

# **TOPS-10/TOPS-20 DDT11 Manual**

AA-M494A-TK

**July 1984**

The DDT11 program is a symbolic debugging program that runs on TOPS-10 and TOPS-20, communicating with PDP-11 based front ends and remote nodes. This manual describes how to use the DDT11 program.

**OPERATING SYSTEM:** TOPS-10 V7.02  
TOPS-20 V5.1

**SOFTWARE:** DDT11 V7E

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center. Outside the United States, orders should be directed to the nearest DIGITAL Field Sales Office or representative.

**Northeast/Mid-Atlantic Region**

Digital Equipment Corporation  
PO Box CS2008  
Nashua, New Hampshire 03061  
Telephone:(603)884-6660

**Central Region**

Digital Equipment Corporation  
Accessories and Supplies Center  
1050 East Remington Road  
Schaumburg, Illinois 60195  
Telephone:(312)640-5612

**Western Region**

Digital Equipment Corporation  
Accessories and Supplies Center  
632 Caribbean Drive  
Sunnyvale, California 94086  
Telephone:(408)734-4915

**First Printing, July 1984**

© Digital Equipment Corporation 1984. All Rights Reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

**digital**™

DEC	MASSBUS	UNIBUS
DECmate	PDP	VAX
DECsystem-10	P/OS	VMS
DECSYSTEM-20	Professional	VT
DECUS	Rainbow	Work Processor
DECwriter	RSTS	
DIBOL	RSX	

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

# CONTENTS

## PREFACE

1	OVERVIEW . . . . .	1
1.1	Initializing Procedures for DDT11 . . . . .	1
1.2	Initial DDT11 Dialog . . . . .	2
1.3	Examining a Running Node . . . . .	2
1.4	Examining Dump Files . . . . .	2
1.5	Examining Unrun System Images . . . . .	3
2	USING DDT11 FOR ANF-10 NODES . . . . .	3
2.1	Loading and Saving ANF-10 Node Symbols . . . . .	3
2.2	Examining a Running ANF-10 Node . . . . .	3
2.3	Examining an ANF-10 Dump . . . . .	4
2.4	Examining an ANF-10 Unrun System Image . . . . .	4
2.5	Examining a Nonrunning Node . . . . .	4
3	USING DDT11 FOR IBMCOM FRONT ENDS . . . . .	5
3.1	Loading and Saving Symbols . . . . .	5
3.2	Examining a DN60 Front End . . . . .	5
3.3	Examining an IBMCOM Front-End Dump . . . . .	6
3.4	Examining an Unrun Image . . . . .	6
4	USING DDT11 FOR DECnet-10/DECnet-20 FRONT ENDS . . . . .	6
4.1	Loading and Saving DN20 Symbols . . . . .	7
4.2	Examining a Running DECnet-10/DECnet-20 MCB Node . . . . .	8
4.3	Examining Dumps from DECnet (MCB) Nodes . . . . .	8
4.4	Examining an MCB Node Unrun System Image . . . . .	9
5	USING DDT11 FOR AN RSX-20F FRONT END . . . . .	9
5.1	Loading and Saving Symbols . . . . .	9
5.2	Examining an RSX-20F Front End . . . . .	10
5.3	Examining RSX-20F Front-End Dumps . . . . .	10
5.4	Examining an RSX-20F Front-End Unrun Image . . . . .	10
6	SETTING OUTPUT MODES . . . . .	11
7	USING DDT11 TO EXAMINE MEMORY . . . . .	12
7.1	Moving the Location Pointer . . . . .	12
7.2	Displaying the Contents of a Location . . . . .	13
7.3	Searching for a Word . . . . .	14
7.4	Using Masks to Search for Matches . . . . .	14
8	USING DDT11 TO DEPOSIT INTO MEMORY . . . . .	15
9	STARTING EXECUTION . . . . .	16
10	MONITORING A LOCATION . . . . .	16
11	DUMPING MEMORY . . . . .	17
12	USING SYMBOLS . . . . .	18
12.1	Defining Symbols . . . . .	18
12.2	Controlling Symbol Typeout . . . . .	18
12.3	Matching Symbols . . . . .	19
12.4	Completing Partial Symbols . . . . .	19
13	USING COMMAND FILES IN DDT11 . . . . .	19
13.1	Command File Echoing . . . . .	19
13.2	Using Macro-Style Arguments . . . . .	20
13.3	Using Tags in Command Files . . . . .	20
14	DDT11 COMMAND FILE FOR AN RSX-20F FRONT END . . . . .	21
15	WRITING A BINARY FILE . . . . .	23

APPENDIX A        SUMMARY OF DDT11 COMMANDS

APPENDIX B        SWITCHES FOR THE INPUT SPECIFICATION

TABLES

A-1        DDT11 Commands . . . . . A-1

## PREFACE

This manual provides information for the moderately to very experienced user of the TOPS-10 or TOPS-20 system. In addition, the reader should be familiar with the configuration of the system network, including both the software and hardware of front ends and remote stations.

The purpose of this manual is twofold: first, to introduce DDT11 to users who have never used DDT11 before; second, to bring new information to experienced users. Therefore, the descriptions in this manual are organized both functionally and in reference format. However, it is recommended that the user become familiar with the standard DDT before attempting to use DDT11.

## CONVENTIONS USED IN THIS MANUAL

The following notations are used in this manual:

- `ESC` Indicates a place where you should press the ESCAPE key.
- `CTRL/x` Indicates a place where you should hold the CTRL key and press the character indicated by x.
- `LF` Indicates a place where you should press the linefeed key.
- `RET` Indicates a place where you should press the RETURN key.

## DOCUMENTS REFERENCED IN THIS MANUAL

You should be familiar with, and have access to, one of the following manuals. Choose the manual appropriate to the communications software you will be using.

TOPS-10 ANF-10 Software Installation Guide

TOPS-10 IBM Emulation/Termination DN61: 2780/3780

TOPS-20 IBM Emulation/Termination

TOPS-20 User's Guide

TOPS-10 Operating System Commands Manual

TOPS-10/TOPS-20 RSX-20F System Reference Manual

TOPS-10 Utilities Manual

(In particular, the section on DDT)

DECnet-10 Network Generation and Installation Procedures

DECnet-20 Network Generation and Installation Procedures

## 1 OVERVIEW

The DDT11 program runs on TOPS-10 and TOPS-20 host systems. It is used by system support specialists to diagnose and fix errors in nodes and front ends, to make modifications to running software, and to examine the running nodes and front ends. The DDT11 program performs some or all of the following functions for DC76, ANF-10, IBM communications, DECnet-10/20, and RSX-20F front ends and remote nodes:

- Examines and deposits memory on a running node.
- Examines dumps.
- Examines and deposits instructions in the system image file.
- Simulates the console terminal, execution of PDP-11 instructions, and the clock functions. The core image can be obtained from any system image file or dump file. Instructions can be executed in a free-running mode or one instruction at a time.

DDT11 runs only on a TOPS-10 or TOPS-20 host system; input and output are directed to and from your controlling terminal, unless you specify otherwise. Although the DDT11 program functions in essentially the same manner for each product listed below, the program requires a different symbol file and a different command to identify the object to be debugged. Therefore, a separate section describing the procedures for initializing DDT11 is devoted to each of the following products:

<u>Product</u>	<u>Section</u>
ANF-10 Nodes	2.0
IBMCOM-10/20 Nodes	3.0
DECnet-10/20 MCB (DN20)	4.0
RJE-20 (DN200) Nodes	4.0
RSX-20F Front End	5.0

### 1.1 Initializing Procedures for DDT11

Before you can use DDT11 to examine and debug a specific node, you should create a version of DDT11 that recognizes the symbols specific to the software running on that node.

The standard PDP-11 and PDP-8 assembly operation codes are defined as symbols in DDT11. You can define additional symbols by loading them with DDT11 from appropriate symbol files. Symbol definitions are different for each node.

## 1.2 Initial DDT11 Dialog

You should start the DDT11 that you created for a specific node or front end. To start the specific DDT11, RUN the program that you created with the SAVE command.

For example, to start the DDT11 for a specific DN82 (node 22) (see Section 2.1), type:

```
.RUN DN8222 (RET)
```

DDT11 prints one line identifying itself by version number and specifying the file from which the symbols were obtained. The response to the above command is:

```
DDT11 7E(114) = DN8222/ SYMBOLS = DSK:DN8222.CRF 14:14 13-JAN-83
```

DDT11 then prompts for what you want to look at:

Input:

Your response to the input prompt is determined by the function you want to perform and the communications product involved.

The following sections describe how to load and examine dumps and system images of running and nonrunning nodes and front ends, and remote stations.

### NOTE

When you specify a file as input to DDT11, the file is loaded into memory. You can perform the same functions on a file as you can perform on a running node. Therefore, for the purposes of this manual, "memory" describes the memory you are working with, whether a running system or a previously loaded file.

## 1.3 Examining a Running Node

To examine a running node, you must first gain access to it. To access the node, you must respond to the Input: prompt by specifying the node specification. This is determined by the communications software that is currently running. You must have privileges to examine a running node or front end.

## 1.4 Examining Dump Files

To examine a dump from one of the nodes, you must specify the dump file specification instead of the node specification. The dump file name must consist of six characters or less. If the file does not exist in your job's current directory path, the file specification must include the PPN of the directory area where the file exists. On TOPS-20, a file must also be specified by a PPN, if it does not exist in the logged-in directory area. You can use the TRANSLATE command to obtain a PPN for the file specification. Refer to the TOPS-20 User's Guide.



## 1.5 Examining Unrun System Images

You can use DDT11 to examine unrun system images. An unrun system image is the file that is used initially to load the front end or node. For examples on using DDT11 to examine a specific node or front end, refer to the section specific to that product.

## 2 USING DDT11 FOR ANF-10 NODES

The following sections show the procedures for loading and saving symbols that are specific to the software generated on a TOPS-10 host system for a DN82 node with a node number of 22. These sections discuss examining a running node, a dump, and an unrun system image.

### 2.1 Loading and Saving ANF-10 Node Symbols

The file name shown here is, of course, specific to this case. You must specify the CREF output file from MACDLX. The DDT11 command sequence, file/SYMBOL, will look for the files with default file extensions .LST and .CRF. If the extension is other than .LST or .CRF, you must specify it. In this example, the file name is DN8222.CRF.

```
.R DDT11 (RET)           ;load a standard copy of DDT11

DDT11 7E(114)           ;DDT11 identifies itself
Input: DN8222/SYMBOL    ;load symbols from the file DN8222.CRF

% loaded nn symbols     ;DDT11 tells how many symbols were
                        ;loaded

Input: (CTRLZ)          ;exit DDT11

EXIT                    ;DDT11 signs off
```

This sequence loads the specific symbols with the standard symbols in DDT11. When DDT11 requests another input file, press <CTRL/Z> to exit from DDT11.

You can now save the loaded program by using the SSAVE command:

```
.SSAVE DN8222 (RET)     ;save the node-specific DDT11
DN8222 saved            ;DN8222.EXE is saved

.                        ;TOPS-10 prompt
```

### 2.2 Examining a Running ANF-10 Node

You can specify an ANF-10 node by typing the /NODE:node-id switch in response to the Input: prompt. Node-id refers to node name or node number in the ANF-10 environment. For example, the following is used to examine node 22, which was used in previous examples:

```
.RUN DN8222 (RET)
DDT11 7E(114) = DN8222/ SYMBOLS = DSK:DN8222.LST 12:03 01-JAN-83

Input:/NODE:22 (RET) or /NODE:CTCH22 (RET)
```

The /NODE switch provides sufficient node specification for all running ANF-10 nodes.

### 2.3 Examining an ANF-10 Dump

To examine a dump from a TOPS-10 ANF-10 node, run the node-specific DDT11 that was built for the node. For input, specify the file name of the dump file. The default file extension is .LSD. The .LSD extension is the default for dumps created by NETLDR. The following example shows how a dump file might be read by a node-specific version of DDT11:

```
.RUN DN8222(RET) ;Run the node-specific DDT11
DDT11 7E(114) = DN8222/ SYMBOLS = DSK:DN8222.LST 12:03 01-JAN-83

Input:DN8222(RET) ;Dump file is DN8222.LSD
```

#### NOTE

Dumps produced by DTELDR from front ends must have /DTELDR appended to the dump file specification. All others do not require a switch.

### 2.4 Examining an ANF-10 Unrun System Image

The dialog used to examine an unrun system image from an ANF-10 node is shown in this example:

```
.RUN DN8222(RET)
DDT11 7E(114) = DN8222/ SYMBOLS = DSK:DN8222.LST 12:03 01-JAN-83

Input: SYS:DN8222.BIN(RET)
[62p core]
[63p core]
[81p core]
```

### 2.5 Examining a Nonrunning Node

You can use DDT11 to look at the memory of a node that is not running network software, but is running only the down-line load ROM. To examine such a node, use the /NODE switch to specify a running node that is adjacent to the nonrunning node in the network. Include the /LINE switch in the input specification. Use /LINE to indicate the line in the running node to which the nonrunning node is connected. In the following example, to examine node 41 from the TOPS-10 host, you must specify node 31. In node 31, line 12 is connected to node 41. The input line looks like this:

```
Input:/NODE:31/LINE:12
```

### 3 USING DDT11 FOR IBMCOM FRONT ENDS

The distribution tape contains files necessary to build the IBMCOM front ends. These files are named D6xyz.EXE to indicate the type of system, and are copies of DDT11 with the correct symbols loaded for that system.

#### 3.1 Loading and Saving Symbols

Unless you have reassembled the front end programs, you can use these .EXE files directly from the distribution tape. If you have reassembled the front end software and included anything new, you should load and save your own DDT11 using the .CRL file produced by the assembler. For example:

```
{@DDT11(RET) ;(TOPS-20) load standard copy of DDT11}
{.R DDT11(RET) ;TOPS-10}

DDT11 7E(114) ;DDT11 identifies itself

Input:D6xyz.CRL/SYMBOL(RET)
loaded nn symbols ;load symbols from the file D6xyz.CRL
;exit DDT11

EXIT

{@SAVE D6xyz.EXE(RET) ;TOPS-20}
{.SSAVE D6xyz.EXE(RET) ;TOPS-10}
```

#### 3.2 Examining a DN60 Front End

To examine front ends that are running DN6x software (any software in the DN60 line), you must specify the /PORT switch. The port specification is the number of the port through which the node communicates with the TOPS-10/20 host system. For more information on the front-end operating system, generically called D6xyz, see the TOPS-20 IBM Emulation/Termination manual. The following example shows how such a node is specified:

```
{@D6xyz(RET) ;TOPS-20}
{.RUN D6xyz(RET) ;TOPS-10}

DDT11 5(54) = DN60/SYMBOLS=DSK:D6xyz.CRL[4,56] 21:35 22-FEB-83

Input:/PORT:n(RET)
```

Where: n is

<u>Operating System</u>	<u>Number</u>
TOPS-10 KL	0-3 on DL-10#0 4-7 on DL-10#1 11-13 on DTE (10 is the console front end)
TOPS-20 KL	11-13 on DTE (10 is the console front end)
TOPS-10/TOPS-20 KS	10-11 on 2020

To enable writing to the node, be sure to include the /PATCH switch in the input specification.

### 3.3 Examining an IBMCOM Front-End Dump

To examine a dump from an IBMCOM front end, you must specify the dump file name as shown in the following example:

```
{@D6xyz(RET) ;TOPS-20}  
{.R D6xyz(RET) ;TOPS-10}
```

```
DDT11 5(54) = D6xyz/SYMBOLS=D6xyz.CRL[4,56] 18:41 13-Sept-82
```

```
Input: D6xyz.DMP(RET)
```

```
[78p core]  
[79p core]  
[110p core]
```

```
highest location is 157777
```

#### NOTE

Dumps produced by DTELDR from front ends must have /DTELDR appended to the dump file specification. All others do not require a switch.

### 3.4 Examining an Unrun Image

To examine an unrun system image from an IBMCOM front end, use the .BIN file from the distribution tape, as shown:

```
{@D6xyz(RET) ;TOPS-20}  
{.R D6xyz(RET) ;TOPS-10}
```

```
Input: D6xyz.BIN(RET)
```

## 4 USING DDT11 FOR DECnet-10/DECnet-20 FRONT ENDS

You can build your own DDT11 node-specific symbol files by using all the .STB files in the directory used to build the front end. The sections that follow show how to load and save DDT11 for DECnet-10/DECnet-20 front ends; examine running front ends, dumps, and unrun system images. Note that MCB is software for DECnet that runs on DN20.

#### 4.1 Loading and Saving DN20 Symbols

The following example contains all the files from the DN20 distribution that you can use to build a DDT11 file containing symbols common to all DECnet front ends.

```
{@DDT11(RET) ;TOPS-20}  
{.R DDT11(RET) ;TOPS-10}
```

DDT11 7E(114)

Input: CEXCOM.STB/STB(RET)  
[47p core]  
[48p core]

Input: DCP.STB/STB(RET)

Input: DMC.STB/STB(RET)  
[49p core]

Input: DMR.STB/STB(RET)

Input: DTE.STB/STB(RET)

Input: INI.STB/STB(RET)

Input: KDP.STB/STB(RET)

Input: MDT.STB/STB(RET)  
[50p core]

Input: NDT.STB/STB(RET)

Input: NML.STB/STB(RET)  
[51p core]  
[52p core]

Input: NMS.STB/STB(RET)

Input: NMX.STB/STB(RET)

Input: NS1.STB/STB(RET)

Input: NS2.STB/STB(RET)

Input: NSP.STB/STB(RET)

Input: RSX11S.STB/STB(RET)  
[53p core]

Input: RSXMS.STB/STB(RET)  
[54p core]  
[55p core]  
[56p core]

Input: SC.STB/STB(RET)  
[57p core]

Input: SC1.STB/STB(RET)

Input: SCX.STB/STB(RET)

Input: TLI.STB/STB(RET)

```

Input: TOP.STB/STB (RET)
Input: XPE.STB/STB (RET)
Input: XPT.STB/STB (RET)
[58p core]

Input: (CTRL/Z)
EXIT
{ @SAVE MCB (RET) ;TOPS-20 }
{ .SSAVE MCB (RET) ;TOPS-10 }
  MCB Saved

```

#### NOTE

The procedures for loading and saving DDT11 for the RJE-20 (DN200) remote station are the same as described for the DECnet-10/DECnet-20 MCB (DN20). There are additional symbol files for the DN200.

### 4.2 Examining a Running DECnet-10/DECnet-20 MCB Node

To examine nodes running DECnet-10/20 MCB software, you must specify the DECnet node name, as shown in the example below (DECnet MCB with node name D2102A):

```

{ @MCB (RET) ;TOPS-20 }
{ .R MCB (RET) ;TOPS-10 }

DDT11 7E(114) = MCB /SYMBOLS=DN20:XPT.STB[4,25] 11:01 8-Dec-82

DDT11 7E(114) ;DDT11 identifies itself
Input: /MCB:D2102A (RET) ;establishes a logical link to the node

```

If you intend to deposit into the node, you must include the /PATCH switch on the input line.

### 4.3 Examining Dumps from DECnet (MCB) Nodes

To examine a dump from a DECnet node, you must specify the node name in the dump file specification followed by the /MCBDMP switch, as shown in the following example:

```

{ @RUN MCB (RET) ;TOPS-20 }
{ .R MCB (RET) ;TOPS-10 }

DDT11 7E(114) = MCB /SYMBOLS=DN20:XPT.STB[4,25] 11:01 8-Dec-82

Input: SYS:D2102A.DMP/MCBDMP (RET)
[62p core]
[63p core]
[205p core]
highest location is 757777

```

#### 4.4 Examining an MCB Node Unrun System Image

You can use DDT11 to examine the file used to initially load an MCB node. The following example shows the dialog used for an MCB node:

```
$RUN MCB (RET)
.RUN MCB (RET)

DDT11 7E(114) = MCB /SYMBOLS=DN20:XPT.STB[4,25] 11:01 8-DEC-82

Input: SYS:D2102A.SYS/MCBSYS (RET)
[62p core]
[63p core]
[205p core]
highest location is 662003
```

### 5 USING DDT11 FOR AN RSX-20F FRONT END

The following sections explain using DDT11 to load and save symbols; and to examine running front ends, dumps, and unrun system images for the RSX-20F front end.

#### 5.1 Loading and Saving Symbols

The RSX-20F front end also contains specific symbols that must be loaded into DDT11. The symbol file that is distributed with the front end software is RSX20F.MAP. Therefore, you should specify this .MAP file when loading the symbols. (Note that this example shows the process as accomplished on a TOPS-20 host system, although the TOPS-10 front end can be loaded in a similar manner.)

```
$DDT11 (RET) ;load a standard copy of DDT11

DDT11 7E(114) ;DDT11 identifies itself
Input: SYSTEM:RSX20F/FESYM ;load symbols from the front-end
(RET) ;file, note that you must use the
; /FESYM switch. If the extension
; is other than .MAP or .SYM, you
; must specify it.

loaded nn symbols ;DDT11 tells how many symbols were
; loaded

Input: (CTRL/Z) ;exit DDT11

EXIT
```

The RSX-20F symbols are now loaded with the standard DDT11 symbols. It is important that you specify the /FESYM switch after the file name.

Now you must save the loaded DDT11, using the SAVE command:

```
$SAVE VB1445 (RET) ;save the front-end-specific DDT11
VB1445 saved ;VB1445.EXE is saved

$
```

## 5.2 Examining an RSX-20F Front End

You can examine the RSX-20F front end by typing the /FE switch for the input specification. The switch /FE:nm requires that you include the CPU number (n) and DTE number (m). If n is not specified, 0 is the default. For TOPS-20 systems, n will always be 0. For TOPS-10 systems, n is the number of the CPU to which the front end is connected. The following example shows the procedure for examining a front end on DTE0 from a TOPS-20 host system:

```
$RUN VB1445 (RET)
DDT11 7E(114) = VB1445/SYMBOLS = FE:RSX20F.MAP[1,8] 7:18
16-JAN-83
```

```
Input:/FE:0 (RET)
```

If you wish to deposit into the memory of the front end, include the /PATCH switch on the same line as /FE.

## 5.3 Examining RSX-20F Front-End Dumps

You can look at a dump from RSX-20F using the same procedure, but you must run the RSX-20F-specific version of DDT11, and the RSX-20F dump file. On TOPS-20, you must remember to rename the dump file to have a file name of six characters or less. Also, be sure to specify the extension (on TOPS-20, "file type"), if different from .LSD. The extension must be three characters or less.

Finally, you must specify the /DTELDR switch after the filespec of the dump file from RSX-20F.

The following example shows the dialog used to initiate examination of a dump file from an RSX-20F front end on TOPS-20.

```
$RUN VB1445 (RET) ;Run the specific version of DDT11
DDT11 7E(114) = VB1445/SYMBOLS = FE:RSX20F.MAP[1,8] 7:18
20-JAN-83
Input: SYSTEM:DUMP11.BIN/DTELDR (RET) ;Type the dump file name
```

In this example, the node-specific version of DDT11 is named VB1445.EXE. The dump file had been renamed to DUMP11.BIN.

## 5.4 Examining an RSX-20F Front-End Unrun Image

The RSX-20F file system contains a file called RSX20F.SYS. This file can be examined to see an unrun copy of RSX-20F. The actual core image of RSX-20F can be found in the file [0,0]CORIMG.SYS on the front-end file system. This is the image that the SAV function modifies when patching the front end.

The following example shows the dialog used to examine unrun system images for an RSX-20F front end:

```
$RUN VB1445 (RET)
DDT11 7E(114) = VB1445/SYMBOLS = FE:RSX20F.MAP[1,8] 7:18
20-JAN-83
Input: RSX20F.SYS/MCBSYS (RET)
```



## 6 SETTING OUTPUT MODES

After the appropriate files are loaded, DDT11 output can take the form of instructions, numbers, bytes, ASCII text, or addresses. You can control the output in either temporary or permanent mode. Temporary mode exists until you press the RETURN key. Permanent mode exists until you change it with a new mode instruction.

Numeric information can be displayed in any numeric radix from 2 through 16 (binary). In hexadecimal, a "." must be typed for decimal input. The default radix is octal. To set the numeric radix temporarily, type:

<ESC>nR           Where n is the temporary radix number.

To set the radix in permanent mode, type:

<ESC><ESC>nR       Where n is the permanent radix number.

You can control the format of output to your terminal by using the following commands. If n is present, it must be entered as an octal number, a decimal number (identified by a decimal point or "8" or "9" within the number), or an expression enclosed by parentheses such as (3\*3). Always include decimal point when typing in a decimal number. The default format is instruction format.

<ESC>nA           Sets absolute address format in temporary mode. In this mode, the address field (right half) of each word is typed out in absolute numeric form. Each typeout will consist of n addresses. If you omit n, the default is one address.

<ESC><ESC>nA       Sets address format in permanent mode.

<ESC>nB           Sets byte format in temporary mode. The two bytes in the word will be switched, interpreted, and the result displayed in numeric. Each typeout will consist of n bytes. If you omit n, the default is two bytes.

<ESC><ESC>nB       Sets byte format in permanent mode.

<ESC>nC           Sets current numeric word format in temporary mode. Numeric word format will be displayed in the current radix. Each typeout will consist of n words. If you omit n, the default is one word.

<ESC><ESC>nC       Sets current numeric word format in permanent mode.

<ESC>nF           Sets radix 50 typeout mode for n bytes.

<ESC><ESC>nF       Sets radix 50 format in permanent mode.

<ESC>nI           Sets IBM format in temporary mode. Each typeout will consist of n EBCDIC bytes. If you omit n, the default is two bytes.

<ESC><ESC>nI       Sets IBM format in permanent mode.

<ESC>nR            Sets output radix to "n" in temporary mode.

<ESC><ESC>nR        Sets output radix in permanent mode.

<ESC>nS            Sets symbolic format in temporary mode. The typeout will consist of instructions of one, two, or three words, depending on the type of instruction. The variable n specifies the number of instructions per typeout. If you omit n, the default is one instruction.

<ESC><ESC>nS        Sets symbolic format in permanent mode.

<ESC>nT            Sets text format in temporary mode. Each typeout will consist of n ASCII characters. If you omit n, the default is two characters.

<ESC><ESC>nT        Sets text format in permanent mode.

## 7 USING DDT11 TO EXAMINE MEMORY

In DDT mode, your terminal input and output from the program take place at a defined location in memory. This location is designated by the location pointer. The location pointer is an invisible mark in memory from which DDT11 will output text at your request, and where your changes are deposited. The symbol ".", by itself, represents the current location.

### 7.1 Moving the Location Pointer

Each address in memory designates a location. DDT11 allows you to examine one location at a time on your terminal. You can access the locations in memory that you wish to examine or change by moving the location pointer to that location, and by opening the location. When a location is "open," you can examine or change its contents.

When you first start DDT11, the location pointer is set to location 0. To move the location pointer to another location, type the octal address of that location, or the symbolic expression in the program, followed by a slash (/). This action moves the pointer, opens the location, and displays the contents of the location. For example:

```
3001/            ;type this to move to location 3001
```

or

```
REV0/           ;type this to move to REV0
```

When you type /, DDT11 will open the location and display its contents on your terminal. You can use [ whenever a / is accepted. The [ always displays contents in the current radix; / displays contents in the current output mode.

After you reach the desired location, you can move your location pointer to the adjacent locations. Press the line-feed key <LF> to move to the next location, open it and display its contents. Press the up-arrow, or circumflex (^), or backspace (^H) to move to the previous location, open it, and display its contents. The up-arrow is echoed on your terminal. <ESC><RET> gets back the previously opened location.

## NOTE

PDP-11 instructions can be 1, 2, or 3 words long (2, 4, or 6 bytes). A line-feed given in symbolic format (<ESC>S) will display the next succeeding instruction regardless of the length of the instruction. The up-arrow will move to the previous address and try to interpret that as an instruction.

In all other formats, line-feed moves down n bytes and up-arrow moves up n bytes.

## 7.2 Displaying the Contents of a Location

You can see the current location in any one of various modes by typing one of the following:

. = (period equal) displays the address of the current location  
; (semicolon) opens and displays the location in current typeout mode  
\_ (underscore) opens and displays the location as an instruction  
= (equal) opens and displays the location in current numeric radix

Many instructions indicate movement to another part of the program. Such instructions may contain a branching statement, or a reference to another location. To move the location pointer to a subroutine or referenced address, press <TAB> at that location. For example:

```
3001/ BR 1466 TAB ;display current location and press  
; <TAB>  
1466 RESET ;DDT11 moves to location of BR and  
; displays contents
```

If the <TAB> key is used on other than a branch instruction, DDT11 sets the location pointer to the address specified by the last quantity in the current location (whether it was DDT11 typeout or input), and will move to that location.

Once at the routine that you branched to, you can move back to the location at which you pressed <TAB> by pressing <ESC> followed by a RETURN. Pressing <ESC> before a circumflex (^) moves back to the location, and then moves to the previous location and displays its contents. Press <ESC> followed by <LF> to move back and then move to the next location.

### 7.3 Searching for a Word

DDT11 has a search facility that will scan a specified area in memory for either a match or a no-match condition on a 16-bit search argument. The formats of the two commands are as follows:

```
loaddr<hiaddr>value (ESC)W ;for all matches in range
loaddr<hiaddr>value (ESC)N ;for non-matches in range
```

Where:

loaddr	is the lower limit of the search area.
<hiaddr>	is the upper limit of the search area.
value	is the search argument (the 16-bit quantity to be searched for).
<ESC>W	is the DDT11 command to search for all words that match the search argument.
<ESC>N	is the DDT11 command to search for all words that do not match the search argument.

If you do not specify search limits, the previous limits are used. The default limits are zero when examining a running node. When reading a dump file, the default limits are the entire file.

### 7.4 Using Masks to Search for Matches

It is possible to specify a partial string, which DDT11 can use to find matches on that string. DDT11 uses a mask facility to provide for the matching of specific bits in a word. The initial value of the mask is 177777 (match on all 16 bits). To change the mask, enter the following:

```
(ESC)M/ 177777 newmask(RET)
```

Where:

<ESC>M/	is the DDT11 command to display the current mask.
177777	is DDT11's reply (the current mask, displayed in octal).
newmask	is the new mask to be in effect.

For example, to perform a search that matches only on the right half of a word (bits 7 through 0), set the mask equal to 377. A match on only the left half of a word, bits 15 through 8, would require a mask of 177400.

## 8 USING DDT11 TO DEPOSIT INTO MEMORY

To patch a program that is running in a node or make changes to a binary file residing in auxiliary storage, run the node-specific DDT11. Note that to patch a running node you must have privileges, and you must specify the /PATCH switch on the input specification line. Use the examine functions to locate and open the location to be patched. For example, the following examine entry displays location 3314. (DDT11 is in numeric single-word typeout mode.)

```
3314/ 12737
```

The location pointer is now set to 3314. If you now enter an expression followed by <RET>, <LF>, or ^, the contents of 3314 will be replaced by this new expression.

If you enter an octal number consisting of more than 16 bits, only the rightmost 16 bits will be deposited.

To enter ASCII text, type a double quote ("), followed by a character to serve as a delimiter. Then type the string of characters, and repeat the delimiter. Each character is stored as 7 bits plus a zero parity bit. The string is stored at the current location.

To type in multiple words or expressions, separate them with commas.

You can use any of the following expression operators:

+	addition
-	subtraction
*	multiplication
'	division
!	inclusive OR
&	and
<space>	usually equivalent to +
<CTRL/X>	exclusive OR

Operators are performed in the following order:

1. Logical operations (from left to right)
2. Multiplication and division (from left to right)
3. Addition and subtraction (from left to right)

During typein, you can use any of the following before you press RETURN, to edit the input line:

DELETE	Deletes the previously typed character.
--------	---

## NOTE

The DELETE key echoes differently, depending on the type of terminal you are using and the type you have specified (for example, /LA36 or /VT52). If a hardcopy terminal, DELETE echoes the deleted characters between slashes. If a display terminal, DELETE erases the deleted characters from the screen.

<CTRL/W>       Deletes the previous expression (refer to your system commands manual for more detail).

<CTRL/U>       Deletes the whole input line and rings the bell.

<CTRL/R>       Reprints the input line.

## 9 STARTING EXECUTION

Starting execution is valid only when examining a dump, system image, or a non-running node. This procedure is not recommended for RSX-20F or DECnet.

When you are finished examining and writing into memory, you can start execution with the <ESC>G command. You can specify an address or symbol in front of <ESC>G to start execution at that point. If you do not specify a starting location, execution starts at the current location. For example, in FOO<ESC>G, FOO is the starting location.

To simulate execution of a single instruction, use the <ESC>X command. As in <ESC>G, you can type an address or symbol in front of <ESC>X to specify the location. To specify the number of instructions to execute, type the number of instructions between <ESC> and X.

To execute a call to a subroutine (without single-stepping the entire subroutine), type <ESC><ESC>X. Execution will continue until control returns to the subroutine caller.

## 10 MONITORING A LOCATION

When you are examining a running node, it may be useful to monitor a particular location for changes. The <ESC>V command can be used to display the contents of an opened location when the contents change. For example, to open location FRECNT and monitor its contents, enter:

```
FRECNT/ 141 (ESC)V
```

DDT11 will then display each observed change in contents as:

```
FRECNT/ 143
FRECNT/ 141
FRECNT/ 142
FRECNT/ 145
FRECNT/ 141
.
.
.
```

To end monitoring, type any character.

#### NOTE

If changes are occurring rapidly, some may not be detected.

The masking feature of the search function is also active during the monitor function. If the initial mask value is in effect (177777), the monitored location is displayed when any one of the 16 bits changes. If, for example, you want to monitor a change in bit 12 of some status word, set the mask as follows:

```
(ESC)M/ oldmask 10000 (RET)
```

Where:

<ESC>M/	is the command to display the current mask and set the location pointer.
oldmask	is the current mask displayed by DDT11.
10000	is the new mask with only bit 12 set on.

#### 11 DUMPING MEMORY

Occasionally, it is desirable to dump portions of memory to a line printer or a disk file. Use the <ESC>D command to initiate a dump and set the dump limits with the following command:

```
loadaddr<hiaddr> (ESC)D
```

Where:

loadaddr	is the lowest address to be dumped.
<hiaddr>	is the highest address to be dumped.
<ESC>D	is the dump command.

DDT11 will then prompt you for a file specification:

FILE:

Enter a file name and extension. If you omit the extension, the default is .LSD.

When the file specification is accepted by DDT11, the file is opened, the dump is recorded, and the file is then closed. Therefore, each range of memory locations that is dumped is recorded in a separate dump file.

If a file of the same name already exists, the new dump will be appended to the old dump. If the dump is part of a command file, the device specification for the command file will be used for the dump file.

The current output format is used for output. To create a file that looks like an assembly listing, use the /DISASSEMBLE switch after the dump file name.

## 12 USING SYMBOLS

DDT11 always has the standard PDP-11 instructions defined as symbols. The current location pointer may be referred to with the period (.). The last word output by DDT11 or input to DDT11 may be referenced with the command <ESC>Q. For example:

```
2002/  MOV 6(R5),@(R5)+  (ESC)Q=6
2020/  BR 1466          (ESC)Q=622
```

When you input PDP-11 instructions, DDT11 defaults to relative addressing. If you want to input absolute addresses, precede the address by the symbol, @#. For example:

```
.R DDT11 (RET)
DDT11 7E(114)

Input: /CORE:4K/GO/PAT (RET)
[56p core]
[60p core]
[61p core]

520<FOO:          FOO=520
500/  HALT        JMP FOO

500/  JMP FOO     =167 14

500/  JMP FOO     JMP @#FOO
500/  JMP @#FOO  =137 520
```

(CTRL/Z)

.

### 12.1 Defining Symbols

You can define a symbol with a value equal to the current location counter by typing the new symbol and terminating it with a colon (:). For example:

```
5570/  RESET  START:
```

You can also directly assign a value to a symbol by typing the value, a left angle bracket (<) and the symbol, terminated by a colon. For example:

```
105<Q:
```

### 12.2 Controlling Symbol Typeout

Type <ESC>K if you wish to make the last symbol typed (by you or DDT11) unavailable for typeout from DDT11. You can also remove the last symbol typed from the DDT11 symbol table by typing <ESC><ESC>K. Typing sym\$K and sym\$\$K will achieve the same results, respectively.



### 12.3 Matching Symbols

If you type a question mark (?) when entering a symbol, DDT11 will search its symbol table for partial matches. A partial match is any symbol that begins the same as the partially typed-in symbol. If any partial matches are discovered, DDT11 will type:

following are partial matches

DDT11 will then type all of the partial matches and their values in the current output radix. If the symbols are suppressed, the value will be followed by:

spd

If no symbols match the partially typed symbol, DDT11 will respond:

no partial matches for symbol

After displaying the partial matches, DDT11 will retype the input and you can continue typing your symbol.

### 12.4 Completing Partial Symbols

If you type CTRL/F with a symbol, DDT11 will search for partial symbol matches. DDT11 will then add as many unambiguous characters as it can to the typed-in symbol. If the symbol has no matches, DDT11 will give a bell response. If the symbol is not ambiguous, DDT11 will fill in the unambiguous portion (if any) and then give a bell response.

For example, type:

FO CTRL/F

DDT11 will search its symbol tables for symbols beginning with FO. If it finds only FOOP, it will complete the symbol for you by finishing it with OP, and then it will give the bell signal.

If, however, DDT11 knows the symbols FOOP and FOOL, it will add one "O" and then give a bell response, because the symbol is ambiguous.

## 13 USING COMMAND FILES IN DDT11

You can use a command file as input to DDT11. To start a command file, type the <ESC>Y command. After you type <ESC>Y, DDT11 will ask where to log responses. You may type a file name for the log file, or TTY: to see output on your terminal. Then DDT11 prints the FILE: prompt. Here you type the name of the command file. (See Appendix B for a sample command file.)

### 13.1 Command File Echoing

Usually, as characters are read from a command file, they are echoed into the log file. If a CTRL/S is encountered in the command file, echoing will be suspended until a CTRL/Q is encountered.

## 13.2 Using Macro-Style Arguments

During the execution of command files, CTRL/A is used to delimit symbols whose values are to be converted to ASCII text. For example, if the symbol FOO has a value of 13, and if the command file contains the string:

```
L CTRL/A FOO CTRL/A BLK
```

the string will be processed as if it were:

```
L13BLK
```

## 13.3 Using Tags in Command Files

Command files may contain tags. Tags are of the form:

```
%tag:
```

Where tag is an alphanumeric string.

The first character after the percent sign (%) should be alphabetic. Only the first five alphanumeric characters are used. Be careful when you choose tag names. A text input string with a percent sign might be confused with a tag elsewhere in the file.

Command files may contain unconditional branches by including %GOTO %TAG commands. The %TAG may be anywhere in the command file.

Conditional branches may be included in the command files. For numeric tests, the expression is a general expression. For DEF or NDF, the expression is a single symbol name. If the condition is true, control will be transferred to %TAG. The conditions which can be tested for are:

DEF	The symbol is tested for existence.
DF	Same as DEF.
E	Same as EQ.
EQ	Expression is tested for a zero value.
ERR	Tests and clears the error flag. The error flag is set if an undefined symbol is used in an expression, or if an illegal command is typed.
G	Same as GT.
GE	The expression is tested for a positive value.

GT           The expression is tested for a positive, nonzero value.

N            Same as NE.

NDF          The symbol is tested for nonexistence.

NE           The expression is tested for a nonzero value.

L            Same as LT.

LE           The expression is tested for a zero or negative value.

LT           The expression is tested for a negative value.

#### 14 DDT11 COMMAND FILE FOR AN RSX-20F FRONT END

The following example is a DDT11 command file for an RSX-20F front end. The notations used are:

\$ = <ESC>  
 ^S = <CTRL/S>  
 ^Q = <CTRL/Q>  
 ^Z = <CTRL/Z>

Example:

```
@ddt11 (RET)
DDT11 7F(106)

Input:      vbl445.map/fesym
[46p core]
[47p core]
[48p core]
[49p core]
[50p core]
[51p core]
[52p core]
[53p core]
[54p core]
[55p core]
%Loaded 1216 symbols.

Input: (CTRL/Z)
EXIT
@save vbl445 (RET)
  VB1445.EXE.8 Saved
@type dumpfe.ddt (RET)
^S""modified by David Weaver"

""^QRSX20F crash analysis command file^S"^Q
^S""^QRSX20F version:^S"$10T.VERNO/^Q
^S""^QCrashed on:
^S"$12R$1B.MON$/$Q377+1=.DAY$/$Q377+1=$C.YEAR/^Q
^S""^QRSX20F crash_code:^S"$3T0/^Q
^S""^QRunning on KL serial -S"$12R$C.CPUSN/^Q
^S""^QCurrent task:-S"$A.CRTSK/^Q
^S""^QBytes left in the Free-pool:^S".FREPL+2[^Q
^S""^QBytes left in Big Buffer:^S".BGBUF+2[^Q
^S""^QNumber of lines locally shut off:^S".S2IDC[^Q
^S""^QNumber of lines with clock requests:^S".TTS2F[^Q
^S""^QSaved stack pointer:^S"$CSPSAV/$Q<ZZ:^Q
^S%IF EQ ZZ %REG
^Q
```

^S""^QStack items:^S"^Q

^S""^QR5:^S"ZZ/\$A./^Q  
^S""^QPC:^S".+2/\$A./^Q  
^S""^QPS:^S".+2/\$A./^Q  
^S""^QR5:^S".+2/\$A./^Q  
^S""^QPC:^S".+2/\$A./^Q  
^S""^QPS:^S".+2/\$A./^Q  
^S  
%REG:  
^Q

^S""^QRegisters:^S"^Q

^S""^QR0:^S"40/\$A./^Q  
^S""^QR1:^S".+2/\$A./^Q  
^S""^QR2:^S".+2/\$A./^Q  
^S""^QR3:^S".+2/\$A./^Q  
^S""^QR4:^S".+2/\$A./^Q  
^S""^QR5:^S".+2/\$A./^Q  
^S""^QSP:^S".+2/\$A./^Q  
^S""^QPC:^S".+2/\$A./^Q

@run vb1445 (RET)  
DDT11 7F(106) = VB1445 /SYMBOLS=DSK:VB1445.MAP[4,144] 14:32  
12-Nov-81

Input: remote.dmp/dte (RET)  
[59p core]  
[60p core]  
[96p core]  
highest location is 157777

\$y  
Log responses where: tty:

FILE: dumpfe

dump made 0:09:26 12-Apr1-83 of /FILE:DSK:REMOTE.DMP[4,144] 14:35  
8-Feb-83  
by DDT11 7F(106) = VB1445 /SYMBOLS=DSK:VB1445.MAP[4,144] 14:32  
12-Nov-81

RSX20F crash analysis command file

RSX20F version: VB14-45h  
Crashed on: 2. 8. 1983.  
RSX20F crash code: TBT  
Running on KL serial 2102.  
Current task: NULTSK  
Bytes left in the Free-pool: 4614  
Bytes left in Big Buffer: 2000  
Number of lines locally shut off: 0  
Number of lines with clock requests: 0  
Saved stack pointer: 25136

Stack items:

```
R5: 4      UC.TTY
PC: 11676  CR$TBT
PS: 301    QI.VER+1
R5: 0      0
PC: 5542   FETBL
PS: 164    H.LUT+34
```

Registers:

```
R0: 71304  71304
R1: 71304  71304
R2: 100    .INTEN
R3: 4676   IO.SSO+276
R4: 177777 K.LUSR
R5: 174412 174412
SP: 11426  EMGSTK
PC: 173606 173606
```

File TTY: . [4,144] 1-Jan-64 written

Finished command file

CTRL/Z

@

## 15 WRITING A BINARY FILE

After DDT11 has read in a binary file, the <ESC>P command can be used to write another binary file. Switches may accompany the file specification. Legal switches include:

```
/BM873   To indicate tape wanted for punching a BM873 ROM.
/IMAGE   To copy a file to a PTP with /I.
/M9301   To make a tape for a M9301 ROM.
/MCBSYS  To patch a task image or an MCB system image.
/PACKED  To make a packed binary file suitable for BOOT11 or
          NETLDR. This is the default switch.
```



**APPENDIX A**  
**SUMMARY OF DDT11 COMMANDS**

The following table contains all the commands (characters and escape sequences) that are significant to DDT11. The first column of Table A-1 shows the command character, and the second column contains a brief description of the function of each command.

**Table A-1: DDT11 Commands**

Character	Function
<CTRL/A>	Delimits symbols whose values are to be converted to ASCII text, in command files.
<CTRL/C>	Aborts DDT11 execution and returns to the operating system.
<CTRL/F>	Requests DDT11 to supply the rest of the symbolic name (if unique).
Backspace	Moves pointer to previous location and prints its contents.
<TAB>	Opens and displays the contents of the branch address, if currently at a branching instruction.
<LF>	Stores a value (if available) in the open location, closes it, then displays the next sequential instruction.
<RET>	Stores a value (if available) in the open location and then closes it.
<CTRL/O>	Suppresses output.
<CTRL/Q>	Resumes output; also, within a command file, <CTRL/Q> resumes logging.
<CTRL/R>	Re-displays a partial command.
<CTRL/S>	Pauses output; also within a command file, <CTRL/S> stops logging.
<CTRL/T>	Forces USESTAT command (if RTCOMP is not set); otherwise, <CTRL/T> is illegal.
<CTRL/U>	Deletes current typein.

SUMMARY OF DDT11 COMMANDS

Table A-1: DDT11 Commands (Cont.)

Character	Function
<CTRL/W>	Deletes last symbol in typein.
<CTRL/X>	Is the logical exclusive OR operator.
<CTRL/Z>	Requests DDT11 to clean up and exits to the operating system.
<ESC>	Depends on next character for meaning.
<SPACE>	Is the same as "+" between symbolic names.
!	Is the logical inclusive OR operator.
"	Initiates text typein.
#	Denotes immediate mode in instruction typein.
\$	Forms symbolic names.
%tag:	Denotes a tag in a command file.
&	Is the logical AND operator.
'	Is the division operator.
(	Is the symbol for start-of-index register or start of expression.
)	Is the symbol for end-of-index register or end of expression.
*	Is the multiplication operator.
+	Is the addition operator.
,	Is the separator for multiword expressions.
-	Is the subtraction operator.
.	Stands for current location pointer if used alone; otherwise, stands for a character in a symbolic name.
addr/	Moves pointer to specified location and prints contents.
0-7	Is the octal digit. Used either as a character in a symbolic name, or as a character in an octal or decimal number.
8-9	Is used either as a character in a symbolic name, or as character in a decimal number.
;	Displays current location in current mode.
val<symb:	Assigns the value (val) to the symbol (symb).



SUMMARY OF DDT11 COMMANDS

Table A-1: DDT11 Commands (Cont.)

Character	Function
=	Retypes previous input or output in current radix.
loadr<hiaddr>	Defines the lower and upper limits for a search, dump, or output.
?	Prints a list of symbols that match.
@	Identifies address as indirect.
@	Denotes absolute addressing in instruction typein.
addr[	Moves pointer to specified location and prints its contents in numeric mode.
\	Opens the new location without changing the position of the location pointer.
^	Moves pointer to previous location and prints its contents.
-	Prints contents of location as instruction.
<DELETE>	Deletes the previous character that you typed. (Also known as RUBOUT.)
<ESC><LF>	Returns pointer to location before <TAB> and prints contents of next location.
<ESC><RET>	Returns pointer to location before <TAB> and prints contents of the location.
<ESC>nA	Sets address format for typeout in temporary mode.
<ESC><ESC>nA	Sets address format in permanent mode.
<ESC>nB	Sets byte format for typeout in temporary mode.
<ESC><ESC>nB	Sets byte format in permanent mode.
<ESC>nC	Sets current word format for typeout in temporary mode.
<ESC><ESC>nC	Sets current word format in permanent mode.
<ESC>D	Dumps memory to a disk file.
<ESC>G	Starts execution.
<ESC>nI	Sets IBM format for typeout in temporary mode.
<ESC><ESC>nI	Sets IBM format in permanent mode.

SUMMARY OF DDT11 COMMANDS

Table A-1: DDT11 Commands (Cont.)

Character	Function
<ESC>K	Makes previously referenced symbol unavailable for output.
<ESC><ESC>K	Removes previously referenced symbol from symbol table.
<ESC>M	Displays the current mask.
<ESC>N	Searches for words that do not match the specified string.
<ESC>P	Outputs a binary file.
<ESC>Q	Types out the last word referenced.
<ESC>nR	Sets numeric radix for typeout in temporary mode.
<ESC><ESC>nR	Sets numeric radix in permanent mode.
<ESC>nS	Sets symbolic format for typeout in temporary mode.
<ESC><ESC>nS	Sets symbolic format in permanent mode.
<ESC>nT	Sets text format for typeout in temporary mode.
<ESC><ESC>nT	Sets text format in permanent mode.
<ESC>V	Types out the current location every time its contents change.
<ESC>W	Searches for words that match the specified string.
<ESC>X	Executes a single instruction.
<ESC><ESC>X	Executes a subroutine call.
<ESC>Y	Starts a command file.
<ESC>^	Returns pointer to location before <TAB> and prints contents of previous location.

**APPENDIX B**  
**SWITCHES FOR THE INPUT SPECIFICATION**

The following is a list of all the switches that you can use in the input specification:

/11	Same as /PDP11.
/8	Same as /PDP8.
/BINARY	Reads a file produced by MACDLX.
/CORE:nK	Allocates the specified amount of core to read a dump or binary file (1 core:28K).
/DTEHDR	Reads the file as a DTEHDR dump. This switch is used when looking at a RSX-20F dump.
/DUMP	Reads the file as a NETLDR dump.
/FE:nm	Indicates that the front end is running an RSX-20F. The value n is the number of the CPU to which the front end is connected. The value m is the DTE to which the front end is connected. If n is not specified, then 0 is implied. You require n only for systems with more than one CPU (SMP).
/FESYMB	Loads symbols from a map file for the RSX-20F front end.
/GO	Uses the core image saved with the DDT11 program.
/HELP	Produces text explaining all the DDT11 switches.
/LA36	Defines your terminal to be a hardcopy terminal. This affects echo of deleted characters.
/LINE:n	Specifies the line number on a running node of an adjacent remote node that you want to examine.
/LSD	Same as /DUMP.
/MCB:node	Examines a running DECnet (MCB) node.
/MCBDMP	Examines a DECnet node dump.
/MCBSYS	Examines an unrun DECnet (or RSX-20F) system image.
/MERGE	Reads a new dump and appends it to the existing dump file.

## SWITCHES FOR THE INPUT SPECIFICATION

/NODE:node-id	Specifies the ANF-10 node that you wish to examine. The node-id can be a node name or node number.
/PATCH	Enables deposits. You must have privileges to deposit into a running node.
/PDP11	Uses only PDP-11 symbols. This is the default setting.
/PDP8	Uses only PDP-8 symbols. You must specify this switch when you examine a PDP-8 node (that is, a DN92) or dump.
/PORT:n	Specifies the port number of the node you wish to examine.
/RELOCA:sym	Adds an offset to symbols loaded from an RSX-20F dump, where sym is the offset number or symbol.
/STB	Reads an MCB symbol table produced by the DECnet NETGEN procedure.
/SYMBOL:n	Loads symbols from a CREF listing file starting at n, where n is the disk blocknumber. The disk block number is optional.
/UBA:n	Specifies the UBA when running standalone on a KS10, where n is the UBA number of the node.
/VT52	Defines your terminal to be a video terminal. This affects echo of deleted characters.

After specifying the node or file specification, DDT11 enters DDT mode, as explained in Section 6.0.

The default file extensions are shown in the following table:

<u>Switch</u>	<u>Default File Extensions</u>
/SYMBOL	., .LST, .CRF
/FESYM	., .MAP, .SYM
/STB	, .STB, .SYM
/MCBSYS	., .SYS, .MCB
/MCBDMP	., .DMP, .SYS
/BINARY	
/LSD	
/DUMP	., .LSD, .BIN
/DTELDR	., .BIN

## INDEX

- Accessing locations in memory, 12
- ANF-10 node
  - examining a dump, 4
  - examining a running, 3
  - examining unrun system image, 4
  - loading and saving symbols, 3
- Binary file
  - making changes, 15
  - writing, 23
- Choosing tag names, 20
- Command file
  - for RSX-20F front end, 21
- Command files
  - using, 19
  - using Macro-style arguments, 20
  - using tags, 20
- Controlling
  - output format, 11
  - symbol typeout, 18
- DDT11 Commands
  - summary, A-1
- DDT11 default
  - relative addressing, 18
  - with PDP-11 instructions, 18
- DDT11 input switches, B-1
- DDT11 mask facility
  - for a partial string, 14
- DDT11 output mode, 11
- DDT11 search facility
  - for match condition, 14
  - for no-match condition, 14
- DECnet front end
  - loading and saving symbols, 7
- DECnet MCB node
  - examining a dump, 8
  - examining a running, 8
  - examining unrun system image, 9
- Default file extension table, B-2
- Defining symbols, 18
- Displaying contents
  - of locations in memory, 13
- Dumping memory, 17
- Examining
  - a dump
    - ANF-10 node, 4
    - DECnet MCB node, 8
    - IBMCOM front end, 6
    - RSX-20F front end, 10
  - a dump file, 2
  - a running
    - ANF-10 node, 3
    - DECnet MCB node, 8
    - RSX-20F front end, 10
  - a running node, 2
- Examining (Cont.)
  - IBMCOM front end running DN6x, 5
  - unrun system image
    - ANF-10 node, 4
    - DECnet MCB node, 9
    - IBMCOM front end, 6
    - RSX-20F, 10
- IBMCOM front end
  - examining a DN60 front end, 5
  - examining a dump, 6
  - examining unrun image, 6
  - loading and saving symbols, 5
- Initializing DDT11
  - ANF-10 Nodes, 1
  - DECnet-10/20 MCB (DN20), 1
  - IBMCOM-10/20 Nodes, 1
  - RJE-20 (DN200) Nodes, 1
  - RSX-20F Front End, 1
- Input specification
  - DDT11 switches, B-1
- Loading and saving symbols
  - ANF-10 node, 3
  - DECnet front end, 7
  - IBMCOM front end, 5
  - RJE-20 (DN200), 8
  - RSX-20F front end, 9
- Location
  - monitoring, 16
- Locations in memory
  - accessing, 12
  - displaying contents, 13
- Macro-style arguments
  - in command files, 20
- Matching symbols, 19
- Memory
  - accessing locations, 12
  - depositing into, 15
  - dumping, 17
  - location pointer, 12
  - using DDT11 to examine, 12
- Monitoring a location, 16
- Output format
  - controlling, 11
  - setting, 11
- Partial symbol matches, 19
- Patching a running node, 15
- PDP-11 instructions
  - defined as symbols, 18
- Permanent output mode, 11
- Relative addressing
  - DDT11 default, 18

RJE-20 (DN200)  
  loading and saving symbols, 8  
RSX-20F front end  
  DDT11 command file, 21  
  examining a dump, 10  
  examining a running, 10  
  examining unrun system image,  
  10  
  loading and saving symbols, 9  
  
Setting output format, 11  
Starting execution  
  restriction for DECnet, 16  
  restriction for RSX-20F, 16  
  when examining a dump, 16  
  when examining a non-running  
  node, 16  
  when examining a system image,  
  16  
Summary  
  DDT11 commands, A-1  
Switches  
  DDT11 input specification, B-1  
Symbol typeout  
  controlling, 18  
Symbols  
  defining, 18  
  matching, 19  
  
Symbols (Cont.)  
  partial matches, 19  
  using, 18  
  
Table  
  of default file extensions, B-2  
Tags  
  choosing names, 20  
  form, 20  
  in command files, 20  
Temporary output mode, 11  
  
Using command files, 19  
Using DDT11, 1  
  for ANF-10 nodes, 3  
  for DECnet front ends, 6  
  for IBMCOM front ends, 5  
  for RJE-20 (DN200), 8  
  for RSX-20F front end, 9  
  initial dialog, 2  
  to deposit into memory, 15  
  to examine a running node, 2  
  to examine dump files, 2  
  to examine memory, 12  
  to examine unrun system images,  
  3  
Using symbols, 18  
  
Writing a binary file, 23

### READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

---

---

---

---

---

---

---

---

---

---

Did you find errors in this manual? If so, specify the error and the page number.

---

---

---

---

---

---

---

---

---

---

Please indicate the type of reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) \_\_\_\_\_

Name \_\_\_\_\_ Date \_\_\_\_\_

Organization \_\_\_\_\_ Telephone \_\_\_\_\_

Street \_\_\_\_\_

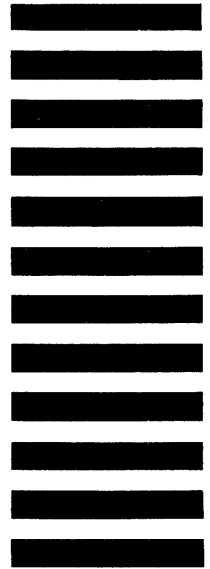
City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_  
or Country

Do Not Tear - Fold Here and Tape

**digital**



No Postage  
Necessary  
if Mailed in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

**SOFTWARE PUBLICATIONS**  
200 FOREST STREET MRO1-2/L12  
MARLBOROUGH, MA 01752

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line