


```

LL      IIIIII  BBBB8888  DDDDDDDD  000000  CCCCCCCC  000000  MM      MM
LL      IIIIII  88888888  DDDDDDDD  000000  CCCCCCCC  000000  MM      MM
LL      II      BB      BB  DD      DD  00      00  CC      CC  00      00  MMMM  MMMM
LL      II      BB      BB  DD      DD  00      00  CC      CC  00      00  MMMM  MMMM
LL      II      BB      BB  DD      DD  00      00  CC      CC  00      00  MM   MM  MM
LL      II      BB      BB  DD      DD  00      00  CC      CC  00      00  MM   MM  MM
LL      II      BBB88888  DD      DD  00      00  CC      CC  00      00  MM      MM
LL      II      88888888  DD      DD  00      00  CC      CC  00      00  MM      MM
LL      II      BB      BB  DD      DD  00      00  CC      CC  00      00  MM      MM
LL      II      BB      BB  DD      DD  00      00  CC      CC  00      00  MM      MM
LL      II      BB      BB  DD      DD  00      00  CC      CC  00      00  MM      MM
LL      II      BB      BB  DD      DD  00      00  CC      CC  00      00  MM      MM
LLLLLLLLLLLL IIIIII  88888888  DDDDDDDD  000000  CCCCCCCC  000000  MM      MM
LLLLLLLLLLLL IIIIII  88888888  DDDDDDDD  000000  CCCCCCCC  000000  MM      MM

```

```

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$DO_COMMAND ( ! Perform a command
2 0002 0
3 0003 0 IDENT = '1-008' ! File: LIBDOCOM.B32 Edit: SBL1008
4 0004 0
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
12 0012 1 * ALL RIGHTS RESERVED. *
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
19 0019 1 * TRANSFERRED. *
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
23 0023 1 * CORPORATION. *
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1 **
33 0033 1 FACILITY: General library
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This module causes the curren program to end and a command
38 0038 1 to be executed.
39 0039 1
40 0040 1 ENVIRONMENT: VAX-11 User Mode
41 0041 1
42 0042 1 AUTHOR: John Sauter, CREATION DATE: 13-NOV-1979
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 1-001 - Original. JBS 26-NOV-1979
47 0047 1 1-002 - Correct entry point name. JBS 27-NOV-1979
48 0048 1 1-003 - Change STR$ codes to LIB$ codes. JBS 22-JAN-1980
49 0049 1 1-004 - Restrict command string to 132 bytes, since that's all that
50 0050 1 DCL can handle. SBL 27-Jan-1981
51 0051 1 1-005 - Remove use of process common, unnecessary. SBL 27-Jan-1981
52 0052 1 1-006 - Enhance to recognize additional classes of string descriptors
53 0053 1 by invoking LIB$ANALYZE_SDESC_R3 to extract length and
54 0054 1 address of 1st data byte of descriptor. RKR 27-MAY-1981.
55 0055 1 1-007 - Redirect jsb's from LIB$ANALYZE_SDESC_R3 to
56 0056 1 LIB$ANALYZE_SDESC_R2. RKR 18-NOV-1981
57 0057 1 1-008 - Use CLISC_SRVDESC. Report LIB$_NOCLI and LIB$_UNECLIERR when

```

LIB\$DO_COMMAND
1-008

J 16
16-Sep-1984 00:49:42
14-Sep-1984 12:38:40

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]LIBDOCOM.B32;1

Page 2
(1)

```
: 58      0058 1 !          necessary. SBL 18-Dec-1981
: 59      0059 1 ! --
: 60      0060 1 !
: 61      0061 1 ! <BLF/PAGE>
```

```

: 63      0062 1  |
: 64      0063 1  | SWITCHES:
: 65      0064 1  |
: 66      0065 1  |
: 67      0066 1  | SWITCHES ADDRESSING MODE
: 68      0067 1  |           (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
: 69      0068 1  |
: 70      0069 1  |
: 71      0070 1  | LINKAGES:
: 72      0071 1  |
: 73      0072 1  |     REQUIRE 'RTLIN:STRLNK';      ! Linkage for LIB$ANALYZE_SDESC_R2
: 74      0257 1  |
: 75      0258 1  | TABLE OF CONTENTS:
: 76      0259 1  |
: 77      0260 1  |
: 78      0261 1  | FORWARD ROUTINE
: 79      0262 1  |     LIB$DO_COMMAND;             ! End this program and do a command
: 80      0263 1  |
: 81      0264 1  |
: 82      0265 1  | INCLUDE FILES:
: 83      0266 1  |
: 84      0267 1  |
: 85      0268 1  | REQUIRE 'RTLIN:RTLPSECT';       ! Macros for defining PSECTS
: 86      0363 1  |
: 87      0364 1  | LIBRARY 'RTLSTARLE';           ! System symbols
: 88      0365 1  |
: 89      0366 1  | REQUIRE 'RTLML:CLIMSG';       ! CLIS_ messages
: 90      0646 1  |
: 91      0647 1  |
: 92      0648 1  | MACROS:
: 93      0649 1  |
: 94      0650 1  |     NONE
: 95      0651 1  |
: 96      0652 1  | EQUATED SYMBOLS:
: 97      0653 1  |
: 98      0654 1  |     NONE
: 99      0655 1  |
: 100     0656 1  | PSECTS:
: 101     0657 1  |
: 102     0658 1  | DECLARE_PSECTS (LIB);         ! Declare psects for LIB$ facility
: 103     0659 1  |
: 104     0660 1  | OWN STORAGE:
: 105     0661 1  |
: 106     0662 1  |     NONE
: 107     0663 1  |
: 108     0664 1  | EXTERNAL REFERENCES:
: 109     0665 1  |
: 110     0666 1  |
: 111     0667 1  | EXTERNAL ROUTINE
: 112     0668 1  |     LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_JSB_LINK, ! Extract length
: 113     0669 1  |                                                                | and address of
: 114     0670 1  |                                                                | 1st data byte
: 115     0671 1  |                                                                | from descrip.
: 116     0672 1  |     SYSSCLI;                 ! Call the command language interpreter
: 117     0673 1  |
: 118     0674 1  |
: 119     0675 1  | !+
```

```
: 120      0676 1 ! The following are the error codes used in this module.
: 121      0677 1 !-
: 122      0678 1
: 123      0679 1 EXTERNAL LITERAL
: 124      0680 1     LIB$_INVARG,      ! Invalid argument
: 125      0681 1     LIB$_NOCLI,      ! No CLI to perform function
: 126      0682 1     LIB$_UNECLERR;    ! Unexpected CLI error
: 127      0683 1
```

```
129 0684 1 GLOBAL ROUTINE LIBSDO_COMMAND (      ! Do a command
130 0685 1
131 0686 1     CMD_TEXT      ! Text of command to execute
132 0687 1
133 0688 1     ) =
134 0689 1
135 0690 1 !++
136 0691 1 FUNCTIONAL DESCRIPTION:
137 0692 1
138 0693 1     Stop running this program and do a command.
139 0694 1
140 0695 1 CALLING SEQUENCE:
141 0696 1
142 0697 1     status.wlc.v = LIBSDO_COMMAND (CMD_TEXT.rt.dx)
143 0698 1
144 0699 1 FORMAL PARAMETERS:
145 0700 1
146 0701 1     CMD_TEXT.rt.dx The text of the command to be executed.
147 0702 1                 max of 132 bytes.
148 0703 1
149 0704 1 IMPLICIT INPUTS:
150 0705 1
151 0706 1     NONE
152 0707 1
153 0708 1 IMPLICIT OUTPUTS:
154 0709 1
155 0710 1     NONE
156 0711 1
157 0712 1 COMPLETION CODES:
158 0713 1
159 0714 1     LIB$_INVARG if string is longer than 132 bytes
160 0715 1                 or error in CMD_TEXT descriptor.
161 0716 1
162 0717 1 SIDE EFFECTS:
163 0718 1
164 0719 1     Does not normally return to its caller.
165 0720 1
166 0721 1 !--
167 0722 1
168 0723 2 BEGIN
169 0724 2
170 0725 2 MAP
171 0726 2     CMD_TEXT : REF BLOCK [8, BYTE];
172 0727 2
173 0728 2 LOCAL
174 0729 2     RET_STATUS: BLOCK [4, BYTE],      ! Status returned
175 0730 2     CLI_REQ_BLOCK : BLOCK [CLI$_SRVDESC, BYTE], ! CLI service block
176 0731 2     DESC_ADDR : REF BLOCK [8, BYTE],    ! Pointer to descriptor
177 0732 2                                     ! in req block
178 0733 2     CMD_TEXT_LEN,                       ! Length of the command
179 0734 2                                     ! line provided.
180 0735 2     CMD_TEXT_ADDR;                     ! Address of 1st data
181 0736 2                                     ! byte in command line
182 0737 2                                     ! provided.
183 0738 2
184 0739 2 !+
185 0740 2 ! Get length and address of 1st data byte in command line provided.
```

```
186 0741 2 ! If descriptor invalid, return the status returned by
187 0742 2 ! LIB$ANALYZE_SDESC_R2.
188 0743 2 !-
189 0744 3     IF NOT (RET_STATUS = LIB$ANALYZE_SDESC_R2 ( .CMD_TEXT :
190 0745 3         CMD_TEXT_LEN,
191 0746 3         CMD_TEXT_ADDR ) )
192 0747 3     THEN RETURN (.RET_STATUS) ;
193 0748 2
194 0749 2 !+
195 0750 2 ! Don't allow too long a string.
196 0751 2 !-
197 0752 2     IF (.CMD_TEXT_LEN GTRU 132) THEN RETURN (LIB$_INVARG);
198 0753 2
199 0754 2 !+
200 0755 2 ! Initialize CLI command request block.
201 0756 2 !-
202 0757 2     CH$FILL (0, CLIS$SRVDESC, CLI_REQ_BLOCK);
203 0758 2     CLI_REQ_BLOCK [CLIS$B_RQTYPE] = CLIS$_CLISERV;           ! Request a
204 0759 2     CLI_REQ_BLOCK [CLIS$W_SERVCOD] = CLIS$_COMMAND;         ! service
205 0760 2                                                         ! Service is
206 0761 2                                                         ! "pass command
207 0762 2                                                         ! line"
208 0763 2     DESC_ADDR = CLI_REQ_BLOCK [CLIS$Q_RQDESC];
209 0764 2
210 0765 2 !+
211 0766 2 ! Fill in the descriptor
212 0767 2 !-
213 0768 2     DESC_ADDR [DSC$W_LENGTH] = .CMD_TEXT_LEN;
214 0769 2     DESC_ADDR [DSC$A_POINTER] = .CMD_TEXT_ADDR;
215 0770 2
216 0771 2     RET_STATUS = SYSS$CLI (CLI_REQ_BLOCK);
217 0772 2
218 0773 2     IF NOT .RET_STATUS
219 0774 2     THEN
220 0775 3     BEGIN
221 0776 3     IF .RET_STATUS [STSS$V_FAC_NO] EQL CLIS$_FACILITY
222 0777 3     THEN
223 0778 4     BEGIN
224 0779 4     IF .RET_STATUS EQLU CLIS$_INVREQTYP
225 0780 4     THEN
226 0781 4     RETURN LIB$_NOCLI
227 0782 4     ELSE
228 0783 4     RETURN LIB$_UNECLIERR;
229 0784 4     END
230 0785 3     ELSE
231 0786 3     RETURN .RET_STATUS;
232 0787 2     END;
233 0788 2
234 0789 2 !+
235 0790 2 ! Now, to get the command to be executed, we must exit.
236 0791 2 !-
237 0792 2     $EXIT (CODE = SSS$_NORMAL);
238 0793 2 !+
239 0794 2 ! Its not likely that we'll get back from a call to $EXIT, but BLISS
240 0795 2 ! doesn't know that, so it insists that we return a value.
241 0796 2 !-
242 0797 2
```


: 243 0798 2 RETURN (0);
: 244 0799 1 END;

! end of LIB\$DO_COMMAND

.TITLE LIB\$DO_COMMAND
.IDENT \1-008
.EXTRN LIB\$ANALYZE_SDESC_R2
.EXTRN SYSSCLI, LIB\$INVARG
.EXTRN LIB\$NOCLI, LIB\$UNECLIERR
.EXTRN SYSSEXIT
.PSECT _LIB\$CODE, NOWRT, SHR, PIC, 2

.ENTRY LIB\$DO_COMMAND, Save R2,R3,R4,R5,R6,R7,R8 : 0684
MOVAB -84(SP), SP : 0744
MOVL CMD_TEXT, R0
JSB LIB\$ANALYZE_SDESC_R2
MOVQ R0, RET_STATUS
MOVL R2, R8
BLBC RET_STATUS, 3\$: 0752
CML CMD_TEXT_LEN, #132
BLEQU 1\$
MOVL #LIB\$INVARG, R0
RET
MOVCS #0, (SP), #0, #84, CLI_REQ_BLOCK : 0757
MOVAB #5, CLI_REQ_BLOCK : 0758
MOVW #5, CLI_REQ_BLOCK+1 : 0760
MOVAB CLI_REQ_BLOCK+8, DESC_ADDR : 0763
MOVW CMD_TEXT_LEN, (DESC_ADDR) : 0768
MOVL CMD_TEXT_ADDR, 4(DESC_ADDR) : 0769
PUSHL SP : 0771
CALLS #1, SYSSCLI
MOVL R0, RET_STATUS
BLBS RET_STATUS, 4\$: 0773
CMPZV #16, #12, RET_STATUS, #3 : 0776
BNEQ 3\$
CML RET_STATUS, #231458 : 0779
BNEQ 2\$
MOVL #LIB\$NOCLI, R0 : 0783
RET
MOVL #LIB\$UNECLIERR, R0 : 0786
RET
MOVL RET_STATUS, R0 : 0792
RET
PUSHL #1 : 0792
CALLS #1, SYSSEXIT : 0798
CLRL R0 : 0799
RET

01FC 00000
5E AC AE 9E 00002
50 04 AC D0 00006
00000000G 00 16 0000A
56 50 7D 00010
58 52 D0 00013
5A 56 E9 00016
00000084 8F 57 D1 00019
08 1B 00020
50 00000000G 8F D0 00022
04 00029
0054 8F 00 6E 00 2C 0002A 1\$:
6E 00031
01 6E 05 90 00032
AE 05 B0 00035
50 08 AE 9E 00039
60 57 B0 0003D
04 A0 58 D0 00040
5E DD 00044
00000000G 00 01 FB 00046
56 50 D0 0004D
24 56 E8 00050
03 0C 10 ED 00053
00038822 8F 19 12 00058
56 D1 0005A
08 12 00061
50 00000000G 8F D0 00063
04 0006A
50 00000000G 8F D0 0006B 2\$:
04 00072
50 56 D0 00073 3\$:
04 00076
00000000G 00 01 DD 00077 4\$:
01 FB 00079
50 D4 00080
04 00082

: Routine Size: 131 bytes, Routine Base: _LIB\$CODE + 0000

: 245 0800 1
: 246 0801 1 END
: 247 0802 1

! end of module LIB\$DO_COMMAND

LIB\$DO_COMMAND
1-008

D 1
16-Sep-1984 00:49:42
14-Sep-1984 12:38:40

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]LIBDOCOM.B32;1

Page 8
(3)

: 248 0803 0 ELUDOM

PSECT SUMMARY

: Name Bytes Attributes
: _LIB\$CODE 131 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

: File Total Symbols Loaded Percent Pages Mapped Processing Time
: _\$255\$DUA28:[SYSLIB]STARLET.L32;1 9776 16 0 581 00:00.8

COMMAND QUALIFIERS

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

: Size: 131 code + 0 data bytes
: Run Time: 00:09.7
: Elapsed Time: 00:41.5
: Lines/CPU Min: 4956
: Lexemes/CPU-Min: 97037
: Memory Used: 103 pages
: Compilation Complete

0206 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

This image displays a grid of 100 small technical diagrams or code snippets, arranged in 10 rows and 10 columns. Each diagram is a small-scale representation of a system component or interface, often featuring a title and some graphical elements like bars or lines. The labels for these diagrams are as follows:

- Row 1: LIBEMODH LIS
- Row 2: LIBEMODU LIS
- Row 3: LIBEF LIS
- Row 4: LIBEMULAT LIS
- Row 5: LIBBFFS LIS
- Row 6: LIBFINCUT LIS
- Row 7: LIBE2AREV LIS
- Row 8: LIBEMODG LIS
- Row 9: LIBEXTV LIS
- Row 10: LIBFAO LIS
- Row 11: LIBEDIV LIS
- Row 12: LIBSTABL LIS
- Row 13: LIBFFC LIS
- Row 14: LIBEMODF LIS
- Row 15: LIBEMUL LIS
- Row 16: LIBFILSCA LIS
- Row 17: LIBEXTZU LIS
- Row 18: LIBEBCASC LIS
- Row 19: LIBFAOL LIS