Critical Release Notice

Publication number: 297-1001-360
Publication release: Standard 06.02

The content of this customer NTP supports the SN06 (DMS) and ISN06 (TDM) software releases.

Bookmarks used in this NTP highlight the changes between the baseline NTP and the current release. The bookmarks provided are color-coded to identify release-specific content changes. NTP volumes that do not contain bookmarks indicate that the baseline NTP remains unchanged and is valid for the current release.

Bookmark Color Legend

Black: Applies to new or modified content for the baseline NTP that is valid through the current release.

Red: Applies to new or modified content for NA017/ISN04 (TDM) that is valid through the current release.

Blue: Applies to new or modified content for NA018 (SN05 DMS)/ISN05 (TDM) that is valid through the current release.

Green: Applies to new or modified content for SN06 (DMS)/ISN06 (TDM) that is valid through the current release.

Attention!
Adobe® Acrobat® Reader™ 5.0 is required to view bookmarks in color.

Publication History

March 2004

Standard release 06.02 for software release SN06 (DMS) and ISN06 (TDM).

Change of phone number from 1-800-684-2273 to 1-877-662-5669, Option 4 + 1.

297-1001-360

DMS-100 Family

Basic Translation Tools

Guide

BASE11 Standard 06.01 January 2002

NORTEL NETWORKS
How the world shares ideas.

DMS-100 Family

Basic Translation Tools

Guide

Publication number: 297-1001-360
Product release: BASE11
Document release: Standard 06.01
Date: January 2002

Publication history

January 2002

BASE11 Standard 06.01

Revisions associated with Prostar EP90199.

August 1999

BASE11 Standard 05.01

Added a note to "Translations verification" for CSR EV90244

April 1999

BASE07 Standard 04.05

Editing changes

December 1997

BASE07 Standard 04.04

Editing changes

March 1997

BASE07 Standard 04.03

Revisions associated with a PRS

September 1996

BASE07 Standard 04.02

Editing changes

August 1996

BASE07 Standard 04.01

Added information associated with the GTTUSAGE command. (feature AR1617)

November 1995

CSP02 Standard 03.02

Editing changes and cleanup

November 1995

CSP02 Standard 03.01

Applicable CSP05 features added.

July 1992

BCS34 Standard 02.01

- Revised "Pending Order Subsystem"; to improve usability.
- Added information on commands DUMPTAB and DMOPRO.
- Added the store file PRINT command.

October 1991

BCS33 Standard 01.01 First release of this document

Contents

About this document

When to use this document

This describes translations and provides an overview of the contents of this book. DMS-100 Family translations involve the reading of selected tuples in designated data tables to determine the screening or routing that a call is to receive. This screening and routing is accomplished through the software of the switch and the datafill of the tables and subtables.

How to check the version and issue of this document

The version and issue of the document are indicated by numbers, for example, 01.01.

The first two digits indicate the version. The version number increases each time the document is updated to support a new software release. For example, the first release of a document is 01.01. In the *next* software release cycle, the first release of the same document is 02.01.

The last two digits indicate the issue. The issue number increases each time the document is revised but rereleased in the *same* software release cycle. For example, the second release of a document in the same software release cycle is 01.02.

More than one version of this document may exist. To determine whether you have the document that applies to the software in your office, check the release information in *North American DMS-100 Northern Telecom Publications Cancellation Index*, 297-1001-002.

References in this document

The following documents are referred to in this document:

- *DMS-100 Family Commands Reference Manual*
- *Servord Reference Manual*

What precautionary messages mean

The types of precautionary messages used in NT documents include danger, warning, and caution messages. Danger, warning, and caution messages indicate possible risks.

Examples of the precautionary messages follow.

DANGER

Possibility of personal injury

DANGER

Risk of electrocution

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

WARNING

Possibility of equipment damage

DANGER

Damage to the backplane connector pins

Align the card before seating it, to avoid bending the backplane connector pins. Use light thumb pressure to align the card with the connectors. Next, use the levers on the card to seat the card into the connectors.

CAUTION

Possibility of service interruption or degradation

CAUTION

Possible loss of service

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

How commands, parameters, and responses are represented

Commands, parameters, and responses in this document conform to the following conventions.

Input prompt (>)

An input prompt (>) indicates that the information that follows is a command:

```
>BSY
```

Commands and fixed parameters

Commands and fixed parameters that are entered at a MAP terminal are shown in uppercase letters:

```
>BSY CTRL
```

Variables

Variables are shown in lowercase letters:

```
>BSY CTRL ctrl_no
```

The letters or numbers that the variable represents must be entered. Each variable is explained in a list that follows the command string.

Responses

Responses correspond to the MAP display and are shown in a different type:

```
FP 3 Busy CTRL 0: Command request has been submitted.
FP 3 Busy CTRL 0: Command passed.
```

The following excerpt from a procedure shows the command syntax used in this document:

1 Manually busy the CTRL on the inactive plane by typing

`>BSY CTRL ctrl_no`

and pressing the Enter key.

where

ctrl_no
is the number of the CTRL (0 or 1)

Example of a MAP response:

```
FP 3 Busy CTRL 0: Command request has been submitted.
FP 3 Busy CTRL 0: Command passed.
```

1 Translations tools overview

This chapter describes translations and provides an overview of the contents of this document.

What are translations?

DMS-100 Family translations involve the reading of selected tuples in designated data tables to determine the screening or routing that a call is to receive. This screening and routing is accomplished through the software of the switch and the datafill of the tables and subtables.

Reference tools

Reference tools are the tools available to the translator that are not part of the software of the switch. Specifically available are Northern Telecom documentation and telecommunications courses offered by the Raleigh Technical Education Center.

Translations tools

Translations tools are the tools available in the DMS-100 Family software. Tools of this type are table editor, Pending Order subsystem, translations verification (TRAVER), Service Order system (SERVORD), and a variety of utility commands.

This guide discusses each of these translations tools with special emphasis on table editor. Some of the tools, such as TRAVER and SERVORD, are discussed in detail in other publications. The purpose of this guide is to serve as a quick reference for some of the most commonly used translations tools and to familiarize the user who is new at translations with the translations environment.

2 Pending Order subsystem

This section describes the Pending Order (PO) subsystem as it applies to the DMS-100 Family switches.

The PO subsystem provides the means for storing and manipulating orders, such as service orders (SO) and data modification orders (DMO), previously created by the user. These orders are classified as pending orders (PO), each with its own unique PO file identifier (POFID).

The PO subsystem enables the user to

- manually activate POs singly or collectively by POF type, POFID, or due date
- display the contents of POs singly by POFID, collectively, or by due date
- delete the POs from the PO subsystem singly or collectively by POF type, POFID, or due date
- place any previously produced DMO file into the PO subsystem.

The command DMOPRO works with the PO subsystem. For instance, after creating a POF, the user can activate it with the DMOPRO command.

Pending order types

Two types of POs are presently handled in the PO subsystem. They are pending service order files (PSOF) and data order files (DOF).

- Pending service order files Order files generated by the Service Order system (SERVORD), including bulk SOs, are placed in a PSOF. In order to process pending SOs, the user must be in the SERVORD environment before accessing the PO subsystem. (Refer to *SERVORD Reference Manual*.)
- Data order files Data order files are DMOs generated in the POF mode of table editor. They can also be externally created files or bulk DMOs such as those created by the command DUMPTAB or in the system store file editor.

Data order files are normally processed into the system using the CI command DMOPRO. If the DOFs are to be processed at a future date or time, the user must convert the DMO file into a POF using the CREATE command while in the pending order environment.

Pending order states

There are three ways a PO may be classified:

- pending - The PO is due to be activated at some future date, as indicated by the date and time associated with its POFID.
- prompt - The PO has its time slot specified in days in the last data field in the POFID. During this time slot, the operating company is notified of the approaching PO due date. (See "Log Reports" below.)
- due - The PO's date and time is due to be activated, according to the operating company. (See "Log Reports" below.)

Log reports

Twice daily, at midnight and noon, a PO audit is recorded and one or both of the following log reports may be generated:

- PEND100 - This log is generated when at least one PO is due or past due within the next 12 hours.
- PEND101 - This log is generated when at least one PO has entered the prompt range.

Note: A list of POs applicable to PEND100 and a list of those applicable to PEND101 can be obtained using the DUE parameter with the DISPLAY command.

File disposition prompts

After the execution of the ACTIVATE and DELETE commands, the system prompts the user to respond to the following two displays regarding the disposition of the store file device (SFDEV) file and the POF:

```
DO YOU WANT TO ERASE SFDEV FILE? (Y/N)
DO YOU WANT TO DELETE POF? (Y/N)
```

To save time when more than one PO is being processed during the same session, the user can suspend these prompts by inserting a no-prompt (NP) parameter in the ACTIVATE and DELETE commands.

Whether the SFDEV file is erased or not is determined by operating company policy. Therefore, manipulation of the SFDEV file is not covered in this document.

Whether the POF file is deleted or not is determined by personnel responsible for the PO. If the POF is not deleted, it is retained in the PO subsystem, but it is no longer pending. The retained order can be deleted later using the DELETE command.

Pending order subsystem store limits

The PO subsystem store capacity is initially set to 45,000 words of protected data store, which allows for approximately 2,000 POs. Table DSLIMIT (Data Store Limit) defines the allocated data store space. If additional memory is required, only technical support personnel are authorized to allocate an increase.

The user enters new POFs in table NPENDING. The DMS automatically updates table DMOTAB. This table has a limit of 4096 tuples, and the user must understand that one POF can create several tuples in table DMOTAB.

Creating a file

Batch files can be created in table editor using the POF command or with the SFDEV EDIT command. In table editor, enter the PO subsystem by typing the command POF. Any further commands are recorded in the POF to be activated or manipulated later. To end a pending order session, type ENDP. The procedure for creating a PO subsystem file while in table editor and an example MAP display follow.

Procedure for creating a PO subsystem while in table editor

At the MAP display:

1 Create a PO file while in table editor by typing

```
>POF SFDEV file_name
```

and pressing the Enter key.

where

file_name
is the name of the pending order SFDEV file you wish to create

2 When finished entering desired table manipulations, end the PO session by typing

```
>ENDP
```

and pressing the Enter key.

Figure 2-1 Example of a MAP display of creating a PO subsystem file while in table editor

```
CI:
 >table clli
 TABLE: CLLI
 >pof sfdev A1234
 >add
 CLLI:
 >apexncout
 ADNUM:
 >98
 TRKGRSIZ:
 >32
 ADMININF:
 >apex_nc_out
 TUPLE TO BE ADDED:
    APEXNCOUT   98   32   APEX_NC_OUT
 ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
 >y
 DMO ADDED TO POF
 >endp
```

A file created while using POF can contain the table editor commands TABLE, SUBTABLE, RETURN, POSITION, ADD, DELETE, REPLACE, and QUIT. It can also contain the symbols % followed by comments, and + to indicate the continuance of a line of input.

After creating a store file using the POF command, the user can verify its presence in store file by executing a LISTSF command. The user can also verify the format of the file by executing a PRINT command followed by the file name. Instead of using POF in table editor, a file may also be created using the store file EDIT command.

Accessing the PO subsystem

Once a file is created using either the store file EDIT command or the table editor POF command, the user can then convert the file into a PO subsystem file. If the file has been built using acceptable DMOPRO commands, the user can verify the file with the DMOVER command and run it immediately with the DMOPRO command. If the file has been created using table editor commands, then the user must convert it by entering the PO subsystem and executing a CREATE command on the file. The procedure for converting a file into PO subsystem format and an example MAP display follows

Procedure for converting a file into PO subsystem format

At the MAP display:

1 Ensure the existence of the store file by typing

`>LISTSF`

and pressing the Enter key.

2 Enter the PO subsystem environment by typing

`>PENDING$`

and pressing the Enter key.

where

$
serves as a null entry, allowing the user to enter the PO subsystem without specifying a POFID

3 Execute the CREATE command by typing

`>CREATE file_name year month day time prompt`

and pressing the Enter key.

where

file_name
is the name of the SFDEV or POF table editor file created

year
is the year the file will run (for example, 1992)

month
is the month the file will run (for example, SEP)

day
is the day the file will run (for example, 10)

time
is the time the file will run (for example, 315)

prompt
is the number of days before the due date that the operating company will be notified (for example, 1)

4 Ensure the created POF by typing

`>DISPLAY POF pofid`

and pressing the Enter key.

where

pofid
is the name of the POF file created. It is the same as the file name used above

Note: What was an SFDEV file name is now a PO subsystem POFID.

5 Exit the PO subsystem by typing

`>LEAVE`

and pressing the Enter key.

Figure 2-2 Example of a MAP display converting a file into PO subsystem format

```
CI:
>listsf
A1234
>pending $
PENDING:
>create a1234 1992 sep 10 315 1
>display pof a1234
PENDING:
           A1234   1992  SEP  10  315  1
    10   DMOPRO A1234
>leave
>
```

Once the file is converted into the PO subsystem, it can be specified and manipulated with any of the PO subsystem commands such as ACTIVATE and DISPLAY. The procedure for accessing the PO subsystem using an existing POFID and an example MAP display follow.

Procedure for accessing the PO subsystem using an existing POFID

At the MAP display:

1 Enter the PO subsystem environment by typing

>PENDING pofid

and pressing the Enter key.

where

pofid
is the name of an existing POF file

Figure 2-3 Example of a MAP display of accessing the PO subsystem using an existing POFID

```
CI:
>pending a1220
PENDING:
      A1220 1992 JUL 14 1201 1
   10  DMOPRO  A1220
>
```

If the user does not enter a POFID parameter, the system prompts for one.

In the preceding MAP display, A1220 appears as the current PO. The entry includes the following information:

year
Year in which the PO is due. The range of valid entries is 1990-2999, inclusive.

month
Month in which the PO is due. Valid entries are JAN, FEB, MAR, APR, MAY, JUN, JULY, AUG, SEP, OCT, NOV, and DEC.

day
Day of the month in which the PO is due. The range of valid entries is 1-31, inclusive.

hour
Time of day (military time) at which the PO is due. The range of valid entries is 0000-2359, inclusive.

prompt
Number of days, prior to the due date, that the operating company is notified of the PO due date.

If the user enters an invalid POFID, the system responds that it cannot find the PO. However, the user can still access the PO subsystem, although without a current PO. The following MAP display shows an example MAP display of this situation.

Figure 2-4 Example of a MAP display of accessing the PO subsystem with an invalid POFID

```
CI:
>pending
PENDING FILE NAME: $
>zyx987
PENDING:
ZYX987 not found
>
```

In the preceding MAP display, the system could not find the POFID entered by the user, zyx987.

Pending order subsystem commands

The commands available in the PO subsystem include ACTIVATE, CREATE, DELETE, DISPLAY, HELP, and LEAVE.

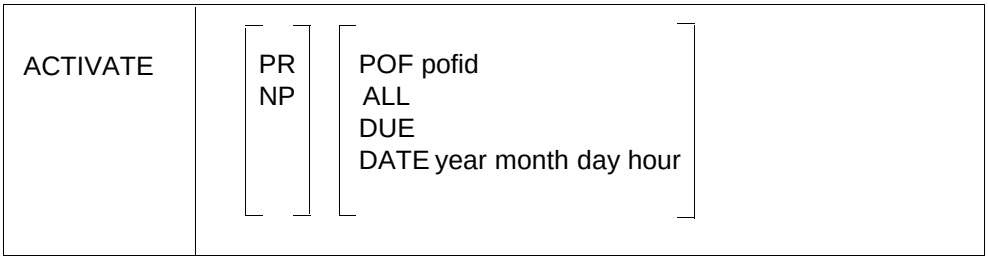
The following sections describe the PO subsystem commands in alphabetical order. An example MAP display is included for each command. However, refer to *SERVORD Reference Manual* for examples that apply specifically to SOs.

In the format used, the command is given in the left box of the diagram. After the command, parameter and variable choices are grouped in brackets. For instance, in the ACTIVATE command, the user chooses PR or NP, and POF with pofid, ALL, DUE, or DATE. The ACTIVATE command can also be executed without parameters.

Note: To use these commands for SO manipulation, the user must be in the SERVORD environment. To enter the SERVORD environment, type SERVORD at the system prompt and press the Enter key.

ACTIVATE

Use the ACTIVATE command to activate the POs, as specified by the following parameters.



An explanation of each parameter follows:

PR The following file disposition prompts will be displayed upon completion of the ACTIVATE command. Each prompt requires a yes or no response from the user.

```
DO YOU WANT TO ERASE SFDEV FILE?(Y/N)
DO YOU WANT TO DELETE POF?(Y/N)
```

> ***Note:*** PR is the default parameter. Use of the PR parameter erases both the SFDEV file and the POF. In addition, the POF is erased when activation of the PO fails.

NP The file disposition prompts will not be displayed upon completion of the ACTIVATE command.

POF Only specified POs in the PSOF are activated.

pofid Enter the identification name of the PO to be activated.

ALL All POs in the PO subsystem are to be deleted sequentially by their entered due date and time.

DUE Specified or all POs due prior to and including the present date and time are to be activated, as determined by the system clock.

DATE Enter the time frame when specified or all POs are due prior to and including the date and time specified. The entry must be in a year-month-day-hour format.

Figure 2-5 Example of a MAP display of the ACTIVATE command with the PR parameter

```
CI:
>pending $
>display all
    AB12345 1992    SEP 30  1201    1
    AB12346 1992    AUG 14  1201    1
    AB12347 1992    SEP 30      1   1
>act pr pof AB12346
ACTIVATING POF: AB12346
COPYING POF INTO SFDEV
   NEW AB12346 82 8 14 PM   6211218 1FR+
   HOST 00 0 19 8 DGT$
COMMAND AS ENTERED
NEW AB12346 82 8 14 PM 6211218 1FR HOST 00 0 19 8 DGT$
DO YOU WANT TO ERASE SFDEV FILE? (Y/N)
>y
DO YOU WANT TO DELETE POF? (Y/N)
>y
AB12346 ERASED FROM POF SYSTEM
>leave
CI:
```

If the ACTIVATE command is used without parameters, the current POF is activated. The procedure for using the ACTIVATE command without parameters and an example MAP display follow.

Procedure for using the ACTIVATE command without parameters

At the MAP display:

1 Enter the PO subsystem environment by typing

>PENDING pofid

and pressing the Enter key.

where

pofid
is the name of an existing POF file

2 Execute the ACTIVATE command by typing

>ACTIVATE

and pressing the Enter key.

Figure 2-6 Example of a MAP display of the ACTIVATE command without parameters

```
CI:
>pending a1234
PENDING:
      A1234 1992 DEC 16 315 1
   10  DMOPRO  A1234
>activate
Activating POF: A1234
Copying POF into sfdev
  DMOPRO A1234
--- TABLE: CLLI -- 1 added, 0 replaced, 0 deleted
In total, 3 commands were processed
DO YOU WANT TO DELETE SFDEV FILE? (Y/N)
>y
DO YOU WANT TO DELETE POF? (Y/N)
>y
A1234 erased from POF system
>
```

CREATE

Use the CREATE command to enter the file name of any previously produced DMO file in the PO subsystem, as specified by the following parameters.

CREATE	file_name year month day time [prompt]

An explanation of each parameter follows:

file_name Enter the name of the previously produced DMO file.

year Enter the year in which the PO is due. The range of valid entries is 1990-2999, inclusive.

month Enter the month in which the PO is due. Valid entries are JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, and DEC.

day Enter the day of the month in which the PO is due. The range of valid entries is 1-31, inclusive.

hour Enter the time of the day at which the PO is due. Valid entries include 0000-2359, inclusive.

prompt Enter the number of days, prior to the due date, that the operating company is notified of the PO due date.

Figure 2-7 Example of a MAP display of the CREATE command

```
>create cutover2 1992 DEC 15 1200 10
>
```

See "Procedure for converting a file to PO subsystem format" for an example of the CREATE command.

DELETE

Use the DELETE command to delete POs as specified by the following parameters.

DELETE	[PR / NP] [POF pofid / ALL / DUE DATE year month day hour]

An explanation of each parameter follows:

PR The following file disposition prompts will be displayed upon completion of the DELETE command. Each prompt requires a yes or no response from the user.

```
DO YOU WANT TO ERASE SFDEV FILE?(Y/N)
 DO YOU WANT TO DELETE POF?(Y/N)
```

Note: PR is the default parameter. Use of the PR parameter erases both the SFDEV file and the POF. In addition, the POF is erased when activation of the PO fails.

NP The above-mentioned file disposition prompts will not be displayed upon completion of the DELETE command.

POF Only specified POs in the PSOF are deleted.

pofid Enter the identification name of the PO to be deleted.

ALL All POs in the PO subsystem are to be deleted sequentially by their entered due date and time.

DUE Specified or all POs due prior to and including the present date and time are to be deleted, as determined by the system clock.

DATE Enter the time frame when specified or all POs are due prior to and including the date and time specified. The entry must be in a year-month-day-hour format.

Figure 2-8 Example of a MAP display of the DELETE command with the NP parameter

```
CI:
>pending a1220
PENDING:
      A1220 1992 JUL 14 1201 1
   10  DMOPRO  A1220
>delete np
DELETING POF A1220
A1220 ERASED FROM POF SYSTEM
>
```

If the DELETE command is used without parameters, the current POF is deleted. The procedure for using the DELETE command without parameters and an example MAP display follow.

Procedure for using the DELETE command without parameters

At the MAP display:

1 Enter the PO subsystem environment by typing

`>PENDING pofid`

and pressing the Enter key.

where

pofid
is the name of an existing POF file

2 Execute the DELETE command by typing

`>DELETE`

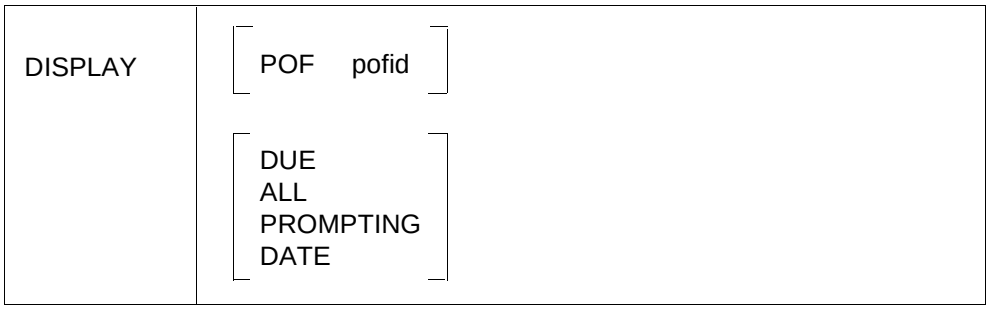
and pressing the Enter key.

Figure 2-9 Example of a MAP display of the DELETE command without parameters

```
CI:
>pending A1234
PENDING:
        A1234 1992 SEP 30 1201 1
    10  DMOPRO A1234
>delete
Deleting POF A1234
Copying POF into SFDEV
DMOPRO A1234
DO YOU WANT TO ERASE SFDEV FILE?  (Y/N)
>y
DO YOU WANT TO DELETE POF?  (Y/N)
>y
A1234 ERASED FROM POF SYSTEM
>leave
CI:
```

DISPLAY

Use the DISPLAY command to display the contents of all or specified POs in the PO subsystem by either the due date or sort option, which uses the parameters TIME and POF.



An explanation of each parameter follows:

POF Only specified POs in the PSOF are to be displayed.

pofid Enter the identification name of the PO to be displayed.

DUE Specified or all POs due prior to and including the present date and time are to be displayed, as determined by the system clock. Information is sorted by time and name.

ALL All POs of the specified type in the PO subsystem are to be displayed sequentially by the entered due date and time. Information is sorted by time and name.

PROMPTING Enter the number of days, prior to the due date, that the pending DMO will enter the prompting range. The range of entries is 0-366, inclusive (0 is the default). Information is sorted by time and name.

DATE Enter the time frame when specified or all POs are due prior to and including the date and time specified. The entry must be in a year-month-day-hour format.

Note: If the user does not include parameters in the DISPLAY command, the system automatically displays the currently accessed PO. An error message is displayed if there is no currently accessed PO.

Figure 2-10 Example of a MAP display of the DISPLAY command:

```
CI:
>pending $
PENDING:
display pof AB12345
      AB12345 1982 SEP 30 1201 1
    1 NEW AN12345 82 9 30 PM 6211217 FR
    2 HOST 00 0 19 7 DGTS
>leave
CI:
```

Figure 2-11 Example of a MAP display of the DISPLAY command with the ALL parameter

```
CI:
>pending $
PENDING:
display all
      AB12345 1982 SEP 30 1201 1
      AB12468 1982 OCT 14 1201 1
      AB12555 1982 SEP 12       1 1
>leave
CI:
```

See "Procedure for converting a file to PO subsystem format" for an example of the DISPLAY command.

HELP

Use the HELP command to display a brief functional description of the desired PO subsystem command, as specified by the following parameters.

HELP	cmd_name

cmd_name Enter the name of the PO command for which descriptive information is requested.

Figure 2-12 Example of a MAP display of the HELP command

```
CI:
>pending $
PENDING:
>help
Gives HELP for Pending System commands:
DISPLAY, ACTIVATE, DELETE, CREATE
Parms: [{DISPLAY,
      DIS,
      ACTIVATE,
      ACT,
      DELETE,
      CREATE (TELCO only),
      CRE (TELCO only),
      PENDING,
      HELP}]
>
```

Note: If the user does not enter a command name, a list of all PO subsystem commands is displayed. The commands presently in the PO system are ACTIVATE, CREATE, DELETE, DISPLAY, HELP, and LEAVE.

LEAVE

Use the LEAVE command to exit the PO subsystem and return to the CI level.

LEAVE	

Figure 2-13 Example of a MAP display of the LEAVE command

```
CI:
>pending ZH55566
PENDING:
ZH55566 NOT FOUND
>leave
CI:
```

See "Procedure for converting a file to PO subsystem format" for an example of the LEAVE command.

Example of automatic PSOF activation

To make PSOF activation automatic create a new file in SFDEV. Datafill table AUTOEXEC to execute file SFDEV.

Figure 2-14 MAP example of automatic PSOF activation

```
NEW FILE EDIT:
>input
INPUT:
>servord
>pending $
>act np psof due
>y
>quit
>quit
>
EDIT:
>save sfdev
EDIT:
>quit
>listsf
SO_POF
>print so_pof
>servord
>pending$
>act np psof due
>y
>quit
>quit
>table autoexec
TABLE: AUTOEXEC
>add autoexec
TIME:
>3 30
DURATION:
>55
USER:
>pending
EXEC:
>so_pof
TUPLE TO BE ADDED
 AUTOEXEC 3 30    55       PENDING        SO_POF
 ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
 >y
 TUPLE CHANGED
 WRITTEN TO JOURNAL FILE AS JF NUMBER 452
```

3 Reference tools

This section describes some of the reference tools available to the user of DMS-100 Family equipment. The foremost reference tool is DMS-100 Family documentation. Another source for learning about DMS-100 Family equipment is the Raleigh Technical Education Center. These reference tools are described in the following sections.

Northern Telecom documentation

The best source for determining individual documentation needs is *DMS-100 and DMS-10 Product Documentation Catalog*, catalog number 50003.15/05-94, Issue 03.01. This catalog contains a list of Northern Telecom publications (NTP) specific to DMS-10 and DMS-100 Family switches and includes a brief description of each document, as well as current prices. To order this product documentation catalog or to get more information, call 1-877-662-5669, Option 4 + 1.

This tools guide is not intended to replace the DMS-10 and DMS-100 product documentation catalog. However, for convenience, brief descriptions of some NTPs that are useful when studying or entering translations are provided below.

- *Translations Guide* This document
  - — provides an overview of the translations system for plain old telephone service (POTS) features and services, as well as information on translations for Common Channel Signaling No. 7 (CCS7) features. Batch change supplement (BCS) change notes and sequence charts in

the document assist operating company personnel in assigning new features.

— describes the translations and service order information for Enhanced 911 Emergency Service (E911).

— provides the procedures for completing service orders, datafilling tables, and datafilling office parameters for all Residential Services base software packages and features.

— provides complete descriptions and datafill for the Meridian Digital Centrex (MDC) features, including system, attendant console, station, and business set features.

— provides an overview of the Remote Switching Center (RSC) table translations. Both integrated services digital network (ISDN) and non-ISDN are described.

— describes the datafill requirements for the Subscriber Module Urban (SMU). Included are a list of the required data tables presented in sequential order; a description of the tables, their fields, and subfields; and specific datafill examples for feature packages SMU-Subscriber Module Urban (NTX387AB) and SMU Special Services (NTX621AB).

For TOPS, the *Translations Guide*

— provides guidelines for those individuals involved with preparation, datafill, and validation activities associated with the DMS TOPS 04 and TOPS MP (Multipurpose).

— provides guidelines for those individuals involved with preparation, datafill, and validation activities associated with the DMS TOPS MPX (Multipurpose Extended) translations.

- See the data schema section of the *Translations Guide* for

— data tables common to DMS-200 (toll) and DMS-100 (local) systems. Each table and its fields are described to assist the operating company in preparing office-dependent data for its switching unit.

— data tables applicable to the Integrated Business Network (IBN). Each table and its fields are described to assist the operating company in preparing office-dependent data for its switching unit.

— data tables applicable to the DMS-100 local system. Each table and its fields are described to assist the operating company in preparing office-dependent data for its switching unit.

  - — data tables applicable to the DMS-200 toll system. Each table and its fields are described to assist the operating company in preparing office-dependent data for its switching unit.
  - — data tables pertaining to the configuration of a DMS switching unit equipped with TOPS. Each table and its fields are described to assist the operating company in preparing office-dependent data for its switching unit.

- *Office Parameters Reference Manual* This document contains the customer data schema tables that pertain to office parameters. It includes sections previously found in Common Customer Data Schema, 297-1001-451.
- *DMS-100 Family Commands Reference Manual* (297-1001-822) This document provides a list of DMS-100 resident non-MAPCI commands together with examples of online help information. It also contains information on translations verification (TRAVER).
- *Feature Description Manual Reference Manual* This document provides feature information for the DMS-100. This feature information is designed to help operating company personnel prepare for insertion of a new BCS load or understand elements of the software. The information would be useful to operating company personnel involved in planning and engineering or maintenance activities.
- *SERVORD Reference Manual* This document describes the procedures for using the Service Order system (SERVORD) and query commands. Service orders provide operating companies with a means of changing directory numbers and adding or removing subscriber lines, line service options, and features. Query commands allow operating company personnel to determine the status of directory numbers.

Raleigh Technical Education Center

The Raleigh Technical Education Center offers over 200 telecommunications courses in support of Integrated Network Systems, Transmission, and Data Network product families. It is another reference tool that allows the translator to gain hands-on experience and professional training in the classroom setting.

To order the catalog of courses offered, call 1-800-992-2303 and ask for Technical Education Course Catalog. This catalog contains further information on the training center and its counterpart in California, names of representatives who will help with determining course needs, and comprehensive information on the courses offered.

4 Service Order system (SERVORD)

Service Order system (SERVORD)

The Service Order system (SERVORD) consists of commands that enable the user to modify existing lines in a DMS. SERVORD is part of the database software of the DMS-100 Family. SERVORD provides the operating company with a rapid and transparent means to perform many operations, such as adding and deleting options from a line, adding or removing lines, and changing aspects of a line such as directory number (DN) or line equipment number (LEN). For detailed information on SERVORD, refer to *SERVORD Reference Manual*.

To enter the SERVORD environment, type SERVORD at the system prompt and press the Enter key. The user can enter this command at any level in the DMS; entering SERVORD at the CI level is not mandatory.

Basic SERVORD commands

This section covers four basic SERVORD commands:

- ADO
- DEO
- OUT
- NEW

With these commands, the user can add and delete options, remove a line from service, and put a line into service.

SERVORD commands can be entered in prompt or no-prompt mode. Refer to "Prompt and no-prompt modes" for an explanation of these terms.

ADO

The ADO (add option) command allows the user to add options to an existing line. After the user enters the ADO command, the system prompts the user for the service order number (SONUMBER). If the user wishes the service order to be immediate, he or she presses the Enter key. If the user wishes to set the service order at a future date, he or she enters the service order number along

with the year, month, and day, plus AM or PM. This entry should follow the format of the SONUMBER prompt. The system then prompts the user for the DN or LEN of the line to which the option is being added.

Finally, the system prompts the user for the option. In the example that follows, the call waiting (CWT) option is added. The option requested determines any further system prompts. Once the addition of an option is completed, the system prompts the user for another option. If the user does not wish to add another option, he or she enters a dollar sign ($).

Figure4-1 SERVORD example of using the ADO command in prompt mode

```
>SERVORD
SO:
>ADO
SONUMBER:  NOW 91 10 17 AM
>$
DN_OR_LEN
>7820019
OPTION:
>CWT
OPTION:
>$
COMMAND AS ENTERED:
ADO NOW 91 10 17 AM 7820019 ( CWT ) $
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT
>Y
1991/10/17 10:39.12.859 THUR. JOURNAL FILE RECORD ID 5
```

Figure4-2 SERVORD example of using the ADO command in no-prompt mode

```
>ADO $ 7820019 CWT $ Y
```

DEO

The DEO (delete option) command allows the user to delete options from an existing line. The system prompts and user actions for the DEO command are very similar to those for ADO. This is shown in the example that follows.

Figure4-3 SERVORD example of using the DEO command in prompt mode

```
>SERVORD
SO:
>DEO
SONUMBER:  NOW 91 10 17 AM
>$
DN_OR_LEN
>7820019
OPTION:
>CWT
OPTION:
>$
COMMAND AS ENTERED:
DEO NOW 91 10 17 AM 7820019 ( CWT ) $
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT
>Y
1991/10/17 10:39.12.859 THUR. JOURNAL FILE RECORD ID 5
```

Figure4-4 SERVORD example of using the DEO command in no-prompt mode

```
>DEO $ 7820019 CWT $ Y
```

OUT

The OUT command allows the user to remove an existing line from service. After a line is removed from service, the DN and LEN that were associated with that line become available for assignment.

When the OUT command is executed, the system prompts the user first for the SONUMBER and then for the DN of the line being removed from service. Next, the user is prompted for the LEN or logical terminal identifier (LTID) of the line being removed from service.

Finally, the system prompts the user for the intercept name, which is the treatment a caller receives if he or she dials the number removed from service. In the example that follows, the intercept name BLDN (blank DN) indicates that the DN is now unassigned.

Figure4-5 SERVORD example of using the OUT command in prompt mode

```
>SERVORD
SO:
>OUT
SONUMBER:  NOW 91 10 17 AM
>$
DN
>7820019
LEN_OR_LTID
>00019
INTERCEPT_NAME:
>BLDN
COMMAND AS ENTERED:
OUT NOW 91 10 17 AM 7820019 HOST 00 0 00 19 BLDN
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT
>Y
1991/10/17 10:42:09.093 THRU.  JOURNAL FILE RECORD ID 6
```

Figure4-6 SERVORD example of using the OUT command in no-prompt mode

```
>OUT $ 7820019 00019 BLDN Y
```

NEW

The NEW command allows the user to establish service to a line. Once a DN and LEN are assigned, they cannot be assigned to other lines.

When the NEW command is executed, the system prompts the user first for the SONUMBER and then for the DN of the line being established. Next, the user is prompted for the line class code (LCC). In the example that follows, 1FR (one flat rate) is used as an LCC. After the LCC prompt, the system prompts the user for the local access and transport area name (LATANAME) and then the line treatment group (LTG). These fields further identify the type of line being established. Finally, the system prompts for the LEN or LTID and for options. In the example that follows, DGT (Digitone) and CWT are added to illustrate how options may be when establishing a line.

Figure4-7 SERVORD example of using the NEW command in prompt mode

```
>SERVORD
SO:
>NEW
SONUMBER:  NOW 91 10 17 AM
>$
DN:
>7820091
LCC:
>1FR
LATANAME
>NILLATA
LTG:  0
>1
LEN_OR_LTID:
>00019
OPTION:
>DGT
OPTION:
>CWT
OPTION:
>$
COMMAND AS ENTERED:
NEW NOW 91 10 17 AM 7820019 1FR NILLATA 1 HOST 00 0 00 19
( DGT ) ( CWT )
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT
>Y
1991/10/17 10:50:47.220 THRU. JOURNAL FILE RECORD ID 7
```

Figure4-8 SERVORD example of using the NEW command in no-prompt mode

```
>NEW $ 7820019 1FR NILLATA 1 00019 DGT CWT $ Y
```

For detailed information on SERVORD, refer to the *SERVORD Reference Manual*.

5 Translations tools

Table editor

This chapter describes table editor. Table editor is a translations tool that is inherent to the DMS-100 Family equipment and allows for the manipulation of tables.

In the DMS switch, the data for a given office is located in software structures known as tables and subtables. Each table in the DMS switch has a unique table structure and contains information relative to its function. For example, Table TERMDEV (Terminal Device) stores information about video display units (VDU), printers, and modems associated with a given switch.

The information contained in tables defines the configuration of a switch. There are over 500 tables at work in the average DMS-100 switch.

The table editor consists of a set of commands that enable the user to create or modify office data table entries. The tables and table editor are part of the database software of the DMS-100 Family.

Table editor commands are entered by typing in commands using the keyboard of a teleprinter (TPR) or the keyboard of a MAP (maintenance and administration position). The results of the commands are displayed on the MAP screen or printed on the TPR.

Table editor functions

Table editor commands allow the user to perform the following functions:

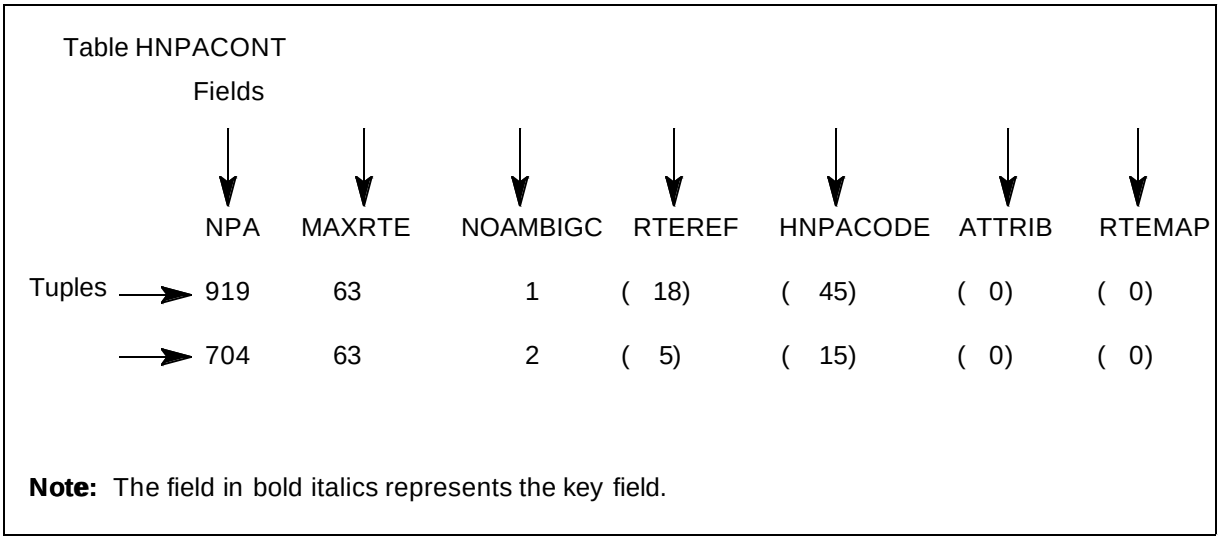
- add, delete, or change tuples or fields in a table or subtable
- list one or more tuples of a table or subtable
- move the cursor to display any tuple in a table or subtable
- display specified valid field values
- search for tuples containing specified field values
- verify table alterations before activating them
- modify subtables of a table

- create pending files from within the table editor
- display the number of used and allocated tuples of a table
- alter data when the central processing unit (CPU) of the DMS switch is out of sync

Data table structures

Tables and subtables consist of horizontal rows called tuples and vertical columns called fields. The information in the fields makes up the tuples (see Figure 5-1). Subtables contain data relative to the the tuple of which they are a part. The names of the fields listed across the top of the table are referred to as the heading.

Figure5-1 Data table structure



Note: The field in bold italics represents the key field.

Fields

The fields in a table or subtable have the following properties:

- Each field has two identifiers, a *field name* and a *field number*. Fields are referenced by either one.
- Fields are numbered consecutively from left to right beginning with the number one (1).
- There is no set number of fields for each table.
- Each field name is eight characters or fewer.
- Field data enclosed by parentheses indicates the existence of a subtable.
- The contents of a field may contain more than one element of information.
- Field data may consist of letters, numbers, or both.

Tuples

The tuples in a table or subtable have the following properties.

- Each tuple has a unique identifier called a *key.* The key for each tuple is usually the first field, but it may be the first few fields out of necessity to uniquely identify the tuple. Because the first field is usually the key, it is referred to as the key field.
- Tuples are numbered consecutively from top to bottom beginning with the number zero (0).
- The number of tuples is limited only by the memory capacity of a particular table. The memory capacity is determined by many factors that are not discussed in this document.
- All of the fields making up a tuple contain information about the key. Any subtables involved relate to the key of the main table.
- Tuples are referenced either by their key or by the table editor cursor placement.

Prompt and no-prompt modes

A prompt is the system's way of indicating to the user that the data entered is incomplete or fails to fall within the parameters defined for the field.

Prompt mode

The prompt mode is a method of entering data in which the system prompts the user for each piece of information needed. See the following MAP display as an example.

When the prompt mode is used, the following environment is present.

- The name of the required field or parameter is displayed.
- The user must input the correct data for each field or parameter as it is displayed.
- If the data that is input is not correct, the system responds with an error message, and the field or parameter prompt is redisplayed. If data is entered incorrectly again, the system responds with an error message and a brief description of what is needed for that field or parameter.
- At any time while in the prompt mode, the user may enter ABORT, which stops the execution of the original command.
- When a field or parameter contains subfields, these subfields are included in the system prompts.
- When the user is prompted for list items, prompting continues until the user enters a dollar sign ($) or until the maximum number of entries for the list is reached.

Figure5-2 MAP display example of prompt mode

```
CI:
>table hnpacont
TABLE:  HNPACONT
>add
NPA:
>919
MAXRTE:
>63
NOAMBIGC:
>1
TUPLE TO BE ADDED:
     919  63  1  (  0)   (  0)  (  0)  (  0)
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE ADDED
WRITTEN TO JOURNAL FILE AS JF NUMBER 233
>
```

No-prompt mode

Once the user is familiar with input for a particular activity, he or she may enter commands more efficiently by using the no-prompt mode. In the no- prompt mode, the user simply enters the command and all the information for each field in a continuous line of data with spaces between information. See the folowing MAP display as an example.

When the no-prompt mode is used, the following environment is present.

- The plus sign (+) is used when the full information for one tuple exceeds one input line. When + is used as the last character on a line, the contents of that line and any following lines are processed as a single input.
- After the system recognizes a valid command, one field value is processed at a time until the end of the input line is reached or an error is found. After identifying an error, the system reverts to the prompt mode at the point of the incorrect field.
- When the system processes a list, subsequent fields are assumed to be part of the list until a field value of $ is encountered, the maximum list length is reached, or the end of the input line is reached.

Figure5-3 MAP display example of no-prompt mode

```
CI:
>table hnpacont
TABLE:  HNPACONT
>add 919 63 1
TUPLE TO BE ADDED:
     919  63  1  (  0)   (  0)  (  0)  (  0)
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE ADDED
WRITTEN TO JOURNAL FILE AS JF NUMBER 233
>
```

Table editor commands

Table editor is accessed from the CI (command interpreter) level of the MAP by entering the command TABLE followed by the name of the table desired. Once in table editor, various commands and their appropriate parameters are available to the user.

After a command string is entered, it is followed by a carriage return. On most keyboards, the carriage return is designated as Enter, Return, or CR. After the user has entered a command followed by a carriage return, the system displays the prompt character (>) to indicate it is ready for another command.

The following sections contain descriptions of the most common table editor commands. Each command has a brief explanation of its function, the procedure for entering the command, and an example of how executing it might appear on the MAP.

Following the procedure for entering table editor, the commands are alphabetized for user convenience.

Entering table editor

The TABLE command is used to enter the table editor subsystem for the table specified.

The procedure for entering table editor and an example MAP display follow.

Procedure5-1 Procedure for entering table editor

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

Figure5-4 Example of a MAP display of the TABLE command

```
CI:
>table hnpacont
TABLE:  HNPACONT
>
```

If the user enters only the command TABLE and fails to provide a table name, the system prompts the user for the table name. If this occurs, enter the table name only.

Note: Reissuing the TABLE command is not recognized as a valid table name by the system, and the user is prompted for the table name again. Therefore, do not enter the TABLE command when prompted for a table name.

Figure5-5 Example of a MAP display of the TABLE command where table name is not specified

```
CI:
>table
TABLE NAME:
>hnpacont
TABLE:  HNPACONT
>
```

If the user enters an unrecognized table name, the system prompts the user for the table name. If this occurs, enter the table name only.

Note: Reissuing the TABLE command is not recognized as a valid table name by the system, and the user is prompted for the table name again. Therefore, do not enter the TABLE command when prompted for a table name.

Figure5-6 Example of a MAP display of the TABLE command where table name is invalid

```
CI:
>table hncont
UNKNOWN TABLE
TABLE NAME:
>hnpacont
TABLE:  HNPACONT
>
```

Note: Upon successful entry into a table, the internal table editor cursor is positioned on tuple 0, which is the uppermost tuple in the table, the override mode is off, and the verify mode is on.

ABORT

The ABORT command is used to cancel a command or input that has been entered by accident or incorrectly.

When a command that contains incorrect parameters is input, the system repeatedly prompts the user for the correct field value. When this occurs, using the ABORT command cancels the previous command and permits the user to try again. If the system continues to prompt the user for the correct field value, enter the ABORT command, then try the INFORM, LIST, and RANGE commands to determine the resident table and the parameters the system is expecting.

The procedure for entering the ABORT command and an example MAP display follow.

Procedure5-2 Procedure for entering the ABORT command

At the MAP display:

1 To cancel an incorrect or unintentional command once it has been entered in table editor, enter the ABORT command by typing

>ABORT

and pressing the Enter key.

The system then aborts the previously entered command.

Figure5-7 Example of a MAP display of the ABORT command

```
CI:
>table hnpacont
TABLE:    HNPACONT
>add
NPA:
>abort
>
```

Note: If the ABORT command is entered out of context with another command, the system responds with Undefined command "ABORT".

ADD

The ADD command is used to add tuples to a table. Depending on the table, the tuple is automatically placed at the bottom of the table or in a specific location in the table. Therefore, the ADD command can be executed from any location in the table.

After entering the ADD command, the user is prompted to add data in the first field of the table. The user is then prompted for any remaining fields, if appropriate. When familiar with the configuration of a table, the user may use the no-prompt mode of adding tuples. If a mistake is made, the system reverts to the prompt mode.

When using the ADD command, it is important to refer to the data schema documentation of the table the user is accessing.

The procedure for entering the ADD command and an example MAP display follow.

Procedure5-3 Procedure for entering the ADD command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To add a tuple, enter the ADD command by typing

`>ADD`

and pressing the Enter key.

The system then prompts you for the fields that need to be filled for the table you are in.

Figure5-8 Example of a MAP display of the ADD command:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>add
EXTPRTNM:
>pots
TUPLE TO BE ADDED:
          POTS       (  0)    (  0)
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE ADDED
WRITTEN TO JOURNAL FILE AS JF NUMBER 233
>
```

Note: If entering a long string of data that does not fit within the boundaries of the terminal screen, use + at the end of the first line, press Enter, and continue the command on the next line. The plus sign tells the system to regard the two lines as one command string.

BOTTOM

The BOTTOM command is used to position the cursor at the bottom of a table on the last tuple. BOT is an abbreviated form of the command.

The procedure for entering the BOTTOM command and an example MAP display follow.

Procedure5-4 Procedure for entering the BOTTOM command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To go to the bottom of a table, enter the BOTTOM command by typing

`>BOTTOM`

and pressing the Enter key.

The system then positions the cursor on the last tuple of the table you are in.

Figure5-9 Example of a MAP display of the BOTTOM command

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>bottom
       POT9    (   21)     (    2)
>
```

When the BOTTOM command is executed in an empty table, the system responds with `TABLE EMPTY.`

CHANGE

The CHANGE command is used to change the value of existing data in a field of a given tuple. It is vital that the cursor be positioned properly on the tuple to be changed before executing the CHANGE command or the user could alter tables incorrectly. CHA is an abbreviated form of the command.

After executing the CHANGE command, the system prompts the user by presenting the first field and the current data filling it. The user can then type in new data or press the Enter key to move to the next field without making a change.

The procedure for entering the CHANGE command and an example MAP display follow.

Procedure5-5 Procedure for entering the CHANGE command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 Position on the tuple you wish to change by typing

`>POS key_field`

and pressing the Enter key.

where

key_field
is the name of the key field of the tuple you wish to change

3 To change the tuple, enter the CHANGE command by typing

`>CHANGE`

and pressing the Enter key.

The system then prompts you for each changeable field in the tuple. Pass over the fields you do not wish to change by pressing the Enter key. When you reach the field you wish to change, enter the new data at the prompt.

Figure5-10 Example of a MAP display of the CHANGE command:

```
CI:
>table clli
TABLE:  CLLI
>pos incoming_trunk
       INCOMING_TRUNK   250    10   INCOMING_IT
>change
ADNUM:   250
>
TRKGRSIZ:  10
>20
ADMININF:  INCOMING_IT
>
TUPLE TO BE CHANGED:
        INCOMING_TRUNK   250    20   INCOMING_IT
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE CHANGED
WRITTEN TO JOURNAL FILE AS JF NUMBER 234
>
```

The user can also go directly to one particular field to change it by entering the CHANGE command followed by the specific field name or number. In tables with many fields, this alternate method can save time if only one change is desired.

This alternate procedure for entering the CHANGE command and an example MAP display follow.

Procedure5-6 Alternate procedure for entering the CHANGE command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 Position on the tuple you wish to change by typing

`>POS key_field`

and pressing the Enter key.

where

key_field
is the name of the key field of the tuple you wish to change

3 To go quickly to the field you wish to change in the tuple, enter the CHANGE command followed by the desired field name or field number by typing

>CHANGE field_name or field_number

and pressing the Enter key.

where

field_name
is the field name you wish to change

field_number
is the number of the field you wish to change

The system then goes directly to the field specified. Enter the new data at the prompt.

Figure5-11 Example of a MAP display of the CHANGE command with specific field:

```
CI:
>table clli
TABLE:  CLLI
>pos incoming_trunk
      INCOMING_TRUNK   250    10   INCOMING_IT
>change 3
TRKGRSIZ:  10
>20
TUPLE TO BE CHANGED:
      INCOMING_TRUNK   250    20   INCOMING_IT
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE CHANGED
WRITTEN TO JOURNAL FILE AS JF NUMBER 234
>
```

COUNT

The COUNT command is used to count the tuples in a table. When the COUNT command is executed without parameters, all the tuples in the table are counted.

The procedure for entering the COUNT command and an example MAP display follow.

Procedure5-7 Procedure for entering the COUNT command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To count the number of tuples in the table, enter the COUNT command by typing

>COUNT

and pressing the Enter key.

The system then counts the number of tuples and displays the number. If the number of tuples in the table is very large, the process of counting them may take several minutes.

Figure5-12 Example of a MAP display of the COUNT command:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>count
BOTTOM
SIZE = 30
>
```

Several types of parameters can be used with the COUNT command. With these parameters, specific pieces of information and combinations of information can be counted rather than all tuples in a table. For example, a user can get a count of all lines with a pretranslator name of pots in Table LINEATTR (Line Attribute), if desired.

The parameters that can be used with the COUNT command and their definitions follow:

EQ
equal to

GE
greater than or equal to

GT
greater than

LE
less than or equal to

LT
less than

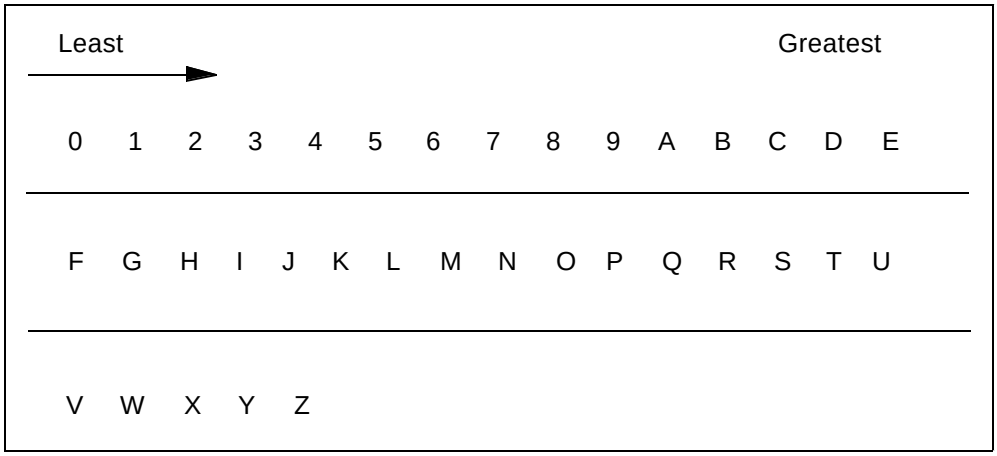
NE
not equal to

or

&
and

When using these parameters, it is important to understand what the system considers greater than or less than with respect to characters. The system bases its hierarchy on the standard ASCII hierarchy of symbols, with the exception that the DMS switch does not differentiate between uppercase and lowercase letters. Figure 5-2 is included to aid the user when using the COUNT command parameters that follow.

Figure5-13 ASCII character hierarchy



EQ

This parameter is used to tell the system to count tuples with fields containing specific information that is equal to what the user defines. The system only counts the tuples that meet the requirements defined by the user.

The procedure for entering the COUNT command with the EQ parameter and an example MAP display follow.

ATTENTION

If the user forgets to close the parentheses in the formula, the system will not respond to any further commands and will continue to display the prompt every time the Enter key is pressed. To regain command ability, the user should enter a right parenthesis and press the Enter key. The system will then respond with INVALID SEARCH ARGUMENT because the programming recognized the return as part of the formula and counted it as invalid to the formula. Once the user closes the parentheses and the system responds, the formula may be entered again correctly.

Procedure5-8 Procedure for entering the COUNT command with the EQ parameter

At the MAP display:

1 Enter the table editor environment by typing

```
>TABLE table_name
```

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To count specific tuples in the table using the EQ parameter, enter the COUNT command by typing

```
>COUNT (field_name or field_number EQ
field_value_desired)
```

and pressing the Enter key.

where

field_name
is the name of the field you wish to single out

field_number
is the number of the field you wish to single out

field_value
is the value of the field you wish to single out _desired

The system then counts the number of tuples with fields equal to the datafill specified and displays the number. If the number of tuples in the table is very large, the process of counting them may take several minutes.

Figure5-14 Example of a MAP display of the COUNT command with the EQ parameter

```
CI:
>table lineattr
TABLE:  LINEATTR
>count (prtnm eq pots)
BOTTOM
SIZE = 62
>


Note:  In this example, the system counts the tuples
with a pretranslator name equal to pots.
```

GE

This parameter is used to tell the system to count tuples with fields containing specific information that is greater than or equal to what the user defines. The system considers greater than as numbers greater than what is defined (2 is greater than 1) and letters greater than what is defined, with A functioning as the lowest letter (B is greater than A). The system only counts the tuples that meet the requirements defined by the user.

The procedure for entering the COUNT command with the GE parameter and an example MAP display follow.

Procedure5-9 Procedure for entering the COUNT command with the GE parameter

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To count specific tuples in the table using the GE parameter, enter the COUNT command by typing

`>COUNT (field_name or field_number GE field_value_desired)`

and pressing the Enter key.

where

field_name
is the name of the field you wish to single out

field_number
is the number of the field you wish to single out

field_value
is the value of the field you wish to single out _desired

The system then counts the number of tuples with fields greater than or equal to the datafill specified and displays the number. If the number of tuples in the table is very large, the process of counting them may take several minutes.

Figure5-15 Example of a MAP display of the COUNT command with the GE parameter

```
CI:
>table lineattr
TABLE:  LINEATTR
>count (prtnm ge pots)
BOTTOM
SIZE = 84
>


Note:  In this example, the system counts the tuples
with a pretranslator namesof pots, as well as
pretranslator names beginning with letters greater
 than pots
```

GT

This parameter is used to tell the system to count tuples with fields containing specific information that is greater than what the user defines. The system considers greater than as numbers greater than what is defined (2 is greater than 1) and letters greater than what is defined, with A functioning as the lowest letter (B is greater than A). The system only counts the tuples that meet the requirements defined by the user.

The procedure for entering the COUNT command with the GT parameter and an example MAP display follow.

Procedure5-10 Procedure for entering the COUNT command with the GT parameter

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To count specific tuples in the table using the GT parameter, enter the COUNT command by typing

>COUNT (field_name or field_number GT field_value_desired)

and pressing the Enter key.

where

field_name
is the name of the field you wish to single out

field_number
is the number of the field you wish to single out

field_value
is the value of the field you wish to single out _desired

The system then counts the number of tuples with fields greater than the datafill specified and displays the number. If the number of tuples in the table is very large, the process of counting them may take several minutes.

Figure5-16 Example of a MAP display of the COUNT command with the GT parameter

```
CI:
>table lineattr
TABLE: LINEATTR
>count (prtnm gt pots)
BOTTOM
SIZE = 22
>

Note: In this example, the system counts pretanslator
names greater than or equal to the name pots.
```

LE

This parameter is used to tell the system to count tuples with fields containing specific information that is less than or equal to what the user defines. The system considers less than as numbers less than what is defined (3 is less than 4) and letters less than what is defined, with A functioning as the lowest letter (C is less than H). The system only counts the tuples that meet the requirements defined by the user.

The procedure for entering the COUNT command with the LE parameter and an example MAP display follow.

Procedure5-11 Procedure for entering the COUNT command with the LE

parameter

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To count specific tuples in the table using the LE parameter, enter the COUNT command by typing

`>COUNT (field_name or field_number LE field_value_desired)`

and pressing the Enter key.

where

field_name
is the name of the field you wish to single out

field_number
is the number of the field you wish to single out

field_value
is the value of the field you wish to single out _desired

The system then counts the number of tuples with fields less than or equal to the datafill specified and displays the number. If the number of tuples in the table is very large, the process of counting them may take several minutes.

Figure5-17 Example of a MAP display of the COUNT command with the LE parameter

```
CI:
>table lineattr
TABLE:  LINEATTR
>count (prtnm le pots)
BOTTOM
SIZE = 70
>


Note: In this example, the system counts pretranslator
names of pots, as well as pretranslator names beginning
with letters less than pots.  If there is no match for
names less than pots, the system only counts tuples
with a pretranslator name of pots.
```

LT

This parameter is used to tell the system to count tuples with fields containing specific information that is less than what the user defines. The system considers less than as numbers less than what is defined (3 is less than 4) and letters less than what is defined, with A functioning as the lowest letter (C is less than H). The system only counts the tuples that meet the requirements defined by the user.

The procedure for entering the COUNT command with the LT parameter and an example MAP display follow.

Procedure5-12 Procedure for entering the COUNT command with the LT parameter

At the MAP display:

1 Enter the table editor environment by typing

```
>TABLE table_name
```

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To count specific tuples in the table using the LT parameter, enter the COUNT command by typing

```
>COUNT (field_name or field_number LT
field_value_desired)
```

and pressing the Enter key.

where

field_name
is the name of the field you wish to single out

field_number
is the number of the field you wish to single out

field_value
is the value of the field you wish to single out _desired

The system then counts the number of tuples with fields less than the datafill specified and displays the number. If the number of tuples in the table is very large, the process of counting them may take several minutes.

Figure5-18 Example of a MAP display of the COUNT command with the LT parameter

```
CI:
>table lineattr
TABLE:  LINEATTR
>count (prtnm lt pots)
BOTTOM
SIZE = 8
>


Note:  In this example, the system counts pretranslator
names beginning with letters less than pots.
```

NE

This parameter is used to tell the system to count tuples with fields containing specific information that is not equal to what the user defines. The system counts all tuples that do not have the datafill specified.

The procedure for entering the COUNT command with the NE parameter and an example MAP display follow.

Procedure5-13 Procedure for entering the COUNT command with the NE parameter

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To count specific tuples in the table using the NE parameter, enter the COUNT command by typing

`>COUNT (field_name or field_number NE field_value_desired)`

and pressing the Enter key.

where

field_name
is the name of the field you wish to single out

field_number
is the number of the field you wish to single out

field_value is
the value of the field you wish to single out _desired

The system then counts the number of tuples with fields not equal to the datafill specified and displays the number. If the number of tuples in the table is very large, the process of counting them may take several minutes.

Figure5-19 Example of a MAP display of the COUNT command with the NE parameter

```
CI:
>table lineattr
TABLE:  LINEATTR
>count (prtnm ne pots)
BOTTOM
SIZE = 30
>


Note:  In this example, the system counts pretranslator
names that are not equal to pots..
```

| (pipe)

The | (pipe) parameter is used to mean *or*. When using this symbol in a counting formula, the user is asking the system to count tuples meeting either one set of criteria or another. The | symbol appears on keyboards above the \ symbol.

The procedure for entering the COUNT command with the | parameter and an example MAP display follow.

Procedure5-14 Procedure for entering the COUNT command with the | parameter

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To count specific tuples in the table using the | parameter, enter the COUNT command by typing

`>COUNT (field_name or field_number xx field_value_desired | field_name or field_number xx field_value_desired)`

and pressing the Enter key.

where

field_name
is the field name you wish to single out

field_number
is the number of the field you wish to single out

field_value
is the value of the field you wish to single out _desired

xx
is one of the count command parameters, such as EQ or GT

The system then counts the number of tuples meeting either the parameters set by the first string or the parameters set by the second string. If the number of tuples in the table is very large, the process of counting them may take several minutes.

Figure5-20 Example of a MAP display of the COUNT command with the | parameter

```
CI:
>table lineattr
TABLE:  LINEATTR
>count (prtnm eq pots | prtnm eq pot1)
BOTTOM
SIZE = 72
>


Note:  In this example, the system counts tuples with
either a pretranslator name equal to pots or a
pretranslator name equal to pot1.
```

& (ampersand)

The & (ampersand) parameter is used to mean *and.* When using this symbol in a counting formula, the user is asking the system to count tuples meeting two sets of criteria.

The procedure for entering the COUNT command with the & parameter and an example MAP display follow.

Procedure5-15 Procedure for entering the COUNT command with the & parameter

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To count specific tuples in the table using the & parameter, enter the COUNT command by typing

`>COUNT (field_name or field_number xx field_value_desired & field_name or field_number xx field_value_desired)`

and pressing the Enter key.

where

field_name
is the field name you wish to single out

field_number
is the number of the field you wish to single out

field_value
is the value of the field you wish to single out _desired

xx
is one of the count command parameters, such as EQ or GT

The system then counts the number of tuples meeting the parameters set by the first string and the parameters set by the second string. If the number of tuples in the table is very large, the process of counting them may take several minutes.

Figure5-21 Example of a MAP display of the COUNT command with the & parameter

```
CI:
>table lineattr
TABLE:  LINEATTR
>count (prtnm eq owts & scrncl eq nscr)
BOTTOM
SIZE = 4
>


Note: In this example, the system counts pretranslator
names equal to owts and screening class names equal to
nscr.  All tuples counted must meet both conditions.
```

DELETE

The DELETE command is used to delete tuples from a table. Some tables are write-protected from deletions because removing tuples from certain tables can (or will) have a drastic effect on service. Therefore, execute the DELETE command with caution. Positioning the cursor correctly is vital. DEL is an abbreviated form of the command.

The procedure for entering the DELETE command and an example MAP display follow.

Procedure5-16 Procedure for entering the DELETE command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 Position on the tuple you wish to delete by typing

>POS key_field

and pressing the Enter key.

where

key_field
is the name of the key field of the tuple you wish to delete

3 To delete the tuple, enter the DELETE command by typing

>DELETE

and pressing the Enter key.

The system then deletes the tuple on which you are positioned.

Figure5-22 Example of a MAP display of the DELETE command

```
CI:
>table clli
TABLE:  CLLI
>position incoming_trunk
      INCOMING_TRUNK   250    20   INCOMING_IT
>delete
TUPLE TO BE DELETED:
      INCOMING_TRUNK   250    20   INCOMING_IT
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE DELETED
WRITTEN TO JOURNAL FILE AS JF NUMBER 245
>
```

The DELETE command can be entered from anywhere in a table if the key field of the tuple to be deleted is specified. Only a user who is familiar with table editor and the table being altered should use this method.

This alternate procedure for entering the DELETE command and an example MAP display follow.

Procedure5-17 Alternate procedure for entering the DELETE command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To delete a tuple from any position in the table, enter the DELETE command by typing

>DELETE key_field

and pressing the Enter key.

where

key_field
is the name of the key field of the tuple you wish to delete

The system then deletes the tuple that you specified.

Figure5-23 Example of a MAP display of the alternate DELETE command

```
CI:
>table clli
TABLE:  CLLI
>delete incoming_trunk
TUPLE TO BE DELETED:
      INCOMING_TRUNK   250    20   INCOMING_IT
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE DELETED
WRITTEN TO JOURNAL FILE AS JF NUMBER 245
>
```

After a tuple is deleted, the table editor cursor is always positioned to the tuple immediately following the deleted tuple. When the tuple deleted is the last one in the table, the cursor is left unpositioned, and the system responds with WARNING: CURRENTLY NOT POSITIONED.

Note: There are many tables that are interdependent. A deletion in one of them will cause data corruption in fields of tables elsewhere. In some cases, a deletion is not allowed. If you have any questions concerning whether a tuple should be deleted or in what sequence it should be deleted, consult the data schema books for that table or a senior translator.

DISPLAY

The DISPLAY command is used to display the tuple currently positioned on without a heading. A heading is the display of field names listed across the top

of a table. See Figure 5-1 for an illustration of a heading. DIS is an abbreviated form of the command.

The procedure for entering the DISPLAY command and an example MAP display follow.

Procedure5-18 Procedure for entering the DISPLAY command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To display a tuple without a heading, enter the DISPLAY command by typing

`>DISPLAY`

and pressing the Enter key.

The system then displays the tuple on which you are positioned but does not display the heading.

Figure5-24 Example of a MAP display of the DISPLAY command

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>display
       POT1      (   50)      (   10)
>
```

If the cursor is not positioned when the DISPLAY command is given, the system responds with `WARNING CURRENTLY NOT POSITIONED.`

DOWN

The DOWN command is used to move the cursor down a specific number of tuples. The tuple repositioned on is displayed without a heading. DOW is an abbreviated form of the command.

The procedure for entering the DOWN command and an example MAP display follow.

Procedure5-19 Procedure for entering the DOWN command

At the MAP display:

1 Enter the table editor environment by typing

`TABLE table_name`

and pressing the Enter key. where

table_name
is the name of the table you wish to enter

2 To move the cursor down a specific number of tuples, enter the DOWN command by typing

`DOWN n`

and pressing the Enter key.

where

n
is the number of tuples you wish to move down

The system then moves the cursor down the specified number of tuples and displays the tuple positioned on without a heading.

Note: If the DOWN command is entered without parameters, the system moves the cursor down only on tuple

Figure5-25 Example of a MAP display of the DOWN command

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>down 3
      OWTS       (   25)     (    5)
>
```

If the number of tuples the user wishes to move down exceeds the number of tuples in the table, the system goes to the bottom of the table and does not present a display to the user (see the following MAP display).

Figure5-26 Example of a MAP display of the DOWN command when the requested number exceeds the number of tuples in the table

```
CI:
>table stdprtct
TABLE: STDPRTCT
>down 3000
BOTTOM
>
```

FIRST

The FIRST command is used to position the cursor on the first tuple in the table. The tuple is displayed without a heading.

The procedure for entering the FIRST command and an example MAP display follow.

Procedure5-20 Procedure for entering the FIRST command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To position the cursor on the first tuple, enter the FIRST command by typing

`>FIRST`

and pressing the Enter key.

The system then repositions the cursor on the first tuple without displaying the tuple.

Figure5-27 Example of a MAP display of the FIRST command

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>bottom
      POT9      (   21)    (    2)
>first
>display
      POTS      (   30)    (    2)
>
```

If the FIRST command is executed in an empty table, the system responds with `TABLE EMPTY.`

HEADING

The HEADING command is used to display the table heading. The tuple positioned on is not displayed.

The procedure for entering the HEADING command and an example MAP display follow.

Procedure5-21 Procedure for entering the HEADING command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To display the table heading, enter the HEADING command by typing

>HEADING

and pressing the Enter key.

The system then displays the table heading.

Figure5-28 Example of a MAP display of the HEADING command

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>heading
     EXTPRTNM     STDPRT    AMAPRT
>
```

HELP

The HELP command is used to display the available table editor commands. For details about a specific command, type HELP followed by the desired command.

The procedure for entering the HELP command and an example MAP display follow.

Procedure5-22 Procedure for entering the HELP command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To display the available table editor commands, enter the HELP command by typing

>HELP

and pressing the Enter key.

The system then displays the table editor command list.

Note: HELP is a table editor command; therefore, you must be in a table to execute it.

3 To display more information on a particular command, enter the HELP command with a command name string by typing

>HELP command_name

and pressing the Enter key.

where

command
is a specific command name _name

The system then displays more information about the command specified.

Figure5-29 Example of a MAP display of the HELP command

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>help
TABLE EDITOR UTILITY
SUBCOMMANDS: SUBtable, FIRST, NEXT, LAST, PREV, LOCate
    ADD, REPlace, CHAnge, DELete, OVErride,
    DISplay, TOP, BOTtom, LISt, DOWn, POSition,
    UP, ASSIGN, HEADING, COUNT, POF, ENDPof,
    PUTPof, RANge, EXTend, EQ, NE, LT, LE, GT,
    GE, INForm, RETurn, VERify, FORMAT, CHECK.
>help first
FIRST
POSITIONS TO FIRST TUPLE IN TABLE (NO DISPLAY)
>
```

INFORM

The INFORM command is used to display the current position of the user in the database. The display shows the table the user is in, the key field positioned on, and the subtable entered, if appropriate. INF is an abbreviated form of the command.

The procedure for entering the INFORM command and an example MAP display follow.

Procedure5-23 Procedure for entering the INFORM command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To display your current position in the database, enter the INFORM command by typing

>INFORM

and pressing the Enter key.

The system then displays your position in the database.

Figure5-30 Example of a MAP display of the INFORM command

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>position pot1
      POT1      (    50)     (    10)
>sub 2
WARNING: CHANGES IN TABLE STDPRT MAY ALTER OFFICE
BILLING. CALL TYPE DEFAULT IS NP. PLEASE REFER TO
DOCUMENTATION.
>inform
TABLE:   STDPRTCT  POT1:  STDPRT
>
```

LAST

The LAST command is used to position the cursor on the last tuple in the table. The tuple is not displayed.

The procedure for entering the LAST command and an example MAP display follow.

Procedure5-24 Procedure for entering the LAST command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key.

where

table_name
is the name of the table you wish to enter

2 To position the cursor on the last tuple, enter the LAST command by typing

>LAST

and pressing the Enter key

The system then repositions the cursor on the last tuple but does not display the tuple.

Figure5-31 Example of a MAP display of the LAST command

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>last
>display
     POT9      (   21)      (    2)
>
```

If the LAST command is executed in an empty table, the system responds with `TABLE EMPTY`.

LIST

The LIST command is used to display the table heading and one or more tuples of the table. If the LIST command is executed without parameters, the table heading and the tuple positioned on are displayed. LIS is an abbreviated form of the command.

The procedure for entering the LIST command and an example MAP display follow.

Procedure5-25 Procedure for entering the LIST command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To display the tuple you are positioned on in a table, enter the LIST command by typing

`>LIST`

and pressing the Enter key

The system then displays the table heading followed by the tuple on which you are positioned.

Figure5-32 Example of a MAP display of the LIST command:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>list
   EXTPRTNM       STDPRT       AMAPRT
        POTS      (   30)      (    2)
>
```

If the LIST command is executed with a number parameter, the system displays the table heading and the specified number of tuples in sequential order, beginning with the tuple on which the user is positioned.

The procedure for entering the LIST command with a number parameter and an example MAP display follow.

Procedure5-26 Procedure for entering the LIST command with a number parameter

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To display two or more tuples in a table, enter the LIST command by typing

`>LIST n`

and pressing the Enter key

where

n
is the number of tuples you wish to list

The system then displays the table heading followed in sequential order by the number of tuples requested, beginning with the tuple on which you are positioned.

Figure5-33 Example of a MAP of the LIST command with a number parameter:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>list 4
  EXTPRTNM     STDPRT       AMAPRT
      POTS    (    30)    (     2)
      P111    (    15)    (     3)
      C111    (    15)    (     3)
      OWTS    (    25)    (     5)
>
```

The same parameters that can be used with the COUNT command can be used with the LIST command. As with the COUNT command, tuples with specific pieces of information and combinations of information can be listed.

LOCATE

The LOCATE command is used to position the cursor on a specific tuple. This tuple is not displayed. LOC is an abbreviated form of the command.

The procedure for entering the LOCATE command and an example MAP display follow.

Procedure5-27 Procedure for entering the LOCATE command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To position on a tuple without displaying it, enter the LOCATE command by typing

`>LOCATE key_field`

and pressing the Enter key

where

key_field
is the name of the key field of the tuple you wish to locate

The system then positions the cursor on the specified tuple but does not display it.

Figure5-34 Example of a MAP display of the LOCATE command:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>locate pot1
>display
   POT1      (   1)      (    2)
>
```

If the user enters the key field incorrectly, the system prompts the user for the key field.

NEXT

The NEXT command is used to position the cursor one tuple below the tuple on which the user is currently positioned. The tuple is not displayed.

The procedure for entering the NEXT command and an example MAP display follow.

Procedure5-28 Procedure for entering the NEXT command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To move to the next tuple without displaying it, enter the NEXT command by typing

>NEXT

and pressing the Enter key

The system then positions the cursor one tuple below the tuple currently positioned on but does not display it.

Figure5-35 Example of a MAP display of the NEXT command:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>pos pot1
   POT1   (  50)    (  10)
>next
>display
   POT2   (  42)    (   5)
>
```

OVERRIDE

The OVERRIDE command is used to cancel the prompt that occurs when the CPU is out of sync, when the journal file is not available, or both. It stops the messages the system sends to the user when changes to tables are attempted. The OVERRIDE command is useful if many changes are planned, but it should not be used randomly by the beginner. OVE is an abbreviated form of the command.

To activate the OVERRIDE command, the user must input the command each time a table is entered. This means that quitting a table deactivates the command.

The procedure for entering the OVERRIDE command and an example MAP display follow.

Procedure5-29 Procedure for entering the OVERRIDE command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To override the system prompts that occur when the CPU is out of sync or the journal file is not available, enter the OVERRIDE command by typing

>OVERRIDE

and pressing the Enter key

The system then overrides the system prompts.

Figure5-36 Example of a MAP display when CPU is out of sync and journal file is unavailable:

```
CI:
>table clli
MACHINES NOT IN SYNC - DMOS NOT ALLOWED
JOURNAL FILE UNAVAILABLE - DMOS NOT ALLOWED
TABLE:  CLLI
>pos outgoing_trunk
   OUTGOING_TRUNK 252  20  OUTGOING_IT
>del
MACHINES NOT IN SYNC - DMOS NOT ALLOWED
JOURNAL FILE UNAVAILABLE - DMOS NOT ALLOWED
ENTER Y TO CONTINUE PROCESSING OR N TO QUIT
>y
TUPLE TO BE DELETED:
   OUTGOING_TRUNK 252  20  OUTGOING_IT
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE DELETED
JOURNAL FILE INACTIVE
>
```

Figure5-37 Example of a MAP display of the OVERRIDE command:

```
CI:
>table clli
MACHINES NOT IN SYNC - DMOS NOT ALLOWED
JOURNAL FILE UNAVAILABLE - DMOS NOT ALLOWED
TABLE:  CLLI
>pos outgoing_trunk
   OUTGOING_TRUNK 252 20 OUTGOING_IT
>override
>del
TUPLE TO BE DELETED:
   OUTGOING_TRUNK 252 20 OUTGOING_IT
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE DELETED
JOURNAL FILE INACTIVE
>
```

POSITION

The POSITION command is used to position the cursor on a specific tuple and display the tuple. POS is an abbreviated form of the command.

The procedure for entering the POSITION command and an example MAP display follow.

Procedure5-30 Procedure for entering the POSITION command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To position on a tuple and display it, enter the POSITION command by typing

`>POSITION key_field`

and pressing the Enter key

where

key_field
is the name of the key field of the tuple on which you wish to position

The system then positions on the requested tuple and displays it.

Figure5-38 Example of a MAP display of the POSITION command:

```
CI:
>table stdprtct
TABLE; STDPRTCT
>position pot1
        POT1    ( 50)     ( 10)
>
```

If the POSITION command is entered without a key field specified, the system prompts the user for the field. It is recommended that the user execute the LIST command when first entering a table. This enables the user to determine how the key field is composed and therefore how it should be entered in order to position correctly. This procedure will help the user avoid frustrating error messages.

QUIT

The QUIT command is used to exit the user from one or all levels in table editor. The QUIT command executed in a subtable results in exiting both the subtable and the main table. If user enters several tables consecutively, the QUIT command results in exiting the current table and returning to the previous table. The cursor retains its position in each table entered until the QUIT command is executed. The QUIT ALL command is used to exit all levels in table editor and return the user to the CI level. QUI is an abbreviated form of the command.

The procedure for entering the QUIT command and an example MAP display follow.

Procedure5-31 Procedure for entering the QUIT command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To exit a table, enter the QUIT command by typing

`>QUIT`

and pressing the Enter key

The system then quits the table you are in and its subtable, if appropriate.

3 To exit table editor completely, enter the QUIT ALL command by typing

>QUIT ALL

and pressing the Enter key

The system then quits table editor and returns you to the CI level.

Figure5-39 Example of a MAP display of the POSITION command:

```
CI:
>table hnpacont
TABLE:  HNPACOUNT
>table clli
 TABLE: CLLI
 >table stdprtct
 TABLE STDPRTCT
 >pos pot1
      POT1 ( 50) ( 10)
 >sub 2
  WARNING: CHANGES IN TABLE STDPRT MAY ALTER OFFICE
  BILLING. CALL TYPE DEFAULT IS NP. PLEASE REFER TO
  DOCUMENTATION
 >table lineattr
 TABLE: LINEATTR
 >quit
 TABLE: STDPRTCT POT1: STDPRT
 >quit
 TABLE:: CLLI
 quit all
  CI:
```

RANGE

The RANGE command displays the fields of the current table by field name and field number and provides a brief description of each. The RANGE command can also be executed on an individual field to display the field name and number and the allowable datafill. RAN is an abbreviated form of the command.

Not all type information can be printed out with the use of the RAN or RANGE command in the table editor. For the output of instance types that have symbolic or string range names, only the type name is output if the total number of characters of the symbolic/string range names exceed 75 characters. However, if all the symbolic/string range names need to be seen, the the operating company can obtain the full output by using the PRINTTYPE command in DDEDIT.

The procedure for entering the RANGE command and an example MAP display follow.

Procedure5-32 Procedure for entering the RANGE command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To display table fields by number and name, enter the RANGE command by typing

`>RANGE`

and pressing the Enter key

The system then displays the table fields by number and name and includes a brief explanation of each field.

3 To display information on a specific field, enter the RANGE command followed by the field name or number by typing

`>RANGE field_name or field_number`

and pressing the Enter key

where

field_name
is the name of the field you wish to single out

field_number
is the number of the field you wish to single out

The system then displays the field by number and name and includes a brief explanation of the field and the acceptable datafill for that field.

Figure5-40 Example of a MAP display of the RANGE command:

```
CI:
>table hnpacont
TABLE:  HNPACONT
>range
1  NPA      THREE_DIGIT_CODE
2  MAXRTE       NUMBER_OF_ROUTE_REFERENCES
3  NOAMBIGC     NUMBER_OF_AMBIGUOUS_CODES
4  RTEREF       TABLE_OWNERSHIP
5  HNPACODE     TABLE_OWNERSHIP
6  ATTRIB       TABLE_OWNERSHIP
7  RTEMAP       TABLE_OWNERSHIP
LOGICAL TUPLE TYPE: LOG_HNPA_CNTL_TUPLE
>range 3
3  NOAMBIGC     NUMBER_OF_AMBIGUOUS_CODES
TYPE IS NUMBER_OF_AMBIGUOUS_CODES {0 TO 255}
>
```

REPLACE

The REPLACE command is used to replace a tuple with the tuple given as the command parameter. The system searches for the correct key field and positions before executing a replace. The system also prompts the user to confirm that the replacement tuple is correct. If the user chooses to edit the replacement tuple, the system displays each field with the option to type in new material. Afterwards, the system asks the user to verify the replacement tuple again. REP is an abbreviated form of the command.

Before executing a REPLACE command, it is recommended that the user execute the RANGE and LIST commands in order to understand exactly what fields the system will expect to be filled.

The procedure for entering the REPLACE command and an example MAP display follow.

Procedure5-33 Procedure for entering the REPLACE command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To replace a tuple, enter the REPLACE command by typing

`>REPLACE key_field tuple_datafill`

and pressing the Enter key

where

key_field
is the name of the key field of the tuple to be replaced

tuple_datafill
is the replacement datafill for the remaining appropriate fields

The system then prompts you with a display of the tuple you have entered and the option to accept, reject, or edit it.

Figure5-41 Example of a MAP display of the REPLACE command:

```
CI:
>table clli
TABLE:  CLLI
>replace newyorkout  118  10  trunk_to_newyork
TUPLE TO BE REPLACED:
     NEWYORKOUT      118  10  TRUNK_TO_NEWYORK
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT.
>y
TUPLE REPLACED
WRITTEN TO JOURNAL FILE AS JF NUMBER 246
>
```

If the REPLACE command is entered with a key field and no tuple datafill, the system prompts the user for each field.

RETURN

The RETURN command is used to leave a subtable and return to the main table. It saves the user from quitting the subtable and main table entirely and entering the main table again. After the user executes the RETURN command, the system displays the tuple that the user is positioned on in the main table. RET is an abbreviated form of the command.

The procedure for entering the RETURN command and an example MAP display follow.

Procedure5-34 Procedure for entering the RETURN command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To enter a subtable, enter the SUBTABLE command by typing one of the following

>SUBTABLE field_name

>field_number

and pressing the Enter key

where

field_name
is the name of the field you wish to single out

field_number
is the number of the field you wish to single out

The system then enters the subtable indicated.

3 To leave a subtable and return to the main table, enter the RETURN command by typing

>RETURN

and pressing the Enter key

The system then returns you to the main table and displays the tuple on which you are positioned.

Figure5-42 Example of a MAP display of the RETURN command:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>pos pot1
   POT1    (   50)    (   10)
>sub 2
WARNING:CHANGES IN TABLE STDPRT MAY ALTER OFFICE BILLING.
CALL TYPE DEFAULT IS NP. PLEASE REFER TO DOCUMENTATION.
>inform
TABLE:   STDPRTCT  POT1:  STDPRT
>return
   POT1    (   50)    (   10)
>
```

SUBTABLE

The SUBTABLE command is used to enter a subtable from a main table. The main table must be entered before any of its subtables can be accessed. If a main table contains more than one subtable, the field name or number associated with the subtable must be specified. SUB is an abbreviated form of the command and is the most common form of the command.

The procedure for entering the SUBTABLE command and an example MAP display follow.

Procedure5-35 Procedure for entering the SUBTABLE command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To enter a subtable, enter the SUBTABLE command by typing one of the following

>SUBTABLE field_name

>field_number

and pressing the Enter key

where

field_name
is the name of the field you wish to single out

field_number
is the number of the field you wish to single out

The system then enters the subtable indicated.

Figure5-43 Example of a MAP display of the SUBTABLE command:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>pos pot1
   POT1     (  50)    (  10)
>subtable 2
>inform
TABLE:   STDPRTCT  POT1:  STDPRT
>
```

It is important to position on the tuple desired before entering a subtable. Information in the subtable relates to the key field of the tuple positioned on in the main table. When the subtable is entered, all table editor commands are still active.

When the SUBTABLE command is executed and no subtable exists, the system responds with SUBTABLE DOES NOT EXIST.

TOP

The TOP command is used to position the cursor on the first tuple in the table and display the tuple.

The procedure for entering the TOP command and an example MAP display follow.

Procedure5-36 Procedure for entering the TOP command

At the MAP display:

1 Enter the table editor environment by typing

`>TABLE table_name`

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To position on the first tuple in the table, enter the TOP command by typing

`>TOP`

and pressing the Enter key

The system then repositions to the first tuple in the table and displays the tuple.

Figure5-44 Example of a MAP display of the TOP command:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>list 4
   EXTPRTNM  STDPRT      AMAPRT
   POTS     (    30)    (    2)
   P111     (    15)    (    3)
   C111     (    15)    (    3)
   OWTS     (    25)    (    5)
>top
   POTS     (    30)    (    2)
>
```

UP

The UP command is used to move the cursor up a specific number of tuples. The tuple repositioned on is displayed without a heading.

The procedure for entering the UP command and an example MAP display follow.

Procedure5-37 Procedure for entering the UP command

At the MAP display:

1 Enter the table editor environment by typing

>TABLE table_name

and pressing the Enter key

where

table_name
is the name of the table you wish to enter

2 To move the cursor up a specific number of tuples, enter the UP command by typing

>UP n

and pressing the Enter key

where

n
is the number of tuples you wish to move up

The system then moves the cursor up the specified number of tuples and displays the tuple positioned on without a heading

Figure5-45 Example of a MAP display of the UP command:

```
CI:
>table stdprtct
TABLE:  STDPRTCT
>list 4
   EXTPRTNM  STDPRT      AMAPRT
   POTS     (    30)    (    2)
   P111     (    15)    (    3)
   C111     (    15)    (    3)
   OWTS     (    25)    (    5)
>up 2
   P111     (    15)    (    3)
>
```

If the UP command is entered without parameters, the system moves the cursor up only one tuple

Miscellaneous commands

RENAMECLLI

The RENAMECLLI command is used to rename a common language location identifier (CLLI) in Table CLLI and in all other corresponding tables. The command can be entered from any location in the switch; it is not necessary to be in Table CLLI.

The procedure for entering the RENAMECLLI command and an example MAP display follow.

Procedure5-38 Procedure for entering the RENAMECLLI command

At the MAP display:

1 To rename a CLLI, enter the RENAMECLLI command by typing

`>RENAMECLLI old_clli new_clli`

and pressing the Enter key

where

old_clli
is the name of the original CLLI

new_clli
is the name of the new CLLI

The system then renames the old CLLI and records the new name in Table CLLI and corresponding tables.

Figure5-46 Example of a MAP display of the RENAMECLLI command:

```
CI:
>renameclli newyorkout newyork2w
DO YOU REALLY WANT TO RENAME IT ?
Please confirm (“YES” or “NO”):
>YES
>
```

TABREF

The TABREF command is used to list all tables that use information found in the table specified. This command is useful when deleting translations from tables.

The procedure for entering the TABREF command and an example MAP display follow.

Procedure5-39 Procedure for entering the TABREF command

At the MAP display:

1 To get a list of tables that use information found in a specific table, enter the TABREF command by typing

`>TABREF table_name`

and pressing the Enter key

where

table_name
is the name of the table you want to reference

The system then lists the tables that use data found in the table specified.

Figure5-47 Example of a MAP display of the TABREF command:

```
CI:
>tabref clli
CLLI, TRKGRP, ANNS, TONES, CUSTHEAD, XLANAME, TRKNAME,
ANNMEMS, DRAMS, DRAMTRK, AUDIO, DRMUSERS, FMRESINV,
NLUPCLLI, TRKMTCE, PXRTE, OFCRTE, FARTE, ACRTE, FTRTE,
NSCRTE, CTRTE, KTGROUP, KTMINMAX, DQMODEM, NSCDEFS,
AMATKOPT, MSGRTE, LENFEAT, DN_LOG, PRECONF, AIODTKN,
LTMAP, ISTRKGRP, PREPLANS, NWMAOCR, SILCNWM, NWMSCPT,
NWMSDPT, MTAHORIZ, SITE, ASCS, CNALDSPK, LPBKMEM,
IBNFEAT, IBNXLA, ATTCONS, FNMAP, IBNTREAT, ESARTE,
TRKLATA, TOPSCOIN, TOPLNDIS, OCGRP, VLMEM, TOPSHDLC,
MCCSOST, TOPEATRK, TOPSACTS, XFROPSEL, EBAFTKPH,
EAMCCSAN, EAACTSAN, BRANDANN, BRANDOPT, PARSDENY,
MCCSNBEC, ACTSNBEC, AABSOST, DATRKOPT, C6LKSET, C6LAYER,
C7LINK, TKCVDATA, WCKCODES, CODECALL, KSETFEAT, CLASSOFC,
RESINV, RESGROUP, ACDSGRP, CLLIMTCE, TREATMENT_SUBTABLE,
DATAOWNR, OFRT, FNPACONT, FNPA_STS, POSITION,
STD_PRETRANS, DNINV, DNROUTE, NEWTHGRP, INWORIRT,
INWTERTE, IBNRTE, IBNRT2, IBNRT3, IBNRT4, NWMCLLI,
ATTSCHED
>
```

6 Translations verification (TRAVER)

Translations verification (TRAVER)

Translations verification (TRAVER) is a tool that allows the user to examine the translation and routing of a particular call. A TRAVER report can display the possible results of a call, the translation of a call, or both.

TRAVER was designed to help the user quickly identify translations errors, oversights, or misdirection while debugging and testing software. This allows the user to correct datafill problems more efficiently.

NORTEL recommends use of TRAVER at the CI level of the MAP to minimize real time usage of call processing resources. With TRAVER, the user can specify the type of call originator, the number being processed or the trunk taken, and the kind of report desired.

For more detailed information on TRAVER, refer to *DMS100 Family Command Reference Manual*, 297-1001-822.

Note: The TRAVER tool does not correctly show the number of digits sent over the trunk for a call routed through a virtual facility group (VFG). For a call routed through a VFG, the number of digits TRAVER can verify is up to and including 18 digits minus the number of prefix digits.

TRAVER command structure

Composing a TRAVER command involves several different pieces of information. These are the code for the call originator, the identification of the originator, the directory number being processed or the outgoing trunk, and the type of report desired. The originator ID and the directory numbers being processed are dependent on the particular configuration of the switch involved. The originator type and report code are part of the TRAVER program.

The fixed parameters that can be used with the TRAVER command and their definitions follow.

L
indicates the originator is a line

TR
indicates the originator is a trunk

T
indicates only a report on the table entries is desired (trace)

NT
indicates only a report on the results is desired (no trace)

B
indicates a report on both table entries and results is desired

Note: Refer to the appendix for brief definitions of the basic translations tables encountered during a call trace.

Line-to-line TRAVER trace

A line-to-line call trace is executed using the L parameter with the TRAVER command. A line-to-line call is a local call originating from one line and terminating to another line without the interaction of trunks.

Procedure6-1 Procedure for entering a line-to-line call trace

At the MAP display:

1 To produce a TRAVER report for a line-to-line call, type:

```
>TRAVER L calling_dn called_dn report_code
```

and pressing the Enter key.

where

calling_dn
is the directory number originating the call

called_dn
is the directory number receiving the call

report_code
is the type of report the user is requesting (NT, T, or B)

Note: When executing the TRAVER command on numbers that include an asterisk, the letter b equals *. The use of an asterisk in dialing is common to Meridian Digital Centrex (MDC) plans.

In the following example, all tables accessed in the call trace are listed, but a summary of the result is not given.

Figure6-1 TRAVER output example for line-to-line call trace with the T parameter

```
>TRAVER L 6211234 6211235 T
TABLE LINEATTR
0 1FR NONE NT NSCR 0 613 P621 L613 TSPS 10 NIL NILSFC NILLATA
     0 NIL NIL 00 N
LCABILL OFF - BILLING DONE ON BASIS OF CALLTYPE
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
TABLE STDPRTCT
P621 (1)  (0)
.SUBTABLE STDPRT
.621 632 N NP 0 NA
.SUBTABLE AMAPRT
.KEY NOT FOUND
.DEFAULT VALUE IS:  NONE OVRNONE N
TABLE HNPACONT
613 128 1 (  43)  (  1)  (  0)  (  0)
.SUBTABLE HNPACODE
.621 621 DN 613 621
TABLE TOFCNAME
613 621
TABLE DNINV
613 621 1235 L HOST 00 0 00 01
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
TABLE LCASCRCN
613 L613 (  25)  OPTL N
.SUBTABLE LCASCR
.621 622
TABLE PFXTREAT
OPTL NP Y NP UNDT

+++ TRAVER:  SUCCESSFUL CALL TRACE +++
```

In the following example, the tables accessed are not given, but a summary of the result is listed. Line three indicates the line where the call terminates. Line five indicates what treatment the caller will receive, if any, if the line dialed is busy. In this case, generalized no circuit (GNCT) treatment is given. If GNCT is unavailable, the next possible treatments are overflow (OFLO) and then lock out (LKOUT).

Figure6-2 TRAVER output example for line-to-line call trace with the NT parameter

```
>TRAVER L 6211234 6211235 NT
DIGIT TRANSLATION ROUTES

1 LINE            6136211235

TREATMENT ROUTES.  TREATMENT IS: GNCT
1 *OFLO
2 LKOUT

+++TRAVER: SUCCESSFUL CALL TRACE +++
```

Line-to-trunk TRAVER trace

A line-to-trunk call trace is executed using the L parameter with the TRAVER command. A line-to-trunk call is a call originating from a line and terminating to a trunk.

Procedure6-2 Procedure for entering a line-to-trunk call trace

At the MAP display:

1 To produce a TRAVER report for a line-to-trunk call, type:

`>TRAVER L calling_dn called_dn report_code`

and pressing the Enter key.

where

calling_dn
is the directory number originating the call

called_dn
is the directory number receiving the call

report_code
is the type of report the user is requesting (NT, T, or B)

In the following example, all tables accessed in the call trace are listed, but a summary of the result is not given.

Figure6-3 TRAVER output example for line-to-trunk call trace with the T parameter

```
>TRAVER L 7820012 13045331235 T
TABLE LINEATTR
3 1FR NONE NT NSCR 3 919 POTS LCA3 CTOP 23 NIL NILSFC NILLATA
     0 NIL NIL 00 N
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
TABLE STDPRTCT
POTS ( 1)  ( 1)
.SUBTABLE STDPRT
.13  19 N DD 1 NA
.SUBTABLE AMAPRT
.KEY NOT FOUND
.DEFAULT VALUE IS:  NONE OVRNONE N
TABLE HNPACONT
919 63 1  (  43)  (  1)  (  0)  (  0)
.SUBTABLE HNPACODE
.304 309 FRTD 12
.SUBTABLE RTEREF
.  12 S D GRNSBONCIT2W
.EXIT TABLE RTEREF
EXIT TABLE HNPACONT
TABLE LCASCRCN
919 LCA3 (  25)  MAND N
.SUBTABLE LCASCR
.TUPLE NOT FOUND.  DEFAULT IS NON-LOCAL
TABLE PFXTREAT
MAND DD N DD UNDT
LATA IS NIL, THEREFORE NOT AN EQUAL ACCESS CALL

+++ TRAVER:  SUCCESSFUL CALL TRACE +++
```

In the following example, the tables accessed are not given, but a summary of the result is listed. Line three indicates the trunk that will carry the outpulsed digits to their next destination. Line five indicates what treatment the caller will receive, if any, if the trunk accessed is busy. In this case, generalized no circuit (GNCT) is given. If GNCT is unavailable, the next possible treatments are overflow (OFLO) and then lock out (LKOUT).

Figure6-4 TRAVER output example for line-to-trunk call trace with the NT parameter

```
>TRAVER L 7820012 13045331235 NT
DIGIT TRANSLATION ROUTES

1 GRNSBONCIT2W      3045331235        ST

TREATMENT ROUTES.  TREATMENT IS: GNCT
1 *OFLO
2 LKOUT

+++TRAVER: SUCCESSFUL CALL TRACE +++
```

Trunk-to-line TRAVER trace

A trunk-to-line call trace is executed using the TR parameter with the TRAVER command. A trunk-to-line call is a call originating from a trunk and terminating to a line.

Procedure6-3 Procedure for entering a trunk-to-line call trace

At the MAP display:

1 To produce a TRAVER report for a trunk–to–line call, type:

`>TRAVER TR  incoming_trunk incoming_dn report_code`

and pressing the Enter key.

where

incoming_trunk
is the trunk that is transmitting the directory number

incoming_dn
is the directory number that is coming in

report_code
is the type of report the user is requestin (NT, T, or B)

In the following example, all tables accessed in the call trace are listed, but a summary of the result is not given.

Figure6-5 TRAVER output example for trunk-to-line call trace with the T parameter

```
>TRAVER TR GRNSBONCIT2W 7820012 T
TABLE TRKGRP
GRNSBONCIT2W IT 0 ELO NCRT 2W N NIL MIDL 704 POTS NSCR 919 000 Y N
TABLE STDPRTCT
POTS (  1)  (  1)
.SUBTABLE STDPRT
.782 782 N NP 0 NA
.SUBTABLE AMAPRT
.KEY NOT FOUND
.DEFAULT VALUE IS:  NONE OVRNONE N
TABLE HNPACONT
919 63 1 (  10)  (  1)  (  0)  (  0)
.SUBTABLE HNPACODE
.782 782 DN 919 782
TABLE TOFCNAME
919 782
TABLE DNINV
919 782 0012 L HOST 00 0 00 12
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND

+++ TRAVER:  SUCCESSFUL CALL TRACE +++
```

In the following example, the tables accessed are not given, but a summary of the result is listed. Line three indicates the line where the call terminates. Line five indicates what treatment the caller will receive, if any, if the line is busy. In this case, generalized no circuit (GNCT) treatment is given. If GNCT is unavailable, the next possible treatment is overflow (OFLO).

Figure6-6 TRAVER output example for trunk-to-line call trace with the NT parameter

```
>TRAVER TR GRNSBONCIT2W 7820012 NT
DIGIT TRANSLATION ROUTES

1 LINE           9197820012        ST

TREATMENT ROUTES.  TREATMENT IS: GNCT
1 *OFLO

+++TRAVER: SUCCESSFUL CALL TRACE +++
```

Trunk-to-trunk TRAVER trace

A trunk-to-trunk call trace is executed using the TR parameter with the TRAVER command. A trunk-to-trunk call is a call originating from one trunk and terminating to another trunk. Tandem calls are trunk-to-trunk.

Procedure6-4 Procedure for entering a trunk-to-trunk call trace

At the MAP display:

1 To produce a TRAVER report for a trunk-to-trunk call, type:

`>TRAVER TR incoming_trunk outgoing_dn report_code`

and pressing the Enter key.

where

incoming_trunk
is the trunk that is transmitting the directory number

outgoing_dn
is the directory number the trunk will outpulse

report_code
is the type of report the user is requestin (NT, T, or B)

In the following example, all tables accessed in the call trace are listed, but a summary of the result is not given.

Figure6-7 TRAVER output example for trunk-to-trunk call trace with the T parameter

```
>TRAVER TR GRNSBONCIT2W 2045661824 T
TABLE TRKGRP
GRNSBONCIT2W IT 0 ELO NCRT 2W N NIL MIDL 704 POTS NSCR 919 000 Y N
TABLE STDPRTCT
POTS (  1)  (  1)
.SUBTABLE STDPRT
.2 6 N NP 0 NA
.SUBTABLE AMAPRT
.KEY NOT FOUND
.DEFAULT VALUE IS:  NONE OVRNONE N
TABLE HNPACONT
919 63 1 (  10)  (  1)  (  0)  (  0)
.SUBTABLE HNPACODE
.204 204 FRTD 1
.SUBTABLE RTEREF
.   1 S D RCHMVAIT1W
.   S D RCHMVAIT2W
.EXIT TABLE RTEREF
EXIT TABLE HNPACONT

+++ TRAVER:  SUCCESSFUL CALL TRACE +++
```

In the following example, the tables accessed are not given, but a summary of the result is listed. Line three indicates the first trunk that will be tried in the route list. Line four indicates the second trunk that will be tried if the first trunk is unavailable to carry the outpulsed digits to their next destination. Line six indicates what treatment the caller will receive, if any, if either of these trunks are busy. In this case, generalized no circuit (GNCT) treatment is given. If GNCT is unavailable, the next possible treatment is overflow (OFLO).

Figure6-8 TRAVER output example for trunk-to-trunk call trace with the NT parameter

```
>TRAVER TR GRNSBONCIT2W 2045661824 NT
DIGIT TRANSLATION ROUTES

1 RCHMVAIT1W          2045661824
2 RCHMVAIT2W          2045661824

TREATMENT ROUTES.  TREATMENT IS: GNCT
1 *OFLO

+++TRAVER: SUCCESSFUL CALL TRACE +++
```

7 Utility commands

Query commands

This section describes the use of some query commands. Query commands help the user extract information from the switch.

QDN

The QDN (query DN) command is used to display information about a subscriber line. The subscriber line is identified in the command by its DN. It is not necessary to be at the CI level when executing this command.

Procedure7-1 Entering the QDN command

At the MAP display:

1 Enter the QDN command by typing

>QDN dn

and pressing the Enter key.

where

dn
is the DN you wish to query

Figure7-1 Example of a MAP display for the QDN command:

```
CI:
>qdn 7820019
------------------------------------------------------------
DN: 7820019
TYPE: SINGLE PARTY LINE
SNPA: 919  SIG: DT  LNATTIDX:   0
LINE EQUIPMENT NUMBER:   HOST 00 0 00 19
LINE CLASS CODE:   1FR
LINE TREATMENT GROUP:   1
CARDCODE:  6X17AB   GND: N  PADGRP: STDLN  BNV: NL MNO: N
PM NODE NUMBER      :    37
PM TERMINAL NUMBER  :    15
OPTION:
DGT  CWT
------------------------------------------------------------
```

The fields shown in the above MAP display are described in the following sections:

DN This field identifies the DN of the subscriber line.

TYPE This field describes the line category.

SNPA This field identifies the serving number plan area, that is, the three–digit area code of the subscriber line.

SIG This field identifies the signaling type. In the previous MAP display, the signaling type is Digitone (DT).

LNATTIDX This field identifies the line attribute index number, which defines translation routing for an LCC in table LINEATTR (Line Attribute).

LINE EQUIPMENT NUMBER This field identifies the site, frame, unit, drawer, and circuit of the line.

LINE CLASS CODE This field defines a line group on the basis of tariff rate.

LINE TREATMENT GROUP This field is the value of field LTG in table LINEATTR.

CARDCODE This field identifies the product engineering code (PEC) for the line card.

GND This field defines whether or not ground is applied to the LEN. In the previous MAP display, N equals loop start.

PADGRP This field identifies transmission level. In the previous MAP display, STDLN equals standard line.

BNV This field indicates the balance network value. In the previous MAP display, NL equals nonloaded.

MNO This field indicates whether or not manual override is possible.

PM NODE NUMBER This field identifies the peripheral module to which the line is connected.

PM TERMINAL NUMBER This field identifies the terminal device (for example, telephone set or line controlling device).

OPTION This field identifies the options and features assigned to the line.

QLEN

The QLEN (query line equipment number) command is used to display the same type of information as QDN. However, in the QLEN command, the line is identified by its LEN. QLEN is useful when the user is querying a line that does not have a unique DN such as a member of a hunt group. It is not necessary to be at the CI level when executing this command.

Procedure7-2 Entering the QLEN command

At the MAP display:

1 Enter the QLEN command by typing

`>QLEN len`

and pressing the Enter key.

where

len
is the line equipment number you wish to query

Figure7-2 Example of a MAP display of the QLEN command:

```
CI:
>qlen 0 0 0 19
------------------------------------------------------------
LEN:     HOST 00 0 00 19
TYPE: SINGLE PARTY LINE
SNPA: 919
DN: 7820019
LINE CLASS CODE:   1FR
SIGNALLING TYPE: DIGITONE
LINE TREATMENT GROUP:   1
LINE ATTRIBUTE GROUP:      0
CARDCODE:  6X17AB   GND: N  PADGRP: STDLN  BNV: NL MNO: N
PM NODE NUMBER      :    37
PM TERMINAL NUMBER  :    15
OPTION:
DGT  CWT
------------------------------------------------------------
```

The fields shown in the above MAP display that are different from the fields in a QDN are described in the following sections.

LEN This field identifies the line equipment number of the subscriber line.

SIGNALLING TYPE This field has the same meaning as the SIG field in a QDN. It identifies the signaling type.

LINE TREATMENT GROUP This field has the same meaning as the LNATTIDX field in a QDN. It identifies the line attribute index number, which defines translation routing for an LCC in table LINEATTR.

QDNWRK

The QDNWRK (query DN working) command is used to query a range of working lines. These lines are identified by their DNs. This command saves the user from executing QDN several times.

Figure7-3 Using the QDNWRK command in prompt mode

```
> QLENWRK
DIRECTORY_NUMBER_RANGE: ALL
> R
FROM_DN:
> 7820012
TO_DN:
> 7820014
LINE_CLASS_CODE: NLCC
> 1FR
OPTION:
> DGT
SUMMARY OR DETAIL:
> S
LINE_DRAWER_NUMBER:
> $
LINE CLASS CODE: NLCC
> 1FR
OPTION:
> DGT
SUMMARY OF DETAIL: S
> S
COMMAND AS ENTERED
QDNWRK  R 7820012 7820014 1FR ( DGT) $ $
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT

> Y
WARNING: QUERIES OF ALL DN'S OR QUERIES OF A LARGE RANGE OF
        DN'S MAY RUN FOR 30 MINUTES OR MORE BEFORE
        PRODUCING ANY OUTPUT
REPORT ON WORKING LINE EQUIPMENT NUMBERS
FROM 7820012 TO 7820014
LCC                 1FR            OPTION               DGT
TOTAL COUNT OF WORKING DN FROM 7820012 TO 7820014: 3
```

Figure7-4 Using the QDNWRK command in no-prompt mode

```
> QDNWRK R 7820012 7820014 1FR DGT $ S
```

Note 1: A dollar sign ($) must be entered after the options in the no–prompt mode.

Note 2: A detailed report lists the DNs in the same format as the QDN command.

The fields shown in the example above are described in the following sections.

DIRECTORY_NUMBER_RANGE This field allows the user to choose whether all numbers will be queried or a range will be given. ALL is the default.

FROM_DN If R (range) is entered in the DIRECTORY_NUMBER_RANGE field, the FROM_DN field appears for the user to indicate the starting DN in the range to be queried.

TO_DN If R (range) is entered in the DIRECTORY_NUMBER_RANGE field, the TO_DN field appears for the user to indicate the ending DN in the range to be queried.

LINE CLASS CODE In this field, the user enters the LCC of the DNs to be queried. In the example shown, the LCC is 1FR. The default is NLCC (no LCC).

OPTION This field allows the user to specify DNs that have a particular option assigned to them.

SUMMARY OR DETAIL In this field, the user can choose the kind of report required. The report can be detailed (D) or a summary (S), which is the default. The output for a detailed report is formatted like the output for a QDN.

QLENWRK

The QLENWRK (query LEN working) command is used to display the same type of information as QDNWRK. However, in the QLENWRK command, the working lines are identified by their LENs.

Figure7-5 Using the QLENWRK command in prompt mode

```
> QLENWRK
LINE_MODULE_RANGE: ALL
> R
FROM_LM:
> 0 0 0 10
TO_LM:
> 0 0 0 11
LINE_DRAWER_RANGE:
> R
LINE_DRAWER_NUMBER:
> 0
LINE_DRAWER_NUMBER:
> 1
LINE_DRAWER_NUMBER:
> $
LINE CLASS CODE: NLCC
> 1FR
OPTION:
> DGT
SUMMARY OF DETAIL: S
> S
COMMAND AS ENTERED
QLENWRK R HOST 00 0 HOST 00 0
R                           ( 0)  ( 1)$ 1FR (   DGT) $ S
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT
> Y
WARNING: QUERIES OF ALL DN'S OR QUERIES OF A LARGE RANGE OF
        DN'S MAY RUN FOR 30 MINUTES OR MORE BEFORE
        PRODUCING ANY OUTPUT
REPORT ON WORKING LINE EQUIPMENT NUMBERS
FROM HOST 00 0  TO HOST 00 0  0  1
LCC                 1FR           OPTION              DGT
COUNT BY LINE DRAWERS

  0   1
LM : HOST 00 0  COUNT : 24

   14  10

           TOTAL                  24
. . . . . .
DRW TOTALS:

   14  10
```

Figure7-6 Using the QLENWRK command in no-prompt mode

```
> QLENWRK R 0 0 0 10 0 0 0 11 R 0 1 $ 1FR DGT $ S
```

Note: A dollar sign ($) must be entered after the options in the no–prompt mode.

The fields shown in the example above are described in the following sections.

LINE_MODULE_RANGE This field allows the user to specify if the line modules to be queried will be given in a range or contain all line modules. The default is ALL.

FROM_LM If R (range) is entered in the LINE_MODULE_RANGE field, the FROM_LM field appears for the user to indicate the starting line module in the range to be queried.

TO_LM If R (range) is entered in the LINE_MODULE_RANGE field, the TO_LM field appears for the user to indicate the ending line module in the range to be queried.

LINE_DRAWER_RANGE In this field, the user chooses whether to query all drawers (ALL) or a range (R).

LINE_DRAWER_NUMBER If R (range) is entered in the LINE_DRAWER_RANGE field, the user must enter the drawer numbers to query. This prompt is repeated until the user enters a dollar sign ($) to indicate an end of line drawer numbers.

LINE CLASS CODE In this field, the user enters the LCC of the LENs to be queried. The default is NLCC (no LCC).

OPTION This field allows the user to specify DNs that have a particular option assigned to them.

SUMMARY OR DETAIL In this field, the user can choose the kind of report required. The report can be detailed (D) or a summary (S), which is the default. The output for a detailed report is formatted like the output for a QLEN.

QHASU

The QHASU (query hardware assigned software unassigned) command is used to determine what hardware might be available for assignment. This command is especially useful when translations need testing.

Figure7-7 Using the QLENWRK command in prompt mode

```
> QHASU
LINE_MODULE_RANGE: ALL
> R
FROM_LM: HOST 00 0
> 0 0
TO_LM: HOST 00 0
> 0 1
LINE_DRAWER_RANGE: ALL
> R
LINE_DRAWER_NUMBER:
> 5
LINE_DRAWER_NUMBER:
> 6
LINE_DRAWER_NUMBER:
> $
CARD CODE: NIL_CTN
> 6X17AC
GND: N
> N
SUMMARY OF DETAIL: S
> S
COMMAND AS ENTERED
QHASU R HOST 00 0 HOST 00 1
R                      ( 5)  ( 6)  $ 6X17AC N S
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT
> Y
SUMMARY OF HARDWARE ASSIGNED SOFTWARE UNASSIGNED
LEN - HASU
FROM HOST 00 0    TO HOST 00 1  5  6
CARDTYPE          6X17AC         OPT  ALL
  COUNT BY LINE DRAWERS

   5  6
LM : HOST 00 0    COUNT : 6
   6  0


                  TOTAL:  6
. . . . . .
. .
DRW TOTALS:

   6  0
```

Figure7-8 Using the QLENWRK command in no-prompt mode

```
> QHASU R 0 0 0 1 R 5 6 $ 6X17AC N S
```

The fields shown in the example above are described in the following sections.

LINE_MODULE_RANGE This field allows the user to specify if the line modules to be queried will be given in a range or contain all line modules. The default is ALL.

FROM_LM If R (range) is entered in the LINE_MODULE_RANGE field, the FROM_LM field appears for the user to indicate the starting line module in the range to be queried.

TO_LM If R (range) is entered in the LINE_MODULE_RANGE field, the TO_LM field appears for the user to indicate the ending line module in the range to be queried.

LINE_DRAWER_RANGE In this field, the user chooses whether to query all drawers (ALL) or a range (R). The default is ALL.

LINE_DRAWER_NUMBER If R (range) is entered in the LINE_DRAWER_RANGE field, the user must enter the drawer numbers to query. This prompt is repeated until the user enters a dollar sign ($) to indicate an end of line drawer numbers.

CARD CODE In this field, the user enters the card code of the card to be queried. The default is NIL_CTN.

GND In this field, the user defines whether or not ground is applied to the LEN. In the example shown, the default N equals loop start.

SUMMARY OR DETAIL In this field, the user can choose the kind of report required. The report can be detailed (D) or a summary (S), which is the default. The output for a detailed report is formatted like the output for a QLEN.

Store file

The store file in the DMS is treated as a device and is referred to as SFDEV (store file device). Information can be sent to SFDEV by utilizing such commands as SEND and COPY.

Programs can be copied to and from SFDEV and external devices such as disk and magnetic tape. When a nonresident program is desired, it can be copied from an external storage device to SFDEV. Once in SFDEV, the store file

system provides commands to execute the program, which can then be erased, if desired.

The following sections describe store file commands and store file EDIT commands. The commands in each section are alphabetized. However, the following list presents some commands in the order they might be used by the beginning store file user:

- EDIT
- INPUT
- SAVE or FILE
- TYPE
- FIND
- CHANGE
- DELETE
- QUIT
- ERASESF

Note: It is important to realize that store file is a tool to be used by operating company personnel at their own discretion.

Store file commands

The following commands are available in store file.

ERASESF

The ERASESF command is executed at the CI level and is used to erase a specified store file. Before executing an ERASESF command, the user must first execute a LISTSF command.

Procedure7-3 Procedure for entering the ERASESF command

At the MAP display:

1 Enter the LISTSF command by typing

`>LISTSF`

and pressing the Enter key.

2 Enter the ERASESF command by typing

`>ERASESF file_name`

and pressing the Enter key.

where

file_name
is the name of the file you wish to erase

LISTSF

The LISTSF command is executed at the CI level and is used to provide the user with a list of, and accessibility to, the files contained in SFDEV.

Procedure7-4 Procedure for entering the LISTSF command

At the MAP display:

1 Enter the LISTSF command by typing

```
>LISTSF ALL
```

and pressing the Enter key.

ALL
is an optional parameter that lists all the store files in SFDEV. If the LISTSF command is executed without a parameter, the system displays only files created by the user.

READ

The READ command is executed at the CI level and is used to run the store file specified. Before executing a READ command, the user must first execute a LISTSF command.

Procedure7-5 Procedure for entering the READ command

At the MAP display:

1 Enter the LISTSF command by typing

```
>LISTSF
```

and pressing the Enter key.

2 Enter the READ command by typing

```
>READ file_name
```

and pressing the Enter key.

where

file_name
is the name of the file you wish to run

Store file EDIT commands

The following commands are available during a store file edit session.

CHANGE

The CHANGE command is used to alter data in the line on which the internal store file pointer is positioned. After positioning on the correct line, the user enters the data to be changed in single quotes and then the new data in single quotes.

Procedure7-6 Procedure for entering the CHANGE command

At the MAP display:

1 Enter the CHANGE command by typing

```
>CHANGE 'xxx' 'yyy'
```

and pressing the Enter key.

where

xxx
is the data string as it currently appears in the line

yyy
is the data string as you wish it to appear

DELETE

The DELETE command is used to delete the line on which the internal pointer is positioned. Once a line is deleted, the internal pointer is positioned on the line below the deleted line.

Procedure7-7 Procedure for entering the DELETE command

At the MAP display:

1 Enter the DELETE command by typing

```
>DELETE
```

and pressing the Enter key.

DOWN

The DOWN command is used to move the internal store file pointer down a specified number of lines. If the number of lines is not specified, the pointer moves down one line.

Procedure7-8 Procedure for entering the DOWN command

At the MAP display:

1 Enter the DOWN command by typing

```
>DOWN n
```

and pressing the Enter key.

where

n
is the number of the line you wish to move down

EDIT

The EDIT command is used to create a new file or to enter the store file editor system for an existing file.

Procedure7-9 Procedure for entering the EDIT command

At the MAP display:

1 Enter the EDIT command by typing

`>EDIT file_name`

and pressing the Enter key.

where

file name
is the name of the file you wish to create or edit

END

The END command is used to move the internal store file pointer to the end of the file. The indicator EOF (end of file) is displayed just above the last line in the store file.

Procedure7-10 Procedure for entering the END command

At the MAP display:

1 Enter the END command by typing

`>END`

and pressing the Enter key.

Figure7-9 Example of a MAP display for the END command:

```
>END
EOF
EDIT:
>
```

FILE

The FILE command is used to save any editing that has occurred in a store file. The system does this by refiling the file. The user may file the store file under the original name (the default) or under a new name by specifying a name after the FILE command.

When the FILE command is executed, the user leaves the EDIT mode as if a QUIT were executed.

Procedure7-11 Procedure for entering the FILE command

At the MAP display:

1 Enter the FILE command by typing

`>FILE dev_type file_name`

and pressing the Enter key.

where

dev_type
is the type of storage device for the file

file_name
is the new name of the file

Note: It is not necessary to specify a file name if you wish to file the store file under its original name. However, if you do not specify a file name, you must specify a storage device name (for example, SFDEV).

FIND

The FIND command is used to move the internal store file pointer to the first line that begins with the specified data. The data is specified within single quotes.

Procedure7-12 Procedure for entering the FIND command

At the MAP display:

1 Enter the FIND command by typing

```
>FIND 'xxx'
```

and pressing the Enter key.

where

xxx
is the string of data you wish to position on

Note: The string of data specified must appear at the beginning of a line of information. The search is conducted from the position of the cursor to the end of the file. Therefore, it is recommended that the user position the internal pointer at the top of the file before executing the FIND command.

INPUT

The INPUT command is used to add lines to a store file. The command can be used when creating a store file for the first time or when adding lines to an existing file. Lines can be added anywhere in the file. When the INPUT command is executed, lines are added below the line on which the internal pointer is positioned.

Procedure7-13 Procedure for entering the INPUT command

At the MAP display:

1 Enter the INPUT command by typing

```
>INPUT n
```

and pressing the Enter key.

where

n
is the number of lines you wish to add and is optional

When you have finished adding lines, press the Enter key twice. The system responds with an edit prompt, indicating it is ready for further store file commands.

LINE

The LINE command is used to move the internal store file pointer to the specified line number. Lines are numbered consecutively from top to bottom, beginning with the number zero (0).

Procedure7-14 Procedure for entering the LINE command

At the MAP display:

1 Enter the LINE command by typing

`>LINE n`

and pressing the Enter key.

where

n
is the number of the line you wish to position on

PRINT

The PRINT command is used to print an SFDEV on screen for viewing. It does not execute any commands in the SFDEV file. Before executing a PRINT command, the user must first execute a LISTSF command.

Procedure7-15 Procedure for entering the PRINT command

At the MAP display:

1 Enter the LISTSF command by typing

`>LISTSF`

and pressing the Enter key.

2 Enter the PRINT command by typing

`>PRINT file_name`

and pressing the Enter key.

where

file_name
is the name of the file you wish to print

QUIT

The QUIT command is used to exit the store file editor.

Procedure7-16 Procedure for entering the QUIT command

At the MAP display:

1 Enter the QUIT command by typing

`>QUIT`

and pressing the Enter key.

SAVE

The SAVE command is used to save any changes the user has made to a file without leaving the EDIT mode. As part of the command, the store file device (SFDEV) must be specified.

Procedure7-17 Procedure for entering the SAVE command

At the MAP display:

1 Enter the SAVE command by typing

>SAVE

and pressing the Enter key.

TOP

The TOP command is used to move the internal store file pointer to the top of the file. The indicator TOF (top of file) is displayed just above the top line in the store file.

Procedure7-18 Procedure for entering the TOP command

At the MAP display:

1 Enter the TOP command by typing

>TOP

and pressing the Enter key.

Figure7-10 Example of a MAP display for the TOP command:

```
>TOP
TOF
EDIT:
>
```

TYPE

The TYPE command is used to display a specified number of lines, beginning with the one on which the internal pointer is positioned. If the number of lines is not specified, the system displays one line.

Procedure7-19 Procedure for entering the TYPE command

At the MAP display:

1 Enter the TYPE command by typing

>TYPE n

and pressing the Enter key.

where

n
is the number of lines you wish to display

UP

The UP command is used to move the internal store file pointer up a specified number of lines. If the number of lines is not specified, the pointer moves up one line.

Procedure7-20 Procedure for entering the UP command

At the MAP display:

1 Enter the UP command by typing

```
>UP n
```

and pressing the Enter key.

where

n
is the number of lines you wish to move up

DUMPTAB AND DMOPRO

DUMPTAB (dump table) and DMOPRO (data modification order processor) are system commands that enable the translator to modify translations tables more quickly. The DUMPTAB command is capable of putting an entire table into an SFDEV file for manipulation by the user. The DMOPRO command reverses the process by applying the changed table existing in the SFDEV file to switch translations.

The procedure for using the DUMPTAB command follows.

Procedure7-21 Procedure for using the DUMPTAB command

At the MAP display:

1 Enter the DUMPTAB command by typing

```
>DUMPTAB table_name EXTERNAL device TOTAL input_type
```

and pressing the Enter key.

where

table_name
is the name of the table you wish to dump

device
is the device to which you are sending the table data. SFDEV is the recommended device.

Input_type
is the command DUMPTAB will place before each tuple entry in the input file. The command is later interpreted when the file is read. The following commands are possible: INP, ADD, REP, and PUT

Note: ADD and REP are table editor commands. INP and PUT are DMO commands.

Figure7-11 Example of a MAP display of the DUMPTAB command:

```
CI:
>dumptab termdev external sfdev total put
DUMPING TABLE: TERMDEV                        TO FILE:
RETURN CODE IS: 0
>
```

Figure7-12 Example of a MAP display of a SFDEV file generated from DUMPTAB:

```
CI:
>listsf
TERMDEV
>print termdev
%EXT
TAB TERMDEV
% TERMDES IOCNO IOCCKTNO TERMTYPE BAUDRATE INTYPE EQPEC
PARITY COMCLASS GUARANTE MDMTYPE
%
EXT 128
PUT      MAP  0 8 VT100 B2400 CL 1X67BC NONE Y NONE
                                                  ALL
PUT      MAP2 1 9 VT100 B2400 CL 1X67BC NONE Y NONE
                                                  ALL
PUT      MAP1 1 8 VT100 B2400 CL 1X67BC NONE Y NONE
                                                  ALL

QUI 5
>
```

Once an SFDEV file is generated using the DUMPTAB command, the user may edit the file using SFDEV editing parameters. This may involve changing the information in a few fields, deleting tuples, or adding several new tuples. Once the file has been edited and filed, the user is ready to apply the edited table using the DMOPRO command.

Note: The DUMPTAB command automatically generates EXT before the table input information. This is an old program command that is no longer valid, and it generates an UND (undefined) error when the DMOPRO command is used to apply the table. This error causes no problems in the system, but the user may wish to delete the EXT command line from the SFDEV file to streamline the DMOPRO process.

If only a main table is specified in the DUMPTAB command string and the control table has subtables, the system dumps all table and subtable information. However, the DUMPTAB command is capable of putting only

subtable information in an SFDEV file. It is not capable of dumping the subtable information of one tuple into an SFDEV file. If a user dumps subtables from table STDPRTCT, all pretranslator subtables are dumped.

The procedure for using the DUMPTAB command to dump subtable information follows.

Procedure7-22 Procedure for entering the DUMPTAB command for subtables

At the MAP display:

1 Enter the DUMPTAB command by typing

```
>DUMPTAB head_table_name ALL subtable_name EXTERNAL
device TOTAL input_type
```

and pressing the Enter key.

where

head_table_name
is the name of the control table

subtable_name
is the name of the subtable you wish to dump

device
is the device to which you are sending the table data. SFDEV is the recommended device.

input_type
is the command DUMPTAB will place before each tuple entry in the input file. The command is later interpreted when the file is read. The following commands are possible: INP, ADD, REP, and PUT.

Note 1: ADD and REP are table editor commands. INP and PUT are DMO commands.

Note 2: The string SUB cannot be substituted for the string ALL in the DUMPTAB command, although the system prompts indicate that it can.

Figure7-13 Example of a MAP display of the DUMPTAB command for subtables:

```
CI:
>dumptab stdprtct all stdprt external sfdev total put
DUMPING TABLE: STDPRTCT  SUBTABLE: STDPRT     TO FILE:
WARNING: CHANGES IN TABLE STDPRT MAY ALTER OFFICE
BILLING.  CALL TYPE DEFAULT IS NP.  PLEASE REFER TO
DOCUMENTATION.
WARNING: CHANGES IN TABLE STDPRT MAY ALTER OFFICE
BILLING.  CALL TYPE DEFAULT IS NP.  PLEASE REFER TO
DOCUMENTATION.
RETURN CODE IS: 0
>
```

Note: For table STDPRTCT, there will be a warning for each standard pretranslator in the table.

The DMOPRO command applies the altered table from the SFDEV file generated by the DUMPTAB command. The input commands appearing before each tuple in the DUMPTAB SFDEV file are implemented according to the rules of table editor and DMO files when the DMOPRO command is executed.

The procedure for using the DMOPRO command follows.

Procedure7-23 Procedure for entering the DMOPRO command

At the MAP display:

1 Enter the DMOPRO command by typing

`>DMOPRO input_file_name JOURNAL`

and pressing the Enter key.

where

input_file_name
is the name of the DUMPTAB SFDEV file you wish to apply. The file name is usually the same name as the dumped table.

Note: JOURNAL is an optional command, but it is highly recommended. When it is used, the DMOPRO transaction is recorded in the journal file, which provides an information backup for the user.

Figure7-14 Example of a MAP display of the DMOPRO command:

```
CI:
>dmopro termdev
UND 128
--- TABLE: TERMDEV -- 0 added, 5 replaced, 0 deleted
In total, 8 commands were processed
1 errors were encountered during processing
>
```

Note: The EXT command in the DUMPTAB record generates the 1 error (UND) that appears in the above message.

In addition to the DUMPTAB and DMOPRO commands, the command DMOVER (DMO verification) is recommended. When DMOVER is executed, it verifies the DUMPTAB SFDEV file and checks it for errors. This command serves as a safety feature before the DMOPRO command is used. It presents the user with the same file information as the DMOPRO command, but it does not enter the table involved into translations.

The procedure for entering the DMOVER command follows.

Procedure7-24 Procedure for entering the DMOVER command

At the MAP display:

1 Enter the DMOVER command by typing

`>DMOVER input_file_name`

and pressing the Enter key.

where

input_file_name
is the name of the DUMPTAB SFDEV file you wish to verify. File name is usually the same name as the dumped table.

Figure7-15 Example of a MAP display of the DMOVER command:

```
CI:
>dmover termdev
UND 128
--- TABLE: TERMDEV -- 0 added, 5 replaced, 0 deleted
In total, 8 commands were processed
1 errors were encountered during processing
>
```

Note: The EXT command in the DUMPTAB record generates the 1 error (UND) that appears in the above message.

DMO commands INPUT and PUT

Two DMO commands can be used in bulk DMO files such as those generated by the command DUMPTAB. The two commands are INPUT and PUT. They are described below.

INPUT This command replaces tuples in a table by looking for a match to the key field or fields. If the system finds a match in the table, it replaces the existing tuples with the changed tuples. INP is an abbreviated form of the command.

PUT This command can add or replace tuples in a table. It will add a tuple if there is no matching key field or fields already in the table. It will replace a tuple if it finds a matching key field or fields.

DMS-100 Family

Basic Translation Tools

Guide

electronic mail: cits@nortelnetworks.com

Publication number: 297-1001-360
Product release: BASE11
Document release: Standard 05.01
Date: August 1999
Printed in the United States of America