

# **Professional Tool Kit DIBOL Message Manual**

**Order No. AA-Z515A-TK**

**October 1983**

**Supersession:** This is Version 1.6 and replaces Version 1.5

**Operating System:** VAX/VMS V3.3  
RSX-11M V4.3  
RSX-11M-PLUS V2.1  
P/OS V1.7

**Software Version:** Professional Developer's Tool Kit V1.7  
PRO/Tool Kit V1.0  
Professional Host Tool Kit DIBOL V1.6  
PRO/Tool Kit DIBOL V1.6

1st Printing, October 1983

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 specifications and drawings, herein, are the property of Digital Equipment Corporation and shall not be reproduced or copied or used in whole or in part as the basis for the manufacture or sale of items without written permission.

Copyright © 1983 by Digital Equipment Corporation. All Rights Reserved

The following are trademarks of Digital Equipment Corporation:

CTI BUS	MASSBUS	RSTS
DEC	PDP	RSX
DECmate	P/OS	Tool Kit
DECsystem-10	PRO/BASIC	UNIBUS
DECSYSTEM-20	Professional	VAX
DECUS	PRO/FMS	VMS
DECwriter	PRO/RMS	VT
DIBOL	PROSE	Work Processor
	Rainbow	

## CONTENTS

	Page
PREFACE	v
CHAPTER 1 INTRODUCTION	1-1
CHAPTER 2 COMPILER MESSAGES	2-1
2.1 MESSAGE FORMAT	2-1
2.2 MESSAGES	2-1
CHAPTER 3 RUN-TIME LIBRARY MESSAGES	3-1
3.1 MESSAGE FORMAT	3-1
3.2 MESSAGES	3-1
CHAPTER 4 DIBOL DEBUGGING TECHNIQUE (DDT) MESSAGES	4-1
4.1 MESSAGE FORMAT	4-1
4.2 MESSAGES	4-1



## PREFACE

This manual lists the messages produced by Professional Tool Kit DIBOL. Unless the meaning of a message is obvious, each message is accompanied by an explanation.

## MANUAL ORGANIZATION

This manual consists of four chapters. The first chapter provides an introduction and some general information. Each remaining chapter lists the messages for a particular utility; see the table of contents. The format of a particular message is described at the beginning of each chapter.

## ASSOCIATED DOCUMENTS

The following manuals comprise the Professional Tool Kit DIBOL documentation set.

- Professional TOOL KIT DIBOL USER'S GUIDE
- Professional Tool Kit DIBOL Message Manual
- Professional Tool Kit DIBOL Release Note and Installation Guide
- Introduction to DIBOL
- DIBOL-83 Language Reference Manual
- DIBOL-83 Compatibility Guide

The following manuals provide information on messages issued by the facilities indicated in parentheses.

- Professional 300 Series, Developer's Tool Kit, Volume 3 (Record Management Services - RMS)
- Professional 300 Series, Developer's Tool Kit, Volume 4 (File Management Services - FMS)
- Professional 300 Series, Developer's Tool Kit, Volume 5 (Task Building)



## INTRODUCTION

The components of Professional Tool Kit include the DIBOL compiler, the DIBOL run-time system modules (DBLRES.STB, DBLRES.TSK), the DIBOL Debugging Technique (DDT) object module (DBLDDT.OBJ), the DIBOL OSSL library module (DBPOSL.OLB), and the DIBOL UESL library module (DBPLIB.OLB). These components are described in the Professional Tool Kit DIBOL User's Guide.

During execution, each of these DIBOL components generates messages that report current status, successful completion, or error detection. The run-time messages (Chapter 3) refer to a running DIBOL program.

A fatal error always terminates execution; a non-fatal error may not terminate execution.

The messages are normally displayed at the terminal from which the program or utility is run. If, however, the program or utility is executed by means of the RUNJB subroutine, the messages appear on the P/OS message board.

In this manual, the following abbreviations are used to indicate the error message severity level:

- W           Warning.    This is a non-fatal error condition that may cause unexpected results. Corrective action may be necessary.
- E           Error.     This is a non-fatal error condition that will cause other errors during execution. Corrective action is necessary.
- T           Trappable.   This is a non-fatal run-time error condition that will cause other errors during DIBOL program execution. Program termination results unless the ONERROR statement is used to trap the error. This error type is associated only with DIBOL run-time errors.
- F           Fatal.     This is a fatal error condition that terminates the utility and returns control to the RSX-11M-PLUS monitor.
- NT          Non-trappable. This is a fatal run-time error condition that terminates DIBOL program execution and returns control to the RSX-11M-PLUS monitor. This error type is associated only with DIBOL run-time errors.





## 2.1 MESSAGE FORMAT

Compiler errors, detected during program compilation, are usually errors of syntax in specific DIBOL statements. The compiler messages describe the nature (what) and location (where) of these errors.

The format of a compiler message is as follows.

```
%DIBOL-L-text
```

where:

DIBOL is the facility identification for compiler messages.

L is the error severity level; see Chapter 1.

text is the explanatory text (what and where).

## 2.2 MESSAGES

The compiler messages are listed below in alphabetical order according to the message text. The first column shows the severity level (L); the second column shows the full text of the message. Below this line, additional information on the error is presented for some messages.

**E** Cannot open .INCLUDE file AAA

AAA represents the name of the file not opened.

**E** Close parenthesis missing

The number of left parentheses in an expression is greater than the number of right parentheses in that expression. Execution of this statement results in a non-trappable run-time error; see run-time error #3, Compilation error.

**W** Close parenthesis missing

The number of left parentheses in an expression is greater than the number of right parentheses in that expression. Execution of this statement does not produce a run-time error.

**E** Comma missing

Comma is expected but not found. Execution of this statement results in a non-trappable run-time error; see run-time error #3, Compilation error.

**W** Comma missing

Comma is expected but not found. Execution of this statement does not produce a run-time error.

**E** Decimal literal too big, truncated to DDD

DDD represents the value to which the literal is truncated.

**W** Decimal variable too big

The size of the decimal field is greater than 18.

**E** Directive error

The directive is not recognized as valid - misspelled?

**W** End quote missing

A decimal or alpha literal does not end with a quote (' or ").

**F** Error reading input file

The format of the input file is not readable.

**W** Extra characters at statement end - AAA

The statement includes data that was not expected. AAA represents a copy of the input that generated the error.

**E** Field too big or zero

An alpha field contains more than 16383 characters.

- F File exists and is protected - AAA**  
AAA represents the name of the file.
- E IF statement error - THEN without ELSE**  
Both statements must be present in a valid program construct.
- E Improper pairing of conditional compiler directives**  
A conditional compiler directive is not paired with an .ENDC statement.
- E .INCLUDE nested too deeply - file ignored - AAA**  
AAA represents the name of the file not included in the compilation.
- W Initialization error - no initialization done**  
A decimal field was initialized with alpha data, or vice versa.
- W Initial value missing**  
A data value does not follow the comma.
- W Initial value not allowed**  
Data initialization not allowed in a redefinition (overlay) RECORD statement.
- W Initial value too long, truncated to AAA**  
AAA represents the value to which the literal is truncated.
- W Initial value too short - AAA**  
AAA represents a copy of the input that generated the error.
- F Input file not found - AAA**  
AAA represents the name of the file.

**W** Invalid channel number

The channel number is greater than 15.

**E** Invalid data type

The data type must be alpha (A) or decimal (D).

**F** Invalid device

A file specification contains an invalid device specification.

**E** Invalid DIBOL statement AAA

AAA represents a copy of the input that generated the error.

**E** Invalid first statement

The first statement in the program must be RECORD, COMMON, or PROC.

**E** Invalid IF statement - AAA

An invalid statement or no statement is on the right side of the IF statement. AAA represents a copy of the input that generated the error.

**E** Invalid left side of assignment statement

Either an invalid variable or no variable is designated as the destination in an assignment statement.

**E** Invalid LPQUE keyword - AAA

AAA represents a copy of the input that generated the error.

**E** Invalid name AAA

AAA represents a copy of the input that generated the error.

**E** Invalid numeric literal - DDD

DDD represents a copy of the input that generated the error.

- E Invalid OPEN keyword - AAA**  
AAA represents a copy of the input that generated the error.
- E Invalid OPEN mode**  
The specified mode is not I, O, or U.
- E Invalid OPEN submode**  
The specified submode is not S, P, R, or I.
- E Invalid operand - AAA**  
AAA represents a copy of the input that generated the error.
- W Invalid PROC statement**  
The format of the PROC statement is incorrect, or the number of blocks specified is greater than 15.
- F Invalid switch**  
Compiler command line contains an invalid qualifier.
- E Invalid variable - AAA**  
AAA represents a copy of the input that generated the error.
- E Label missing**  
A program statement refers to a label that does not exist.
- W Label name too long - truncated to AAA**  
AAA represents the value to which the label is truncated.
- E Label not allowed**  
A statement label is associated with a compiler directive. This is not allowed.

- W**    **Label out of context block - AAA**
- AAA represents a copy of the input that generated the error.
- W**    **Line too long**
- A line can consist of 512 or fewer characters.
- F**    **Listing out of room**
- There is not enough space for the listing file on the device or account specified.
- W**    **Mismatched parentheses**
- The number of left parentheses does not equal the number of right parentheses.
- E**    **Missing or invalid operator**
- No comma appeared in the data formatting statements, or an invalid operator is detected within an expression.
- E**    **Missing or invalid subroutine name**
- The subroutine name is missing from a SUBROUTINE statement or an XCALL statement.
- E**    **Missing type or length**
- A field definition has no data type (A or D) or size specification.
- W**    **Multiply defined label**
- A statement label has been used with more than one DIBOL statement.
- E**    **Multiply defined symbol**
- A field name has been used more than once.
- F**    **No command on line**
- A blank command line was passed to the compiler.

**E No DO in DO-UNTIL statement**

Both statements must be present in a valid program construct.

**E No END for BEGIN**

Both statements must be present in a valid program construct.

**E No ENDUSING in USING statement**

Both statements must be present in a valid program construct.

**E No field statements in previous record**

A RECORD or a COMMON statement must include at least one field definition.

**F No file for input**

No input file was specified in the command line.

**E No FROM in FOR statement**

Both statements must be present in a valid program construct.

**W Non-standard syntax**

RSX-11M-PLUS-specific extensions to the DIBOL-83 standard are being used.

**E No PROC statement**

A PROC statement must be present in a DIBOL program.

**E No SELECT in USING statement**

Both statements must be present in a valid program construct.

**E No THRU in FOR statement**

Both statements must be present in a valid program construct.

- E** No UNTIL in DO-UNTIL statement
- Both statements must be present in a valid program construct.
- F** Object out of room
- There is not enough space for the object file on the device or account specified.
- W** Open parenthesis missing
- The number of left parentheses in an expression is less than the number of right parentheses in that expression.
- E** Overlay error
- There is no previously defined record to be overlaid.
- E** Premature termination of line
- A line contained less information than was expected.
- E** Record too big
- The size of the named record exceeds 16383 characters, or an overlay record exceeds the size of the record it overlays.
- W** Relational operand types do not match
- The operands in a relational operation are of different types; that is, one is alpha and the other is decimal. Both operands are assumed alpha.
- E** RMS error code - DDD
- DDD represents the RMS-11 error code.
- F** RMS file access error - file name AAA
- AAA represents the name of the file.



- E Stack overflow**  
Nesting of IF statements, BEGIN-END statements, or subroutine calls (or some combination) is too complex.
- E Statement too complex**  
Break the statement into two or more simpler statements.
- W Subroutine dummy argument not zero**  
The data division of an external subroutine contains a field definition with field size greater than zero.
- E Subscript too complex**  
Subscript has been nested to more than 18 levels.
- W Symbol name too long - truncated to AAA**  
AAA represents the truncated name.
- E Too many subscripts**  
More than two subscripts have been used.
- E Too many symbols or labels**  
More than 1023 symbols or 1023 labels have been used.
- E Too much data**  
More than 32767 bytes of data and literals were found in the program.
- E Undefined label - AAA**  
AAA represents a copy of the input that generated the error.

**E Undefined symbol - AAA**

A variable that is not defined in the Data Division is used in the Procedure Division. AAA represents a copy of the input that generated the error.

**W Unprocessed input files**

An END statement was encountered before all of the input files were compiled.

**E Variable or expression missing**

One or more required parameters is missing from a statement.

**E Wrong data type - AAA**

AAA represents a copy of the input that generated the error.

This chapter describes the run-time errors, detected during program execution, and the messages displayed in connection with these errors.

### 3.1 MESSAGE FORMAT

When DIBOL program execution is terminated by an error, a message is displayed indicating the cause of termination.

The general format of the message is as follows.

```
?Err nnn at line xxx in routine yyy message text
```

where:

Err	represents the word Error.
nnn	is the identification number of the error message.
xxx	is the line number of the statement at which the error occurred.
yyy	is the name of the routine in which the error occurred.
text	is the explanatory text.

"Err", "at line", "in routine", and the message text are obtained from the message file and are translatable.

This general form is followed, where applicable, by traceback information lines of the following format.

```
at line xxx in routine yyy
```

where xxx and yyy are the line number and the routine name as described above.

#### NOTE

The messages are normally displayed at the terminal from which the program is run. If, however, the program was executed by means of the RUNJB subroutine, the messages appear on the P/OS message board. -- The message described in the following paragraph does not appear in this case.

The following message is displayed after the set of one or more run-time messages.

Please write down the above message -- press RESUME to continue

You must press the <RESUME> key to clear the screen and continue processing.

In the following example, program execution is halted by the non-trappable (fatal) error whose number is 2 and whose message text is Return without CALL. The error occurred at line 37 in routine SUB3; SUB3 was called from line 192 of SUB2, and SUB2 was called from line 21 of the main program, PROG.

```
?Err 2 at line 37 in routine SUB3: Return without CALL
  at line 192 in sub2
  at line 21 in PROG
```

Please write down the above message -- press RESUME to continue

### 3.2 TRAPPABLE AND NON-TRAPPABLE ERRORS

Run-time errors are fatal and automatically result in program termination. Some, called trappable run-time errors, can be trapped with the DIBOL ONERROR statement; thus avoiding program termination. Run-time errors that can not be trapped are called non-trappable.

### 3.3 RUN-TIME MESSAGES

The run-time messages are described below in numerical order according to their error number (#) in the first column. The second column indicates whether the error is trappable (T) or non-trappable (NT). The third column shows the text of the message. Below the text, additional information on the error is presented.

#### 1 T End of File

The logical end-of-file was detected:

- when reading an input file (except via ACCEPT dfield).
- when requesting an ISAM record whose key value exceeds the greatest key already in the file.

#### 2 NT Return without CALL

The program executed a RETURN statement but no CALL or XCALL was in effect.

#### 3 NT Compilation Error

The program attempted to execute a statement line that contains an error previously detected by the compiler.

4 NT DIBOL Stack Overflow

Subroutine calls with too many arguments or subexpression nesting too deep.

5 NT Recursive External Call

An external subroutine attempted to call itself (XCALL statement) either directly or indirectly through another external subroutine.

6 NT Incorrect Number of Args

The number of arguments passed to an external subroutine is not equal to the number of arguments specified under the subroutine's SUBROUTINE statement.

7 T Subscript Error

This error can occur under either of the following conditions:

- The subscript value of a variable referenced by array subscripting specified a data element outside the program's data area, was zero, or was negative.
- The first subscript of a variable referenced by substring subscripting specified a value greater than that of the second subscript, one of the subscripts was a negative value, or the subscript specified a data element outside the program's data area.

8 NT Writing into a Literal

An external subroutine attempted to store data in a literal passed as an argument to the subroutine call.

9 T Not Enough Memory

Insufficient memory for one of the following: the

- file buffers in response to an OPEN statement
- file buffers in response to a call to the DELET or RENAM external subroutines
- file buffers in response to a STORE statement in ISAM
- message being sent in response to a SEND statement

10 NT Illegal Channel Number

The channel number specified in an OPEN, ACCEPT, DISPLAY, FORMS, WRITE, WRITES, READ, or READS, statement is out of the range for legal channel numbers (1-15).

11 NT Channel not Open

I/O operations were attempted before the specified channel was opened with an OPEN statement.

12-14 [reserved]

15 T Number Too Long

The program performed a calculation producing a result greater than 18 digits in size.

16 NT DIBOL Channel in Use

The program attempted to issue an OPEN statement to a channel that is already open.

17 T Bad File Specification

Incorrect syntax for the file specification in an LPQUE statement, an OPEN statement, or in an external call to RENAM.

18 T File Not Found

One of the following has occurred:

- The file specified in an OPEN statement does not exist.
- The file specified in an external call to RENAM does not exist.
- The file specified as the optional argument of a STOP statement is not installed.
- The target task in a SEND statement is not installed.

19 [reserved]

20 T Bad Digit

The alpha value being converted to a decimal value consisted of characters other than: space, +, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

21 T Bad OPEN

One of the following was attempted:

- a READ or a WRITE to a sequential file
- a WRITES to an ISAM file
- a WRITE or WRITES to a file in I mode
- a display to other than a terminal or a file opened in O:P mode
- an XCALL to TTSTS (dfield,ch) where the device on the channel is not the terminal
- a FORMS to a file opened in I, SI, U, or SU mode

22 T I/O error

The system detected bad data during an input or output operation. If the operation being performed is sequential (READS or WRITES), attempts to retry may cause loss or duplication of data.

23 T Line too Long

Either of the following has occurred:

- The data record being input is larger than the space reserved for it in memory by the associated RECORD statement in the data division.
- the file consists of fixed length records and the program attempted to write a record of improper length.

24 T No Space for File

The number of blocks of storage requested in the file specification of an OPEN statement is not available on this device.

25 [reserved]

26 NT Field or Record too Long

A record or field was referenced using subscripts which resulted in a size greater than 16,383 characters.

27 T Update of Non-File Device

The program attempted to issue an OPEN statement in update mode to a file that is not a relative file or to a non-file structured device (e.g., line printer or terminal).

28 T Illegal Record Number

A READ or WRITE statement specified a record number argument that is negative, zero, or greater than the number of records in the file.

29 NT Incompatible Compiler

The program was compiled using a version of the compiler that is incompatible with the run-time system.

30 T Divide by 0

The program performed an arithmetic operation that resulted in division by zero.

31 NT Argument Wrong Size

The program attempted to call one of the external utility subroutines supplied by DIGITAL using one or more arguments of the wrong length.

32 T Superseding Existing File

The program attempted to create an output file having the same name as an existing file. Detection of this condition is device-dependent.

- With disks, this condition is detected only if enabled by the FLAGS external utility subroutine (see the DIBOL-83 Language Reference Manual).

- With magnetic tape, this condition is always detected

33-36 [reserved]

37 T Device in Use

The program attempted to submit a print file via LPQUE to the printer while it was busy. This error is detected during execution of the statement.

38 T File in Use

The program attempted to open an output file (O mode) that was already opened.



39 [reserved]

40 T Record Locked

A READ, WRITE, ISAM READS, or STORE statement specified a group of records that is currently being accessed by another program (or by the same program on a different channel).

41 [reserved]

42 T No Device

An attempt was made to open a file on a device which was off line.

43-51 [reserved]

51 NT Too many messages

52 T Illegal Key

The specified ISAM key was not properly defined in the corresponding record in the data section of the program. That is, the key position was wrong. This message occurs with the READ, STORE, and WRITE statements in conjunction with ISAM.

53 T Key not Same

One of the following ISAM error conditions has occurred:

- The key of the record with an indexed READ does not match the key requested. The next higher approximate key is returned with this message.
- The key specified by an indexed WRITE or DELETE does not match the key of the record that was last retrieved (read). The WRITE or DELETE is not executed, and the retrieved record is still locked.
- The key specified by an indexed WRITE specifies a record that is either not locked or has not been read.

54 T No Duplicates

The program attempted to store a record with a duplicate key to an ISAM file where duplicate keys are not accepted.

- 55 [reserved]
- 56 T Not ISAM File
- The specified file was not an ISAM file and an attempt was made to open it in SI or SU mode. The message occurs with the OPEN statement.
- 57 [reserved]
- 58 T Job Startup Error
- One of the following has occurred:
- There is not enough memory.
  - The program specified with RUNJB or a STOP 'filename' was already running.
- 59 [reserved]
- 60 NT No Eligible Receiver
- The program attempted to send a message to a program that has not been installed as part of the application.
- 61 T No Current Record
- Results from an attempt to WRITE an indexed file opened in SU mode following an UNLOCK, even if the record had been read before the unlock.
- 62 T Protection Violation
- This message results from one of the following:
- an attempted illegal operation
  - an attempt to access a locked file
  - the user was not privileged
- 63 NT No Such Operation
- The RMS module for the type of file organization in use was not linked with the user program.

64 T Record Not Found

Results from attempting to read a nonexistent record from an RMS relative file.

65 [reserved]

66 NT R6 Stack Overflow

Subroutine calls have been nested too deeply.

67 NT Operating System Error

A request to the P/OS system by the OTS has failed. Refer to the application documentation or contact the application vendor.

68-76 [reserved]

77 T Arguments out of Order

The order of the arguments in the PAK/UNPAK subroutine call do not match the record definition.

78 T Argument out of Record Limit

An argument passed in the PAK/UNPAK subroutine call is not part of the record being used.

79 T Argument Count

The call lacks the minimum number of arguments required by the PAK/UNPAK subroutine.

80 T Field not Packed

The UNPAK subroutine was asked to unpack a record that was not previously packed.

81-83 [reserved]

84 T Illegal Record Size

A file was opened in block mode (O:B, I:B, or U:B) and an attempt was made to read a record that was not a multiple of 512 bytes.

85 [reserved]

86 NT I/O Services Error

An unexpected error occurred as a result of a QIO operation in the interpreter. The numbers following the text are the two values found in IOSB. Please submit an SPR.

87 T Argument Missing

An attempt was made to operate on an optional argument that was not passed.

88-99 [reserved]

100 NT Unexpected RMS Error

An unexpected error occurred as a result of an RMS operation in the interpreter. The number following the text is the specific RMS error code. Please submit an SPR.

101-103 [reserved]

104 NT Value out of Range

A numeric value has been specified as an argument and it is outside the permissible range of values.

105-112 [reserved]

113 T Invalid ALLOC value in OPEN

The initial file allocation size in blocks specified in an OPEN statement was outside the allowable range of 1 to the maximum free blocks on the system.

114 T Invalid BKTSIZ value in OPEN

The bucket size specified for a relative file being created via an OPEN statement was outside the valid range of 1 to 32.

115 T Invalid BLKSIZ value in OPEN

The block size specified in an OPEN statement is outside the valid range.

116 T Invalid BUFSIZ value in OPEN

The I/O buffer size specified via an OPEN statement is outside the valid range of 1 to 63.

117 T Invalid RECSIZ value in OPEN

The record size specified in an OPEN statement was outside the valid range of 1 to 65535.

118-127 [reserved]

128 Stop

The program has executed a STOP statement. This message will be followed by traceback information without a request to press RESUME to continue.

129 Please write down the above message -- press RESUME to continue

This message appears as the last line following traceback. It indicates that the application has failed. Please forward the information to the application vendor.

140 NT Fatal initialization error #nnn

An error occurred during system initialization. The directive status word code is indicated by "nnn". A possible cause could be the interpreter was unable to start a task because checkpointing was disabled.



## DIBOL DEBUGGING TECHNIQUE (DDT) MESSAGES

This chapter describes the DDT utility messages which are generated during the execution of a DIBOL program with DDT.

#### 4.1 MESSAGE FORMAT

DDT generates messages that report the current status of program execution with DDT; and error messages that describe incorrect use of DDT.

The general format of the message is as follows.

text

where:

text is the explanatory text.

Run-time errors and error messages may be encountered during program execution with DDT. See Chapter 3 for the general format and context of these messages.

#### 4.2 MESSAGES

The DDT messages are listed in alphabetical order according to the message text. Below the text, additional information which elaborates the message may be presented.

##### Argument not passed to subroutine

You attempted to examine an argument in a DIBOL subroutine when the argument is optional and is not being used.

##### Literal?

You have attempted to modify a variable name which appears to be a literal.

##### No such variable

The variable entered has no match in the program.

## Subscript error

You have attempted to manipulate a variable using subscripts to access part of a field but you have not conformed to the rules for subscripts. See the DIBOL-83 Language Reference Manual.

## Too many breakpoints

There can be only one breakpoint in the main program and in each of the subroutines. A maximum of eight breakpoints is allowed at any one time.

## What?

This error can occur under one of the following conditions:

- The number was not specified in an iteration count.
- The specified routine name in which to set a breakpoint was not followed with a colon.
- A decimal value was expected but was not entered.



READER'S COMMENTS

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

Did you find errors in this manual? If so, specify by page.

---

---

---

---

---

---

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

---

---

---

---

---

---

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

---

---

---

---

---

---

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

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Non-programmer interested in computer concepts and capabilities

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

Organization \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_

or  
Country

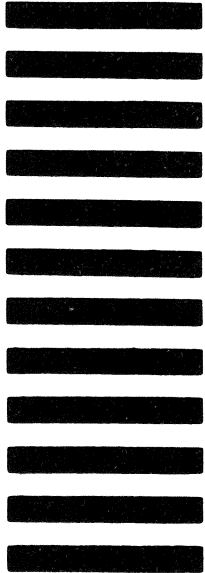
Please cut along this line.

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

DIGITAL EQUIPMENT CORPORATION  
Applied Commercial Engineering MK1-2/H32  
Continental Boulevard  
Merrimack N.H. 03054

ATTN: Documentation Supervisor

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line