

```
UUU      UUU      TTTTTT      IIIIIII      LLL      33333333      22222222
UUU      UUU      TTTTTT      IIIIIII      LLL      33333333      22222222
UUU      UUU      TTTTTT      IIIIIII      LLL      33333333      22222222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUU      UUU      TTT      TTT      LLL      333      333      222      222
UUUUUUUUUUUUUUUUUUUU      TTT      IIIIIII      LLLLLLLLLLLLLLLLL      333      333      222      222
UUUUUUUUUUUUUUUUUUUU      TTT      IIIIIII      LLLLLLLLLLLLLLLLL      33333333      22222222
UUUUUUUUUUUUUUUUUUUU      TTT      IIIIIII      LLLLLLLLLLLLLLLLL      33333333      22222222
```



```
RRRRRRRR      TTTTTTTTTT  BBBB BBBB
RRRRRRRR      TTTTTTTTTT  BBBB BBBB
RR      RR      TT      BB      BB
RR      RR      TT      BB      BB
RR      RR      TT      BB      BB
RR      RR      TT      BB      BB
RRRRRRRR      TT      BBBB BBBB
RRRRRRRR      TT      BBBB BBBB
RR      RR      TT      BB      BB
RR      RR      TT      BB      BB
RR      RR      TT      BB      BB
RR      RR      TT      BB      BB
RR      RR      TT      BBBB BBBB
RR      RR      TT      BBBB BBBB
```

```
....
....
....
....
```

```
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
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```



```
1 0001 0 MODULE RTB      (XTITLE 'Write RT-11 bootstrap'
2 0002 0                  MAIN = RTB
3 0003 0                  IDENT = 'V04-000'
4 0004 0                  ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 *   ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 *   TRANSFERRED.
20 0020 1 *
21 0021 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 *   CORPORATION.
24 0024 1 *
25 0025 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1
32 0032 1 ++
33 0033 1 FACILITY:
34 0034 1     System build.
35 0035 1
36 0036 1 ABSTRACT:
37 0037 1     This is a program to write RT-11 bootstrap blocks that is required to
38 0038 1     build console media.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1     VAX/VMS user mode. Sufficient privilege to execute write logical block
42 0042 1     is required for the output volume.
43 0043 1 --
44 0044 1
45 0045 1 AUTHOR: M. Jack, CREATION DATE: 06-Sep-1981
46 0046 1
47 0047 1 MODIFIED BY:
48 0048 1
49 0049 1 **
50 0050 1
51 0051 1
52 0052 1 LIBRARY 'SYSS$LIBRARY:STARLET';
53 0053 1 PSECT PLIT = $CODE$;
```



```

55 0054 1 ROUTINE RTB=
56 0055 1
57 0056 1 ++
58 0057 1
59 0058 1 FUNCTIONAL DESCRIPTION:
60 0059 1 This is the main routine for the RTB utility.
61 0060 1
62 0061 1 INPUT PARAMETERS:
63 0062 1 Standard VMS activation parameters (not used).
64 0063 1
65 0064 1 IMPLICIT INPUTS:
66 0065 1 Logical names RTB$INPUT and RTB$OUTPUT for the input file and output
67 0066 1 volume to be processed.
68 0067 1
69 0068 1 OUTPUT PARAMETERS:
70 0069 1 NONE
71 0070 1
72 0071 1 IMPLICIT OUTPUTS:
73 0072 1 NONE
74 0073 1
75 0074 1 ROUTINE VALUE:
76 0075 1 Standard VMS completion code.
77 0076 1
78 0077 1 SIDE EFFECTS:
79 0078 1 Five blocks of the input file are copied to the output volume.
80 0079 1
81 0080 1 --
82 0081 1
83 0082 2 BEGIN
84 0083 2 LOCAL
85 0084 2 BUFFER: VECTOR[512*5,BYTE], ! I/O buffer
86 0085 2 INPUT_FAB: $FAB_DECL, ! FAB used for input file lookup
87 0086 2 INPUT_CHAN: WORD, ! Channel assigned to input file
88 0087 2 OUTPUT_CHAN: WORD, ! Channel assigned to output volume
89 0088 2 STATUS, ! General status variable
90 0089 2 IOSB: VECTOR[4,WORD]; ! I/O status block
91 0090 2
92 0091 2
93 0092 2 ! Open the file referenced by the logical name RTB$INPUT, which must be
94 0093 2 equated to CONSOL.SYS in the appropriate directory.
95 0094 2
96 P 0095 2 $FAB_INIT(FAB=INPUT_FAB,
97 P 0096 2 FNA=UPLIT BYTE('RTB$INPUT:'),
98 P 0097 2 FNS=%CHARCOUNT('RTB$INPUT:'),
99 0098 2 FOP=UFO);
100 0099 2 STATUS = $OPEN(FAB=INPUT_FAB);
101 0100 2 IF NOT .STATUS THEN RETURN .STATUS;
102 0101 2 INPUT_CHAN = .INPUT_FAB[FAB$S_L_STV];
103 0102 2
104 0103 2
105 0104 2 ! Assign a channel to the volume referenced by the logical name RTB$OUTPUT,
106 0105 2 which is the console medium being built.
107 0106 2
108 0107 2 STATUS = $ASSIGN(DEVNAM=$DESCRIPTOR('RTB$OUTPUT:'), CHAN=OUTPUT_CHAN);
109 0108 2 IF NOT .STATUS THEN RETURN .STATUS;
110 0109 2
111 0110 2
```

```

: 112      0111 2 ! Read the first five blocks of the input file.
: 113      0112 2 !
: 114      P 0113 2 STATUS = $QIOW(
: 115      P 0114 2     FUNC=IOS_READVBLK,
: 116      P 0115 2     CHAN=.INPUT_CHAN,
: 117      P 0116 2     IOSB=IOSB,
: 118      P 0117 2     P1=BUFFER,
: 119      P 0118 2     P2=512*5,
: 120      0119 2     P3=1);
: 121      0120 2 IF .STATUS THEN STATUS = .IOSB[0];
: 122      0121 2 IF NOT .STATUS THEN RETURN .STATUS;
: 123      0122 2
: 124      0123 2 ! Write virtual block 1 of the input file into logical block 0 of the output
: 125      0124 2 ! volume, and write virtual blocks 2 through 5 into logical blocks 2 through 5.
: 126      0125 2 !
: 127      0126 2 !
: 128      P 0127 2 STATUS = $QIOW(
: 129      P 0128 2     FUNC=IOS_WRITEBLK,
: 130      P 0129 2     CHAN=.OUTPUT_CHAN,
: 131      P 0130 2     IOSB=IOSB,
: 132      P 0131 2     P1=BUFFER,
: 133      P 0132 2     P2=512,
: 134      0133 2     P3=0);
: 135      0134 2 IF .STATUS THEN STATUS = .IOSB[0];
: 136      0135 2 IF NOT .STATUS THEN RETURN .STATUS;
: 137      P 0136 2 STATUS = $QIOW(
: 138      P 0137 2     FUNC=IOS_WRITEBLK,
: 139      P 0138 2     CHAN=.OUTPUT_CHAN,
: 140      P 0139 2     IOSB=IOSB,
: 141      P 0140 2     P1=BUFFER+512,
: 142      P 0141 2     P2=512*4,
: 143      0142 2     P3=2);
: 144      0143 2 IF .STATUS THEN STATUS = .IOSB[0];
: 145      0144 2 IF NOT .STATUS THEN RETURN .STATUS;
: 146      0145 2
: 147      0146 2
: 148      0147 2 ! Return success.
: 149      0148 2 !
: 150      0149 2 $$$_NORMAL
: 151      0150 1 END;
```

```
.TITLE RTB Write RT-11 bootstrap
.IDENT \V04-000\
```

```
.PSECT $CODE$,NOWRT,2
```

```
3A 3A 54 55 50 4E 49 24 42 54 52 00000 P.AAA:
3A 54 55 50 54 55 4F 24 42 54 52 0000A P.AAC:
                                00015
                                0000000B 00018 P.AAB:
                                00000000 0001C
```

```
.ASCII \RTB$INPUT:\
.ASCII \RTB$OUTPUT:\
.BLKB 3
.LONG 11
.ADDRESS P.AAC
```

```
.EXTRN SYSS$OPEN, SYSS$ASSIGN
.EXTRN SYSS$QIOW
```

```
007C 00000 RTB: .WORD Save R2,R3,R4,R5,R6
```

```
: 0054
```


0050	8F	00	56 00000000G	00 9E 00002	MOVAB SYSSQIOW, R6	
			5E F5A4	CE 9E 00009	MOVAB -2652(SP), SP	
			6E	00 2C 0000E	MOVCS #0, (SP), #0, #80, \$RMS_PTR	0098
			0C	AE 00015		
			5003	8F B0 00017	MOVW #20483, \$RMS_PTR	
			00020000	8F D0 0001D	MOVL #131072, \$RMS_PTR+4	
				02 90 00025	MOVB #2, \$RMS_PTR+22	
				02 90 00029	MOVB #2, \$RMS_PTR+31	
				AF 9E 0002D	MOVAB P.AAA, \$RMS_PTR+44	
			B0	0A 90 00032	MOVB #10, \$RMS_PTR+52	
			0C	AE 9F 00036	PUSHAB INPUT_FAB	0099
			00	01 FB 00039	CALLS #1, SYSSOPEN	
			5F	50 E9 00040	BLBC STATUS, 1\$	0100
			52	18 AE B0 00043	MOVW INPUT_FAB+12, INPUT_CHAN	0101
				7E 7C 00047	CLRQ -(SP)	0107
			08	AE 9F 00049	PUSHAB OUTPUT_CHAN	
			A9	AF 9F 0004C	PUSHAB P.AAB	
			00	04 FB 0004F	CALLS #4, SYSS\$ASSIGN	
			77	50 E9 00056	BLBC STATUS, 2\$	0108
				7E 7C 00059	CLRQ -(SP)	0119
			7E	01 7D 0005B	MOVQ #1, -(SP)	
			7E	8F 3C 0005E	MOVZWL #2560, -(SP)	
			0A00	AE 9F 00063	PUSHAB BUFFER	
			70	7E 7C 00066	CLRQ -(SP)	
			24	AE 9F 00068	PUSHAB IOSB	
				31 DD 0006B	PUSHL #49	
			7E	52 3C 0006D	MOVZWL INPUT_CHAN, -(SP)	
				7E D4 00070	CLRL -(SP)	
			66	0C FB 00072	CALLS #12, SYSSQIOW	
			58	50 E9 00075	BLBC STATUS, 2\$	0120
			50	04 AE 3C 00078	MOVZWL IOSB, STATUS	
			51	50 E9 0007C	BLBC STATUS, 2\$	0121
				7E 7C 0007F	CLRQ -(SP)	0133
				7E 7C 00081	CLRQ -(SP)	
			7E	8F 3C 00083	MOVZWL #512, -(SP)	
			70	AE 9F 00088	PUSHAB BUFFER	
				7E 7C 0008B	CLRQ -(SP)	
			24	AE 9F 0008D	PUSHAB IOSB	
				20 DD 00090	PUSHL #32	
			7E	28 AE 3C 00092	MOVZWL OUTPUT_CHAN, -(SP)	
				7E D4 00096	CLRL -(SP)	
			66	0C FB 00098	CALLS #12, SYSSQIOW	
			32	50 E9 0009B	BLBC STATUS, 2\$	0134
			50	04 AE 3C 0009E	MOVZWL IOSB, STATUS	
			2B	50 E9 000A2	BLBC STATUS, 2\$	0135
				7E 7C 000A5	CLRQ -(SP)	0142
			7E	02 7D 000A7	MOVQ #2, -(SP)	
			7E	8F 3C 000AA	MOVZWL #2048, -(SP)	
			0800	CE 9F 000AF	PUSHAB BUFFER+512	
			0270	7E 7C 000B3	CLRQ -(SP)	
			24	AE 9F 000B5	PUSHAB IOSB	
				20 DD 000B8	PUSHL #32	
			7E	28 AE 3C 000BA	MOVZWL OUTPUT_CHAN, -(SP)	
				7E D4 000BE	CLRL -(SP)	
			66	0C FB 000C0	CALLS #12, SYSSQIOW	
			0A	50 E9 000C3	BLBC STATUS, 2\$	0143
			50	04 AE 3C 000C6	MOVZWL IOSB, STATUS	

RTB
V04-000

Write RT-11 bootstrap

N 11
16-Sep-1984 02:19:46
14-Sep-1984 13:25:24

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[UTIL32.SRC]RTB.B32;1

Page 5
(2)

03
50

50 E9 000CA
01 D0 000CD
04 000D0 2\$:

BLBC STATUS, 2\$
MOVL #1, R0
RET

: 0144
: 0150
:

; Routine Size: 209 bytes, Routine Base: \$CODE\$ + 0020

RTB
V04-000

Write RT-11 bootstrap

B 12
16-Sep-1984 02:19:46
14-Sep-1984 13:25:24

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[UTIL32.SRC]RTB.B32;1 Page 6 (3)

: 153
: 154
0151 1 END
0152 0 ELUDOM

PSECT SUMMARY

:
: Name Bytes Attributes
: \$CODE\$ 241 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

:
: File Total Symbols Loaded Percent Pages Mapped Processing Time
: _\$255\$DUA28:[SYSLIB]STARLET.L32;1 9776 51 0 581 00:01.0

COMMAND QUALIFIERS

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

: Size: 209 code + 32 data bytes
: Run Time: 00:06.8
: Elapsed Time: 00:11.6
: Lines/CPU Min: 1341
: Lexemes/CPU-Min: 31341
: Memory Used: 106 pages
: Compilation Complete

0429 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

