


```

BBBBBBBBB      AAAAAA      SSSSSSSS      RRRRRRRR      AAAAAA      DDDDDDDD      5555555555      000000
BBBBBBBBB      AAAAAA      SSSSSSSS      RRRRRRRR      AAAAAA      DDDDDDDD      5555555555      000000
BB          BB  AA          AA  SS          RR          RR  AA          AA  DD          DD  55          00          00
BB          BB  AA          AA  SS          RR          RR  AA          AA  DD          DD  55          00          00
BB          BB  AA          AA  SS          RR          RR  AA          AA  DD          DD  555555      00          0000
BB          BB  AA          AA  SS          RR          RR  AA          AA  DD          DD  555555      00          0000
BBBBBBBBB      AA          AA          SSSSSS      RRRRRRRR      AA          AA          DD          DD          55          00          00
BBBBBBBBB      AA          AA          SSSSSS      RRRRRRRR      AA          AA          DD          DD          55          00          00
BB          BB  AAAAAAAAAA      SS          RR          RR  AAAAAAAAAA      DD          DD          55          0000      00
BB          BB  AAAAAAAAAA      SS          RR          RR  AAAAAAAAAA      DD          DD          55          0000      00
BB          BB  AA          AA          SS          RR          RR  AA          AA          DD          DD          55          00          00
BB          BB  AA          AA          SS          RR          RR  AA          AA          DD          DD          55          00          00
BBBBBBBBB      AA          AA          SSSSSSSS      RR          RR  AA          AA          DDDDDDDD      555555      000000
BBBBBBBBB      AA          AA          SSSSSSSS      RR          RR  AA          AA          DDDDDDDD      555555      000000

```

```

LL          IIIIII      SSSSSSSS
LL          IIIIII      SSSSSSSS
LL          II          SS
LL          II          SS
LL          II          SS
LL          II          SS
LL          II          SSSSSS
LL          II          SSSSSS
LL          II          SS
LL          II          SS
LL          II          SS
LL          II          SS
LLLLLLLLLLL IIIIII      SSSSSSSS
LLLLLLLLLLL IIIIII      SSSSSSSS

```

```

1 0001 0 MODULE BASSRAD50 ( ! RADIX50 conversion
2 0002 0 IDENT = '1-005' ! File: BASRAD50.B32
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: BASIC-PLUS-2 Miscellaneous
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains the BASIC RAD$ function,
36 0036 1 which converts a 16-bit word to three ASCII characters.
37 0037 1
38 0038 1 ENVIRONMENT: VAX-11 User Mode
39 0039 1
40 0040 1 AUTHOR: John Sauter, CREATION DATE: 01-MAY-1979
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original.
45 0045 1 1-002 - The BASSRAD synonym must be global! JBS 02-MAY-1979
46 0046 1 1-003 - Change LIB$$ and OTS$$ to STR$. JBS 21-MAY-1979
47 0047 1 1-004 - Change the call to STR$COPY. JBS 16-JUL-1979
48 0048 1 1-005 - Improve a comment. JBS 07-NOV-1979
49 0049 1 --
50 0050 1
51 0051 1 !<BLF/PAGE>
    
```

```

53 0052 1 |
54 0053 1 | SWITCHES:
55 0054 1 |
56 0055 1 |
57 0056 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
58 0057 1 |
59 0058 1 |
60 0059 1 | LINKAGES:
61 0060 1 |
62 0061 1 |     NONE
63 0062 1 |
64 0063 1 | TABLE OF CONTENTS:
65 0064 1 |
66 0065 1 |
67 0066 1 | FORWARD ROUTINE
68 0067 1 |     BASSRAD50 : NOVALUE;           ! convert word to three chars
69 0068 1 |
70 0069 1 |
71 0070 1 | INCLUDE FILES:
72 0071 1 |
73 0072 1 |
74 0073 1 | REQUIRE 'RTLIN:RTLPSECT';       ! Macros for defining psects
75 0168 1 |
76 0169 1 | LIBRARY 'RTLSTARLE';           ! System symbols
77 0170 1 |
78 0171 1 |
79 0172 1 | MACROS:
80 0173 1 |
81 0174 1 |     NONE
82 0175 1 |
83 0176 1 | EQUATED SYMBOLS:
84 0177 1 |
85 0178 1 |
86 0179 1 | GLOBAL BIND
87 0180 1 |     ROUTINE
88 0181 1 |     BASSRAD = BASSRAD50;       ! Synonym used by compiled code
89 0182 1 |
90 0183 1 |
91 0184 1 | PSECTS:
92 0185 1 |
93 0186 1 | DECLARE_PSECTS (BAS);         ! Declare psects for BASS$ facility
94 0187 1 |
95 0188 1 | OWN STORAGE:
96 0189 1 |
97 0190 1 |     NONE
98 0191 1 |
99 0192 1 | EXTERNAL REFERENCES:
100 0193 1 |
101 0194 1 |
102 0195 1 | EXTERNAL ROUTINE
103 0196 1 |     R50ASC : NOVALUE,           ! Convert radix50 to ASCII
104 0197 1 |     STR$COPY_R,                 ! Copy a string by reference
105 0198 1 |     LIB$STOP : NOVALUE;        ! Signal a fatal error
106 0199 1 |

```

```

: 108 0200 1 GLOBAL ROUTINE BASSRAD50 (           ! Convert a word to three characters
: 109 0201 1     RESULT,                          ! Descriptor of resultant string
: 110 0202 1     NUMBER,                          ! Value to convert
: 111 0203 1     ) : NOVALUE =
: 112 0204 1
: 113 0205 1 ++
: 114 0206 1 FUNCTIONAL DESCRIPTION:
: 115 0207 1
: 116 0208 1     Convert a 16-bit word into an ASCII string. The word is
: 117 0209 1     encoded as RADIX50.
: 118 0210 1
: 119 0211 1     This function should not be used for new development;
: 120 0212 1     someday we hope to do away with this extra character set.
: 121 0213 1
: 122 0214 1 FORMAL PARAMETERS:
: 123 0215 1
: 124 0216 1     RESULT.wt.dx      A string containing the 3 characters in the word
: 125 0217 1     NUMBER.rw.v     The 16-bit word into which the characters are
: 126 0218 1                     packed.
: 127 0219 1
: 128 0220 1 IMPLICIT INPUTS:
: 129 0221 1
: 130 0222 1     NONE
: 131 0223 1
: 132 0224 1 IMPLICIT OUTPUTS:
: 133 0225 1
: 134 0226 1     NONE
: 135 0227 1
: 136 0228 1 ROUTINE VALUE:
: 137 0229 1 COMPLETION CODES:
: 138 0230 1
: 139 0231 1     NONE
: 140 0232 1
: 141 0233 1 SIDE EFFECTS:
: 142 0234 1
: 143 0235 1     NONE
: 144 0236 1
: 145 0237 1 --
: 146 0238 1
: 147 0239 2 BEGIN
: 148 0240 2
: 149 0241 2 LOCAL
: 150 0242 2     THREE_CHARS : VECTOR [3, BYTE];
: 151 0243 2
: 152 0244 2 ++
: 153 0245 2 ! Call the FORTRAN-compatibility routine to convert radix-50 to ASCII.
: 154 0246 2 --
: 155 0247 2     R5OASC (%REF (3), NUMBER, THREE_CHARS [0]);
: 156 0248 2
: 157 0249 2 ++
: 158 0250 2 ! Copy the three characters back to the caller's descriptor.
: 159 0251 2 --
: 160 0252 1     STR$COPY_R (.RESULT, %REF (3), THREE_CHARS [0]);
:                                     ! end of BASSRAD50

```

```

.TITLE BASSRAD50
.IDENT \1-005\

```

```

                                .EXTRN R50ASC, STR$COPY_R
                                .EXTRN LIB$STOP
                                .PSECT _BAS$CODE, NOWRT, SHR, PIC, 2
                                .ENTRY BAS$RAD50, Save nothing
0000 00000                                : 0200
      5E                                :
      04 08 C2 00002                        :
      04 AE 9F 00005                        :
      08 AC 9F 00008                        :
      08 03 D0 0000B                        :
      08 AE 9F 0000F                        :
00000000G 00 03 FB 00012                    :
      04 AE 9F 00019                        :
      04 03 D0 0001C                        :
      04 AE 9F 00020                        :
00000000G 00 04 AC DD 00023                    :
      03 FB 00026                        :
      04 0002D                        :
                                .SUBL2 #8, SP
                                PUSHAB THREE CHARS
                                PUSHAB NUMBER
                                MOVL #3, 8(SP)
                                PUSHAB 8(SP)
                                CALLS #3, R50ASC
                                PUSHAB THREE CHARS
                                MOVL #3, 4(TSP)
                                PUSHAB 4(SP)
                                PUSHL RESULT
                                CALLS #3, STR$COPY_R
                                RET
                                : 0252

```

; Routine Size: 46 bytes, Routine Base: _BAS\$CODE + 0000

```

: 161      0253 1
: 162      0254 1 END
: 163      0255 1
: 164      0256 0 ELUDOM

```

! end of module BAS\$RAD50

BAS\$RAD== BAS\$RAD50

PSECT SUMMARY

```

:
: Name          Bytes          Attributes
:
: _BAS$CODE     46 NOVEC, NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)
:

```

Library Statistics

```

:
: File          Total  Symbols  Percent  Pages  Processing
:               -----  Loaded  -----  Mapped  Time
:
: _$255$DUA28:[SYSLIB]STARLET.L32;1  9776      0      0      581    00:01.0
:

```

COMMAND QUALIFIERS

BASRAD50
1-005

H 3
16-Sep-1984 01:00:48
14-Sep-1984 11:56:33

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BASRAD50.B32;1

Page 5
(3)

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:BASRAD50/OBJ=OBJ\$:BASRAD50 MSRC\$:BASRAD50/UPDATE=(ENH\$:BASRAD50)

; Size: 46 code + 0 data bytes
; Run Time: 00:03.3
; Elapsed Time: 00:10.0
; Lines/CPU Min: 4726
; Lexemes/CPU-Min: 12627
; Memory Used: 36 pages
; Compilation Complete

0030 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BasRAD50 LIS

BasRSET LIS

BasRPUT LIS

BasRECPRO LIS

BasRESTAR LIS

BasRANDOM LIS

BasREMAP LIS

BasRESTOR LIS

BasRIGHT LIS