

```

FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFFFFF
FFFFFFFFFFFFFFF
FFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF

```

```

    111
    111
    111
  111111
  111111
  111111
    111
    111
    111
    111
    111
    111
    111
    111
    111
    111
    111
    111
  111111111
  111111111
  111111111

```

```

    111
    111
    111
  111111
  111111
  111111
    111
    111
    111
    111
    111
    111
    111
    111
    111
    111
    111
    111
  111111111
  111111111
  111111111

```

```

          AAAAAAAAA
          AAAAAAAAA
          AAAAAAAAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAA          AAA
        AAA          AAA

```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```

0001 0 MODULE MAKSTR (
0002 0     LANGUAGE (BLISS32),
0003 0     IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1
0008 1 *****
0009 1 *
0010 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 *  ALL RIGHTS RESERVED.
0013 1 *
0014 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 *  TRANSFERRED.
0020 1 *
0021 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 *  CORPORATION.
0024 1 *
0025 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 **
0032 1
0033 1 FACILITY: F11ACP Structure Level 1
0034 1
0035 1 ABSTRACT:
0036 1
0037 1     This routine converts a RAD-50 file name block into the
0038 1     equivalent ASCII name string.
0039 1
0040 1 ENVIRONMENT:
0041 1
0042 1     STARLET operating system, including privileged system services
0043 1     and internal exec routines.
0044 1
0045 1 --
0046 1
0047 1
0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 3-Jan-1977 11:11
0049 1
0050 1 MODIFIED BY:
0051 1
0052 1     V02-001 ACG0186 Andrew C. Goldstein, 4-Feb-1981 21:19
0053 1     Fix garbage in high word of returned length
0054 1
0055 1     V02-000 ACG0167 Andrew C. Goldstein, 16-Apr-1980 19:27
0056 1     Previous revision history moved to f11B.REV
0057 1 **

```

MAKSTR  
V04-000

C 3  
16-Sep-1984 01:10:09  
14-Sep-1984 12:29:44

VAX-11 Bliss-32 V4.0-742  
DISK\$VMMASTER:[F11A.SRC]MAKSTR.B32;1 Page 2 (1)

```
: 58      0058 1
: 59      0059 1
: 60      0060 1 LIBRARY 'SYSSLIBRARY:LIB.L32';
: 61      0061 1 REQUIRE 'SRCS:FCPDEF.B32';
```

```

63 0376 1 GLOBAL ROUTINE MAKE_STRING (NAMEBLOCK, STRING) =
64 0377 1
65 0378 1 |++
66 0379 1
67 0380 1 FUNCTIONAL DESCRIPTION:
68 0381 1
69 0382 1     This routine converts a RAD-50 file name block into the
70 0383 1     equivalent ASCII name string.
71 0384 1
72 0385 1 CALLING SEQUENCE:
73 0386 1     MAKE_STRING (ARG1, ARG2)
74 0387 1
75 0388 1 INPUT PARAMETERS:
76 0389 1     ARG1: address of file name block
77 0390 1
78 0391 1 IMPLICIT INPUTS:
79 0392 1     NONE
80 0393 1
81 0394 1 OUTPUT PARAMETERS:
82 0395 1     ARG2: address of buffer for string
83 0396 1
84 0397 1 IMPLICIT OUTPUTS:
85 0398 1     NONE
86 0399 1
87 0400 1 ROUTINE VALUE:
88 0401 1     length of string generated
89 0402 1
90 0403 1 SIDE EFFECTS:
91 0404 1     NONE
92 0405 1
93 0406 1 |--
94 0407 1
95 0408 2 BEGIN
96 0409 2
97 0410 2 MAP
98 0411 2     NAMEBLOCK      : REF BBLOCK,      ! name block argument
99 0412 2     STRING         : REF VECTOR [,BYTE]; ! string buffer arg
100 0413 2
101 0414 2 LOCAL
102 0415 2     BLOCKP         : REF VECTOR [,WORD], ! pointer into name block
103 0416 2     STRINGP        : REF VECTOR [,BYTE], ! pointer into string
104 0417 2     CHARS          : VECTOR [3, BYTE], ! holding place for characters
105 0418 2     STRINGD        : VECTOR [2];      ! string descriptor for FAO
106 0419 2
107 0420 2 BIND
108 0421 2     FORMAT         = DESCRIPTOR ('!SW'), ! format string for FAO
109 0422 2     DELIMITER      = UPLIT BYTE (';.'),
110 0423 2     : VECTOR [,BYTE]; ! type and version delimiters
111 0424 2
112 0425 2
113 0426 2 ! Set up the pointers. Then start up the outer loop, which iterates
114 0427 2 ! over name and type fields.
115 0428 2
116 0429 2
117 0430 2 BLOCKP = NAMEBLOCK[NMBSW_NAME];
118 0431 2 STRINGP = .STRING;
119 0432 2

```

```

120 0433 2  DECR K FROM 2 TO 1 DO
121 0434 2  BEGIN
122 0435 2  :
123 0436 2  : The next loop iterates over the RAD-50 words in the name block.
124 0437 2  : There are 3 words for name, 1 for type. Expand each word into
125 0438 2  : the 3 RAD-50 characters.
126 0439 2  :
127 0440 2  :
128 0441 3  DECR I FROM (IF .K THEN 1 ELSE 3) TO 1 DO
129 0442 4  BEGIN
130 0443 4  CHARS[0] = .BLOCKP[0] / (40*40);
131 0444 4  CHARS[1] = .BLOCKP[0]/40 MOD 40;
132 0445 4  CHARS[2] = .BLOCKP[0] MOD 40;
133 0446 4  :
134 0447 4  : Now convert each character into the correct ASCII code and store it
135 0448 4  : in the string buffer if it is not null.
136 0449 4  :
137 0450 4  :
138 0451 4  INCR J FROM 0 TO 2 DO
139 0452 4  IF .CHARS[J] NEQ 0
140 0453 4  THEN
141 0454 5  BEGIN
142 0455 5  STRINGP[0] =
143 0456 6  (
144 0457 6  IF .CHARS[J] LSS 30
145 0458 6  THEN .CHARS[J] - 1 + 'A'
146 0459 6  ELSE .CHARS[J] - 30 + '0'
147 0460 5  );
148 0461 5  STRINGP = .STRINGP + 1;
149 0462 4  END;
150 0463 4  BLOCKP = .BLOCKP + 2;          ! move to next word
151 0464 4  END;                      ! end of word loop
152 0465 4  :
153 0466 3  : At the end of each field, insert the appropriate field delimiter.
154 0467 3  :
155 0468 3  :
156 0469 3  STRINGP[0] = .DELIMITER[K];
157 0470 3  STRINGP = .STRINGP + 1;
158 0471 2  END;                      ! end of outer loop
159 0472 2  :
160 0473 2  : Now build a descriptor for the remainder of the string buffer and
161 0474 2  : call FAO to convert the version number.
162 0475 2  :
163 0476 2  :
164 0477 2  STRINGD[0] = 6;
165 0478 2  STRINGD[1] = .STRINGP;
166 0479 2  $FAO (FORMAT, STRINGD[0], STRINGD[0], .BLOCKP[0]);
167 0480 2  :
168 0481 2  RETURN .STRINGP + .STRINGD[0] - .STRING;      ! final byte count
169 0482 2  :
170 0483 1  END;                      ! end of routine MAKE_STRING

```

```

.TITLE MAKSTR
.IDENT \V04-000\
.PSECT $CODE$,NOWRT,2

```

```

57 53 21 00000 P.AAB: .ASCII \.SW\
          00003 .BLKB 1
          00000003 00004 P.AAA: .LONG 3
          00000000' 00008 .ADDRESS P.AAB
2E 3B 20 0000C P.AAC: .ASCII \ ;.\

```

```

FORMAT= P.AAA
DELIMITER= P.AAC
          .EXTRN SYSS$FAO

```

			003C	00000	.ENTRY	MAKE_STRING, Save R2,R3,R4,R5	:	0376
		5E	0C	C2 00002	SUBL2	#12, SP	:	
50	04	AC	06	C1 00005	ADDL3	#6, NAMEBLOCK, BLOCKP	:	0430
		52	08	AC D0 0000A	MOVL	STRING, STRINGP	:	0431
		53	02	D0 0000E	MOVL	#2, K	:	0433
		05	53	E9 00011	1\$: BLBC	K, 2\$	:	0441
		55	01	D0 00014	MOVL	#1, R5	:	
			03	11 00017	BRB	3\$	:	
		55	03	D0 00019	2\$: MOVL	#3, R5	:	
			55	D6 0001C	3\$: INCL	1	:	
			52	11 0001E	BRB	9\$	:	
		51	60	3C 00020	4\$: MOVZWL	(BLOCKP), R1	:	0443
		51	00000640	8F C6 00023	DIVL2	#1600, R1	:	
		6E	51	90 0002A	MOVB	R1, CHARS	:	
		51	60	3C 0002D	MOVZWL	(BLOCKP), R1	:	0444
		51	28	C6 00030	DIVL2	#40, R1	:	
7E	00	51	01	7A 00033	EMUL	#1, R1, #0, -(SP)	:	
51	51	8E	28	7B 00038	EDIV	#40, (SP)+, R1, R1	:	
		01	51	90 0003D	MOVB	R1, CHARS+1	:	
		51	60	3C 00041	MOVZWL	(BLOCKP), R1	:	0445
7E	00	51	01	7A 00044	EMUL	#1, R1, #0, -(SP)	:	
51	51	8E	28	7B 00049	EDIV	#40, (SP)+, R1, R1	:	
		02	51	90 0004E	MOVB	R1, CHARS+2	:	
			51	D4 00052	CLRL	J	:	0451
		54	6E41	9A 00054	5\$: MOVZBL	CHARS[J], R4	:	0452
			11	13 00058	BEQL	8\$	:	
		1E	54	91 0005A	CMPB	R4, #30	:	0457
			06	1E 0005D	BGEQU	6\$	:	
		54	40	A4 9E 0005F	MOVAB	64(R4), R4	:	0458
			03	11 00063	BRB	7\$	:	
		54	12	C0 00065	6\$: ADDL2	#18, R4	:	0459
		82	54	90 00068	7\$: MOVB	R4, (STRINGP)+	:	0456
		51	02	F3 0006B	8\$: AOBLEQ	#2, J, 5\$	:	0452
E5		50	02	C0 0006F	ADDL2	#2, BLOCKP	:	0463
		AB	55	F5 00072	9\$: SOBGTR	1, 4\$	:	0441
		82	84	AF43 90 00075	MOVB	DELIMITER[K], (STRINGP)+	:	0469
		94	53	F5 0007A	SOBGTR	K, 1\$	:	0433
	04	AE	06	D0 0007D	MOVL	#6, STRINGD	:	0477
	08	AE	52	D0 00081	MOVL	STRINGP, STRINGD+4	:	0478
		7E	60	3C 00085	MOVZWL	(BLOCKP), -(SP)	:	0479
			08	AE 9F 00088	PUSHAB	STRINGD	:	
			0C	AE 9F 0008B	PUSHAB	STRINGD	:	
			FF63	CF 9F 0008E	PUSHAB	FORMAT	:	
	00000000G	00	04	FB 00092	CALLS	#4, SYSS\$FAO	:	
		52	04	AE C0 00099	ADDL2	STRINGD, R2	:	0481
		52	08	AC C2 0009D	SUBL2	STRING, R2	:	

50 52 D0 000A1 MOVL R2, R0  
04 000A4 RET

: 0483

: Routine Size: 165 bytes, Routine Base: \$CODE\$ + 000F

: 171 0484 1  
: 172 0485 1 END  
: 173 0486 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	180	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	3 0	1000	00:02.0

COMMAND QUALIFIERS

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

: Size: 165 code + 15 data bytes  
: Run Time: 00:07.4  
: Elapsed Time: 00:23.4  
: Lines/CPU Min: 3951  
: Lexemes/CPU-Min: 14959  
: Memory Used: 93 pages  
: Compilation Complete



The image displays a grid of 100 terminal windows, arranged in 10 rows and 10 columns. Each window contains a program name followed by the letters 'LIS'. The programs are: Row 1: REQUEL LIS, RWATTR LIS; Row 2: MOOTFY LIS; Row 3: SCHFCB LIS; Row 4: MAKACC LIS; Row 5: MPWIND LIS; Row 6: MAPUBN LIS, PMS LIS, RDHEDR LIS, RWJB LIS; Row 7: RETDIR LIS; Row 8: ROBLOK LIS; Row 9: SMALOC LIS; Row 10: MAKMBE LIS, MAKSTR LIS, MXTHOR LIS. The background of each window is dark with light-colored text and some graphical elements like bar charts.