



```

SSSSSSSS  CCCCCCCC  RRRRRRRR  IIIIII  NN  NN  PPPPPPPP  UU  UU  TTTTTTTTTT
SSSSSSSS  CCCCCCCC  RRRRRRRR  IIIIII  NN  NN  PPPPPPPP  UU  UU  TTTTTTTTTT
SS  CC  RR  RR  RR  II  II  PP  PP  UU  UU  TT
SS  CC  RR  RR  RR  II  II  PP  PP  UU  UU  TT
SS  CC  RR  RR  RR  II  II  PP  PP  UU  UU  TT
SSSSSS  CC  RRRRRRRR  II  II  NN  NN  PPPPPPPP  UU  UU  TT
SSSSSS  CC  RRRRRRRR  II  II  NN  NN  PPPPPPPP  UU  UU  TT
SS  CC  RR  RR  RR  II  II  NN  NN  PP  UU  UU  TT
SS  CC  RR  RR  RR  II  II  NN  NN  PP  UU  UU  TT
SS  CC  RR  RR  RR  II  II  NN  NN  PP  UU  UU  TT
SSSSSSSS  CCCCCCCC  RR  RR  IIIIII  NN  NN  PP  UU  UU  TT
SSSSSSSS  CCCCCCCC  RR  RR  IIIIII  NN  NN  PP  UU  UU  TT
UUUUUUUUUU  TT  ....
UUUUUUUUUU  TT  ....

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 %TITLE 'SCR$INPUT - Input routines for the screen package'  
2 0002 0 MODULE SCR$INPUT (  
3 0003 0 IDENT = 'V04-000' ! File: SCRINPUT.B32 Edit: PLL1001  
4 0004 0 ) =  
5 0005 1 BEGIN  
6 0006 1  
7 0007 1 *****  
8 0008 1 *  
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *  
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *  
11 0011 1 * ALL RIGHTS RESERVED. *  
12 0012 1 *  
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *  
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *  
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *  
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *  
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *  
18 0018 1 * TRANSFERRED. *  
19 0019 1 *  
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *  
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *  
22 0022 1 * CORPORATION. *  
23 0023 1 *  
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *  
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *  
26 0026 1 *  
27 0027 1 *  
28 0028 1 *****  
29 0029 1  
30 0030 1  
31 0031 1 **  
32 0032 1 FACILITY: Screen Management  
33 0033 1  
34 0034 1 ABSTRACT:  
35 0035 1  
36 0036 1 This module contains routines which perform input from the screen  
37 0037 1 via keyboard or keypad.  
38 0038 1  
39 0039 1 ENVIRONMENT: User mode, Shared library routines.  
40 0040 1  
41 0041 1 AUTHOR: P. Levesque, CREATION DATE: 18-Oct-1982  
42 0042 1  
43 0043 1 MODIFIED BY:  
44 0044 1  
45 0045 1 1-001 - Original. PLL 18-Oct-1982  
46 0046 1 --  
47 0047 1
```

```

49 0048 1 %SBTTL 'Declarations'
50 0049 1
51 0050 1 : SWITCHES:
52 0051 1 :
53 0052 1
54 0053 1 :
55 0054 1 : LINKAGES:
56 0055 1
57 0056 1 : NONE
58 0057 1
59 0058 1 : TABLE OF CONTENTS:
60 0059 1 :
61 0060 1
62 0061 1 FORWARD ROUTINE
63 0062 1
64 0063 1 SCR$GET_SCREEN; ! Get text from screen
65 0064 1
66 0065 1 :
67 0066 1 : The following is defined via a GLOBAL BIND.
68 0067 1 :
69 0068 1 : LIB$GET_SCREEN = SCR$GET_SCREEN
70 0069 1 :
71 0070 1
72 0071 1 :
73 0072 1 : INCLUDE FILES
74 0073 1 :
75 0074 1
76 0075 1 REQUIRE 'SRC$:SCRPROLOG'; ! defines psects, macros, tcb,
77 0150 1 ! wcb, & terminal symbols
78 0151 1 :
79 0152 1 : EXTERNAL REFERENCES
80 0153 1 :
81 0154 1
82 0155 1 EXTERNAL ROUTINE
83 0156 1 SCR$$GET_TYPE_R3 : GET_TYPE_LINK, ! Get device type
84 0157 1 LIB$GET_INPUT; ! Get a line of text from SY$$INPUT.
85 0158 1
86 0159 1 EXTERNAL
87 0160 1 SCR$L_FLINKHEAD, ! Head of chain of TCB's
88 0161 1 SCR$L_CUROUTPUT; ! Pointer to current TCB
89 0162 1
90 0163 1 !<BLF/PAGE>

```

```

: 92 0164 1 XSBTTL 'LIB$GET_SCREEN - Get Text from Screen'
: 93 0165 1 GLOBAL ROUTINE LIB$GET_SCREEN (
: 94 0166 1     INPUT_TEXT : REF BLOCK [,BYTE],
: 95 0167 1     PROMPT_STR : REF BLOCK [,BYTE],
: 96 0168 1     OUT_LEN   : REF VECTOR [,WORD]
: 97 0169 1 ) =
: 98 0170 1 ++
: 99 0171 1 FUNCTIONAL DESCRIPTION:
100 0172 1
101 0173 1     This routine accepts input from the terminal into a
102 0174 1     user-specified buffer. An optional string may be given to be
103 0175 1     used for a prompt. If a third parameter is present, it will
104 0176 1     be set to the length of the string read.
105 0177 1
106 0178 1 CALLING SEQUENCE:
107 0179 1
108 0180 1     ret_status.wlc.v = LIB$GET_SCREEN (INPUT_TEXT.wt.dx
109 0181 1     [,PROMPT_STR.rt.dx
110 0182 1     [,OUT_LEN.wwu.r]])
111 0183 1
112 0184 1 FORMAL PARAMETERS:
113 0185 1
114 0186 1     INPUT_TEXT.wt.dx     Address of buffer descriptor.
115 0187 1
116 0188 1     PROMPT_STR.rt.dx    Optional. Address of descriptor of
117 0189 1     prompt string.
118 0190 1
119 0191 1     OUT_LEN.wwu.r       Optional. Length of input string read.
120 0192 1
121 0193 1 IMPLICIT INPUTS:
122 0194 1
123 0195 1     NONE
124 0196 1
125 0197 1 IMPLICIT OUTPUTS:
126 0198 1
127 0199 1     NONE
128 0200 1
129 0201 1 COMPLETION STATUS:
130 0202 1
131 0203 1     SSS_NORMAL         Normal successful completion
132 0204 1
133 0205 1 SIDE EFFECTS:
134 0206 1
135 0207 1     NONE
136 0208 1 --
137 0209 1
138 0210 1 +
139 0211 1 Equate to SCR$ entry point.
140 0212 1 -
141 0213 1
142 0214 1 GLOBAL BIND ROUTINE LIB$GET_SCREEN = SCR$GET_SCREEN ;
143 0215 1 !<BLF/PAGE>

```

```

145 0216 1 %SBTTL 'SCR$GET_SCREEN - Get Text from Screen'
146 0217 1 GLOBAL ROUTINE SCR$GET_SCREEN (
147 0218 1     INPUT_TEXT : REF BLOCK [,BYTE],
148 0219 1     PROMPT_STR  : REF BLOCK [,BYTE],
149 0220 1     OUT_LEN     : REF VECTOR [,WORD]
150 0221 1 ) =
151 0222 1 +-+
152 0223 1 FUNCTIONAL DESCRIPTION:
153 0224 1
154 0225 1     This routine accepts input from the terminal into a
155 0226 1     user-specified buffer. An optional string may be given to be
156 0227 1     used for a prompt. If a third parameter is present, it will
157 0228 1     be set to the length of the string read.
158 0229 1
159 0230 1 CALLING SEQUENCE:
160 0231 1
161 0232 1     ret_status.wlc.v = SCR$GET_SCREEN (INPUT_TEXT.wt.dx
162 0233 1     [,PROMPT_STR.rt.dx
163 0234 1     [,OUT_LEN.wwu.r]])
164 0235 1
165 0236 1 FORMAL PARAMETERS:
166 0237 1
167 0238 1     INPUT_TEXT.wt.dx     Address of buffer descriptor.
168 0239 1     PROMPT_STR.rt.dx    Optional. Address of descriptor of
169 0240 1     prompt string.
170 0241 1
171 0242 1     OUT_LEN.wwu.r       Optional. Length of input string read.
172 0243 1
173 0244 1 IMPLICIT INPUTS:
174 0245 1
175 0246 1     NONE
176 0247 1
177 0248 1 IMPLICIT OUTPUTS:
178 0249 1
179 0250 1     NONE
180 0251 1
181 0252 1 COMPLETION STATUS:
182 0253 1
183 0254 1     Statuses returned by LIB$GET_INPUT.
184 0255 1
185 0256 1 SIDE EFFECTS:
186 0257 1
187 0258 1     NONE
188 0259 1
189 0260 1 --
190 0261 1
191 0262 2 BEGIN
192 0263 2
193 0264 2 BUILTIN
194 0265 2 AP,
195 0266 2 CALLG ;
196 0267 2
197 0268 2 LOCAL
198 0269 2     TCB : REF BLOCK [, BYTE] ,      ! Current terminal control block
199 0270 2                                     ! Needed by $SCR$GET_TYPE
200 0271 2     TERM TYPE,                      ! dummy arg - device type
201 0272 2     STATOS;

```

```

: 202 0273 2
: 203 0274 2 STATUS = SCR$$GET_TYPE_R3 (SCR$C_GET_SCREEN; TCB, TERM_TYPE) ;
: 204 0275 2 ! this will call foreign terminal
: 205 0276 2 ! code if necessary
: 206 0277 2
: 207 0278 2 IF .TCB [SCR$B_TYPE] NEQ VTForeign
: 208 0279 2 THEN
: 209 0280 2 RETURN ( CALLG ( .AP, LIB$GET_INPUT))
: 210 0281 2 ELSE
: 211 0282 2 RETURN (.STATUS); ! foreign terminal handler does its
: 212 0283 2 ! own input
: 213 0284 2
: 214 0285 1 END; ! End of routine SCR$GET_SCREEN

```

.TITLE SCR\$INPUT SCR\$INPUT - Input routines for the sc  
reen packa

.IDENT \V04-000\

.EXTRN SCR\$\$GET\_TYPE\_R3  
.EXTRN LIB\$GET\_INPUT, SCR\$L\_FLINKHEAD  
.EXTRN SCR\$L\_COROUTPUT

.PSECT \_LIB\$CODE, NOWRT, SHR, PIC, 2

```

          50          00000000G 01 D0 00002
          04          0A          A1 91 0000B
00000000G 00          07 13 0000F
          6C ^A 00011
          04 00018 1$:

```

```

.ENTRY SCR$GET_SCREEN, Save R2,R3
MOVL #1, R0
JSB SCR$$GET_TYPE_R3
CMPB 10(TCB), #4
BEQL 1$
CALLG (AP), LIB$GET_INPUT
RET

```

: 0217  
: 0274  
: 0278  
: 0280  
: 0285

: Routine Size: 25 bytes, Routine Base: \_LIB\$CODE + 0000

: 215 0286 1 !<BLF/PAGE>

SCR\$INPUT  
V04-000

SCR\$INPUT - Input routines for the screen packa  
SCR\$GET\_SCREEN - Get Text from Screen

K 8  
16-Sep-1984 02:30:29  
14-Sep-1984 13:34:43

VAX-11 Bliss-32 V4.0-742  
[VMSLIB.SRC]SCRINPUT.B32;1

Page 6  
(5)

: 217 0287 1 END  
: 218 0288 1  
: 219 0289 0 ELUDOM

! End of module LIB\$SCREEN

LIB\$GET\_SCREEN== SCR\$GET\_SCREEN

PSECT SUMMARY

Name	Bytes	Attributes
_LIB\$CODE	25	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	1	0	581	00:01.0
_\$255\$DUA28:[VMSLIB.OBJ]SCRLIB.L32;1	62	2	3	7	00:00.1

COMMAND QUALIFIERS

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

: Size: 25 code + 0 data bytes  
: Run Time: 00:03.5  
: Elapsed Time: 00:04.5  
: Lines/CPU Min: 4926  
: Lexemes/CPU-Min: 6937  
: Memory Used: 36 pages  
: Compilation Complete

SCR  
V04

000  
000



0437

AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

[Screen 1]	[Screen 2]	[Screen 3]	[Screen 4]	[Screen 5]	[Screen 6]	[Screen 7]	[Screen 8]	[Screen 9]	[Screen 10]	[Screen 11]	[Screen 12]	[Screen 13]	[Screen 14]	[Screen 15]	[Screen 16]	[Screen 17]	[Screen 18]	[Screen 19]	[Screen 20]	[Screen 21]	[Screen 22]	[Screen 23]	[Screen 24]	[Screen 25]	[Screen 26]	[Screen 27]	[Screen 28]	[Screen 29]	[Screen 30]	[Screen 31]	[Screen 32]	[Screen 33]	[Screen 34]	[Screen 35]	[Screen 36]	[Screen 37]	[Screen 38]	[Screen 39]	[Screen 40]	[Screen 41]	[Screen 42]	[Screen 43]	[Screen 44]	[Screen 45]	[Screen 46]	[Screen 47]	[Screen 48]	[Screen 49]	[Screen 50]	[Screen 51]	[Screen 52]	[Screen 53]	[Screen 54]	[Screen 55]	[Screen 56]	[Screen 57]	[Screen 58]	[Screen 59]	[Screen 60]	[Screen 61]	[Screen 62]	[Screen 63]	[Screen 64]	[Screen 65]	[Screen 66]	[Screen 67]	[Screen 68]	[Screen 69]	[Screen 70]	[Screen 71]	[Screen 72]	[Screen 73]	[Screen 74]	[Screen 75]	[Screen 76]	[Screen 77]	[Screen 78]	[Screen 79]	[Screen 80]	[Screen 81]	[Screen 82]	[Screen 83]	[Screen 84]	[Screen 85]	[Screen 86]	[Screen 87]	[Screen 88]	[Screen 89]	[Screen 90]	[Screen 91]	[Screen 92]	[Screen 93]	[Screen 94]	[Screen 95]	[Screen 96]	[Screen 97]	[Screen 98]	[Screen 99]	[Screen 100]
------------	------------	------------	------------	------------	------------	------------	------------	------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	--------------

SCRINPUT  
LIS

SCRMSC  
LIS

SCRRED  
LIS

SETPRV  
LIS

SHRMSG  
LIS

TPARSE  
LIS

SCRIB  
LIS

SCRVECTOR  
LIS

SSMSG  
LIS

SYSCMSG  
LIS