set to zero, the buffer is deallocated.
clear	This parameter clears the default display option.
delete	This parameter, if included, removes the session containing the data once the display is completed.
device	This variable specifies the device to display data on.
display	This parameter displays all of the tracing data collected by TIMECALL. See the section on TIMECALL in the “Interpreting CallTrak Output” chapter for information on what will be displayed.

-continued-

timecall

timecall command parameters and variables (continued)

Parameters and variables	Description
displayopts	This parameter controls the default display options.
delete	If included, the session containing the data will be removed once the display is completed.
file	This parameter specifies that a file will be worked with.
filename	This variable specifies the name of the file to be worked with.
fullavg	This parameter averages the collected calls together if they are the same type of call. Only the agent types are checked when averaging. For example, call of the same type, but between different directory numbers would be averaged together.
noavg	This parameter stops the averaging of calls.
noreply	This parameter, if included, prohibits the system from informing the the user when the display is complete.
nowait	This parameter creates a copy of the session and displays the data from the copy. The nowait option may be used with any other option to display a session in the background.
off	This parameter disables TIMECALL. This command does not clear or deallocate the TIMECALL buffer.
on	This parameter enables TIMECALL. If a TIMECALL buffer has not been allocated, the buffer is set to 256 entries.
partavg	This parameter averages the collected calls together only if the agents in the list are the same. For example, calls of the same type, but between different DNs would not be averaged together.
query	This parameter queries the status.
set	This parameter is used in conjunction with an option to set the default display option.
summary	This parameter averages calls in the same way that fullavg does; however, the calls are not displayed. Only a count of the number of calls and the call types are shown.

-end-

Qualifications

Changing the buffer size causes a new buffer to be allocated and the old buffer to be deallocated. TIMECALL attempts to copy data into the new buffer from the old buffer before deallocating the old buffer. When the buffer is reallocated to a smaller size, TIMECALL indicates the number of data entries lost.

If an option is not specified when the data is displayed, the display will default to the option set by the displayopt command.

Examples

Examples of the timecall command

Example	Task, response, and explanation
timecall ↵	
	Task: Query the status of TIMECALL. **Response:** `TIMECALL:  Enabled` **Explanation:** TIMECALL has been enabled.
timecall bufsize 500 ↵ *where* 500 *is the number of entries to allocate to the buffer.*	
	Task: Allocate a buffer of 500 entries. **Response:** `26 Data entries have been lost` **Explanation:** The previous buffer held 526 data entries. 26 entries could not be transferred to the new buffer.
timecall displayopts set summary ↵	
	Task: Set the display option to summary. **Response:** `Display opts:  SUMMARY` **Explanation:** The display option has been set to summary.

-continued-

timecall

Examples of the timecall command (continued)

Example	Task, response, and explanation
timecall display ↵	
	Task: Display the data collected by TIMECALL.
	Response: `See the chapter clarifying TIMECALL output for a sample response.`
	Explanation: See the chapter clarifying TIMECALL output for an explanation of the output.

-end-

Responses

Responses for the timecall command

MAP output	Meaning and action
`TIMECALL: Enabled`	
	Meaning: TIMECALL has been enabled.
	Action: None
`TIMECALL: Disabled`	
	Meaning: TIMECALL has been disabled.
	Action: None

`TIMECALL: On`
`Buffersize: 256`
`#Entries: 0`
`Display Opts: SUMMARY`	

	Meaning: The status of TIMECALL is being shown, including the current buffersize, the number of entries collected to date, and the display options.
	Action: None

-continued-

timecall

Responses for the timecall command (continued)

MAP output	Meaning and action
`Buffersize:  256`	**Meaning:** The current buffersize is 256 entries. **Action:** None
`nn Data entries have been lost`	**Meaning:** When the buffer was reallocated to a smaller size, nn entries were not transferred. **Action:** None
`Tracing started - use STOP first`	**Meaning:** TIMECALL may not be enabled or disabled while any tool is tracing calls. **Action:** Use the stop command to stop call tracing. Then retry the command.
`Cannot allocate enough memory`	**Meaning:** The system could not allocate enough memory (DS temp) for the data buffer. **Action:** Try allocating a smaller user buffer, then retry the command.

`**WARNING - Firmware timing is in use`
`**  You will be able to SELECT and REMOVE terminals`
`**  but you cannot start tracing until the other`
`**  tools is turned off.`
	Meaning: The timing base is already in use.
	
	Action: Wait until the other tool stops.

`** ERROR: only one option can be selected`	**Meaning:** Only one TIMECALL display option may be selected at a time. **Action:** Retry the command, selecting only one option.

-continued-

timecall

Responses for the timecall command (continued)

MAP output	Meaning and action
`Display Opts: aaaaaaa`	**Meaning:** The current display option is shown. **Action:** None
`Nothing to Display`	**Meaning:** TIMECALL has not collected any data. **Action:** Verify that call tracing has been started, and the proper terminals have been selected. Then retry the command.
`Tracing has been stopped`	**Meaning:** Call tracing has been stopped. The data output will follow. **Action:** None

-end-

vidtotid

Function

Use the vidtotid command to map the specified VID to the corresponding TID and extension byte.

vidtotid command parameters and variables

Command	Parameters and variables	
vidtotid	*vnode*	*vterm*

Parameters and variables	Description
vnode	This variable specifies the virtual node number.
vterm	This variable specifies the virtual terminal number.

Qualifications

The vidtotid command is used to find the terminal identifier of the loop on which an EBS or ISDN set is resident. Once found, the TID may be used in conjunction with the tidtovid command to find all of the other VIDs on the same TID.

Examples

Examples of the vidtotid command

Example **Task, response, and explanation**

vidtotid ***98 4*** ↵
where

98 is the virtual node number.
4 is the virtual terminal number.

Task: Map an EBS VID to the appropriate TID.

Response: `See the first example in the Responses section.`

Explanation: The output shows the TID, the extension byte, and a description.

vidtotid

Responses

Responses for the vidtotid command

MAP output	Meaning and action
VID <==> TID ExtByte Description 98 4 <==> 26 49 3 LEN HOST 00 0 02 16 DN 7227000	**Meaning:** The VID was successfully mapped to the TID. **Action:** None
Invalid VID	**Meaning:** The node number and terminal number specified do not comprise a valid VID. **Action:** Check the VID and retry the command.
ERROR: Could not convert VID to TID	**Meaning:** The VID could not be mapped to the TID. **Action:** None

List of terms

alltools

A command used by CallTrak to enable or disable all of the available tools.

AMA

Automatic message accounting

automatic message accounting (AMA)

An automatic recording system that documents all the necessary billing data of subscriber-dialed long distance calls.

batch change supplement (BCS)

A DMS-100 Family software release.

BCS

Batch change supplement

Bell-Northern Research (BNR)

Part of the tri-corporate structure consisting of Bell Canada, Northern Telecom, and Bell-Northern Research.

BNR

Bell-Northern Research

busy signal

1. An audible signal, a flashing signal, or both, often 60 impulses per minute, indicating that the called number is unavailable. 2. A signal, transmitted at 120 impulses per minute, indicating that all voice paths are temporarily unavailable.

call

In a DMS, any demand to set up a connection through the switch. Also used as a unit of telephone traffic. Synonymous with cue.

call duration

The interval of time between the moment a connection is established between the calling and called stations and the moment the calling station

gives the clearing signal (or the moment that the connection is taken down by the operator).

called number

The number of the party being called.

calling number

The number of the party initiating the call.

call processing (CP)

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

card

A plug-in circuit pack containing components. In a DMS, card is the preferred term for a printed circuit pack or a printed circuit board.

carrier

1. In a DMS, the communications links between switching offices. 2. The protocol by which these links communicate.

CC

Central control

central control (CC)

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

CI

Command interpreter level

CLLI

Common language location identifier

CM

Communications module, computing module, connection memory

CP

Call processing

command

1. A control signal. 2. In user interface language, the specification of an expected action or function by the system.

command interpreter (CI) level

Initial MAP level where commands are entered.

common language location identifier (CLLI)

A standard identification method for trunk groups in the form:

aaaa bb xx yyyy

Where:

aaaa=City code
bb=Province or state code
xx=Trunk group identity
yyyy=Trunk number

See also short common language location identifier.

computing module (CM)

The processor or memory complex of DMS SuperNode.

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.

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.

display

A command used by CallTrak, MSGTRACE, PGMTRACE, and TIMECALL to show the output.

DMS

Digital Multiplex System

DMS SuperNode

A central control complex for the DMS-100. The two major components of DMS SuperNode are the computing module and the message switch. Both are compatible with the current network module, the input/output controller, and XMS-based peripheral modules.

DN

Directory number

EAEO

Equal access end office

EBS

Electronic business set

electronic business set (EBS)

A telephone set that provides subscribers with push-button access to various business features.

electronic telephone set (ETS)

An alternate name for electronic business set.

equal access end office (EAEO)

A central office that provides access to several long distance carriers.

ETS

Electronic telephone set

IBN

Integrated Business Network

IBN EBS

Integrated Business Network electronic business set

Integrated Business Network (IBN)

Now known as Meridian Digital Centrex. A special DMS business services package that utilizes the data-handling capabilities of a DMS-100 Family office to provide a centralized telephone exchange service. Many optional features also are available.

Integrated Business Network electronic business set (IBN EBS)

A control device with addressable points (for example, directory number keys, feature keys, and display units). Business sets can support multiple simultaneous calls. They also can support premium voice features and low-speed data service.

integrated services digital network (ISDN)

A set of standards proposed by the CCITT to establish compatibility between the telephone network and various data terminals and devices. ISDN is a communications network that provides access to voice, data, and imaging services from a single type of connector.

ISDN

Integrated services digital network

LEN

Line equipment number

line equipment number (LEN)

A 7-digit function reference used to identify line circuits.

logical terminal

The datafilled instance of an abstract terminal that is provided with a subset of the features and services (service profile) datafilled in the access termination for the abstract terminal.

logical terminal identifier (LTID)

The unique identifier that is assigned to a logical terminal when it is datafilled in the ISDN access termination.

log report

A message from the DMS whenever a significant event has occurred in the switch or one of its peripherals. A log report includes state and activity reports as well as reports on hardware and software faults, test results, and other events or conditions likely to affect the performance of the switch. A log report can be generated in response to a system or manual action.

loop (LP)

1. A local circuit between a central office and a subscriber telephone station. Synonymous with subscriber loop and local loop. 2. A signaling method whereby on-hook/off-hook signals are transmitted by bridging the loop on a two-wire trunk or circuit. Signals are received by detecting the flow of loop current. In a trunk, LP signaling occurs in one direction at a time.

LP

Loop

LTID

Logical terminal identifier

MAP

The maintenance and administration position. MAP is a group of components that provides a user 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.

MAPCI

MAP command interpreter

MAP command interpreter (MAPCI)

A MAP level for accessing maintenance and other functional levels.

message (MSG)

The unit of information transfer between nodes in the DMS-100 system. A message is incoming if it is sent from a peripheral to central control (CC) and outgoing if it is sent from CC to a peripheral.

A message is a type of control mechanism used in the input/output message system of the DMS-100 Family. The MSG byte specifies that the information to come is a data message.

message type

Identifies the function of a message. Stimulus call control has only one message type-information. Functional call control has a number of message types related to call establishment, call dis-establishment, and call status.

MSG

Message

msgtrace

A CallTrak command used to monitor incoming and outgoing messages, both to and from the terminals involved in a traced call.

MSGTRACE

A CallTrak tool used to monitor incoming and outgoing messages, both to and from the terminal involved in a traced call.

node

The terminating point of a link. Node is a relative term in that its meaning depends entirely on the context in which it is used. For example, a circuit can be a node in the context of another circuit within a module; the module itself can be a node in the context of another component of the network, and so forth. Some common applications are

- in network topology, a terminal of any branch of a network or a terminal common to two or more branches of a network
- in a switched communications network, the switching points, including patching and control facilities
- in a data network, the location of a data station that interconnects data transmission lines
- a unit of intelligence within a system; in a DMS, includes the central processing unit, network module, and peripheral modules

Northern Telecom Publication (NTP)

A document that contains descriptive information about DMS-100 Family hardware and software modules and performance-oriented practices for testing and maintaining the system. These documents are supplied as part of the standard documentation package provided to an operating company.

NTP

Northern Telecom Publication

off-hook

1. In telephone operations, the condition existing when the receiver or handset is removed from its hook-switch. 2. One of two possible signaling states: tone or no-tone; ground connection or battery connection. 3. The active state (closed loop) of a subscriber or PBX line loop. *See also* on-hook.

on-hook

1. In telephone operation, the condition existing when the receiver or handset is resting on its hook-switch. 2. One of two possible signaling states such as tone or no-tone or ground connection or battery connection. 3. The idle state (open loop) of a subscriber or PBX line loop. *See also* off-hook.

peripheral module (PM)

A generic term referring to all hardware modules of DMS-100 Family systems that provide interfaces with external line, trunk, or service facilities. A PM contains peripheral processors, which perform local routines, thus relieving the load on the central processing unit.

peripheral module intercept system test (PMIST)

A debugging tool that traces messages between the peripheral modules.

pgmtrace

A CallTrak command used to trace call processes through portions of the program code.

PGMTRACE

A CallTrak tool used to trace call processes through portions of the program code.

plain ordinary telephone service (POTS)

Basic conventional telephone service. In the context of service screening, POTS is a pseudo-service that is derived from the combination of a bearer service of speech with no supplementary services.

PM

Peripheral module

PMIST

Peripheral module intercept system test

POTS

Plain ordinary telephone service

process entry module

A module that contains a procedure where a process begins running after initialization.

PROTEL

Procedure-oriented type enforcing language

qdn

A command used at the CI level to query the directory number of a terminal.

qlen

A command used at the CI level to query the LEN of a terminal.

quit

A CallTrak command used to leave the CallTrak environment and return to the CI level.

realtime

The actual time during which the CPU (NT40) or DMS-Core SuperNode performs its functions. The time is divided into two main categories: call processing time and noncall processing time.

remove

A CallTrak command used to deselect originating terminals.

run time

In a DMS, the time during which the central processing unit is allocated to a process.

select

A CallTrak command used to select originating terminals.

service order system (SERVORD)

A user interface used to change, add, or delete a subscriber line. Standard telephone industry command format is used.

SERVORD

Service order system

session

A CallTrak command used to control the virtual session capability.

SOS

Support operating system

start

A CallTrak command used to begin tracing activity originating from the selected terminals.

status

A CallTrak command used to display a list of the terminals selected, and the tools available.

stop

A CallTrak command used to stop call tracing.

support operating system (SOS)

The software that sets up the environment for loading and executing the application software in the DMS-100 Family system. SOS includes the nucleus, file system, command interpreter, and loader.

terminal

1. The point of origination or termination in a communications network. 2. Any device capable of sending information, receiving information, or both over a communication channel. 3. In a DMS, the smallest unit of address space within the input/output system.

terminal ID (TID)

In DMS software, the TID uniquely identifies anything on which a call can be originated or terminated.

TID

Terminal ID

tidtovid

A CallTrak command used to map a specified TID to one or more associated VIDs.

timecall

A CallTrak command used to collect and output call timing information.

TIMECALL

A CallTrak tool used to collect and output call timing information.

TOPS

Traffic operator position system

traffic operator position system (TOPS)

A call processing system made up of a number of operator positions. Each operator position consists of a visual display unit (VDU), a controller, a keyboard, and a headset. TOPS is a trademark of Northern Telecom.

tuple

The horizontal row of a table.

user

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

VID

Virtual identifier.

vidtotid

A CallTrak command used to map a specified VID to the corresponding TID.

virtual circuit

In packet switching, a network facility used for transferring data between those data stations emulating physically-connected stations.

Virtual identifier

A node and terminal number used to identifier agents on loops containing multiple TIDs per terminal.

DMS-100 Family

CallTrak User Guide

Technical Assistance Manual

Publication number: TAM-1001-012
Product release: BCS34 and up
Document release: Standard 01.01
Date: July 1992

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