

A large, stylized letter 'A' is formed using the characters 'S' and 'Y'. The letter is symmetrical and composed of multiple rows of these characters. The top of the 'A' is formed by a row of 'S's, followed by rows of 'Y's and 'S's. The middle section consists of a row of 'Y's, followed by rows of 'S's and 'Y's. The bottom of the 'A' is formed by a row of 'S's, followed by rows of 'Y's and 'S's. The overall shape is a wide, triangular 'A' with a central vertical stem.

[illegible]

(2) 59
(3) 86
(4) 211
(5) 248
(6) 280

DECLARATIONS
CVTFILNAM - CONVERT FILE NAME FROM ASCII TO RAD50
STORER50BYTE - CONVERT AND STORE ASCII BYTE TO RAD50
PACKRAD50 - PACK 3 BYTES OF RAD50 CHARACTERS INTO A WORD
ASCII TORAD50 - CONVERT ASCII CHARACTER TO RAD50

```
0000 1 .TITLE CVTFILNAM - Converts ASCII to RAD50
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 * ALL RIGHTS RESERVED.
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 * TRANSFERRED.
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 * CORPORATION.
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *****
0000 27
0000 28 ++
0000 29 FACILITY:
0000 30
0000 31 Used by BOOT58 and FILEREAD
0000 32
0000 33 ABSTRACT:
0000 34
0000 35 Converts an ASCII string file name, type, and version number
0000 36 to a 6-word RAD50 file name and type, and a 16-bit binary
0000 37 version number.
0000 38
0000 39 ENVIRONMENT:
0000 40
0000 41 any
0000 42
0000 43 AUTHOR:
0000 44
0000 45 Carol Peters 22 March 1979
0000 46 Extracted from the Release 1 module FILEREAD.MAR; all
0000 47 code and definitions are unchanged from Release 1 form.
0000 48
0000 49 MODIFIED BY:
0000 50
0000 51 V03-002 LJK0254 Lawrence J. Kenah 6-Dec-1983
0000 52 Change LIB$CVT_DTB to FIL$CVT_DTB.
0000 53
0000 54 V03-001 TCM0001 Trudy C. Matthews 15-Nov-1982
0000 55 Added G^ to call to LIB$CVT_DTB.
0000 56
0000 57 --
```



```
0000 59      .SBTTL  DECLARATIONS
0000 60
0000 61  :
0000 62  : Include files
0000 63  :
0000 64
0000 65      $$$DEF                      ; Status code definitions
0000 66
0000 67  :
0000 68  : Equated symbols
0000 69  :
0000 70
00000061 0000 71      LCA      =      ^0141      ; LOWER CASE A
0000007A 0000 72      LCZ      =      ^0172      ; LOWER CASE Z
00000041 0000 73      UCA      =      ^0101      ; UPPER CASE A
0000005A 0000 74      UCZ      =      ^0132      ; UPPER CASE Z
00000030 0000 75      ZERO     =      ^060       ; ASCII ZERO
00000039 0000 76      NINE     =      ^071       ; ASCII NINE
0000002E 0000 77      DOT      =      ^056       ; ASCII PERIOD
0000003B 0000 78      SEMI     =      ^073       ; ASCII SEMICOLON
0000 79
0000 80  :
0000 81  : PSECT declaration
0000 82  :
0000 83
00000000 84      .PSECT  YFILEREAD,BYTE,EXE
```

```
0000 86 .SBTTL CVTFILNAM - CONVERT FILE NAME FROM ASCII TO RAD50
0000 87
0000 88 :++
0000 89 : FUNCTIONAL DESCRIPTION:
0000 90 :
0000 91 : THIS ROUTINE CONVERTS THE ASCII FILE NAME (NAME, TYPE, VERSION)
0000 92 : TO THE FILE NAME BLOCK FORM OF 3 WORDS RAD50 NAME, 1 WORD OF TYPE
0000 93 : AND 1 WORD OF VERSION.
0000 94 :
0000 95 : CALLING SEQUENCE:
0000 96 :
0000 97 : CALLG ARGLIST,FIL$CVTFILNAM
0000 98 :
0000 99 : INPUT PARAMETERS:
0000 100 :
0000 101 : FILNAM(AP) = ;STRING DESCRIPTOR OF FILE NAME STRING
0000 102 : FILNAMBLK(AP) = ;ADDRESS OF 5 WORD BLOCK
0000 103 : 3 WORD RAD50 FILE NAME
0000 104 : 1 WORD RAD50 FILE TYPE
0000 105 : 1 WORD BINARY VERSION
0000 106 :
0000 107 : IMPLICIT INPUTS:
0000 108 :
0000 109 : NONE
0000 110 :
0000 111 : OUTPUT PARAMETERS:
0000 112 :
0000 113 : RO = SYSTEM STATUS CODE
0000 114 : FILNAME BLOCK FILLED IN
0000 115 :
0000 116 : IMPLICIT OUTPUTS:
0000 117 :
0000 118 : NONE
0000 119 :
0000 120 : COMPLETION CODES:
0000 121 :
0000 122 : SS$ NORMAL SUCCESSFUL COMPLETION
0000 123 : SS$ _BADFILENAME SYNTAX ERROR IN FILE NAME
0000 124 :
0000 125 : SIDE EFFECTS:
0000 126 :
0000 127 : NONE
0000 128 :
0000 129 : EQUATED SYMBOLS:
0000 130 :
0000 131 : OFFSETS FROM AP
0000 132 :
0000 133 : FILNAM = 4 ;DESCRIPTOR OF ASCII FILE NAME STRING
0000 134 : FILNAMBLK = 8 ;ADDRESS OF 5 WORD FILE NAME BLOCK
0000 135 :
0000 136 : OFFSETS FROM FP
0000 137 :
0000 138 : TYPE = -3 ;BEGINNING OF TYPE FIELD
0000 139 : NAME = -12 ;BEGINNING OF NAME FIELD
0000 140 :
0000 141 : --
0000 142
```

00000004
00000008FFFFFFFFD
FFFFFFFF4


```
0000 143 FIL$CVTFILNAM::
007C 0000 144 .WORD ^M<R2,R3,R4,R5,R6>
7E 7C 0002 145 CLRQ -(SP) ;RESERVE AND ZERO A
7E D4 0004 146 CLRL -(SP) ;12 BYTE NAME STRING
55 04 BC 7D 0006 147 MOVQ @FILNAM(AP),R5 ;R5 = SIZE, R6 = ADR OF STRING
54 F4 AD 9E 000A 148 MOVAB NAME(FP),R4 ;ADDRESS TO STORE NAME
53 09 D0 000E 149 MOVL #9,R3 ;UP TO 9 CHARACTERS
OF 11 0011 150 BRB 40$
0013 151 ;
0013 152 ; STORE UP TO 9 CHARACTERS OF RAD50 IN THE BYTE ARRAY ADDRESSED BY R4
0013 153 ; IN PREPARATION FOR CONVERTING INTO THE PACKED RAD50 FORMAT
0013 154 ;
0013 155 20$:
50 86 90 0013 156 MOVB (R6)+,R0 ;GET THE NEXT CHARACTER
50 2E 91 0016 157 CMPB #DOT,R0 ;IS IT A DOT?
50 0C 13 0019 158 BEQL FILETYPE ;BRANCH IF YES
50 3B 91 001B 159 CMPB #SEMI,R0 ;IS IT A SEMICOLON
24 13 001E 160 BEQL VERSION ;BRANCH IF YES
59 10 0020 161 BSBB STORER50BYTE ;CONVERT AND STORE THE CHARACTER
EE 55 F4 0022 162 40$:
2F 11 0025 163 SOBGEQ R5,20$ ;COUNT THE CHARACTERS IN THE STRING
0027 164 BRB BUILDFNB ;END OF STRING, NO TYPE, NO VERSION
0027 165 ;
0027 166 ; NOW SET UP THE 3 BYTE ARRAY OF FILE TYPE
0027 167 ;
0027 168 FILETYPE:
54 FD AD 9E 0027 169 MOVAB TYPE(FP),R4 ;ADDRESS OF BYTE ARRAY
53 03 D0 002B 170 MOVL #3,R3 ;MAX SIZE OF FILE TYPE
OF 11 002E 171 BRB 40$
0030 172 20$:
50 86 90 0030 173 MOVB (R6)+,R0 ;GET NEXT CHARACTER
50 3B 91 0033 174 CMPB #SEMI,R0 ;IS IT A SEMICOLON?
50 0C 13 0036 175 BEQL VERSION ;BRANCH IF YES
50 2E 91 0038 176 CMPB #DOT,R0 ;DOT FOR VERSION TOO
07 13 003B 177 BEQL VERSION ;BRANCH IF VERSION DELIMITER
3C 10 003D 178 BSBB STORER50BYTE ;CONVERT AND STORE THE BYTE
EE 55 F4 003F 179 40$:
12 11 0042 180 SOBGEQ R5,20$ ;LOOP THROUGH THE CHARACTERS
0044 181 BRB BUILDFNB ;END OF STRING, NO VERSION
0044 182 ;
0044 183 ; VERSION DELIMITER FOUND, CONVERT THE VERSION
0044 184 ;
0044 185 VERSION:
7E 08 AC 08 C1 0044 186 ADDL3 #4*2,FILNAMBLK(AP),-(SP) ;ADDRESS TO STORE VERSION
00000000'GF 7E 55 7D 0049 187 MOVQ R5, -(SP) ;PUSH ADDRESS, PUSH SIZE OF VERSION STRING
2A 50 FB 004C 188 CALLS #3,G^FIL$CVT_DTB ;CONVERT DECIMAL VERSION STRING TO BINARY
E9 0053 189 BLBC R0,BADFILNAM ;BRANCH IF BAD FILE NAME
0056 190 ;
0056 191 ; NOW PACK THE TEMPORARY BYTE STRING INTO THE RAD50 WORDS
0056 192 ;
0056 193 BUILDFNB:
54 08 AC D0 0056 194 MOVL FILNAMBLK(AP),R4 ;ADDRESS TO STORE PACKED RAD50
53 F4 AD 9E 005A 195 MOVAB NAME(FP),R3 ;ADDRESS OF NAME STRING
63 95 005E 196 TSTB (R3) ;ANY NAME GIVEN?
06 13 0060 197 BEQL 20$ ;BRANCH IF NO
26 10 0062 198 BSBB PACKRAD50 ;1ST 3 CHARACTERS
24 10 0064 199 BSBB PACKRAD50 ;2ND 3 CHARACTERS
```

CVTFILNAM
V04-000

- Converts ASCII to RAD50

N 14

15-SEP-1984 23:58:44

VAX/VMS Macro V04-00

Page 5
(3)

CVTFILNAM - CONVERT FILE NAME FROM ASCII

5-SEP-1984 03:41:08

[SYS.SRC]CVTFILNAM.MAR;1

54	08	AC	06	C1	0066	200	BSBB	PACKRAD50	;3RD 3 CHARACTERS
	53	FD	AD	9E	0068	201			
			63	95	0068	202	ADDL3	#3*2,FILNAMBLK(AP),R4	;ADDRESS TO STORE FILE TYPE
			02	13	006D	203	MOVAB	TYPE(FP),R3	;ADDRESS OF TYPE STRING
			13	10	0071	204	TSTB	(R3)	;ANY FILE TYPE PRESENT?
					0073	205	BEQL	40\$	;BRANCH IF NOT
					0075	206	BSBB	PACKRAD50	;CONVERT 3 CHARACTERS
					0077	207			
50	01	3C	0077	208	40\$:		MOVZWL	#SS\$_NORMAL,R0	;INDICATE SUCCESS
		04	007A	209			RET		


```
007B 211 .SBTTL STORER50BYTE - CONVERT AND STORE ASCII BYTE TO RAD50
007B 212 :++
007B 213 : FUNCTIONAL DESCRIPTION:
007B 214 :
007B 215 : CONVERT ASCII BYTE TO RAD50 AND STORE IN THE SPECIFIED BYTE ARRAY
007B 216 :
007B 217 : CALLING SEQUENCE:
007B 218 :
007B 219 : BSBB STORER50BYTE
007B 220 :
007B 221 : INPUT:
007B 222 :
007B 223 : R0 = ASCII BYTE
007B 224 : R3 = SIZE OF BYTE ARRAY TO STORE IN
007B 225 : R4 = ADDRESS OF BYTE ARRAY TO STORE IN
007B 226 :
007B 227 : OUTPUT:
007B 228 :
007B 229 : RSB IF SUCCESSFUL, RET WITH ERROR CODE IN R0 IF ERROR
007B 230 : R3, R4 UPDATED TO REFLECT THE ADDITIONAL BYTE STORED
007B 231 :
007B 232 :--
007B 233 :
007B 234 : .ENABL LSB
007B 235 :
007B 236 STORER50BYTE:
007B 237 BSBB ASCII TORAD50 ;CONVERT TO RAD50
007B 238 SOBGEQ R3,20$ ;BRANCH IF ROOM TO STORE THIS CHARACTER
0080 239 BADFILNAM:
0080 240 MOVZWL #SS$_BADFILENAME,R0 ;RETURN ERROR CODE
0085 241 RET
0086 242 20$:
0086 243 MOVB R0,(R4)+ ;STORE IT
0089 244 RSB ;AND RSB SUCCESSFULLY
008A 245
008A 246 .DSABL LSB
```

26 10 007B 237
06 53 F4 007D 238
50 0818 8F 3C 0080 239
04 0085 241
84 50 90 0086 242
05 0089 243
008A 245
008A 246


```
008A 248 .SBTTL PACKRAD50 - PACK 3 BYTES OF RAD50 CHARACTERS INTO A WORD
008A 249 :++
008A 250 : FUNCTIONAL DESCRIPTION:
008A 251 :
008A 252 : PACK 3 BYTES OF RAD50 CHARACTERS INTO A WORD
008A 253 :
008A 254 : CALLING SEQUENCE:
008A 255 :
008A 256 : BSBB PACKRAD50
008A 257 :
008A 258 : INPUTS:
008A 259 :
008A 260 : R3 = RAD50 BYTE STRING ADDRESS
008A 261 : R4 = ADDRESS OF WORD ARRAY TO STORE IN
008A 262 :
008A 263 : OUTPUTS:
008A 264 :
008A 265 : R3 UPDATED TO POINT TO NEXT GROUP OF 3 CHARACTERS
008A 266 : R4 UPDATED TO POINT AT NEXT WORD TO STORE IN
008A 267 :
008A 268 :--
008A 269 :
008A 270 PACKRAD50:
50 50 83 9A 008A 271 MOVZBL (R3)+,R0 ;BUILD PACKED WORD IN R0
0640 8F A4 008D 272 MULW #40*40,R0 ;START ACCUMULATING IN R0
51 83 9A 0092 273 MOVZBL (R3)+,R1 ;SECOND CHARACTER TO TEMP
51 28 A4 0095 274 MULW #40,R1 ;MULTIPLY BY THE RADIX
50 51 A0 0098 275 ADDW R1,R0 ;AND ACCUMULATE
51 83 9A 009B 276 MOVZBL (R3)+,R1 ;LAST CHARACTER
84 50 51 A1 009E 277 ADDW3 R1,R0,(R4)+ ;ACCUMULATE AND STORE RESULT
05 00A2 278 RSB
```



```
00A3 280 .SBTTL ASCII TORAD50 - CONVERT ASCII CHARACTER TO RAD50
00A3 281 :++
00A3 282 : FUNCTIONAL DESCRIPTION:
00A3 283 :
00A3 284 : CONVERT ASCII CHARACTER TO ITS RAD50 EQUIVALENT
00A3 285 :
00A3 286 : CALLING SEQUENCE:
00A3 287 :
00A3 288 : BSBB ASCII TORAD50
00A3 289 :
00A3 290 : INPUTS:
00A3 291 :
00A3 292 : R0 = CHARACTER TO CONVERT
00A3 293 :
00A3 294 : OUTPUTS:
00A3 295 :
00A3 296 : RSB IF SUCCESSFUL, RET WITH ERROR CODE IN R0 IF ERROR
00A3 297 : R0 = RAD50 EQUIVALENT
00A3 298 : R1,R2,R3 PRESERVED
00A3 299 :
00A3 300 :--
00A3 301 :
00A3 302 ASCII TORAD50:
7A 8F 50 91 00A3 303 CMPB R0,#LCZ :LOWER CASE ALPHA?
D7 1A 00A7 304 BGTR BADFILNAM :BRANCH IF BAD RAD50
61 8F 50 91 00A9 305 CMPB R0,#LCA
05 19 00AD 306 BLSS 10$ :BRANCH IF NOT
50 60 8F 82 00AF 307 SUBB #LCA-1,R0 :CONVERT TO ALPHA
05 00B3 308 RSB :AND RETURN SUCCESSFULLY
00B4 309 10$:
5A 8F 50 91 00B4 310 CMPB R0,#UCZ :UPPER CASE ALPHA?
C6 14 00B8 311 BGTR BADFILNAM :BRANCH IF BAD RAD50
41 8F 50 91 00BA 312 CMPB R0,#UCA
05 19 00BE 313 BLSS 20$ :BRANCH IF NOT ALPHA
50 40 8F 82 00C0 314 SUBB #UCA-1,R0 :CONVERT TO RAD50 ALPHA
05 00C4 315 RSB :AND RETURN SUCCESSFULLY
00C5 316 20$:
39 50 91 00C5 317 CMPB R0,#NINE :NUMERIC?
B6 14 00C8 318 BGTR BADFILNAM :BRANCH IF BAD RAD50
30 50 91 00CA 319 CMPB R0,#ZERO
B1 19 00CD 320 BLSS BADFILNAM :BRANCH IF BAD RAD50
50 12 82 00CF 321 SUBB #ZERO-30,R0 :CONVERT TO RAD50 NUMERIC
05 00D2 322 RSB :AND RETURN SUCCESSFULLY
00D3 323
00D3 324 .END
```


CVTFILNAM
Symbol table

- Converts ASCII to RAD50

E 15

15-SEP-1984 23:58:44
5-SEP-1984 03:41:08

VAX/VMS Macro V04-00
[SYS.SRC]CVTFILNAM.MAR;1

Page 9
(6)

ASCII TORAD50	000000A3	R	02
BADFILNAM	00000080	R	02
BUILDFNB	00000056	R	02
DOT	= 0000002E		
FIL\$CVTFILNAM	00000000	RG	02
FIL\$CVT DTB	*****	X	02
FILETYPE	00000027	R	02
FILNAM	= 00000004		
FILNAMBLK	= 00000008		
LCA	= 00000061		
LCZ	= 0000007A		
NAME	= FFFFFFFF		
NINE	= 00000039		
PACKRAD50	0000008A	R	02
SEMI	= 0000003B		
SS\$_BADFILENAME	= 00000818		
SS\$_NORMAL	= 00000001		
STORERSOBYTE	0000007B	R	02
TYPE	= FFFFFFFD		
UCA	= 00000041		
UCZ	= 0000005A		
VERSION	00000044	R	02
ZERO	= 00000030		

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes														
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
\$ABSS	00000000 (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				
YFILEREAD	000000D3 (211.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.07	00:00:01.18
Command processing	105	00:00:00.47	00:00:05.11
Pass 1	198	00:00:04.06	00:00:13.83
Symbol table sort	0	00:00:00.63	00:00:00.96
Pass 2	68	00:00:01.01	00:00:02.55
Symbol table output	4	00:00:00.04	00:00:00.04
Psect synopsis output	1	00:00:00.03	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	408	00:00:06.31	00:00:23.71

The working set limit was 1200 pages.
22998 bytes (45 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 428 non-local and 9 local symbols.
324 source lines were read in Pass 1, producing 13 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

\$255\$DUA28:[SYS.OBJ]LIB.MLB;1
- \$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

0
4
4

469 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:CVTFILNAM/OBJ=OBJ\$:CVTFILNAM MSRC\$:CVTFILNAM/UPDATE=(ENH\$:CVTFILNAM)+EXECML\$/LIB

0373

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY