BASIC/RT-11 V01B

Release Notes

Order No. DEC-11-LBRNA-B-D

ABSTRACT

This document describes the differ-
ences between BASIC/RT-11 V01B and
BASIC/RT-11 V01-05. It also de-
scribes the problems and solutions
associated with generating and run-
ning BASIC/RT-11 V01B systems. The
system manager should read this en-
tire document before generating the
system as described in the RT-11
System Generation Manual (DEC-11-
ORGMA-A-D).

digital equipment corporation · maynard, massachusetts

The postage prepaid READER'S COMMENTS form on the last page of this
document requests the user's critical evaluation to assist us in pre-
paring 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-10
DECCOMM	DECsystem-20	TYPESET-11

CONTENTS

1.0 INTRODUCTION

This document describes the differences between BASIC/RT-11 V01B and
BASIC/RT-11 V01-05. It also describes the problems and solutions as-
sociated with generating and running BASIC/RT-11 V01B systems. The
system manager should read this entire document before generating the
system as described in the RT-11 System Generation Manual (DEC-11-
ORGMA-A-D). For information on related RT-11 documents, see the RT-11
Documentation Directory (DEC-11-ORDDA-A-D).

1.1 Documentation Conventions

The following are the conventions and abbreviations used throughout
this document:

1. All numbers are listed in octal unless otherwise in-
    dicated.

2. <CR> indicates that the RETURN key should be typed
    at that place in the dialogue.

3. Underlining indicates computer output when it is
    necessary to differentiate output from user input.

4. ^C indicates that a CTRL C should be typed at that
    place in the dialogue.

2.0 DIFFERENCES BETWEEN BASIC/RT-11 V01B AND BASIC/RT-11 V01-05

Outlined below are the major differences between BASIC/RT-11 V01B and
BASIC/RT-11 V01-05; users upgrading from V01-05 to V01B should be
aware of these differences.

1. Problems with BASIC/RT-11 V01-05 for which an article
    and/or patch was published in the Digital Software
    News.

2. Problems with BASIC/RT-11 V01-05 for which no article
    was ever published.

3. The Lab LPS support and the graphic GT-44 support are
    not included in the BASIC/RT-11 kit.

Note that in all cases in both categories, the problems have been cor-
rected in BASIC/RT-11 V01B; published patches should not be applied to
BASIC/RT-11 V01B.

2.1 Problems in V01-05 That Are Corrected in V01B

The following are problems that existed in BASIC/RT-11 V01-05 for which
an article and/or patch was published in the Digital Software News.

Problem/Article Name	D.S.N. Sequence #
1. Erroneous Results When Using Virtual Files	5
2. Overlay Command Produces "?ULN" Error	6

1

3.	Overlay Appears to Scratch Data Area (Note that it is no longer necessary to follow overlay statements with a RESTORE statement.)	7
4.	Reference to 128-Character String Virtual File Causes BASIC to HALT	10, 11
5.	"?ETC" from Long Programs Running in Machines Greater Than 16K	12

2.2 Unpublished Problems in V01-05 That Are Corrected in V01B

The following is a list of problems that existed in BASIC/RT-11 V01-05 for which no article or patch was ever published. Where applicable, a Section from the BASIC/RT-11 Language Reference Manual is given for reference.

1. BASIC virtual memory files, as produced by BASIC/RT-11 V01B, are completely compatible with FORTRAN/RT-11 direct access files.

2. The BASIC "POS (X$,Y$,Z)" function now produces correct results for all values of Z. (Section 6.3)

3. The "OVERLAY" command no longer changes the current program name.

4. BASIC now handles input lines up to 132 characters.

5. The user-defined function evaluation algorithm has been changed to correct a problem that occasionally generated incorrect re-sults. The problem occurred when the user-defined function used more than one argument, and the same argument names were used in the model statement (DEF) and the executable statement. For ex-ample:

        10 DEF FNA(X,Y)=X+Y
        20 X=2
        30 PRINT FNA(10,X)

    These statements caused 20, rather than 12, to be printed.

6. The power fail support in BASIC, which was minimal, has been re-moved. (Section 1.1)

7. A problem that occurred in overlaid versions of BASIC, due to the initial value of the stack pointer, has been corrected. The prob-lem caused an area of memory, from approximately 23706 to 23776, to be overwritten since it was used by the overlay processor as stack space.

8. When input from the high speed reader is used, no "^" prompt char-acter is typed. (Section 5.5.2.1)

3.0 CHANGES AND RESTRICTIONS

BASIC/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. In addition to continued surveillance of the above documents, the user should be aware of the following changes and restrictions at installation time.

Patches made with RT-11 PATCH, as described in this chapter, assume a non-overlaid version of BASIC is being patched. The following procedure is used when patching overlaid versions of BASIC:

```
.R PATCH<CR>

PATCH Version number

FILE NAME --              [use BAS8K or name of overlaid ver-
*BAS8K.SAV/O<CR>          sion being patched]

?BOTTOM ADDR WRONG?

*400;B<CR>               [use 400 or bottom address specified
                          to LINK with /B if different]
*nnnnnn;OR<CR>           [nnnnnn is BASICR address from link
                          map]
*  .                     [rest of procedure is same as for
   .                      non-overlaid versions]
   .
```

The error message ?BOTTOM ADDR WRONG? is expected. It is generated as a result of the way BASIC initializes the stack pointer in overlaid versions.

3.1 Syntax Errors in Multiple Statement GOSUB or in DEF Statement

If a syntax error appears in a multiple statement line after a GOSUB, the line number reported in the error message is not the line number containing the error. Instead, it is the line number of the RETURN statement which returned control after the GOSUB. For example,

```
10   GOSUB 30  \ERROR
20   STOP
30   RETURN
```

When RUN, the message ?SYN AT LINE 30 will appear instead of ?SYN AT LINE 10.

Similarly, if a syntax error appears in the DEF statement for a user-defined function, the line number reported in the error message is not the line number containing the error. Instead, it is the line number of the statement which referenced the user-defined function. For example:

```
10   DEF FNA(X)=ERROR
20   A=FNA(10)
30   STOP
```

When RUN, the message "?SYN AT LINE 20" will be printed instead of "SYN AT LINE 10".

3

3.2 SAVE Command to Magtape and Cassette

When doing a SAVE to magtape or cassette, BASIC/RT-11 does not check to see if there is already a file on the magtape or cassette that has the same name as the file being saved. If a SAVE is done to disk or DECtape, BASIC checks for such a file and issues a "?RPL" message, indicating that the REPLACE command must be used. For magtape or cassette, the SAVE always deletes the old file and puts the new file on the tape.

3.3 Bottom Address When Linking BASIC

All distributed versions of BASIC/RT-11 V01B, except BAS8K.SAV, are linked with a bottom address of 500_8; this is because the area below 500_8 can be used for device interrupt vectors. If the user's configuration has no devices with vectors above 400_8, he can link any of the BASIC versions with a bottom address of 400_8. The distributed version of BAS8K.SAV is standardly linked with a bottom address of 400_8 to allow maximum user area in small machines. Users with 8K, whose configurations have interrupt vectors above 400_8, should relink BAS8K.SAV with a bottom address of 500_8.

3.4 Column Width

BASIC/RT-11 V01-05 had a fixed-column size of 72 characters for all devices. For BASIC/RT-11 V01B this column size may be changed for all devices by patching a single location. Attempting to PRINT on any device past the highest column number, as specified by this location, causes BASIC to print an automatic carriage return/line feed and continue output on the next line.

The following example shows how to change the column width to 132 *for all devices* (including terminal).

```
.R PATCH<CR>

PATCH Version number

FILE NAME --
*BASIC.SAV<CR>
*500;0R<CR>                [use 500 or BASICR address from link
                            map, if different]

*0,10/  110     204<CR>    [column size in octal]
*E
```

The column size is stored in location COLSIZ, which is a global and appears in the link map. It is at an offset of 10 (octal) from the beginning of BASICR. Note that once the patch is made, any BASIC program that depends on BASIC's printing an automatic carriage return/line feed at column 72 will generate different output. COLSIZ initially contains 72. If its contents are changed to 'n', the TAB function is then done in modulo n rather than modulo 72.

3.5 Stack Size

BASIC/RT-11 uses part of the user area for its working stack. A portion of this stack is reserved for interrupt level routines (device handlers) that require stack space. The remainder is used by BASIC and

user assembly language routines. It is possible that some user-written modules, or interrupt level routines, will require more stack space than is currently reserved for them. Therefore, the following procedure can be used to increase the total amount of stack space, or the portion of that total reserved for interrupt level routines, or both. The two locations involved are STKEXT and STKSIZ. The first, STKEXT, is the number of bytes (must be even) of stack space reserved for interrupt level routines; the second, STKSIZ, is the *total* number of bytes (must be even) of stack space that may be used by BASIC, by interrupt level routines, and by user routines (includes STKEXT).

Initially, the value of STKEXT is 40_8 and STKSIZ 200_8. Since STKSIZ includes STKEXT, this means that the amount of stack space available for BASIC and user assembly language routines is 140_8. To increase the amount of stack space available to interrupt level routines, both values should be increased by the desired number of bytes of stack space to be added. Locations STKEXT and STKSIZ are globals and will show up in the link map. They are at an offset of 14_8 and 12_8, respectively, from the start of BASICR.

The following example shows how to increase the stack space available to interrupt level routines (by 20_8 bytes) by adding 20_8 to the current values stored in STKSIZ and STKEXT:

```
    .R PATCH<CR>

    PATCH Version number

    FILE NAME --
    *BASIC.SAV<CR>
    *500;0R<CR>              [use 500 or BASICR address from link
                             map, if different]
    *0,12/    200    220<CR> [STKSIZ]
    *0,14/    40     60<CR>  [STKEXT]
    *E
```

If the user has the sources of BASIC/RT-11, these patches can be made by defining the variables:

```
    $STKSZ=220
    $STKEX=60
```

in an assembly parameter module and reassembling BASIC with this parameter module as described in Appendix F of the BASIC/RT-11 manual.

                              NOTE

            If the value of STKSIZ is increased by
            20_8 bytes and the value of STKEXT is not
            changed, no additional space is reserved
            for interrupt level routines, but an ad-
            ditional 20_8 bytes of stack space are re-
            served for BASIC and user assembly lan-
            guage routines.

3.6   String Storage Garbage Collection Pause

Versions of BASIC/RT-11 with string support periodically appear to hang for some time. This is usually noticed when using assembly language routines to drive some type of graphics display. This hang results

from the fact that each call of an assembly language routine causes a
new string to be created (consisting of the name of the assembly lan-
guage routine). When string storage becomes filled, BASIC performs a
garbage collect operation, causing it to become internally compute-
bound for a period of time.

This problem affects programs doing real-time data acquisition, since
BASIC can service interrupts and collect data during the garbage
collect operation, but it can neither process the data collected nor
write the data out to a file until the garbage collect is completed.
For very fast sampling rates, the interrupt service must be able to
buffer a large amount of data during a garbage collect operation in
order to avoid data loss.

The problem can be eliminated by linking the assembly language routines
with the versions of BASIC that do not support strings.

If string support is required, the following patch can be made to elim-
inate the creation of a new string each time an assembly language rou-
tine is called. Note, however, that *with this patch installed*, string
variables and string expressions are not allowed as the name of an as-
sembly language routine in a CALL statement. Only a literal string,
enclosed in quotes, may be used for the name. In addition, to prevent
BASIC from hanging, users should ensure that no string constants,
string variables, or string expressions appear inside a time-critical
data collection loop (except for the name of the assembly language rou-
tine in the CALL statement). The patch is made to location STRFND in
BASICR, which is a global and appears in the link map. The contents
of this location are changed from the address of subroutine FNDSTR to
that of subroutine FNDSTL. Both these routines are globals and their
addresses appear in the link map to facilitate changes back and forth.

The patch is as follows:

```
        .R PATCH<CR>

        PATCH Version number

        FILE NAME --
        *BASIC.SAV<CR>
        *500;0R<CR>              [use 500 or BASICR address from link
                                 map, if different]
        *0,16/  35160    3044<CR>[use 3044 or FNDSTL address from
                                 link map, if different]
        *E
```

Note that the old contents of location STRFND may vary with the version
of BASIC being patched; 35160 is used for this example only.

3.7 Module Addresses in Distributed SAV Files

The following table lists the addresses of important modules in the
distributed versions of BASIC.

.SAV File	Bottom Address	BASICR	FPMP	BASICE	BASICX	BASICH
BAS8K	400	1136	6072	12374	12374	24212
BASIC	500	500	5776	12276	24666	37242

3.8 Background Display with F/B Monitor

When using BASIC with a background display routine (such as the LPS-11
display routine) running as the background job under the F/B Monitor,
the display is not refreshed when BASIC is waiting for keyboard input.
This is because BASIC issues a .TTINR request and if no input is avail-
able, BASIC refreshes the display. This is acceptable under S/J, but
under F/B, control does not return from the .TTINR until a character is
available, so BASIC cannot refresh the display. This problem can be
avoided by setting bit 6 of the Job Status Word (location 44) for BASIC.
This causes .TTINR under F/B to act as it does under the S/J Monitor.

For example:

        .R PATCH<CR>

        PATCH Version number

        FILE NAME --
        *BASLPS.SAV<CR>
        *44/   0   100<CR>
        *E
        .

For overlaid versions:

        .R PATCH<CR>

        PATCH Version number

        FILE NAME --
        *BGTLPO.SAV/O<CR>
        ?BOTTOM ADDR WRONG?
        *500;B<CR>
        *44/   1000   1100<CR>
        *E
        .

INDEX

Abbreviations, 1
APUT routine, 1

BAS8K module addresses, 12
BASIC module addresses, 12
Bottom address when linking
    BASIC, 6
Buffer storage overflow, 3

Changes and restrictions, 3
Changing column width, 7
Column width,
  changing, 7
<CR> (carriage return), 1
CTRL C, 1

DEF statement, 2, 5
.DEVICE list, 10
Documentation conventions, 1

ERAS loops, 3
Erroneous line numbers, 5

File processing, 3
FNA statement, 2
FORTRAN/RT-11 direct access
    file, 2

GOSUB statement, 5
GTON.SAV, 3

Input lines, 2
INT(X) function, 3
Interrupt level device support,
  assembly language, 6

Linking BASIC, 6

Module addresses in .SAV
    files, 12

Octal numbers, 1
OLD 'PR:', 3
Overlay command, 2

Patching BASIC, 4, 8
POS function, 2
Power fail support, 2
Problems in V01-05, 1

Restrictions and changes, 3

SAVE command,
  to magtape, 6
  to cassette, 6
Stack,
  pointer, 3
  size, 7
String storage,
  garbage collection pause, 8
  overflow, 3
String support, 8
Syntax errors, 5

Underlining, 1

Virtual files, 2

READER'S COMMENTS

NOTE: This form is for document comments only. Problems
        with software should be reported on a Software
        Performance Report (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

If you require a written reply, please check here.   ☐

-- Fold Here --

------------------------------------ Do Not Tear - Fold Here and Staple ------------------------------------