


```

LL      IIIIII  BBBB8888  RRRRRRRR  AAAAAA  RRRRRRRR  IIIIII  AAAAAA  NN      NN
LL      IIIIII  88888888  RRRRRRRR  AAAAAA  RRRRRRRR  IIIIII  AAAAAA  NN      NN
LL      II      BB      BB  RR      RR  AA      AA  RR      RR  II      AA      AA  NN      NN
LL      II      BB      BB  RR      RR  AA      AA  RR      RR  II      AA      AA  NN      NN
LL      II      BB      BB  RR      RR  AA      AA  RR      RR  II      AA      AA  NNNN     NN
LL      II      BBBB8888  RRRRRRRR  AA      AA  RRRRRRRR  II      AA      AA  NN      NN
LL      II      88888888  RRRRRRRR  AA      AA  RRRRRRRR  II      AA      AA  NN      NN
LL      II      BB      BB  RR  RR  AAAAAAAAAA  RR  RR  II      AAAAAAAAAA  NN      NNNN
LL      II      BB      BB  RR  RR  AAAAAAAAAA  RR  RR  II      AAAAAAAAAA  NN      NNNN
LL      II      BB      BB  RR      RR  AA      AA  RR      RR  II      AA      AA  NN      NN
LL      II      BB      BB  RR      RR  AA      AA  RR      RR  II      AA      AA  NN      NN
LLLLLLLLLLLL  IIIIII  88888888  RR      RR  AA      AA  RR      RR  IIIIII  AA      AA  NN      NN
LLLLLLLLLLLL  IIIIII  88888888  RR      RR  AA      AA  RR      RR  IIIIII  AA      AA  NN      NN

```

```

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
LLLLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLLLL  IIIIII  SSSSSSSS

```

```

1 0001 0 MODULE LIB_MAIN ( . Librarian command processor main routine
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000',
4 0004 0 MAIN = LIBRARIAN_START
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1
9 0009 1
10 0010 1
11 0011 1 *
12 0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
13 0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
20 0020 1 * TRANSFERRED.
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
24 0024 1 * CORPORATION.
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
28 0028 1 *
29 0029 1 *
30 0030 1
31 0031 1
32 0032 1 ++
33 0033 1
34 0034 1 FACILITY: Library command processor
35 0035 1
36 0036 1 ABSTRACT:
37 0037 1
38 0038 1 The VAX/VMS librarian is invoked by DCL to process the LIBRARY
39 0039 1 command. It utilizes the librarian procedure set to perform
40 0040 1 the actual modifications to the library.
41 0041 1
42 0042 1 ENVIRONMENT:
43 0043 1
44 0044 1 VAX native, user mode.
45 0045 1
46 0046 1 --
47 0047 1
48 0048 1
49 0049 1 AUTHOR: Benn Schreiber, CREATION DATE: 11-June-1979
50 0050 1
51 0051 1 MODIFIED BY:
52 0052 1
53 0053 1 V03-001 CWH0001 CW Hobbs 19-Aug-1982
54 0054 1 Insert a gross kludge to allow RMSDEF.MDL to be converted to
55 0055 1 RMSDEF.MSG. MSG .LITERAL directives don't make very good
56 0056 1 BLISS field extraction macros.
57 0057 1

```

LIB MAIN
V04=000

M 16
16-Sep-1984 02:00:57 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:38:06 [LIBRAR.SRC]LIBRARIAN.B32;1

Page 2
(1)

: 58 0058 1 !
: 59 0059 1 !
: 60 0060 1 !
: 61 0061 1 !--
: 62 0062 1
: 63 0063 1

V02-003 BLS0029 Benn Schreiber 23-Dec-1980
Convert to message compiler. Add library of shareable image
symbol tables.

```
: 65      0064 1 LIBRARY
: 66      0065 1 'SYSS$LIBRARY:STARLET.L32';      !System data structure definitions
: 67      0066 1 REQUIRE 'PREFIX';              !Macro defs, etc.
: 68      0067 1 REQUIRE 'LIBDEF';              !Librarian command interface definitions
: 69      0251 1 REQUIRE 'LIBDEF';
: 70      0252 1
: 71      0540 1
: 72      0541 1 EXTERNAL ROUTINE
: 73      0542 1     lib_get_command,           !Get a command line
: 74      0543 1     lib_process_cmd;          !Process command line
: 75      0544 1
: 76      0545 1 FORWARD ROUTINE
: 77      0546 1     lib_handler;              !Handler to pick up highest severity
: 78      0547 1
: 79      0548 1 OWN
: 80      0549 1     exitvalue : BLOCK [1] INITIAL !Value to exit with
: 81      0550 1     (ss$_normal);
```

```

: 83 0551 1 GLOBAL ROUTINE LIBRARIAN_START (arglist) : NOVALUE =
: 84 0552 2 BEGIN
: 85 0553 2
: 86 0554 2 |++
: 87 0555 2 |
: 88 0556 2 | FUNCTIONAL DESCRIPTION:
: 89 0557 2 |
: 90 0558 2 |     This is the entry point to the librarian command processor.  It calls
: 91 0559 2 |     lib_get_command to get the command line and then processes it.
: 92 0560 2 |
: 93 0561 2 |
: 94 0562 2 | CALLING SEQUENCE:
: 95 0563 2 |
: 96 0564 2 |     librarian_start(arglist)
: 97 0565 2 |
: 98 0566 2 | INPUT PARAMETERS:
: 99 0567 2 |
: 100 0568 2 |     arglist           is the argument list passed from CLI
: 101 0569 2 |
: 102 0570 2 | IMPLICIT INPUTS:
: 103 0571 2 |     NONE
: 104 0572 2 |
: 105 0573 2 | OUTPUT PARAMETERS:
: 106 0574 2 |     NONE
: 107 0575 2 |
: 108 0576 2 | IMPLICIT OUTPUTS:
: 109 0577 2 |     NONE
: 110 0578 2 |
: 111 0579 2 | ROUTINE VALUE:
: 112 0580 2 |     NONE
: 113 0581 2 |
: 114 0582 2 | SIDE EFFECTS:
: 115 0583 2 |     NONE
: 116 0584 2 |
: 117 0585 2 | --
: 118 0586 2 | LOCAL
: 119 0587 2 |     status;
: 120 0588 2 |
: 121 0589 2 | BUILTIN
: 122 0590 2 |     FP;
: 123 0591 2 |
: 124 0592 2 | .FP = lib_handler;           !Set handler
: 125 0593 2 | IF lib_get_command (arglist-4) !Parse the command
: 126 0594 2 |     THEN lib_process_cmd ();   !And process it
: 127 0595 2 |
: 128 0596 2 | RETURN (status = .exitvalue OR sts$m_inhib_msg); !Exit with no message
: 129 0597 1 | END;                          !Of librarian_start

```

```

.TITLE LIB MAIN
.IDENT \V04-000\

.PSECT $OWNS,NOEXE,2

00000001 00000 EXITVALUE:
.LONG 1

```

LIB_MAIN
V04-000

K 16
16-Sep-1984 02:00:57 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:38:06 [LIBRAR.SRC]LIBRARIAN.B32;1

Page 5
(3)

```

                                .EXTRN LIB_GET_COMMAND
                                .EXTRN LIB_PROCESS_CMD
                                .PSECT $CODE$,NOWRT,2
                                .ENTRY LIBRARIAN_START, Save nothing
0000 00000                                MOVAB LIB_HANDLER, (FP)
6D 0000V CF 9E 00002                                PUSH: AP
                                5C DD 00007                                : 0593
0000G CF 01 FB 00009                                CALLS #1, LIB_GET_COMMAND
05 50 E9 0000E                                BLBC R0, 1$
0000G CF 00 FB 00011                                CALLS #0, LIB_PROCESS_CMD
50 0000' CF 10000000 8F C9 00016 1$: BISL3 #268435456, EXITVALUE, STATUS
                                04 00020                                RET
                                : 0594
                                : 0596
                                : 0597
```

; Routine Size: 33 bytes, Routine Base: \$CODE\$ + 0000

```
131 0598 1 ROUTINE lib_handler (sigargs, mechargs) =
132 0599 2 BEGIN
133 0600 2
134 0601 2 : This routine is a condition handler called when ever a
135 0602 2 : SIGNAL is done by the librarian. It merely remembers the
136 0603 2 : most severe error for an exit status.
137 0604 2
138 0605 2 MAP
139 0606 2 sigargs : REF BBLOCK,
140 0607 2 mechargs : REF BBLOCK;
141 0608 2
142 0609 2 BIND
143 0610 2 signame = sigargs [chf$l_sig_name] : BBLOCK; !Name of signal
144 0611 2
145 0612 2 IF NOT .signame !If its an error signal
146 0613 4 AND ((.signame [sts$v_severity] ! and severity is worse than it was
147 0614 4 GEQU .exitvalue [sts$v_severity])
148 0615 4 OR .exitvalue [sts$v_severity]) ! or we haven't had any errors
149 0616 3 THEN exitvalue = .signame; ! then remember it for exiting
150 0617 2
151 0618 2 IF .signame [sts$v_severity] EQL sts$k_error
152 0619 2 OR .signame [sts$v_severity] EQL sts$k_severe
153 0620 3 THEN BEGIN
154 0621 3 sigargs[chf$l_sig_args] = .sigargs[chf$l_sig_args] - 2; !Decrement PC/PSL pair
155 0622 3 $putmsg(msgvec=sigargs[chf$l_sig_args]); !Output the message
156 0623 3 mechargs [chf$l_mch_savr0] = .signame OR sts$m_inhib_msg; !Set inhib_msg in return R0
157 0624 3
158 0625 3 : Temporary code to set inhib_msg in all messages in the signal vector
159 0626 3
160 0627 4 BEGIN
161 0628 4 LOCAL
162 0629 4 ptr;
163 0630 4
164 0631 4 BIND
165 0632 4 sigargvec = signame : VECTOR[,LONG];
166 0633 4
167 0634 4 ptr = 0;
168 0635 4 WHILE .ptr LSSU .sigargs[chf$l_sig_args] !While there are more args
169 0636 5 DO BEGIN
170 0637 5 BIND
171 0638 5 cursigname = sigargvec[.ptr] : BBLOCK; !Name the current message
172 0639 5
173 0640 5 sigargvec[.ptr] = .sigargvec[.ptr] OR sts$m_inhib_msg; !Set inhib in current msg
174 0641 5 IF .cursigname[sts$v_fac_no] LEQ 1 !If this is RMS message (or system message)
175 0642 6 THEN BEGIN
176 0643 6
177 0644 6 !+
178 0645 6 GROSS KLUDGE - When RMSDEF.MDL was converted to RMSDEF.MSG, the symbol
179 0646 6 RMSSV_STVSTATUS was defined as a literal - this works for MACRO-32, but
180 0647 6 BLISS needs more. We therefore give BLISS what it wants. When the MESSAGE/SDL
181 0648 6 is made smarter this will die with another obscure message.
182 0649 6 -
183 0650 6 IF .cursigname[0,rms$v_stvstatus,1,0] ! with an STV status
184 0651 7 THEN BEGIN
185 0652 7 sigargvec[.ptr+1] = .sigargvec[.ptr+1] OR sts$m_inhib_msg; !Set inhib in it
186 0653 7 ptr = .ptr + 2; !Skip two longwords
187 0654 7 END
```



```

: 188      0655 6      ELSE ptr = .ptr + 1;           !Otherwise just skip one
: 189      0656 6      END
: 190      0657 5      ELSE ptr = .ptr + .sigargvec[.ptr+1] + 2; !Not RMS, skip msg + count + lw of count
: 191      0658 4      END;
: 192      0659 3      END;           !Of hack to set inhib_msg
: 193      0660 3      sigargs[chf$l_sig_args] = .sigargs[chf$l_sig_args] + 2;
: 194      0661 3      IF .signame [sts$V_severity] EQL sts$k_error
: 195      0662 4      THEN BEGIN
: 196      0663 4          SETUNWIND(%REF(1));           !Unwind to caller
: 197      0664 4          RETURN ss$_continue
: 198      0665 4      END
: 199      0666 3      ELSE RETURN ss$_resignal
: 200      0667 3      END
: 201      0668 3
: 202      0669 2      ELSE RETURN ss$_resignal;           !Resignal to get the error printed
: 203      0670 1      END;           !Of lib_handler

```

.EXTRN SYS\$PUTMSG

```

                                003C 00000 LIB_HANDLER:
                                .WORD      Save R2,R3,R4,R5           : 0598
                                55      0000'  CF  9E 00002      MOVAB      EXITVALUE, R5
                                5E      04      04  C2 00007      SUBL2      #4, SP
                                53      04      AC  D0 0000A      MOVL      SIGARGS, R3           : 0610
                                52      04      A3  9E 0000E      MOVAB      4(R3), R2
                                12      62      E8 00012      BLBS      (R2), 2$           : 0612
50      65      03      00  EF 00015      EXTZV     #0, #3, EXITVALUE, R0
50      62      03      00  ED 0001A      CMPZV     #0, #3, (R2), R0           : 0614
                                03      03      1E 0001F      BGEQU     1$
                                03      65      E9 00021      BLBC      EXITVALUE, 2$           : 0615
                                65      62      D0 00024 1$:      MOVL      (R2), EXITVALUE           : 0616
                                54      D4 00027 2$:      CLRL     R4           : 0618
02      62      03      00  ED 00029      CMPZV     #0, #3, (R2), #2
                                04      12 0002E      BNEQ     3$
                                54      D6 00030      INCL     R4
                                07      11 00032      BRB      4$
04      62      03      00  ED 00034 3$:      CMPZV     #0, #3, (R2), #4           : 0619
                                6F      12 00039      BNEQ     9$
                                63      02  C2 0003B 4$:      SUBL2     #2, (R3)           : 0621
                                7E      7C 0003E      CLRQ     -(SP)           : 0622
                                7E      D4 00040      CLRL     -(SP)
                                53      DD 00042      PUSHL    R3
                                00      04  FB 00044      CALLS    #4, SYS$PUTMSG
                                50      08      AC  D0 0004B      MOVL     MECHARGS, R0           : 0623
0C      A0      62 10000000 8F  C9 0004F      BISL3    #268435456, (R2), 12(R0)
                                50      D4 00058      CLRL     PTR           : 0634
                                63      50  D1 0005A 5$:      CMPL     PTR, (R3)           : 0635
                                31      1E 0005D      BGEQU     8$
                                03 A240 10  C8 0005F      BISL2    #16, 3(R2)[PTR]           : 0640
                                02 A240 DF 00064      PUSHAL   2(R2)[PTR]           : 0641
01      9E      0C      00  ED 00068      CMPZV     #0, #12, @(SP)+, #1
                                15      14 0006D      BGTR     7$
                                6240 DF 0006F      PUSHAL   (R2)[PTR]           : 0650
                                0A      9E      0E  E1 00072      BBC      #14, @(SP)+, 6$
                                07 A240 10  C8 00076      BISL2    #16, 7(R2)[PTR]           : 0652

```

```

50          02 C0 0007B      ADDL2  #2, PTR      : 0653
           DA 11 0007E      BRB    5$         : 0650
           50 D6 00080 6$:  INCL  PTR         : 0655
           D6 11 00082      BRB    5$         : 0641
51          04 A240 DC 00084 7$:  MOVL  4(R2)[PTR], R1 : 0657
50          02 A140 9E 00089  MOVL  2(R1)[PTR], PTR
           CA 11 0008E      BRB    5$         : 0635
63          02 C0 00090 8$:  ADDL2  #2, (R3)    : 0660
14          54 E9 00093      BLBC   K4, 9$     : 0661
           7E D4 00096      CLRL  -(SP)      : 0663
04 AE          01 D0 00098      MOVL  #1, 4(SP)
           04 AE 9F 0009C      PUSHAB 4(SP)
00000000G 00          02 FB 0009F      CALLS #2, SYSSUNWIND
50          01 D0 000A6      MOVL  #1, R0     : 0666
           04 000A9      RET
50          0918 8F 3C 000AA 9$:  MOVZWL #2328, R0 : 0669
           04 000AF      RET                    : 0670

```

: Routine Size: 176 bytes, Routine Base: \$CODE\$ + 0021

```

: 204          0671 1
: 205          0672 1 END      ! Of module
: 206          0673 0 ELUDOM

```

.EXTRN SYSSUNWIND

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	4	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	209	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	36	0	581	00:01.0

COMMAND QUALIFIERS

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

LIB MAIN
V04=000

C 1
16-Sep-1984 02:00:57

VAX-11 Bliss-32 V4.0-742

Page 9

LI
VO

: Size: 209 code + 4 data bytes
: Run Time: 00:11.5
: Elapsed Time: 00:23.8
: Lines/CPU Min: 3523
: Lexemes/CPU-Min: 43659
: Memory Used: 117 pages
: Compilation Complete

41
6F

20
21

