BASIC-PLUS-2 Documentation Supplement

Order No. AA-P102A-TK

December 1982

This supplement presents an overview of Professional Tool Kit BASIC-PLUS-2 and describes the differences between BASIC-PLUS-2 on the Professional Developer's Tool Kit and BASIC-PLUS-2 on RSX systems. Tool Kit BASIC-PLUS-2 can be used with the Professional 300 Series Developer's Tool Kit to create application software for use on the Professional Operating System (P/OS).

OPERATING SYSTEM AND VERSION:	VAX/VMS V3.0 RSX-11M V4.0 RSX-11M-PLUS V2.0
SOFTWARE VERSION:	Professional Developer's Tool Kit V1.0 Professional Tool Kit BASIC-PLUS-2 V2.0

DIGITAL EQUIPMENT CORPORATION
Maynard, Massachusetts 01754

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.

The following are trademarks of Digital Equipment Corporation:

CTI BUS
DEC
DECmate
DECsystem-10
DECSYSTEM-20
DECUS
DECwriter
DIBOL
digital

MASSBUS
PDP
P/OS
PRO/BASIC
Professional
PRO/FMS
PRO/RMS
PROSE
Rainbow

RSTS
RSX
Tool Kit
UNIBUS
VAX
VMS
VT
Work Processor

CONTENTS

FIGURE

TABLES

BASIC-PLUS-2
DOCUMENTATION SUPPLEMENT

1.0 RELATED DOCUMENTATION

This manual supplements the manual *BASIC on RSX-11M/M-PLUS Systems* (Order No. AA-L338A-TC). Both manuals are included in the documentation set shipped to Tool Kit users who have ordered BASIC-PLUS-2. Also included in that set are the *BASIC Reference Manual* (Order No. AA-L334A-TK) and the *BASIC User's Guide* (Order No. AA-L335A-TK). See these three manuals for additional information on BASIC and BASIC-PLUS-2 on RSX systems. The *BASIC Pocket Reference Guide* (Order No. AV-L341A-TK), available separately, is useful for quick reference.

For information on the current release of Professional Tool Kit BASIC-PLUS-2, its installation and known bugs, see the *Professional Tool Kit BASIC-PLUS-2 Installation Guide and Release Notes* (Order No. AA-P104A-TK).

2.0 PROFESSIONAL TOOL KIT BASIC-PLUS-2

BASIC-PLUS-2 is a programming language that uses familiar words and mathematical notations. With Tool Kit BASIC-PLUS-2 you can create applications on an RSX-11M/M-PLUS or VAX/VMS host system that run on the Professional Operating System (P/OS). This manual describes the differences between BASIC-PLUS-2 on RSX systems and Tool Kit BASIC-PLUS-2.

The material here is intended for programmers experienced with BASIC-PLUS-2 on RSX systems.

3.0 DEVELOPMENT CYCLE

The development cycle for Tool Kit BASIC-PLUS-2 applications (Figure 1) consists of the following steps:

1. Write source program.
2. Compile source program, creating object module file.
   - □ Compile with /DEBUG to debug at run time.
   - □ Compile without /DEBUG when program is error-free.
3. Build command file and overlay descriptor language file. Edit command file and descriptor file as needed.
4. Task build to create executable task image file.
5. Write application installation file; move it and the task to the Professional.
6. Run task on the Professional. If desired, debug using REDIRECT command and terminal attached to the Professional workstation.
7. Run Application Diskette Builder when program is error-free, creating final diskette.

The following sections present specific details of application development in Tool Kit BASIC-PLUS-2. For a complete description of each stage in Tool Kit application development, you should refer to the *Tool Kit User's Guide.*

3.1 Writing the Source Program

The BASIC-PLUS-2 source program can include external subroutine calls to access P/OS facilities from your program. The external subroutine calls are described in the *Tool Kit User's Guide* and the *CORE Graphics Library Manual.* Facilities exist in the following areas:

1. P/OS user interface support: the Tool Kit BASIC-PLUS-2 application has access to P/OS menu, help, message, and other system services. The ability to display error messages, status, on-line Help, and menus on the screen is provided through the P/OS Service Routines Library.
2. Extended software functions: the Tool Kit BASIC-PLUS-2 application can invoke callable system services, such as PROSE (the P/OS text editor) and the Communications Facility.

P/OS facilities are accessed through the register 5 (R5) calling sequence with the BASIC-PLUS-2 CALL BY REF statement.

3.1.1 Differences and Unsupported Features—Table 1 summarizes features that will influence your source code. Some operate differently than on RSX systems, others are unsupported. Do not use the unsupported features listed here when you write an application using Tool Kit BASIC-PLUS-2.

For more information on editing the command and overlay descriptor files, see Sections 3.5 and 3.6 of this supplement. For a complete list of Tool Kit BASIC-PLUS-2 error messages, See Section 4.

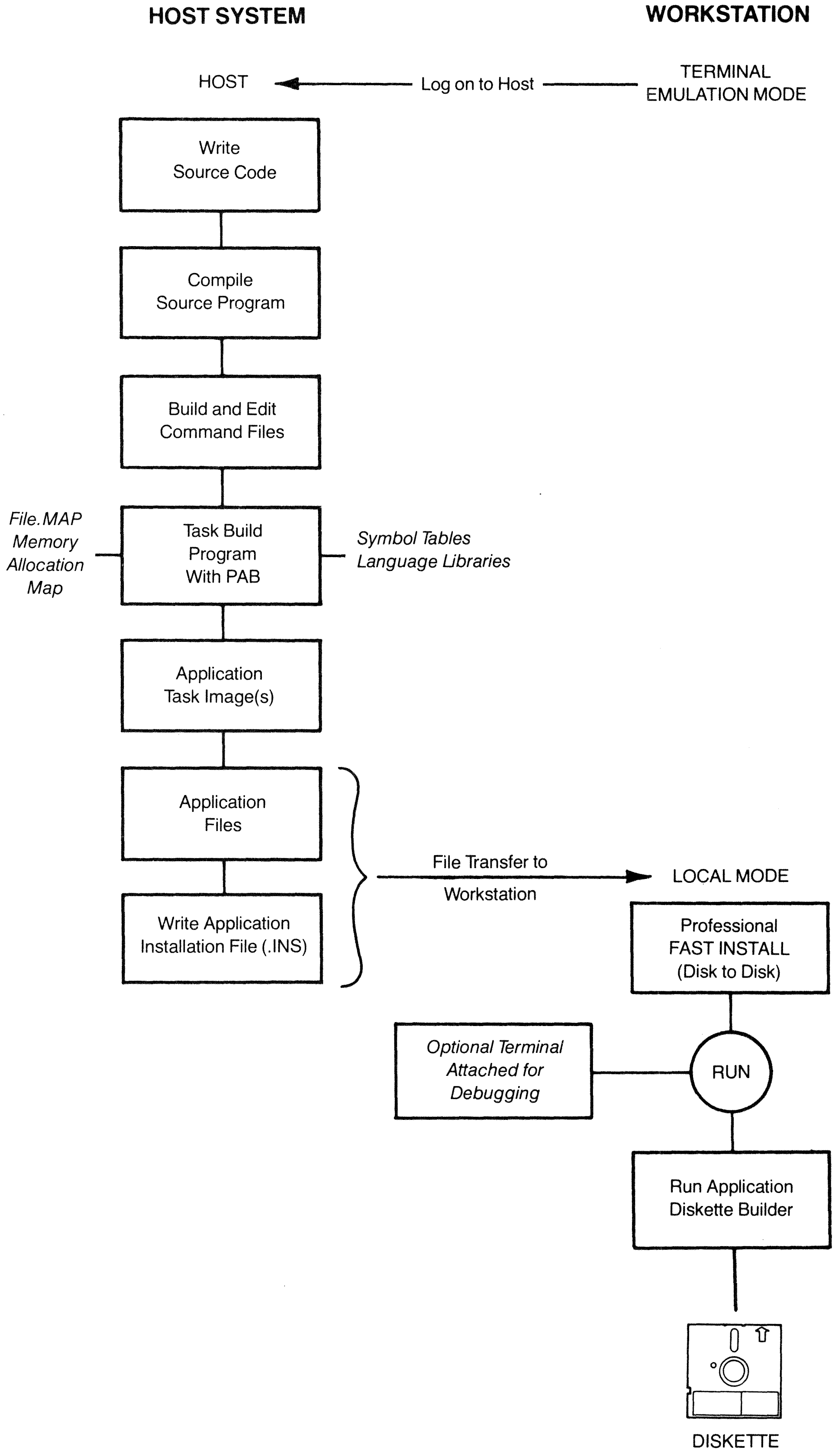
HOST SYSTEM
WORKSTATION
HOST
Log on to Host
TERMINAL EMULATION MODE
Write Source Code
Compile Source Program
Build and Edit Command Files
File.MAP Memory Allocation Map
Task Build Program With PAB
Symbol Tables Language Libraries
Application Task Image(s)
Application Files
Write Application Installation File (.INS)
File Transfer to Workstation
LOCAL MODE
Professional FAST INSTALL (Disk to Disk)
Optional Terminal Attached for Debugging
RUN
Run Application Diskette Builder
DISKETTE

Figure 1 Tool Kit BASIC-PLUS-2 Development Cycle

Table 1
Tool Kit BASIC-PLUS-2—Items of Interest

Feature	*Please Note*
Compiler Commands	
RUN	Not in the Tool Kit compiler.
LOAD	Not in the Tool Kit compiler.
BUILD	.CMD file must be edited. .ODL file may need to be edited.
BUILD/DUMP	Unsupported.
BUILD/NOSEQUENTIAL	Unsupported for programs that do not access RMS files.
SET DUMP	Unsupported.
SET NOSEQUENTIAL	Unsupported for programs that do not access RMS files.
Debugger Commands	
REDIRECT	Allows debugging during application execution from a terminal attached to the Professional. Does not accept terminal device number.
Error Messages	A subset of error messages for BASIC-PLUS-2 on RSX. You must press **RESUME** to continue after receiving error message.
Immediate Mode	Not in Tool Kit compiler.
Magnetic Tape Operations	Unsupported.
RMS Support	Must not be removed from task image.
Statements	
CHAIN	Accepts installed task name, same as name in application installation file (not file specification).
FSS$ Function	Unsupported for named directories.
EDIT$	Do not use any odd integer as argument to EDIT$ if the string to be operated on is an 8-bit character: high-order bit will be cleared.
Time Format	Twelve-hour AM/PM format unsupported.

3.2 Character Sets

Tool Kit BASIC-PLUS-2 supports the DEC Multinational Character Set in string literals, comments, and DATA statements when you compile the application, and as input or output when you run the application on the Professional. (See the *Terminal Subsystem Manual* for a description of the DEC Multinational Character Set.)

The compiler does not allow European alphabetics in variable names.

Do not use any odd integer as the argument to EDIT$ if the string to be operated on is an 8-bit character: the high-order bit will be cleared.

3.3 Compiling the Source Program

When you install Tool Kit BASIC-PLUS-2 on your host system, you select the Extended Instruction Set (EIS) compiler, the Floating Point Unit (FPU) compiler, or both, as part of the installation process. Your choice should reflect the environment in which your applications will run. Floating Point hardware is optional on the Professional. Your host development system does not require an FPU to run the Tool Kit FPU compiler. Build your programs with the FPU compiler only if your applications will run on Professionals which have the Floating Point option.

To invoke the EIS compiler on RSX-11M/11M-PLUS, type:

```
>RUN $PBE
```

To invoke the FPU compiler, type:

```
>RUN $PBF
```

To invoke the EIS compiler on VAX/VMS, use the command:

```
$ RUN SYS$SYSTEM:PBE
```

To invoke the FPU compiler, use the command:

```
$ RUN SYS$SYSTEM:PBF
```

Compile your source program using the BASIC-PLUS-2 COMPILE command. The compiler will check each line of the source program for errors, returning an appropriate message if an error is found. You can then correct the program as necessary and recompile it. Program compilation produces an object module file (file.OBJ) that can be task built on the host system.

BASIC-PLUS-2 has its own debugger. To use the debugger with your application, compile your source program using the /DEBUG qualifier. When you execute the application on the Professional, you can issue commands to the debugger to control and monitor application execution.

3.4 Generating the Command and Descriptor Files

After compiling the program, use the BASIC-PLUS-2 BUILD command to generate a command file (file.CMD) and overlay descriptor file (file.ODL) for your application. The Professional Application Builder (PAB) uses these files to define how libraries are referenced, and to specify special-purpose buffers, logical unit numbers (LUNs), and event flags (EFNs).

You must edit the command file before you task build. You may also need to edit the overlay descriptor file. The following sections describe the information that must be contained in the command and overlay descriptor files.

3.4.1 Rebuilding an Application—If you are rebuilding an application to use the FPU OTS, having first built it for the EIS OTS, you must change all occurences of PBE to PBF in the command and overlay descriptor files generated by the EIS compiler. You must also add the /FP switch to the task image output file specification in the command file.

3.5 Editing the Command File

You must edit the command file generated by the Tool Kit BASIC-PLUS-2 compiler.

A command file output by the Tool Kit BASIC-PLUS-2 EIS compiler for an application named "TEST1" would look like this:

```
SY:TEST1/CP=SY:TEST1/MP
UNITS = 15
ASG = TI:13:15
ASG = SY:5:6:7:8:9:10:11:12
EXTTSK= 952
CLSTR=PBESML,RMSRES:RO
//
```

The command file automatically references the language OTS, clustered with RMS. However, you must edit the command file to include the P/OS Service Routines library, or you will get undefined symbols when you task build. You should also edit the command file if you want to reference any libraries other than the language OTS and RMS, such as FMS-11 or Graphics.

You must make the following edits to your .CMD file:

1. **Task Name** —The command file must assign a task name to the task image. The task name is assigned with a command line of the form:

   ```
   TASK = task-name
   ```

   The value task-name is a 1– to 6–character name identifying the task. The task-name is the installed name of the task.

2. **Clusterable Libraries** —You must edit the command file to reference the POSRES library. List all referenced clusterable libraries on the line beginning "CLSTR". Note that the default library must be the Tool Kit BASIC-PLUS-2 OTS: you must list this library first. After it, list libraries in the order of greatest use. For example:

   ```
   CLSTR = PBESML,RMSRES,POSRES:RO
   ```

   This line identifies the BASIC-PLUS-2 EIS OTS (PBESML) as the default library, and RMS and POSRES as referenced libraries, with RMS being referenced more often than POSRES. All libraries are read-only.

3. **Additional Logical Units** —Since P/OS will require additional logical units, change the UNITS number in the second line to 18:

   ```
   UNITS = 18
   ```

4. **Extend Section Commands.**—If your application uses P/OS user interface services in the POSRES library, allocate buffer space with extend section (EXTSCT) commands in the command file. See the section on task building in Chapter 2 of the *Tool Kit User's Guide* for instructions on calculating buffer sizes for P/OS user interface services.

5. **Logical Unit Numbers.**—P/OS user interface services require logical unit numbers (LUNs) to perform file I/O to menu, help, and message definition files. LUN assignments for P/OS I/O can be made with global definition commands (GBLDEF) in the command file. In assigning LUNs, remember to account for LUNs needed by the BASIC-PLUS-2 OTS. You must include the LUN for BASIC-PLUS-2 OTS messages:

   ```
   GBLDEF  = MS$LUN:16       ; message frame file
   ```

   If one type of P/OS service is never used, such as help services, specify 0 in the global definition line for message files. Do not omit any global definition lines.

6. **Event Flags.**—P/OS user interface routines also require an event flag (EFN) to perform terminal I/O. The event flag assignment must not conflict with event flag assignments made for other purposes by the application. Language OTS routines use DIGITAL reserved event flags (flags 25-32). The event flag assigned with the TT$EFN line should not be a DIGITAL reserved flag.

   Note: The assign (ASG) commands and UNITS commands list decimal numbers. Extend section (EXTSCT) and global definition (GBLDEF) lines specify octal numbers.

The following list shows the required extend section and global definition lines that are added just below the "CLSTR" line in the command file. The values for the extend section lines (EXTSCT) are calculated for the largest possible frame of each type. That is, if every field on a menu, help, and message frame were filled in with the maximum amount of data, the respective buffers must be allocated the sizes shown.

```
                                ; DEFINE BUFFER SIZES
EXTSCT  = MN$BUF:4540           ; static single-choice menu
EXTSCT  = DM$BUF:4540           ; dynamic single-choice menu
EXTSCT  = MM$BUF:1000           ; multi-choice menu
EXTSCT  = HL$BUF:3410           ; help text or help menu
EXTSCT  = FL$BUF:4310           ; file selection/specification
                                ; for OLDFIL and NEWFIL routine

                                ; DEFINE LUN ASSIGNMENTS
GBLDEF  = MN$LUN:20             ; menu frame file
GBLDEF  = HL$LUN:21             ; help frame file
GBLDEF  = MS$LUN:16             ; message frame file
GBLDEF  = TT$LUN:15             ; terminal I/O
GBLDEF  = TT$EFN:1              ; terminal I/O event flag
GBLDEF  = WC$LUN:22             ; directory searches for OLDFIL and NEWFIL
                                ; routine or callable print services
```

The edited command file for TEST1 would look something like this (new or edited lines are shaded):

```
SY:TEST1/CP=SY:TEST1/MP
TASK = task-name
UNITS = 18
ASG = TI:13:15
ASG = SY:5:6:7:8:9:10:11:12
EXTTSK= 952
CLSTR = PBESML,RMSRES,POSRES:RO
                          ; DEFINE BUFFER SIZES
EXTSCT   = MN$BUF:4540    ; static single choice menu
EXTSCT   = DM$BUF:4540    ; dynamic single choice menu
EXTSCT   = MM$BUF:1000    ; multi-screen menu
EXTSCT   = HL$BUF:3410    ; help text/menu
EXTSCT   = FL$BUF:4310    ; file selection/specification
                          ; for ODFIL and NEWFIL routine
                          ; DEFINE LUN ASSIGNMENTS
GBLDEF   = MN$LUN:20      ; menu frame file
GBLDEF   = HL$LUN:21      ; help frame file
GBLDEF   = MS$LUN:16      ; message frame file
GBLDEF   = TT$LUN:15      ; terminal I/O
GBLDEF   = TT$EFN:1       ; terminal I/O event flag
GBLDEF   = WC$LUN:22      ; directory searches for OLDFIL and NEWFIL
                          ; routine or callable print services
 //
```

3.6 Editing the Overlay Descriptor File

The Tool Kit BASIC-PLUS-2 EIS compiler would produce this overlay descriptor file for the Tool Kit application TEST1:

```
        .ROOT   BASIC2-RMSROT-USER,RMSALL
USER:   .FCTR   SY:TEST1-LIBR
LIBR:   .FCTR   LB:[1,5]PBEOTS/LB
@LB:[1,5]PBEIC1
@LB:[1,5]RMSRLX
        .END
```

If you want to reference any Tool Kit object libraries facilities, such as FMS-11, you should edit the overlay descriptor file. See *BASIC on RSX-11M/M-PLUS Systems* for information on editing BASIC-PLUS-2 .ODL files.

3.7 Building the P/OS Task Images

Use the Professional Application Builder (PAB) to create an application task image (file.TSK). See the *Tool Kit User's Guide* for complete information on Professional Application Builder commands.

3.8 Debugging During Execution

Use the BASIC-PLUS-2 debugger to debug the program. (See your BASIC documentation for more information on the BASIC-PLUS-2 debugger.) In addition, when you execute a task on the Professional, you can also connect a terminal to the printer port on the Professional for debugging. (See the *Tool Kit User's Guide* for details.) Use the Tool Kit BASIC-PLUS-2 REDIRECT command to display debugger I/O and issue debugger commands on that terminal. Input is accepted from and output goes to the printer port. Note that the REDIRECT command does not accept a terminal device number. All other I/O, including graphics or forms, will remain unaffected.

4.0 TOOL KIT BASIC-PLUS-2 RUN-TIME ERROR MESSAGES

Tool Kit BASIC-PLUS-2 run-time error messages are a subset of BASIC-PLUS-2 run-time error messages on RSX systems. The Tool Kit BASIC-PLUS-2 run-time error messages are preceded by a number to help you report the error. If you use ERT$ to reference an error not in the Tool Kit BASIC-PLUS-2 subset, ERT$ will return the following message:

```
Can't access LB:[1,2]BASIC2.ERR or can't find frame <id>
```

Frame <id> will identify the error number of the ERT$ you tried to reference.

If the debugger is not present, the following notice will appear on the screen after the error message is displayed:

```
Please write down the above message -- press RESUME to continu
```

You must press the **RESUME** key to clear the screen and continue.

Table 2
Tool Kit BASIC-PLUS-2 Error Messages

Number	*Text*
1	?Bad directory for device
2	?Illegal file name
3	?Account or device in use
4	?No room for user on device
5	?Can't find file or account
6	?Not a valid device
7	?I/O channel already open
8	?Device not available
9	?I/O channel not open
10	?Protection violation

Table 2 *Continued*
Tool Kit BASIC-PLUS-2 Error Messages

Number	*Text*
11	?End of file on device
12	?Fatal system I/O failure
13	?User data error on device
14	?Device hung or write locked
15	?Keyboard wait exhausted
19	?Disk block is interlocked
28	?Programmable ^C trap
31	?Illegal byte count for I/O
34	?Reserved instruction trap
35	?Memory management violation
43	?Virtual array not on disk
44	?Matrix or array too big
45	?Virtual array not yet open
46	?Illegal I/O Channel
47	?Line too long
48	%Floating point error
50	%Data format error
51	%Integer error
52	?Illegal number
53	%Illegal argument in LOG
54	%Imaginary square roots
55	?Subscript out of range
56	?Can't invert matrix
57	?Out of data
58	?ON statement out of range
59	?Not enough data in record
60	?Integer overflow, FOR loop
61	%Division by 0
63	?FIELD overflows buffer
64	?Not a random access device
72	?RETURN without GOSUB
73	?FNEND without function call
88	?Arguments don't match
89	?Too many arguments
97	?Too few arguments
104	?RESUME and no error
105	?Redimensioned array

Table 2 *Continued*
Tool Kit BASIC-PLUS-2 Error Messages

Number	*Text*
116	?PRINT-USING format error
126	?Maximum memory exceeded
127	%SCALE factor interlock
130	?Key not changeable
131	?No current record
132	?Record has been deleted
133	?Illegal usage for device
134	?Duplicate key detected
135	?Illegal usage
136	?Illegal or illogical access
137	?Illegal key attributes
138	?File is locked
139	?Invalid file options
140	?Index not initialized
141	?Illegal operation
142	?Illegal record on file
143	?Bad record identifier
144	?Invalid key of reference
145	?Key size too large
147	?RECORD number exceeds maximum
148	?Bad RECORDSIZE value on OPEN
149	?Not at end of file
150	?No primary key specified
151	?Key field beyond end of record
152	?Illogical record accessing
153	?Record already exists
154	?Record/bucket locked
155	?Record not found
156	?Size of record invalid
157	?Record on file too big
158	?Primary key out of sequence
159	?Key larger than record
160	?File attributes not matched
161	?Move overflows buffer
162	?Cannot open file
164	?Terminal format file required
166	?Negative fill or string length

Table 2 *Continued*
Tool Kit BASIC-PLUS-2 Error Messages

Number	*Text*
167	?Illegal record format
168	?Illegal ALLOW clause
170	?Index not fully optimized
171	?RRV not fully updated
173	?Invalid RFA field
180	?No support for op in task
183	?REMAP overflows buffer
184	%Unaligned REMAP variable
185	?RECORDSIZE overflows MAP buffer
186	?Improper error handling
246	?Error trap needs RESUME
247	?Illegal RESUME to subroutine
248	?Illegal return from subroutine
250	?Not implemented
251	?Recursive subroutine call
252	?FILE ACP failure
253	?Directive error

BASIC-PLUS-2
Documentation Supplement
Order No. AA-P102A-TK

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 ____________
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

Professional 300 Series Publications
DIGITAL EQUIPMENT CORPORATION
146 MAIN STREET
MAYNARD, MASSACHUSETTS 01754

--- Do Not Tear - Fold Here --

Cut Along Dotted Line