

March 1978

This document describes the differences between BASIC/RT-11 V01B and BASIC-11/RT-11 V2. It also describes how to install BASIC-11 on an RT-11 system.

BASIC-11/RT-11 Installation Guide

Order No. DEC-11-LIBTA-A-D, DN1

SUPERSESSION/UPDATE INFORMATION:	This document supersedes the <i>BASIC/RT-11 V01B Release Notes</i> , Order No. DEC-11-LBRNA-B-D, published July, 1976. This document includes Update Notice No. 1.
OPERATING SYSTEM AND VERSION:	RT-11 V03
SOFTWARE VERSION:	BASIC-11/RT-11 V2

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754.

digital equipment corporation • maynard, massachusetts

First Printing, September 1977
Revised: March 1978

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.

Copyright © 1977, 1978 by Digital Equipment Corporation

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.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DEctape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10

CONTENTS

	Page
1.0 INTRODUCTION	1
2.0 CHANGES FROM BASIC/RT-11 V01B	1
3.0 RESTRICTIONS IN BASIC-11/RT-11 V2	9
3.1 Floating Point Operations	9
3.2 READ Statement	10
3.3 REM Statement	10
3.4 Subscripts	10
3.5 GOSUB Statement	10
3.6 COMPILE Command	10
3.7 File Types	10
3.8 Device Handlers	11
3.9 OPEN Statement	11
3.10 Reading a String	11
4.0 INSTALLATION PROCEDURE	11
4.1 Linking BASIC	12
4.2 Using the Dialogue Program	12
4.2.1 Overlay Types	14
4.2.2 Arithmetic Hardware Support	15
4.2.3 Foreground Execution	15
4.2.4 Assembly Language Routines	15
4.2.5 Optional Features	16
4.3 Sample Dialogue	16
4.4 Installation Instructions	17
5.0 THE CONVERTING PROGRAM	20
5.1 Starting the Converting Program	21
5.2 Interacting with the Converting Program	22
5.3 Converting a Sample Program	23
5.4 Compiling the Converting Program	26

FIGURES

FIGURE 1	The Dialogue Text	13
2	Sample Dialogue for Foreground	16
3	Sample Dialogue for Background	19
4	Linking BASIC	20
5	V01B BASIC Program	24
6	Converting Program Dialogue	25
7	V2 BASIC Program	26
8	Compiling the Converting Program	27

TABLES

TABLE 1	New Features of BASIC-11/RT-11 V2	2
2	Changes in Statements from BASIC/RT-11 V01B to BASIC-11/RT-11 V2	3
3	Changes in Functions from BASIC/RT-11 V01B to BASIC-11/RT-11 V2	5

TABLES (Cont.)

TABLE 4	Changes in Commands from BASIC/RT-11 V01B to BASIC-11/RT-11 V2	6
5	Changes in Miscellaneous Features from BASIC/RT-11 V01B to BASIC-11/RT-11 V2	8

BASIC-11/RT-11 INSTALLATION GUIDE

1.0 INTRODUCTION

This document describes the differences between BASIC/RT-11 V01B and BASIC-11/RT-11 V2. It also lists some restrictions you should be aware of when using BASIC-11/RT-11 V2. This document describes the procedure you should follow to install BASIC-11 on your RT-11 V03 system. Lastly, it shows how to use the program SUCVT to convert BASIC/RT-11 V01B programs to a format suitable for BASIC-11/RT-11 V2.

The following documentation conventions are used throughout this manual:

1. Square brackets [] enclose optional items.
2. Braces { } enclose items from which you must choose only one.
3. Upper case letters represent keywords that you must type as shown.
4. Lower case letters represent variables for which you supply values.
5. Ellipses ... indicate that the immediately preceding item can be repeated.
6. User input is underlined where necessary to distinguish it from computer printout.
7. The abbreviation "expr" represents expression.
8. The symbol RET represents a carriage return.

2.0 CHANGES FROM BASIC/RT-11 V01B

This section summarizes the differences between BASIC/RT-11 V01B and BASIC-11/RT-11 V2. You can find more information on BASIC-11/RT-11 V2 in the following two documents:

BASIC-11 Language Reference Manual
Order No. DEC-11-LIBBB-A-D

BASIC-11/RT-11 User's Guide
Order No. DEC-11-LIBUA-A-D

For more information on the RT-11 V03 operating system, consult the following guide:

RT-11 Documentation Directory
Order No. DEC-11-ORDDB-A-D

Table 1 lists the new features of BASIC-11/RT-11 V2.

BASIC-11/RT-11 INSTALLATION GUIDE

Table 1
New Features of BASIC-11/RT-11 V2

Category	Name
Integer Data Type	-
Double Precision Data Type	-
Statements	COMMON DIM# KILL LINPUT NAME ON GOTO ON GOSUB ON THEN PRINT USING RESET
Functions	ABORT CTRLC LOG10 PI RCTRLC RCTRLO SYS TTYSET
Commands	APPEND BYE COMPILE DEL LENGTH RESEQ SUB UNSAVE CTRL/Q CTRL/S

Some statements, functions, and commands are not entirely new in BASIC-11/RT-11 V2, but have syntactical or functional differences from their counterparts in BASIC/RT-11 V01B. Tables 2, 3, and 4 list both the old and new statements, functions, and commands. The tables also provide a brief explanation of the changes.

BASIC-11/RT-11 INSTALLATION GUIDE

Table 2
Changes in Statements from
BASIC/RT-11 V01B to BASIC-11/RT-11 V2

Version	Statement
V01B	CALL "routine name" [(argument list)]
V2	CALL ["]routine name["] [(argument list)] In V2, the quotation marks delimiting the routine names are optional.
V01B	CHAIN "string" LINE number
V2	CHAIN string [LINE expression] In V2, the line number specification can be an expression.
V01B	CLOSE $\left[\begin{array}{l} \{VF_n\} \\ \{#n\} \end{array} \right]$
V2	CLOSE [[#] expr1, [#] expr2, [#] expr3, ...] In V2, the pound sign (#) is optional. You can specify more than one file, and the file specification can be an expression. The VF _n syntax is no longer permitted.
V01B	DATA list
V2	DATA list In V2, unquoted strings are permitted. The DATA statement must be the last statement on a multi-statement line.
V01B	DEF FNletter (var1 [,var2, ...,var5])=expression
V2	DEF FNletter $\left[\begin{array}{l} \{ \$ \} \\ \{ \% \} \end{array} \right]$ (var1 [,var2, ...,var5])=expression In V2, if the function returns a string, the function name (FNletter) must end with a dollar sign (\$). If the function returns an integer, the function name must end with a percent sign (%).
V01B	IF relational expression $\left\{ \begin{array}{l} \text{THEN} \\ \text{GO TO} \end{array} \right\}$ line number IF END #n $\left\{ \begin{array}{l} \text{THEN} \\ \text{GO TO} \end{array} \right\}$ line number

(continued on next page)

BASIC-11/RT-11 INSTALLATION GUIDE

Table 2 (Cont.)
Changes in Statements from
BASIC/RT-11 V01B to BASIC-11/RT-11 V2

Version	Statement
V2	<p>IF relational expression { THEN statement THEN line number GO TO line number }</p> <p>IF END #expr { THEN statement THEN line number GO TO line number }</p> <p>In V2, the THEN keyword can be followed by a statement as well as by a line number.</p>
V01B	<p>INPUT variable1[,variable2,...]</p>
V2	<p>INPUT [#expr,]variable1[,variable2,...]</p> <p>In V2, you can type more than one string on a line.</p>
V01B	<p>INPUT #expr: variable1[,variable2,...]</p>
V2	<p>INPUT [#expr,]variable1[,variable2,...]</p> <p>DIGITAL recommends that you follow the expression by a comma (,) rather than by a colon (:), even though the colon is still permitted.</p>
V01B	<p>[[LET] variable=expression</p> <p>[[LET] VF<i>n</i>(<i>i</i>)=expression</p>
V2	<p>[[LET] variable=expression</p> <p>In V2, the VF format is not permitted.</p>
V01B	<p>OPEN file [FOR { INPUT }] [(b)] AS FILE #digit [DOUBLE BUF]</p> <p>OPEN file FOR [{ INPUT }] [(b)] AS FILE VFdigit (dimension)=string length</p>
V2	<p>OPEN string [{ FOR INPUT }] AS FILE [#] expr [DOUBLE BUF]</p> <p>[,RECORDSIZE expr] [,MODE expr] [,FILESIZE expr]</p> <p>In V2, the file specification can be any string expression. The pound sign (#) is optional. Specify a file size by using the FILESIZE keyword. The VF format is not permitted.</p>

(continued on next page)

BASIC-11/RT-11 INSTALLATION GUIDE

Table 2 (Cont.)
Changes in Statements from
BASIC/RT-11 V01B to BASIC-11/RT-11 V2

Version	Statement
V01B	OVERLAY "file descriptor"
V2	OVERLAY string [LINE expression] In V2, the line number specification can be an expression.
V01B	PRINT #expr: list
V2	PRINT [#expr,] [list] DIGITAL recommends that you follow the expression by a comma (,) rather than by a colon (:), even though the colon is still permitted.
V01B	RANDOMIZE
V2	RANDOMIZE In V2, the random number generator starts at a new number each time the statement is executed.
V01B	REM comment
V2	REM comment In V2, you can terminate a comment by a backslash (\).

Table 3
Changes in Functions from
BASIC/RT-11 V01B to BASIC-11/RT-11 V2

Version	Function
V01B	ASC (string)
V2	ASC (string) In V2, this function returns an 8-bit (rather than 7-bit) value.
V01B	BIN [()string ()]

(continued on next page)

BASIC-11/RT-11 INSTALLATION GUIDE

Table 3 (Cont.)
 Changes in Functions from
 BASIC/RT-11 V01B to BASIC-11/RT-11 V2

Version	Function
V2	BIN(string) In V2, the parentheses delimiting the argument are required.
V01B	CHR\$(expr)
V2	CHR\$(expr) In V2, this function accepts an 8-bit (rather than 7-bit) value.
V01B	OCT [()string ()]
V2	OCT(string) In V2, the parentheses delimiting the argument are required.
V01B	STR\$(expr)
V2	STR\$(expr) In V2, this function can return more than six digits (with double precision arithmetic).

Table 4
 Changes in Commands from
 BASIC/RT-11 V01B to BASIC-11/RT-11 V2

Version	Command
V01B	LIST[NH] LIST[NH]line number LIST[NH]-line number LIST[NH]line number - [END] LIST[NH]line number - line number
V2	LIST[NH] [line specification1,line specification2,...] In V2, you can list more than one line range specification. The END specification is no longer allowed. The other V01B forms of the command are still valid in V2.

(continued on next page)

BASIC-11/RT-11 INSTALLATION GUIDE

Table 4 (Cont.)
Changes in Commands from
BASIC/RT-11 V01B to BASIC-11/RT-11 V2

Version	Command
V01	NEW "program name" OLD "file specification" RENAME "program name" REPLACE "file specification" SAVE "file specification"
V2	NEW [program name] OLD [file specification] RENAME program name REPLACE file specification SAVE [file specification]
	In V2, these commands must not contain quotation marks that delimit the file specifications.
V01B	RUN RUNNH
V2	RUN [NH] RUN [NH] file specification
	In V2, you can include a file specification in the RUN and RUNNH commands.
V01B	CTRL/C
V2	CTRL/C In V2, CTRL/C returns control to BASIC (in V01B, it returned control to the RT-11 monitor). CTRL/C also prints the line number of the line that was executing when it interrupted the program.

Table 5 lists changes in miscellaneous features from BASIC/RT-11 V01B to BASIC-11/RT-11 V2.

BASIC-11/RT-11 INSTALLATION GUIDE

Table 5
 Changes in Miscellaneous Features from
 BASIC/RT-11 V01B to BASIC-11/RT-11 V2

Feature	Change
Maximum line number	In V2, the maximum line number is 32767 (in V01B, it was 65532).
String concatenation operator	In V2, the string concatenation operator can be either the ampersand (&) or the plus sign (+) (in V01B it was the ampersand).
Virtual arrays	In V2, virtual arrays are like arrays in memory (except that they are not zeroed at run time); they do not have special variable names. Virtual arrays can have one or two subscripts (V01B allowed just one subscript). The string length can have any value between 1 and 255; it does not have to be a power of 2. The default string length is 16 (in V01B, it was 32).
Terminal width	In V2, the terminal width can be set from 1 to 255 characters. See the BASIC-11/RT-11 User's Guide for a description of the TTYSET function.
Closing files	In V2, files are closed by the CLOSE or END statement (or by execution of the last line of the program) but are left open by the STOP statement. Any open output files are deleted by the RUN, SCR, OLD, and NEW commands. (In V01B, STOP, RUN, SCR, OLD, and NEW closed files.)
Deleting a character	In V2, the DELETE (or RUBOUT) key deletes a character; the backarrow (or underscore) key does not. (Both DELETE and backarrow were valid in V01B.)
Deleting a line	In V2, the CTRL/U command deletes a line; the ESCAPE (or ALTMODE) key does not. (Both CTRL/U and ESCAPE were valid in V01B.)
Lower case	In V2, lower case keywords and variable names are always converted to upper case. Lower case string constants and remarks are permitted.

(continued on next page)

BASIC-11/RT-11 INSTALLATION GUIDE

Table 5 (Cont.)
Changes in Miscellaneous Features from
BASIC/RT-11 V01B to BASIC-11/RT-11 V2

Feature	Change
Arithmetic hardware support	<p>BASIC-11/RT-11 V2 supports the following hardware (V01B also supported this hardware):</p> <p>EAE Kell-A and Kell-B Extended Arithmetic Element</p> <p>EIS Kell-E Extended Instruction Set</p> <p>FPU FP11-A, FP11-B, and FP11-C Floating Point Unit</p> <p>Support of the following hardware is new in V2:</p> <p>FIS Kell-F Floating Point Instruction Set KEV-11 Extended Arithmetic Element</p>
CALL interface	<p>In V2, the CALL interface is similar to the FORTRAN IV CALL interface. Note that arrays are stored with the second subscript varying faster; they are stored independently of the scalar with the same name. Many internal table formats and offsets have changed. See the <u>BASIC-11/RT-11 User's Guide</u> for a complete description of the CALL interface.</p>

3.0 RESTRICTIONS IN BASIC-11/RT-11 V2

BASIC-11/RT-11 users should always keep abreast of BASIC-related notices published by DIGITAL. Changes published in the Software Performance Summary need be made only once and should be made immediately. Changes published in the Digital Software News should be made as soon as possible to systems in use. The following sections summarize the restrictions in BASIC-11/RT-11 V2 of which you should be aware.

3.1 Floating Point Operations

Since whole floating point numbers are not stored as integers, execution of floating point operations can be slower than in V01B, in certain circumstances. To avoid this problem, convert the data from floating point to integer by specifying the percent sign (%) wherever possible.

BASIC-11/RT-11 INSTALLATION GUIDE

3.2 READ Statement

Since a list created by a data statement is stored in ASCII format (rather than in a binary internal format), the READ statement requires additional time to translate data items.

3.3 REM Statement

A REM statement terminated by a backslash (\) gets longer by one space each time a program containing it is saved by the SAVE command. If a program containing such a line is called and saved (with the OLD and SAVE commands) several times, the line can become too long for BASIC to handle properly. Use the SUB command to delete the extra spaces before the line gets too long.

3.4 Subscripts

BASIC calculates subscripts by performing an INT function rather than by using true truncation. This method of calculating subscripts causes no problems with positive subscripts. It also causes no difficulty with subscripts of -1 or less, since BASIC does not support negative subscripts. However, BASIC does give an error (?SUBSCRIPT OUT OF BOUNDS) for A(-.5).

3.5 GOSUB Statement

An immediate mode GOSUB to a routine that has an INPUT or LINPUT statement can crash BASIC, under certain circumstances. This can occur because BASIC uses the same area of memory to process both immediate mode statements and terminal input. When BASIC returns control to the line with the immediate mode GOSUB statement, that line has been overwritten by the terminal input (from the INPUT or LINPUT statement). To avoid this problem, use an immediate mode GOTO statement in place of the GOSUB statement. This causes BASIC to execute the subroutine, print a warning, and then print the READY message.

3.6 COMPILE Command

The COMPILE command generates a file with a .BAC file type under BASIC with single precision arithmetic. Similarly, it generates a file with a .BAX file type under BASIC with double precision arithmetic. If you use an OLD, RUN, or CHAIN command to call a .BAC file under double precision BASIC or a .BAX file under single precision BASIC, BASIC can crash. You must explicitly specify the wrong file type to create this problem. BASIC executes properly when you rely on default file types.

3.7 File Types

When you call a file (with an OLD, RUN, or CHAIN command), BASIC looks first for a compiled image, and then for a .BAS image. If, for example, you specify OLD X.BAC and X.BAC does not exist, BASIC-11/RT-11 V2 continues to search for the .BAS file.

BASIC-11/RT-11 INSTALLATION GUIDE

3.8 Device Handlers

If a BASIC program that contains an array resides on a device whose handler is not resident, references you make to that device (with the OPEN statement, or the CHAIN command, for example) can cause the system to crash. Be sure to load the appropriate device handler to avoid this problem.

3.9 OPEN Statement

When BASIC opens an integer virtual array, and the subscript you specify in the DIM# statement is 1 or 2 less than a multiple of 256, BASIC creates a file that is one block too long.

3.10 Reading a String

If you place a null string as the last item in a DATA statement or an input line, indicate it by two adjacent quotation marks. The following example shows two null strings: one is indicated by two adjacent commas, and the other by two adjacent quotation marks.

```
10 DATA 5,ABC,6,,DEF,""
```

4.0 INSTALLATION PROCEDURE

The BASIC-11/RT-11 V2 software kit contains three running versions of BASIC:

1. BASIC.SAV
2. BAS8K.SAV
3. BASICD.SAV

The BASIC.SAV version includes all the optional features except the CALL statement. It is overlaid (it has overlay type 3) and does not include support for arithmetic hardware. Its base address is 1000.

The BAS8K.SAV version can execute in 8K words of memory. It should only be used for 8K RT-11 systems. It does not include the following optional features:

1. PRINT USING statement
2. SUB command
3. CALL statement
4. RESEQ command
5. Long error messages

BAS8K is overlaid (it has overlay type 5) and does not include support for arithmetic hardware. In addition, BAS8K has a base address of 700.

BASICD.SAV has the same features and overlay type as BASIC.SAV. It supports double precision floating point arithmetic and has a base address of 1200.

BASIC-11/RT-11 INSTALLATION GUIDE

4.1 Linking BASIC

If BASIC.SAV, BASICD.SAV, or BAS8K.SAV is suitable for your application, you do not need to relink BASIC. However, you can link BASIC to make any of these changes:

1. Change the overlay type
2. Add support for arithmetic hardware
3. Add CALL support and assembly language routines
4. Add or remove any optional features

To link BASIC, you must run the program SUCNFG.BAS. This program asks you questions about BASIC's features and optional support. Then it creates an indirect command file that you must execute. This indirect file links BASIC according to the instructions you supplied to SUCNFG. The indirect file creates an executable version of BASIC (with any name you specify) that is tailored to your requirements.

Read Sections 4.2 and 4.3 to learn how to use the dialogue program, SUCNFG. Then read Section 4.4, which describes how to transfer the necessary files for linking BASIC from the software kit to your system.

4.2 Using the Dialogue Program

To invoke the dialogue program, you must first run BASIC. Then run SUCNFG:

```
.R BASIC (RET)
BASIC-11/RT-11 V02-02
OPTIONAL FUNCTIONS (ALL, NONE, OR INDIVIDUAL)? (RET)

READY
RUN SUCNFG (RET)
BASIC-11/RT-11 V02 Configuration Program

Type "?<ret>" after any prompt for HELP.

.SAV and .COM file name (default is BASIC):
```

When the dialogue prompts you for a .SAV and .COM file name, enter the name of the BASIC version you want to create. If you type only the RETURN key, the new version of BASIC is called BASIC.SAV. Note that if the new version is called BASIC.SAV, it replaces the version you are currently running. The listing shown in Figure 1 contains all the questions in the dialogue program, along with brief explanations.

BASIC-11/RT-11 INSTALLATION GUIDE

.SAV and .COM file name (default is BASIC):

Enter the name in 6 or less characters for the name to be given to the .SAV image of BASIC and the indirect file to be used to build BASIC.

Do you want a map? (Y or N; default is N):

Type Y for yes or N for no. If the answer is Y, the next question will ask for a name.

.MAP name (default is same as .SAV name):

Enter <ret> or any legal file name.

Background or Foreground ? (B or F; default is B):

B creates a .SAV image to be run in the background. F creates a .REL image to be run in the foreground.

Do you want PRINT USING? (Y or N; default is Y):

Y includes the PRINT USING option, N does not.

Do you want CALL support? (Y or N; default is N):

Y includes CALL support, N does not. If CALL support is included the next question will handle the modules to be used with CALL in this BASIC.

Enter module names (one per line):

On each line enter the name of a module to be included. Include your modified BSCLI, your subroutines, and BSCLLB, if it is needed. When all the modules have been entered, type <ret> alone. For more information, please see the documentation on CALL.

Single or Double precision math? (S or D; default is S):

S creates a BASIC that supports the 2-word floating point representation. D creates a BASIC that supports the 4-word floating point representation.

Arithmetic hardware (NONE, EAE, EIS, FIS, or FPU; default is NONE):

Arithmetic Hardware support is defined as follows:

NONE - no arithmetic hardware is supported
EAE - KE11A Extended Arithmetic Element is supported
EIS - KE11E Extended Instruction Set is supported
FIS - KE11F or KEV11F Floating Instruction Set is supported
FPU - FPP11A, B, or C Floating Point Unit is supported

Do you want the transcendental functions? (Y or N; default is Y):

Y includes the transcendental functions, N does not. The transcendental functions are SQR, SIN, COS, ATN, LOG, LOG10, and EXP.

Figure 1 The Dialogue Text

BASIC-11/RT-11 INSTALLATION GUIDE

Do you want the SUB command? (Y or N, default is Y):

Y includes the SUB command, N does not. SUB allows editing without retyping the entire BASIC statement.

Do you want the RESEQ command? (Y or N, default is Y):

Y includes the RESEQ command, N does not. RESEQ allows the resequencing of line numbers in a BASIC program.

Do you want long error messages? (Y or N; default is Y):

Y includes long error messages, N does not. Long messages are several word explanations of an error condition. Short error messages are three character abbreviations of the long messages.

Overlay structure type (1, 2, 3, 4, or 5; default is 1):

The overlay structures are described as follows:

Type 1 - BASIC is not overlaid.

Type 2 - The error messages are overlaid with a segment of the edit phase.

Type 3 - The error message phase, edit phase, and the execute phase are overlaid.

Type 4 - Similar to type 3 except that the edit and execute phases are overlaid internally.

Type 5 - Similar to type 4 except that the math package, the I/O routines, and some other routines normally in the root segment are overlaid.

The higher the overlay structure type number, the less memory is required by BASIC. The saving in memory requirements is made at the expense of performance. Type 2, however, has very minimal effect on the performance of BASIC, with a substantial saving in memory if long error messages are used. Please see the documentation for more information.

Figure 1 The Dialogue Text (Cont.)

NOTE

If you interrupt SUCNFG (with CTRL/C, for example) you must restart it from the very beginning by typing RUN SUCNFG.

4.2.1 Overlay Types - The dialogue program lets you specify how your version of BASIC is to be overlaid. The listing reproduced in Figure 1 summarizes the overlay types. Use overlay type 1 if conserving memory is not a concern on your system. Overlay type 2 is suitable for RK05 and RK06 systems where speed is more important than memory conservation. DIGITAL recommends type 2 for diskette-based systems. Overlay type 3 is suitable for RK05 and RK06 systems where memory

BASIC-11/RT-11 INSTALLATION GUIDE

conservation is more important than speed. Use type 3 for diskette-based systems if memory is at a premium. Use overlay type 4 if you must strictly conserve memory. BAS8K.SAV, the smaller version of BASIC supplied in the software kit, uses overlay type 5. It has many small overlay sections and, therefore, executes slowly. Use type 5 only for 8K systems.

4.2.2 Arithmetic Hardware Support - The versions of BASIC provided in the software kit do not support arithmetic hardware. Specify EAE if your system has the Kell-A or Kell-B Extended Arithmetic Element. Specify EIS to support the Kell-E Extended Instruction Set on a PDP-11/35 or PDP-11/40. Use EIS also for any PDP-11/34, 11/45, 11/50, 11/55, or 11/70 without the FP11 Floating Point Unit. Specify FIS if your system has the Kell-F or KEV-11 Floating Instruction Set. Specify FPU to include support for the FP11 Floating Point Unit. Use FPU also for any PDP-11/60 even if it does not have the FP11-E Floating Point Unit.

4.2.3 Foreground Execution - To link BASIC to execute in the foreground with either the FB or XM monitor, simply type an F in response to the dialogue program's "Background or Foreground?" prompt. To execute BASIC in the foreground, use the FRUN command with the /N option. You must use this option to ensure that your program has adequate memory space for foreground execution. Section 4.3 contains an example of linking BASIC and executing it in the foreground.

4.2.4 Assembly Language Routines - The versions of BASIC provided in the software kit do not support the CALL statement for assembly language routines. To include CALL support, you must first add the names of your assembly language routines to BSCLI.MAC, as described in the BASIC-11/RT-11 User's Guide. Then assemble the three .MAC files together, supplying any file name for the resulting object file:

```
.MACRO/OBJECT:MYCLI BSMAC+BSASM+BSCLI (RET)
ERRORS DETECTED: 0
```

Next, assemble your assembly language routines. You can place more than one routine in an object file. Type a Y in response to the dialogue program's "Do you want CALL support?" prompt. The dialogue next requests the module names. On the first line, enter the name of the .OBJ file you specified in the MACRO assembly shown above. On the next lines, enter the names of the .OBJ files that contain your assembly language routines. On the last line, enter BSCLLB if your assembly language routines use any routines supplied by BASIC (see the BASIC-11/RT-11 User's Guide). Finally, type only the RETURN key to terminate the list of module names. The following example shows the CALL support section of the dialogue:

```
Do you want CALL support? (Y or N; default is N): Y (RET)
```

```
Enter module names (one per line): MYCLI (RET)
Module name: MOD1 (RET)
Module name: MOD2 (RET)
Module name: MOD3 (RET)
Module name: MOD4 (RET)
Module name: BSCLLB (RET)
Module name: (RET)
```

BASIC-11/RT-11 INSTALLATION GUIDE

4.2.5 **Optional Features** - You can link BASIC to add or delete support for any optional features. See the BASIC-11/RT-11 User's Guide for a discussion of the optional features. The text of the dialogue program (see Figure 1) lists the optional features you can add or delete at link time.

4.3 Sample Dialogue

The sample dialogue reproduced in Figure 2 shows how to link BASIC to run in the foreground. This example also requests a wide map listing (with /W), and adds support for EIS arithmetic hardware. The example then shows how to execute the indirect file and how to run the new version of BASIC in the foreground.

```
.R BASIC (RET)
BASIC-11/RT-11 V02-02
OPTIONAL FUNCTIONS (ALL, NONE, OR INDIVIDUAL)? A (RET)

READY
RUN SUCNFG (RET)
BASIC-11/RT-11 V02 Configuration Program

Type "?<ret>" after any prompt for HELP.

.SAV and .COM file name (default is BASIC): MYBAS (RET)

Do you want a map? (Y or N; default is N): Y (RET)

.MAP name (default is same as .SAV name): /W (RET)

Background or Foreground ? (B or F; default is B): F (RET)

Do you want PRINT USING? (Y or N; default is Y): Y (RET)

Do you want CALL support? (Y or N; default is N): N (RET)

Single or Double Precision math? (S or D; default is S): D (RET)

Arithmetic hardware (NONE, EAE, EIS, FIS, or FPU; default is NONE): EIS (RET)

Do you want the transcendental functions? (Y or N; default is Y): Y (RET)

Do you want the SUB command? (Y or N, default is Y): Y (RET)

Do you want the RESEQ command? (Y or N, default is Y): Y (RET)

Do you want long error messages? (Y or N; default is Y): Y (RET)

Overlay structure type (1, 2, 3, 4, or 5; default is 1): 3 (RET)

The following GLOBALS will be undefined. These
errors may be ignored.
..UAC$, ..NRC$, and ..MSP$ (from no CALL).

To create the BASIC that you have just defined,
type "BYE<ret>" after BASIC prints "READY". When the system
prompts for input, type "@MYBAS<ret>" to link the new BASIC.
See the User's Guide for instructions on running
BASIC in the Foreground.
```

Figure 2 Sample Dialogue for Foreground

BASIC-11/RT-11 INSTALLATION GUIDE

```
READY
BYE (RET)

.@MYBAS (RET)

.R LINK
*MYBAS,/W=/R:1200//
*BSPTRD,BSPAT,BSROD
*BSR1D,BSPRO
*BSCLIN
*SUO1ID,BSOTOD.EIS,BSOT1D.EIS
*SUIOPR,SUIOPD
*SUXID,BSX0A,BSX0B,BSX1A,BSX1B/O:1
*BSPR1,BSCLSN
*SUX2ID,BSX2/O:1
*SUE1ID,BSE0,BSE1,BSKEYS/O:1
*BSCMP,BSSUB,BSRSQ
*SUDTCM/O:1
*BSERR,BSERML/O:1
*SUIIMP,BSFUNC,SUOPT/O:2
*//
^LINK-W-Undefined slobals:
..UAC$
..NRC$
..MSP$
*^C

.FRUN MYBAS/N:3000.(RET)

F>
BASIC-11/RT-11 V02-03
OPTIONAL FUNCTIONS (ALL, NONE, OR INDIVIDUAL)? (CTRL/P A) (RET)

READY
10 PRINT "WE ARE RUNNING IN FG NOW" (RET)
RUNNH (RET)

WE ARE RUNNING IN FG NOW

READY
BYE (RET)

B>
UNLOAD FG (RET)

*
```

Figure 2 Sample Dialogue for Foreground (Cont.)

4.4 Installation Instructions

The BASIC-11/RT-11 software kit is distributed in the following forms:

- 1 RK05 disk
- 1 RK06 disk
- 1 RL01 disk
- 1 9-track magtape
- 2 diskettes
- 2 diskettes

BASIC-11/RT-11 INSTALLATION GUIDE

When you receive the software kit, mount the distribution media (write-protected, if possible) and copy all the files onto an empty disk or tape. Then store the original distribution media in a safe place. Use your own copy of the software for the following installation procedure.

If a version of BASIC provided in the software kit (BAS8K.SAV, BASICD.SAV, or BASIC.SAV) is suitable for your application and you do not need to modify it, simply transfer the appropriate file from your copy of the distribution medium to your system device. Use a command similar to this one:

```
.COPY DX1:BAS8K.SAV DK:(RET)
```

You can now execute BASIC by typing:

```
.R BAS8K (RET)
```

If a version of BASIC provided in the software kit is not suitable for your application, you must use the program SUCNFG to relink BASIC. If your system device is an RK05 or RK06 disk, transfer all the files from your copy of the distribution medium to your system device. Use a command similar to this one:

```
.COPY/SETDATE RK1:*. * SY:(RET)
```

for non-magtape distribution media, and

```
.COPY RK1:*. * SY:(RET)
```

for magtape distribution medium.

Run the dialogue program as described in Section 4.2. Then delete all the new files on your system device, except the new version of BASIC that you just created. Use this command to delete all the new files except for the new BASIC file:

```
.DELETE/NEWFILES/EXCLUDE { BASIC.SAV } (RET)  
                                  filnam.tape
```

If your system device is a diskette or DECTape, your BASIC software kit consists of two volumes. The first volume contains all the .SAV, .BAS, .DAT, and .MAC files. The second volume contains all the .OBJ, .EAE, .EIS, .FIS, and .FPU files.

NOTE

If your system device is DECTape, follow this procedure for diskette-based systems, but substitute DT for DX in the examples that follow.

First, mount the first volume in drive 1. Assign DK: to that unit, as follows:

```
.ASSIGN DX1: DK:(RET)
```

Next, run BASIC and SUCNFG, and answer the questions from the dialogue program. A sample run is reproduced in Figure 3.

BASIC-11/RT-11 INSTALLATION GUIDE

```
.R BASIC (RET)
BASIC-11/RT-11 V02-02
OPTIONAL FUNCTIONS (ALL, NONE, OR INDIVIDUAL)? A (RET)

READY
RUN SUCNFG (RET)
BASIC-11/RT-11 V02 Configuration Program

Type "?<ret>" after any prompt for HELP.

.SAV and .COM file name (default is BASIC): BASSM (RET)
Do you want a map? (Y or N; default is N): N (RET)
Background or Foreground ? (B or F; default is B): B (RET)
Do you want PRINT USING? (Y or N; default is Y): N (RET)
Do you want CALL support? (Y or N; default is N): N (RET)
Single or Double precision math? (S or D; default is S): S (RET)
Arithmetic hardware (NONE, EAE, EIS, FIS, or FPU; default is NONE): NONE (RET)
Do you want the transcendental functions? (Y or N; default is Y): N (RET)
Do you want the SUB command? (Y or N; default is Y): N (RET)
Do you want the RESEQ command? (Y or N; default is Y): N (RET)
Do you want long error messages? (Y or N; default is Y): N (RET)
Overlay structure type (1, 2, 3, 4, or 5; default is 1): 2 (RET)

The following GLOBALS will be undefined. These
errors may be ignored.
..PRU$ (from no PRINT USING).
..UAC$, ..NRC$, and ..MSF$ (from no CALL).
SQRT, SIN, COS, ATAN, ALOG10, EXP, and
ALOG (from no transcendental functions).
..SUB$ (from no SUB command).
..RES$ (from no RESEQ command).

To create the BASIC that you have just defined,
type "BYE<ret>" after BASIC prints "READY". When the system
prompts for input, type "@BASSM<ret>" to link the new BASIC.
You may type "R BASSM<ret>" to run it.

READY
```

Figure 3 Sample Dialogue for Background

When the dialogue program finishes, exit from BASIC and copy the new indirect file to your system device:

```
BYE (RET)
COPY BASSM.COM DXO:BASSM.COM (RET)
```

Now remove the first volume from drive 1 and mount the second volume there. Execute the indirect file to link BASIC, as Figure 4 shows.

BASIC-11/RT-11 INSTALLATION GUIDE

```
.@DX0:BASSM.COM (RET)
```

```
.R LINK
*BASSM=/B:1000//
*BSPTRS,BSFAT,BSROS
*BSRIS,BSFRON
*BSCLIN
*SUC1ID,BSOTOS
*SUIOPR,SUIOP0,SUDTCM
*BSX0A,BSX0B,BSX2,BSX1A
*BSX1B,BSPR1N,BSCLSN
*BSEO,BSCMP
*BSSUBN,BSRSQN
*SUE1ID,BSE1,BSKEYS/O:1
*BSERR,BSERMS/O:1
*SUIMP,BSFUNC,SUOFT/O:2
*//
?LINK-W-Undefined globals:
..UAC$
..SUB$
..RES$
..PRU$
..NRC$
..MSP$
*^C
```

Figure 4 Linking BASIC

Next, copy the resulting .SAV (or .REL for foreground) file to your system device, and reassign DK: to DX0:.

```
.COPY BASSM.SAV DX0:BASSM.SAV (RET)
```

```
.ASSIGN DX0: DK: (RET)
```

You can now execute your new version of BASIC:

```
.R BASSM (RET)
BASIC-11/RT-11 V02-03
OPTIONAL FUNCTIONS (ALL, NONE, OR INDIVIDUAL)?
```

5.0 THE CONVERTING PROGRAM

The BASIC software kit contains a converting program, SUCVT, that helps you modify programs written under BASIC/RT-11 V01B so that they execute properly under BASIC-11/RT-11 V2. Use this program to modify all V01B BASIC programs that contain any of the following items:

- Virtual arrays
- User-defined functions that return strings
- REM statements that include a backslash (\)

BASIC-11/RT-11 INSTALLATION GUIDE

- BIN or OCT functions that do not contain parentheses
- Line numbers greater than 32767
- DATA statements on multi-statement lines
- CHR\$ or STR\$ functions
- RANDOMIZE statements
- CALL statements

See Section 2.0 of this document for a summary of how the items listed above have changed in V2 BASIC.

5.1 Starting the Converting Program

To use the converting program, first make sure that the following two programs, which are provided in the software kit, reside on your system device:

1. SUCVT.BAS
2. SUCVT1.BAS

Then start BASIC. Respond to the READY prompt by typing:

```
RUN SUCVT (RET)
```

The converting program first requests the name of the file you need to convert. Enter the input file specification in the standard RT-11 format (dev:filnam.typ). Then the program requests the name to be assigned to the resulting converted program. Enter the output file specification. If you type only the RETURN key for the output file name, the converting program examines the input file but does not create a converted output file. If you omit the file type in the input or output file specification, the converting program uses .BAS. The following is an example of the initial dialogue with the converting program:

```
RUN SUCVT (RET)
```

```
INPUT FILE NAME? CVTEST (RET)  
OUTPUT FILE NAME? CVOUT (RET)
```

Next, the converting program requests the items you need to check for V2 BASIC compatibility (a complete list of the possible items is reproduced at the beginning of Section 5.0):

```
WHICH FEATURES WOULD YOU LIKE TO CHECK (ALL, INDIVIDUAL SELECTION)?
```

To check all possible items for V2 BASIC compatibility, type an A. Typing only the RETURN key is equivalent to typing an A. If you type an A (or a RETURN), the converting program begins to examine the input file.

If you type an I, the converting program lists each feature separately. If you need to examine a particular item, type a Y (or a RETURN). If you do not need to check a feature, type an N. The following is a sample dialogue:

BASIC-11/RT-11 INSTALLATION GUIDE

```
WHICH FEATURES WOULD YOU LIKE TO CHECK (ALL, INDIVIDUAL SELECTION)? I (RET)  
VIRTUAL ARRAYS(Y OR N) ? Y (RET)  
DEF STATEMENTS(Y OR N) ? Y (RET)  
REM STATEMENTS(Y OR N) ? N (RET)  
BIN AND OCT USAGE(Y OR N) ? Y (RET)  
STR# AND CHR# USAGE(Y OR N) ? Y (RET)  
DATA STATEMENTS(Y OR N) ? Y (RET)  
RANDOMIZE STATEMENTS(Y OR N) ? N (RET)  
CALL STATEMENTS(Y OR N) ? Y (RET)
```

As a result of the dialogue shown above, the converting program would check all items except REM and RANDOMIZE statements. When you respond to the last line of the dialogue, the converting program begins to examine the input file.

5.2 Interacting with the Converting Program

As the converting program examines the input file, it searches for the items that you requested in the dialogue. When it finds an occurrence of a requested item, it prints the item (in most cases) and gives you an opportunity to modify a program line or add new program lines.

When the converting program finds a virtual array OPEN statement, it prints the existing array name and requests a new array name. If you type just the RETURN key, the converting program repeats the request. If you type LIST, the converting program prints the program line that contains the OPEN statement, then repeats the request for a new array name. When you type a new virtual array name, the converting program replaces all occurrences of the existing name with the new name. Make sure that the new name you type is not used already as a variable in the program. After you enter a new virtual array name, the converting program requests a channel number for that array. If you type only the RETURN key, the converting program repeats the request. Make sure the channel number you enter is in the range 1 to 12 and is not used in any OPEN statement already in the program.

When the converting program finds a DEF statement, it prints the existing function name and requests a new function name. If you type just the RETURN key, the converting program retains the existing function name and continues processing the input file. If you type LIST, the converting program prints the program line that contains the DEF statement, then repeats the request. When you type a new function name, the converting program replaces all occurrences of the existing name with the new name. Again, make sure that the new name you type is not used in any other DEF statement in the program.

The converting program automatically corrects program lines that contain REM statements with a backslash (\) and those that use the BIN or OCT functions without parentheses. For REM statements, the converting program substitutes the words "BACK-SLASH" for the actual backslash character (\). For BIN and OCT functions, the converting program supplies the missing parentheses. The converting program does not request any information from you when it examines REM statements and BIN and OCT functions.

The converting program always checks the line numbers in your program to see if there are line numbers greater than 32767. If the converting program detects such a line number, it prints the line identified by that line number and requests a new line number for it. Respond by typing a new line number; do not retype the entire statement on that line. Note that if you do not enter a new line number that is less than or equal to 32767, the program cannot execute under V2 BASIC.

BASIC-11/RT-11 INSTALLATION GUIDE

When the converting program finds a CHR\$ or STR\$ function, a CALL or a RANDOMIZE statement, or a DATA statement that is on a multi-statement line and is not the last item, it first prints the appropriate warning message from the following list:

```
WARNING! FUNCTION CHR$ HAS CHANGED
WARNING! FUNCTION STR$ HAS CHANGED
CALL INTERFACE HAS CHANGED
FUNCTION OF RANDOMIZE STATEMENT HAS CHANGED
DATA STATEMENT NOT LAST
```

Then the converting program prints the line that contains the function or statement and requests a new line. If you type only the RETURN key, the converting program leaves the existing program line unchanged. If you type a new program line, the converting program deletes the original line and inserts the new line. The converting program continues to request new lines so that you can properly modify your V01B program. Make sure that the new lines you type do not have the same line numbers as existing program lines, except for the line number of the deleted line. When you are finished, respond to the new line request by typing just the RETURN key.

When the converting program finishes examining the input file it asks if you need to convert another program:

```
CONVERSION COMPLETE.
ANOTHER CONVERSION? (Y OR N) ?
```

Type a Y (or the RETURN key) to restart the converting program. Type an N to terminate the program and return control to BASIC.

5.3 Converting a Sample Program

This section contains a sample V01B BASIC program, a reproduction of the converting program dialogue, and a listing of the resulting converted file. Read the dialogue carefully. Then compare the V01B BASIC program with the V2 BASIC version.

Figure 5 is a listing of a V01B BASIC program called CVTEST.BAS.

BASIC-11/RT-11 INSTALLATION GUIDE

```
100 DEF FNV(S)=CHR$(S)
200 DEF FNX(T)=T+T^2
300 OPEN 'TEST1.DAT' AS FILE #6
400 OPEN 'TEST2.DAT' FOR OUTPUT(4) AS FILE #4
500 OPEN 'TEST3.DAT' FOR OUTPUT AS FILE VF1(25)
600 OPEN 'TEST4.DAT' FOR OUTPUT(10) AS FILE VF2%(25)
700 OPEN 'TEST5.DAT' FOR OUTPUT AS FILE VF5$(64)
750 OPEN 'TEST6.DAT' AS FILE VF6$(6)=16
800 REM TEST PROGRAM FOR V1B
900 REM CONTAINS A \ FOLLOWEDBYMORETEXT
1000 DATA 'NOT','AT','END','OF','A','LINE'\GO TO 1100
1100 K=OCT('707')
1200 PRINT K
1250 PRINT K+VF1(4)
1300 K=OCT('707')
1400 PRINT K
1450 PRINT K+VF2(3)
1500 K=BIN('101010')
1600 PRINT K
1700 K=BIN'101010'
1800 PRINT K
1900 K=BIN'10101'+12
2000 PRINT K
2100 P$='GT Y'&FNV(5)
2200 PRINT FN X(9)
2300 PRINT FNV(6),FN X(VF1(7))
2400 LET VF5(5)='NOW IS THE '&STR$(FN X(1))
2500 RANDOMIZE
2600 CALL 'HYPOT'(G,K,Z)
2800 CLOSE VF1
2900 CLOSE VF5
60000 END
```

Figure 5 V01B BASIC Program

Figure 6 is a sample dialogue with the converting program. Note that all the possible features are to be checked for V2 BASIC compatibility.

BASIC-11/RT-11 INSTALLATION GUIDE

```

READY
RUN SUCVT (RET)

INPUT FILE NAME? CVTEST (RET)
OUTPUT FILE NAME? CVOUT (RET)
WHICH FEATURES WOULD YOU LIKE TO CHECK (ALL, INDIVIDUAL SELECTION)? I (RET)
VIRTUAL ARRAYS(Y OR N) ? Y (RET)
DEF STATEMENTS(Y OR N) ? Y (RET)
REM STATEMENTS(Y OR N) ? Y (RET)
BIN AND OCT USAGE(Y OR N) ? Y (RET)
STR$ AND CHR$ USAGE(Y OR N) ? Y (RET)
DATA STATEMENTS(Y OR N) ? Y (RET)
RANDOMIZE STATEMENTS(Y OR N) ? Y (RET)
CALL STATEMENTS(Y OR N) ? Y (RET)
FUNCTION NAME: FNV
NEW NAME: ? FNV$ (RET)
FUNCTION NAME: FNX
NEW NAME: ? LIST (RET)
200 DEF FNX(T)=T+T^2
NEW NAME: ? (RET)
VIRTUAL ARRAY NAME VF1
NEW NAME ? C (RET)
CHANNEL NUMBER ? 1 (RET)
VIRTUAL ARRAY NAME VF2%
NEW NAME ? C% (RET)
CHANNEL NUMBER ? 2 (RET)
VIRTUAL ARRAY NAME VF5$
NEW NAME ? C$ (RET)
CHANNEL NUMBER ? 3 (RET)
VIRTUAL ARRAY NAME VF6$
NEW NAME ? C1$ (RET)
CHANNEL NUMBER ? 5 (RET)
LINE NUMBER GREATER THAN MAX
OLD LINE: 60000 END
NEW LINE NUMBER: ? 32000 (RET)
WARNING! FUNCTION CHR$ HAS CHANGED.
OLD LINE : 100 DEF FNV$(S)=CHR$(S)
NEW LINE : ? (RET)
DATA STATEMENT NOT LAST
OLD LINE : 1000 DATA 'NOT','AT','END','OF','A','LINE'\GO TO 1100
NEW LINE : ? 1000 GOTO 1100 (RET)
NEW LINE : ? 1010 DATA 'NOT','AT','END','OF','A','LINE' (RET)
NEW LINE : ? (RET)
WARNING! FUNCTION STR$ HAS CHANGED.
OLD LINE : 2400 LET C$(5)='NOWIS THE '&STR$(FNX(1))
NEW LINE : ? (RET)
FUNCTION OF RANDOMIZE STATEMENT HAS CHANGED
OLD LINE : 2500 RANDOMIZE
NEW LINE : ? 50 RANDOMIZE (RET)
NEW LINE : ? (RET)
CALL INTERFACE HAS CHANGED
OLD LINE : 2600 CALL 'HYPOT'(G,K,Z)
NEW LINE : ? (RET)
CONVERSION COMPLETE.
ANOTHER CONVERSION? (Y OR N) ? N (RET)

READY

```

Figure 6 Converting Program Dialogue

Figure 7 is the listing of the resulting converted program, called CVOUT.BAS:

BASIC-11/RT-11 INSTALLATION GUIDE

OLD CVOUT (RET)

READY

LIST (RET)

CVOUT 17-AUG-77 11:37:19

```
50 RANDOMIZE
100 DEF FNV$(S)=CHR$(S)
200 DEF FNX(T)=T+T^2
300 OPEN 'TEST1.DAT' AS FILE #6
400 OPEN 'TEST2.DAT' FOR OUTPUT AS FILE #4, FILESIZE 4
500 OPEN 'TEST3.DAT' FOR OUTPUT AS FILE #1 \ DIM #1,C(25)
600 OPEN 'TEST4.DAT' FOR OUTPUT AS FILE #2, FILESIZE 10 \ DIM #2,C%(25)
700 OPEN 'TEST5.DAT' FOR OUTPUT AS FILE #3 \ DIM #3,C$(64)=32
750 OPEN 'TEST6.DAT' AS FILE #5 \ DIM #5,C1$(6)=16
800 REM TEST PROGRAM FOR V1B
900 REM CONTAINS A "BACK-SLASH" FOLLOWEDBYMORETEXT
1000 GO TO 1100
1010 DATA 'NOT','AT','END','OF','A','LINE'
1100 K=OCT('707')
1200 PRINT K
1250 PRINT K+C(4)
1300 K=OCT('707')
1400 PRINT K
1450 PRINT K+C%(3)
1500 K=BIN('101010')
1600 PRINT K
1700 K=BIN('101010')
1800 PRINT K
1900 K=BIN('10101')+12
2000 PRINT K
2100 P$='GT'&FNV$(5)
2200 PRINT FNX(9)
2300 PRINT FNV$(6),FNX(C(7))
2400 LET C$(5)='NOW IS THE '&STR$(FNX(1))
2600 CALL 'HYPOT'(G,K,Z)
2800 CLOSE 1
2900 CLOSE 3
32000 END
```

READY

Figure 7 V2 BASIC Program

5.4 Compiling the Converting Program

If you need to use the converting program frequently, compile the program so that it executes more efficiently. See Figure 8 for the commands to do this.

BASIC-11/RT-11 INSTALLATION GUIDE

OLD SUCVT (RET)

READY
COMPILE (RET)

READY
OLD SUCVT1 (RET)

READY
COMPILE (RET)

READY

Figure 8 Compiling the Converting Program

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

Fold Here

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 3
MAYNARD, MA

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
146 Main Street ML5-5/E39
Maynard, Massachusetts 01754

